# Representation Analysis Methods to Model Context for Speech Technology


**Anna Ollerenshaw**

Supervisor: Prof Thomas Hain

Department of Computer Science

University of Sheffield

This dissertation is submitted for the degree of

*Doctor of Philosophy*

Speech and Hearing Research Group

July 2023

# Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Anna Ollerenshaw

July 2023

# Acknowledgements

I would like to thank my supervisor Professor Thomas Hain for his supervision and support throughout my PhD. Also to thank my panel members Professor Jon Barker and Professor Eleni Vasilaki for their comments and advice.

I thank Alex Ustinov and Walter Bachtiger from VoiceBase and Liveperson, both for their support and for providing resources throughout my time working in the Research Center.

Particular thanks to Md Asif Jalal who spent countless days, evenings and nights toiling with me in coding and writing and whose unwavering friendship maintained my sanity. I am also thankful for walks and cakes with Rosanna Milner, I really appreciated the smiles, the patience and the support. I am very grateful to them for reviewing my research and providing helpful comments. I would also like to thank my friends and colleagues in the MINI and SPandH groups and to everyone whose discussions, adventures and friendships throughout these years I will truly cherish.

I wish to thank my parents for their support and patience throughout my life, especially during these few years of uncertainty. I found much needed respite in their generosity and woodland walks. I am forever grateful to my extended family in Italy whose love, hospitality and encouragement bring me much needed happiness. Finally, my partner, to whom I am indebted, for her love, kindness and never-tiring patience but particularly for taking care of me when I would surely have failed.

# Abstract

Speech technology has developed to levels equivalent with human parity through the use of deep neural networks. However, it is unclear how the learned dependencies within these networks can be attributed to metrics such as recognition performance. This research focuses on strategies to interpret and exploit these learned context dependencies to improve speech recognition models. Context dependency analysis had not yet been explored for speech recognition networks.

In order to highlight and observe dependent representations within speech recognition models, a novel analysis framework is proposed. This analysis framework uses statistical correlation indexes to compute the coefficiency between neural representations. By comparing the coefficiency of neural representations between models using different approaches, it is possible to observe specific context dependencies within network layers. By providing insights on context dependencies it is then possible to adapt modelling approaches to become more computationally efficient and improve recognition performance. Here the performance of End-to-End speech recognition models are analysed, providing insights on the acoustic and language modelling context dependencies. The modelling approach for a speaker recognition task is adapted to exploit acoustic context dependencies and reach comparable performance with the state-of-the-art methods, reaching 2.89% equal error rate using the Voxceleb1 training and test sets with 50% of the parameters. Furthermore, empirical analysis of the role of acoustic context for speech emotion recognition modelling revealed that emotion cues are presented as a distributed event. These analyses and results for speech recognition

applications aim to provide objective direction for future development of automatic speech recognition systems.

# Table of contents

# List of figures

# List of tables

# Nomenclature

### General Notation

| | |
|---|---|
| $f(\cdot)$ | mapping function |
| $\alpha$ | attention weight |
| $\gamma$ | learning rate |
| $\hat{y}$ | hypothesised output of model |
| $\mathbf{e}_{t,i}$ | bahdanau attention |
| $\mathbf{h}_t$ | hidden state at time step |
| $\mathbf{v}$ | column vector denoted by bold lowercase letter |
| $\mathbf{W}$ | matrix denoted by upper case letter |
| $\mathbf{x}_t$ | input feature at time step |
| $\mathcal{L}$ | cost function |
| $\pi$ | alignment path |
| $\rho$ | correlation coefficient |
| $\sigma$ | activation function |

| | |
|---|---|
| $\mathbf{b}$ | bias |
| $d$ | dimensionality of embedding |
| $max$ | maximum |
| $P(\cdot)$ | probability distribution |
| $p(\cdot)$ | probability density function |
| $s$ | scalar denoted by plain lowercase letter |
| $T$ | dimensionality of time dimension |
| $y$ | ground truth label |
| $z$ | activation output of neuron |
| $\mathbf{K}$ | Key matrix |
| $\mathbf{Q}$ | Query matrix |
| $\mathbf{V}$ | Value matrix |

**Acronyms / Abbreviations**

| | |
|---|---|
| ASR | Automatic speech recognition |
| BLSTM | Bi-directional Long Short-Term Memory Network |
| BPE | Byte Pair Encoding |
| CCA | Canonical Correlation Analysis |
| CE | Cross-Entropy |
| CER | Character Error Rate |

CKA Centered-Kernel Alignment

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

CV Consonant-Vowel

Dconv Dynamic Convolution

DCT Discrete Cosine Transform

DNN Deep Neural Network

EER Equal Error Rate

FAR False Acceptance Rate

FRR False Rejection Rate

GMM Gaussian Mixture Model

HMM Hidden Markov Model

HSIC Hilbert-Schmidt Independence Criterion

KER Keyword Error Rate

KL divergence Kullback-Leibler Divergence

LER Label Error Rate

LF-MMI Lattice-Free Maximum Mutual Information

LM Language Model

LSTM Long Short-Term Memory Network

Mband Multi-Band

Mchan Multi-Channel

MFCC Mel Cepstral Coefficient

MoE Mixture of Experts

MSE Mean Squared Error

PLDA Probabilistic Linear Discriminant Analysis

PWCCA Projected-Weighted Canonical Correlation Analysis

ReLU Rectified Linear Unit

RNN-T Recurrent Neural Network Transducer

RNN Recurrent Neural Network

SER Speech Emotion Recognition

SVD Singular Value Decomposition

SV Speaker Verification

TDNN Time-Delay Neural Network

UA Unweighted Accuracy

UBM Universal Background Model

WA Weighted Accuracy

WER Word Error Rate

WIL Word Information Lost

# Chapter 1

# Introduction

## Contents



## 1.1 Speech Technology

Speech is a fundamental mode of human communication, enabling the exchange of information, thoughts and ideas. Until the 1970s, speech technology lacked the human skills to be able to listen, understand and learn speech. They instead relied upon input methods such as keyboards, joysticks and other physical hardware. The introduction of automated systems instead propelled communication between humans and machines to occur as autonomously as possible. Key fields that encompass speech technology are automatic speech recognition (ASR), speaker recognition, diarisation, speech emotion recognition (SER) and speech synthesis. The topics that are focused on within this research are the latest developments

within ASR, SER and speaker recognition. These areas are interconnected topics within the field of speech technology with distinct objectives, but share common methodologies and applications, where advancements in one area often benefit the others. Combinations of these systems are also often used in industries such as finance, retail, hospitality and manufacturing for automated customer interaction, telemarketing optimisation, speech analytics and product insights.

Modern speech technology systems have utilised distinct computational modules to compute specific task objectives, for example, pronunciation and language modelling. The majority of systems typically include signal processing techniques, acoustic modelling, language modelling and hypothesis searches to enable the interface for recognition. These techniques have relied heavily on domain knowledge of linguistics, signal processing techniques and computational model engineering. Modelling domain knowledge includes a scientific understanding of speech, sound and language while utilising concepts of probability and pattern recognition to deconstruct or construct the speech signals.

Specifically, ASR systems attempt to encode speech representations of the domain knowledge so that it can be understood by a human (such as text transcriptions) or for a down-stream task. Regarding the task speech-to-text recognition, this understanding is represented by the production of text in the target language that could be understood by a human reading it. The performance of these ASR systems is widely measured by the word error rate (WER) or character error rate (CER) which compares the predicted output of the network with the ground truth.

Speaker recognition systems attempt to recognise and identify speakers from their speech or voice. This can include non-speech vocalisations produced that does not have linguistic context such as coughing, laughing, groaning or clicking. The components of a speaker recognition system include speaker modelling and feature extraction to represent a speaker based on the extracted speech features. The performance of a speaker recognition system

is widely measured using false acceptance rate (FAR), false rejection rate (FRR) and equal error rate (EER). EER is where FAR is measured against FRR.

ASR and speaker recognition are closely related and can benefit from each other, for example, ASR systems can adapt models to individual speaker features to improve their performance [18]. Speaker segmentation techniques can also be crucial for ASR systems to adapt to multi-speaker scenarios or where there is overlapping speech present, which is referred to as diarisation. By identifying different speakers within an audio stream, ASR systems can be trained to output appropriate labels to each speaker's speech [19].

Speech emotion recognition (SER) systems attempt to recognise and classify the emotion states from a speakers speech or voice, including non-speech vocalisations. Emotions play an essential role in human communication and affects the understanding of context and meaning. Harmonics of vowels and consonant sounds add paralinguistic cues. Emotion perception can be culturally or linguistically dependent and is typically evaluated with either a categorical or dimensional metric.

ASR and SER systems both attempt to extract meaningful information from the speech signal. ASR models have been shown to contribute to improved SER model performance by jointly training both models [20]. A correlation between SER and ASR systems has also been explored where features learned in some model layers for both tasks were found to be applicable to each other, particularly in the initial model layers [21]. Speaker recognition systems also intersect in relation to SER systems as they both use similar techniques to attempt to extract voice characteristics and features for different classifiers.

Following huge advances in the fields of computational technology (high performance cluster computing, GPU architectures etc.) and speech technology (parallel computation, accessible libraries etc.), the robustness of modelling approaches to noise and conversational speech is arguably reaching "human parity" [22, 23], thus resulting in an explosion of scientific and public interest. As well as combined systems, modern speech technology advances

have increased industry market interest with developments including personal assistants (Amazon's Alexa , Apple's Siri, Microsoft's Cortana, Google's Assitant etc.), dictation systems, information retrieval systems, and military and security systems. These systems typically use various combinations of neural networks, modelling techniques and modern hardware. Due to the computational requirements of training ASR, SER and speaker recognition systems, many model components can be pre-trained and saved for later integration into systems.

The focus of current research within the field of ASR, SER and speaker recognition is to increase recognition performance, build systems for multiple domains and to reduce some of the modelling requirements for domain knowledge. The pre-trained models provide the domain knowledge and are able to tune models to achieve higher performance without needing to train the entire system from scratch. This has given rise to frameworks referred to as End-to-End. The aim of End-to-End frameworks is to train model components jointly from only the input speech, without relying on pre-trained aspects, which is useful for applications where low latency is required or where domain knowledge is limited. In practice, many End-to-End frameworks partly integrate pre-trained components, such as language models, to improve performance. End-to-End approaches are not yet able to achieve the same performance as models that use mostly pre-trained components; performance decreases as the vocabulary size grows and where there are speaker variations, such as accents, dialects, emotive and interrupted speech. There are also considerations that are still non-trivial to model within End-to-End approaches, such as context and semantic constraints, without requiring computational capabilities that are currently unfeasible.

Context modelling in speech technology refers to the incorporation of contextual information to improve the system robustness or accuracy. This can consider linguistic or acoustic context in order for the system to infer the correct interpretation. Typically this leverages information gathered from previous or succeeding words, features or speech patterns to

overcome ambiguity and improve accuracy. Context modelling within speech applications has been improved by the integration of techniques, such as attention mechanisms and deep neural networks (DNNs). Attention-based models are dependent upon the mapping of non-linear dependencies across the input speech and DNNs are often regarded as “black-boxes”. How the network arrives at the output and the relationship of neuron dependencies to map the latent representation space are typically not well understood. Due to a lack of understanding, modelling approaches not easily adapted for different tasks. The performance of speech technology is still highly dependent on increasing the training data resources and parameter space to capture higher quality representations.

## 1.2 Hypotheses and Research Questions

Due to the nature of human speech and communication, context within the speech signal is able to provide a model with more information for specific tasks, such as speaker specific dependencies or linguistic context. Models for ASR, speaker recognition and SER that are able to utilise these variations and exploit them, are more robust to noise and variability across tasks and domains. For End-to-End frameworks, the task of modelling context dependencies is an ongoing focus of research that is still in the early stages of development and there has been little analysis done. By exploring the dependencies within End-to-End frameworks, it may be possible to utilise these insights to improve modelling approaches across the domains of speaker recognition and SER.

To determine the limitations of models for End-to-End frameworks, there is no defined solution. A point of contention within the scientific community has been "what determines an End-to-End framework?” As this term is not explicitly defined, there are several interpretations and published models, with some approaches incorporating pre-trained modules, while others have attempted to completely remove the reliance on external domain knowledge and train the models with only one optimisation strategy. The dependencies of modelling

approaches on pre-trained modules is not clearly understood with regard to performance metrics and also interpretability. For example, how does the model adapt the latent representation space with the incorporation of a pre-trained module?

Despite the variation in definition and approaches, a further question poses, does the specification of End-to-End matter for specific recognition tasks? On one hand, the specification becomes important to distinguish when publishing approaches with regards to specific attributes, such as model size or resource requirements. On the other hand, parallel GPU optimisation and developments in computational technology are exponentially driving models with increased parameters and more computationally demanding algorithms, allowing for "on-the-fly" training of expanded models. These developments will continue to improve model performance, while still being within the domain of End-to-End. One of the major benefits that remains of reducing the need for reliance on external domain data and pre-training is that the models are able to be used for resource constrained tasks, such as where there are limited labelled datasets.

As the use of attention within modelling approaches is a relatively recent research development, little work has been done regarding the effect of modelling approaches and dependent neural representations upon the recognition performance of the system and ability to observe the changes in latent space.

The research questions that this work aims to address are:

- Is it possible to analyse the latent space of neural representations of ASR networks to provide some interpretation of how the representations relate to recognition performance? This is explored in Chapter 3.

- Can analysis of the dependencies of neural representations be used to understand the impact of specific modelling approaches for speech technology? These techniques and dependencies are discussed and evaluated in Chapters 4 and 5.

- Is it possible to exploit acoustic or linguistic context dependencies modelled by attention-based techniques in order to improve model performance without increasing the computational requirements? Experiments are presented in Chapters 5 and 6 that evaluate this for End-to-End ASR and speaker verification.

- How does the choice of technique for modelling context dependencies affect the performance across speech applications, such as speaker recognition or emotion recognition? This hypothesis is explored for speaker verification in Chapter 6 and for SER in Chapter 7.

## 1.3 Contributions

In order to explore dependencies within End-to-End approaches, an analysis framework to conduct statistical correlation analysis of representations within neural networks for ASR has been developed (presented in Chapter 3, Section 3.4). This framework uses state-of-the-art End-to-End ASR modelling approaches but can be incorporated with other neural network applications.

In order to determine how to analyse the representations within neural networks, a comparison study of statistical correlation indexes to measure the coefficiency of neural representations for ASR is presented in Chapter 3, Section 3.6.3.

Using both of these contributions, Chapter 4, Section 4.3 presents an interpretative analysis of state-of-the-art acoustic modelling techniques for End-to-End ASR has been conducted, to derive insights of the properties of modelling parameters. These insights can be used to develop new models and adapt techniques for downstream tasks. Further analysis of state-of-the-art acoustic modelling for cross-corpora End-to-End ASR has provided knowledge of cross-corpora dependencies, which is presented in Chapter 4, Section 4.4. These results indicate that specific layers within the models generalise latent spaces for ASR tasks.

To explore linguistic context modelling, language modelling with End-to-End ASR model integration is presented in Chapter 5, Sections 5.3.3 and 5.3.3. This resulted in improving recognition performance using interpretative analysis to identify cross-domain dependencies.

Where the insights and results from exploring the context dependencies and modelling approaches have been utilised to improve speaker recognition performance for a speaker verification task. A novel approach for speaker recognition modelling is proposed (Chapter 7, Section 6.4) to improve accuracy, and with a focus on reducing computational requirements (results shown in Chapter 6, Section 6.6.

Finally, building off the exploration of interpretative analysis in Chapters 4, an analysis approach is developed and presented to observe that current trends in speech recognition modelling should incorporate changes in acoustic context when classifying emotions (presented in Chapter 7, Section 7.5.

Supporting publications regarding the contributions are:

- [24] Ollerenshaw, A., Jalal, M. A., Hain, T. "Insights on Neural Representations for End-to-End Speech Recognition." Proc. Interspeech 2021, pages 4079-4083.

- [25] Ollerenshaw, A., Jalal, M. A., Hain, T. "Insights of Neural Representations in Multi-Banded and Multi-Channel Convolutional Transformers for End-to-End ASR." 2022 30th European Signal Processing Conference (EUSIPCO). IEEE, 2022.

- [26] Ollerenshaw, A., Jalal, M. A., Hain, T. "Probing Statistical Representations for End-to-End ASR." 2023 31st European Signal Processing Conference (EUSIPCO). IEEE 2023.

- [27] Ollerenshaw, A., Jalal, M. A., Hain, T. "Dynamic Kernels and Channel Attention with Multi-Layer Embedding Aggregation for Speaker Verification." submitted to 2023 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), IEEE.

- Ollerenshaw, A., Jalal, M. A., Milner, R., Hain, T. "Empirical Interpretation of the Relationship Between Speech Acoustic Context and Emotion Recognition." submitted to 2023 Frontiers in Artificial Intelligence.

## 1.4 Thesis Organisation

The beginning of each chapter includes a description of the research area and specific gaps that chapter will explore. Each section of the chapter presents the individual research contributions, which are structured with a subsection depicting the background to the topic, related works to the modelling or analysis approach, experiments, results and discussion. Each chapter is summarised with the key findings and discussion points that lead to subsequent chapters.

Figure 1.1 shows the outline of the thesis structure after the introductory chapter. Chapter 2 discusses the background and foundations of ASR modelling and introduces the model structures that modern frameworks are comprised of. The different approaches for End-to-End ASR are detailed and the limitations of these techniques. Chapter 3 presents the state-of-the-art End-to-End ASR frameworks and evaluates them on a conversational speech recognition task. This chapter presents model recognition output assessments to outline the motivations of the neural representation analysis. The analysis framework is also detailed in Chapter 3, which explores current statistical indexes. The analysis framework in Chapter 3 is used for the analysis experiments in Chapters 4 and 5. Chapter 4 presents an assessment of representations learned by End-to-End ASR modelling approaches with different acoustic contexts. Chapter 5 details the integration of language modelling into End-to-End ASR frameworks and provides analysis regarding the layerwise dependencies of the models. The analysis in Chapter 5 is also used to adapt model parameters to improve domain specific recognition. Chapter 6 describes a proposed modelling approach for improving context modelling for speaker recognition. The proposed modelling approach improves verification

results without increasing computational resources required. Chapter 7 describes a proposed context modelling strategy for speech emotion recognition and details analysis across different domains. Finally, Chapter 8 concludes the thesis and proposes future work that could be conducted from the presented research.

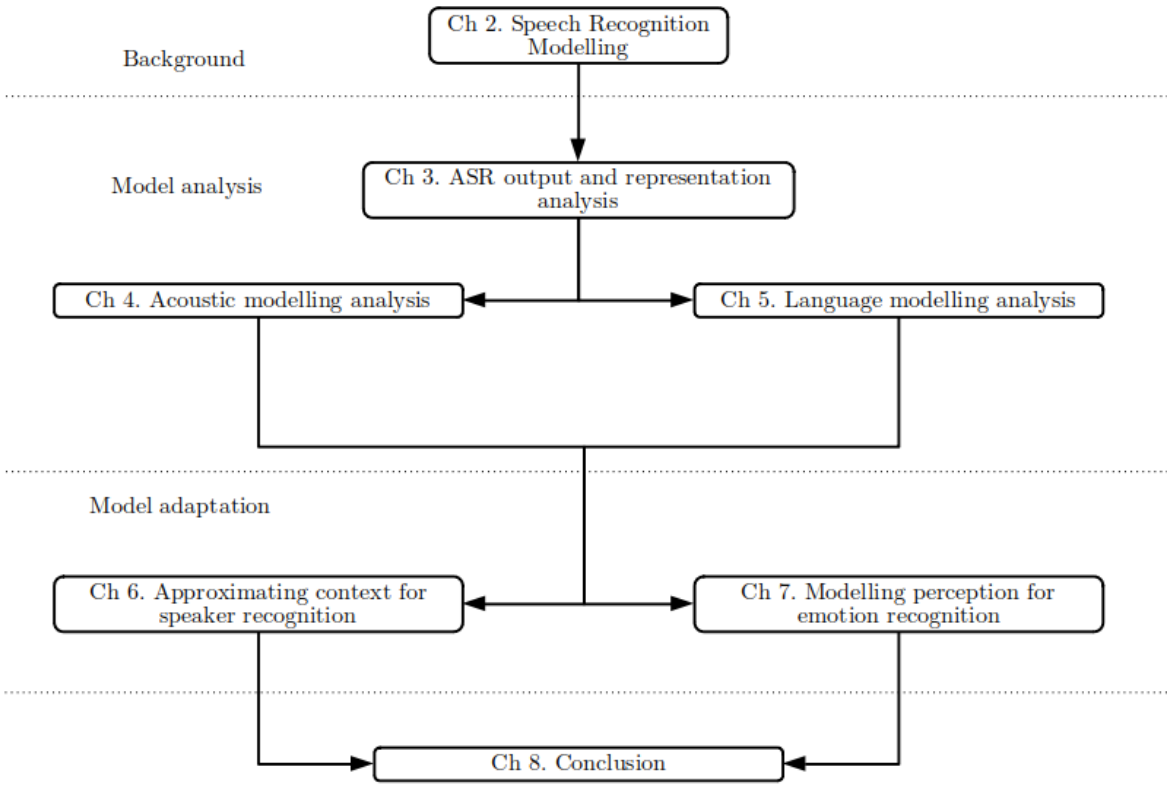



Fig. 1.1 Outline of thesis structure

# Chapter 2

# Speech Technology

## Contents

## 2.1 Introduction

This Chapter presents a review of speech modelling approaches and introduces the key background to acoustic and linguistic context. Neural networks for speech recognition are defined and outlined, which are the foundations to current speech modelling approaches. A thorough review of End-to-End frameworks is also given, to introduce the terminology for subsequent Chapters.

## 2.2 Speech Modelling

Speech modelling is considered a challenging task to automate due to the variability of speech signals. Automated systems have been built to specialise in recognising a particular speaker, language, style of speech or within certain environmental constraints. Robustness is the term for a system's ability to retain recognition performance despite the introduction of variables such as noise or unseen data. Accuracy is the measure of the ability of a system to recognise speech according to an evaluation metric.

There are many nuances to speech that require the recogniser to interpret contextual information in order to understand and comprehend the meaning of the speech. Words can have the same pronunciation but their meaning differs given context (homophones), such as *red* and *read*. Additionally there exists context variability at phonemic levels, such as the phoneme */ee/* within the words *beat* and *meet*. Context variability can become increasingly difficult to interpret as the velocity of speech deliverance increases, which is typical of

spontaneous conversational speech. Variations in speech velocity can be challenging to recognise when attempting to model spectral features and temporal information [28]. For continuous speech recognition tasks, the error rate is typically higher for conversational speech than for read speech [29]. Conversational speech also consists of more emotive speech [30–32] and varies in factors such as amplitude, emphasis and articulation style [33, 34] across cultures [35]. [36] explored another factor affecting the modelling difficulty, which is speaker variability; where the physical differences between a person's vocal tract, age, gender, health, dialect and other individualities can cause variations in speech, challenging the robustness of recognition systems.

In order to develop an ASR system that is robust and accurate with these variables and context, the following discussed modelling approaches have been developed that are targeted to perform a specific aspect of the recognition within a pipeline.

### 2.2.1 Front-end and Acoustic Modelling

Acoustic modelling for speech processing is mainly concerned with the physical properties that govern the propagation of sound waves from the vocal tract, as speech is a sound wave created by vibrations. These theories typically consider variations of the vocal tract shape through time, excitation of sound, energies of the vocal tract and dispersion across the other articulators [37]. Human perception of acoustic signals is also motivated by the typical auditory system [38, 39] and several methods have been developed in order to model acoustic information for speech recognition applications.

ASR systems need to be able to handle variable length inputs and to integrate the context variables from speech into the system. There are a vast number of variables to consider in the implementation of an ASR system, such as speaker specific characteristics: gender; health and stress, and the speech style within the corpus: formality; language; dialect; accents; environmental noise; channel distortion; availability of appropriate data and whether there is

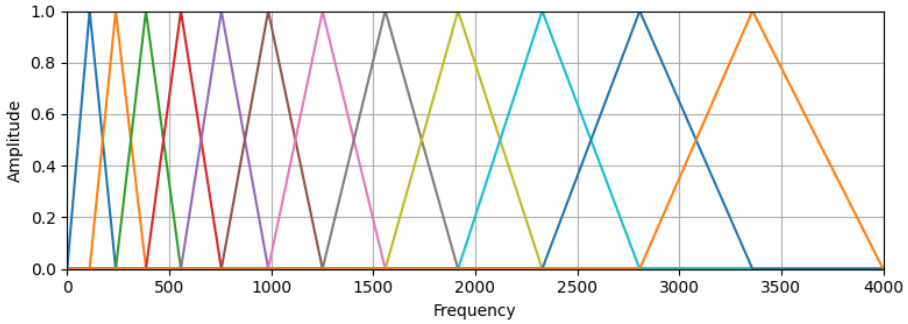



Fig. 2.1 Graph showing triangular filter banks on the Mel scale

a written form of the language [40]. There are signal processing techniques that transform raw speech audio signals and process them to retain speech, whilst simultaneously removing distortion and noise.

Factors such as noisy environments, channel distortion and mismatch are typically addressed using speech enhancement and feature extraction techniques [41]. [42] focused on techniques that attempt to fix the length of the sequences by clipping, padding or overlapping. Clipping, padding or overlapping the vectors has a trade-off as performance of the model decreases, while these methods are relatively computationally inexpensive [43, 44].

Techniques for feature extraction have been developed to transform the speech signal into vectors to attempt to model speech acoustics [45–47]. Feature vectors, filter banks or Mel Cepstral Coefficients (MFCCs) can be extracted by converting the audio signal into the frequency domain. Mel-filter banks [48, 47] are derived by a short-term Fourier transform across a sliding window between 20 and 40ms in order to capture enough samples from the frames of the signal and to get a representative spectral estimate. The power spectrum of each frame is calculated to simulate a human cochlea, which is able to determine the different frequencies. The spectrum is sorted into filter banks to calculate energy in frequency ranges according to the Mel scale, a non-linear scale, which simulates human auditory perception. The frequency is converted to the Mel scale using the formula $m = 2959log_{10}(1+\frac{f}{700})$ where $f$ is the frequency in Hertz, which restricts the higher frequencies and expands the lower

frequencies. This matches theories of human perception of speech, whereby the auditory system discriminates more at lower frequencies than higher frequencies. Figure 2.1 shows an example of triangular filters distributed across the scale.

The MFCC [49, 46, 47] is described as a cepstrum of a signal's window. As perception is not a linear scale, the log amplitudes of the filter bank features can be transformed with a discrete cosine transform (DCT) to result in the cepstral vectors and ensure there are no overlaps, while typically only retaining 13 of the coefficients. The other coefficients are considered more redundant for speech recognition applications. Recent research [50] has suggested that a minimum of 25 coefficients should be used for more modern modelling approaches. [51] suggested that MFCCs are less susceptible to variation, however [52] found that filter bands may carry useful information for specific ASR tasks, which could be lost when converting them into MFCCs.

More recently, deep learning techniques [53] have been used to attempt to capture higher level features that are more robust. These deep learning models are described in further detail in Section 2.3.

### 2.2.2 Language Modelling

Linguistic context refers to the relationships and dependencies between words and phrases within a given language. To provide this linguistic context and aid recognition, knowledge of vocabulary, pronunciation, syntax and the semantics of the language are required [54, 55]. [56] showed that greater End-to-End ASR model accuracy can be attained with the use of a language model (LM) where training corpus size is limited, as this method is able to introduce domain specific information; thereby increasing the entropy [57].

There are several different approaches to model linguistic context, while for ASR applications, statistical language modelling is a more common-place solution [58]. A statistical LM can be introduced to predict the probability of a whole word sequence or the next word

within a sequence by observing the previous word sequences in a text corpus. The sequence of words (**w**) is typically referred to as N-gram where the history of words is restricted to $(N-1)$ words. The simplest LM, where $N=1$, is known as a unigram LM. The probability of a word ($w$) would be determined by $\prod_{N=1} P(w_N)$. This type of model assumes that the probability of each word is independent and only provides the statistical probability of the word occurrence among all words $\{w_1,...,w_k\}$ in the training corpus. The probability of the next word in the word sequence given the previous $N$ words can be described by:

$$P(w_1^N) \approx \prod_{n=1}^{N} P(w_n|\mathbf{w}_{n:N-1}) \tag{2.1}$$

To improve the perplexity of the LM, a large amount of text is required for training [59]. However, the LM is limited in vocabulary to the corpus and sequences of words may exist that are unknown and therefore have no probability. This can be handled by allocating a proportion of the overall sequence probability for unseen sequences [60].

Byte Pair Encoding (BPE), proposed by [61], was originally a data compression technique that has been utilised for sub-word tokenisation. Frequently occurring sub-word pairs are merged in order to combine character and word level representation sequences, which is able to handle rare occurring vocabulary that is potentially not present in the training data.

Sub-word LMs are a common choice for training End-to-End ASR models [62, 3, 63] to reduce computational complexity by only keeping the most frequent words and splitting the rest into sub-words, again only keeping the most frequent sub-words. All remaining tokens are separated into characters, to enable conversion with little information loss as proposed in [64].

### Decoding Strategies

For a sequence-to-sequence ASR system to determine the most likely word sequence, given the observed input, a hypothesis search is required. The hypothesis search utilises the LM

and acoustic model predictions, where applicable, given the input and thereby models the most probable output word or character sequence.

Using a greedy classifier, the most likely prediction can be determined very naively per time-step. However, a greedy classifier does not always find the actual most probable prediction as it cannot compute the sum of probability paths over targets [65]. Another method to find the most probable prediction is prefix search decoding [66], which allows searching through the whole input. However, the search space grows exponentially and is computationally expensive to calculate. The current most commonly used method is beam search decoding [67], as this keeps the computation window within the number of $n$ best candidates so it does not become too expensive to calculate.

### 2.2.3 Speaker Modelling

The task of a speaker recognition system is to recognise and identify speakers from their speech or voice [68]. This can include non-speech vocalisations such as laughing, coughing or exclamations. Speaker recognition is differentiated into speaker identification tasks and speaker verification tasks. Speaker identification attempts to recognise a speaker from a known set of speakers in a one-to-many mapping, while verification attempts to authenticate whether the speaker of an utterance is the target speaker without prior knowledge in a one-to-one mapping [69]. Speaker identification is typically a closed-set task, as the enrolled speakers are known [70], while speaker verification is open-set as the system has no prior information regarding the speakers and simply provides an estimate of whether the utterances belong to the same speaker. Verification is typically computed using a similarity score with a threshold that determines whether the speaker is the target speaker [71]. Both recognition and verification of speakers can be text-dependent or text-independent, where text-dependent techniques utilise transcripts for the target speakers [72]. Text-independent techniques do not have constraints on the words permitted to be spoken by the speakers [73].

A more detailed discussion regarding speaker recognition and the state-of-the-art modelling approaches is explored in Chapter 6, whereby a new approach that builds upon this previous research is proposed to approximate context for speaker specific modelling.

### 2.2.4 Speech Emotion Modelling

Within the field of speech recognition, emotion recognition may be considered a branch that involves similar signal processing, feature extraction and modelling approaches, while differing in the target output classifiers. Emotion recognition is considered a useful aspect of speech recognition systems to capture intention and context behind the literal words. This context information is produced when the speaker varies acoustic properties, such as tone, speed and emphasis, or linguistics, such as negative or positive sentiment.

Speech emotion understanding between humans and emotion recognition with machines and humans are complex research areas that contain elements that are not well understood or agreed upon in the scientific community [74–77]. Current automated modelling approaches focus on adaption to speech variability, while reducing redundancy in acoustic and linguistic perceptual cue recognition. These approaches are particularly challenging to develop because the target labels or the perceived emotion states can be considered very subjective or biased by cultural and linguistic perception differences. Speech emotion, within the domain of SER, is typically represented by two approaches: categorical and dimensional. Speech acoustic segments can be treated as a categorical entity consisting of discrete emotions such as *happy*, *sad*, *fear*, etc [78]. In the categorical approach, annotators label audio segments as emotion categories and use them to model speech emotion. The dimensional approach proposes two fundamental dimensions, valence and arousal, to represent emotion at a given time [79].

Linguistic and cognitive theories regarding a human's perceived speech emotion and the currently developed computational modelling methods are explored in further detail in Chapter 7.

### 2.2.5 Evaluation of Accuracy Measures

A performance indicator for the majority of ASR research is a measurement of the amount of errors produced by the system. Word error rate (WER) and character error rate (CER) have become standard measures of performance. CER is calculated by comparing the predicted output characters of a network against the ground-truth labels, and computing the minimum amount of editing operations to transform the prediction into the true output. The editing operations are defined as insertions $i$, substitutions $s$ and deletions $d$ over the total number of ground-truth characters $n$, whereby:

$$CER = [(i+s+d)/n] * 100 \tag{2.2}$$

This is also referred to as the edit or Levenshtein distance of the model output. WER is similarly calculated by summing the amount of inserted, substituted and deleted words, then dividing over the total amount of words in the ground-truth. These metrics are fairly limited when attempting to analyse and interpret the performance of ASR models and determine the reason behind the wrong hypothesis output compared to the ground-truth [80, 81]. These metrics also do not differentiate between whether an error was potentially worse than another error or whether despite the error, a human could still differentiate the ground-truth from the model output.

As frameworks have become larger, more complex and specialised to the domain, the relationship between internal dependencies on the performance can become less visible. Therefore it can be useful to understand and extract further information in order to develop models with improved recognition performance or for specific speech analytic applications. There has been some research to target more informative performance metrics for ASR such as Keyword Error Rate (KER), weighted WER and Word Information Lost (WIL). KER [82] attempted to evaluate the model using keywords rather than words. Keywords are

determined as being domain-specific verbs or nouns and if a word occurs more frequently than in a non-domain-specific text. The metric is defined as the sum of falsely recognised keywords and missed keywords, over the total number of keywords. However [83] found that there were no significant differences between WER and KER as metrics for speech understanding applications. Similarly, the weighted WER [84], weights the impact of types errors on the overall error metric, based upon the hypothesis that specific keywords have a higher impact on the information retrieval. This metric was developed for speech query tasks, where each word is weighted during evaluation, to approximate the dependency upon the intention. The estimation of the weighting approximation is determined by the information retrieval degradation ratio score for all queries. For many speech recognition tasks, this evaluation metric is considered to be computationally expensive and is more suited to retrieval applications. WIL [85] approximates the mutual information between the hypothesised output and the ground-truth using Shannon Entropy $H$, where:

$$WIL = 1 - \frac{H^2}{(H+s+d)(H+s+i)} \tag{2.3}$$

Entropy is a measure of the average amount of uncertainty in a probability distribution and quantifies the amount of information contained within the distribution. This metric is suited to ASR evaluation where the error rates are higher. Despite the development of informative metrics, currently WER is the most widely accepted metric to evaluate the performance across the majority of ASR tasks.

For speaker recognition tasks, the evaluation is determined using metrics such as false acceptance rate (FAR), false rejection rate (FRR) and equal error rate (EER), described in [86]. EER measures the similarity between the FAR and FRR of the classifications. FAR is the percentage of speakers recognised that are incorrectly accepted as the target speaker, while FAR is the percentage of recognition instances where the target speaker is incorrectly rejected. EER is the point where FAR is equal to FRR.

Speech emotion recognition modelling accuracy is typically evaluated using weighted accuracy and unweighted accuracy. Unweighted accuracy calculates the proportion of correctly classified instances across all emotion classes without considering the distribution of the class. Each class of emotion is measured equally, providing a general measure of the system. While weighted accuracy portions the accuracy by weighting the class distribution of the dataset based on each classes prevalence. It is common for datasets to contain more class labels for certain emotions, therefore this metric attempts to measure the accuracy of the classification more acutely, providing a representation of performance to counter imbalanced class distribution where there may be classes that are underrepresented by the data.

The metrics for SER accuracy do not address the granularity of emotion labelling, whereby it is a complex task for humans to determine a distinct emotion label [87]. A sample of human annotators is sometimes used to provide a distribution range for the labelling to attempt to alleviate some subjective bias.

## 2.3 Neural Networks for Speech Recognition

Historically, research has attempted to tackle front-end processing issues within speech modelling systems, such as variable length vectors, continuous speech and audio signal variability. Since the creation and adoption of early mathematical models of neuron behaviour in the field of artificial intelligence [88–90]; machine learning research for enabling complex speech modelling expanded.

The development of neural networks has vastly expanded each area of speech recognition, speaker recognition and speech emotion recognition due to their ability to learn complex representations and patterns within speech data. The ability to use neural networks for speech modelling allows for parameter estimation in order to best approximate the function from the input signal to the output transcription [91]. The following Section defines and explains neural networks that are used for speech modelling. This covers the different types of neural

networks, activation functions between network layers, and training objectives. ASR, SER and recognition systems are comprised of these elements and sometimes combinations of these elements. As speech modelling involves capturing the underlying representations and statistical properties of the speech signal, utilising these techniques enables systems to understand and classify spoken language, speakers and emotions more accurately.

### 2.3.1 Feed Forward Neural Networks

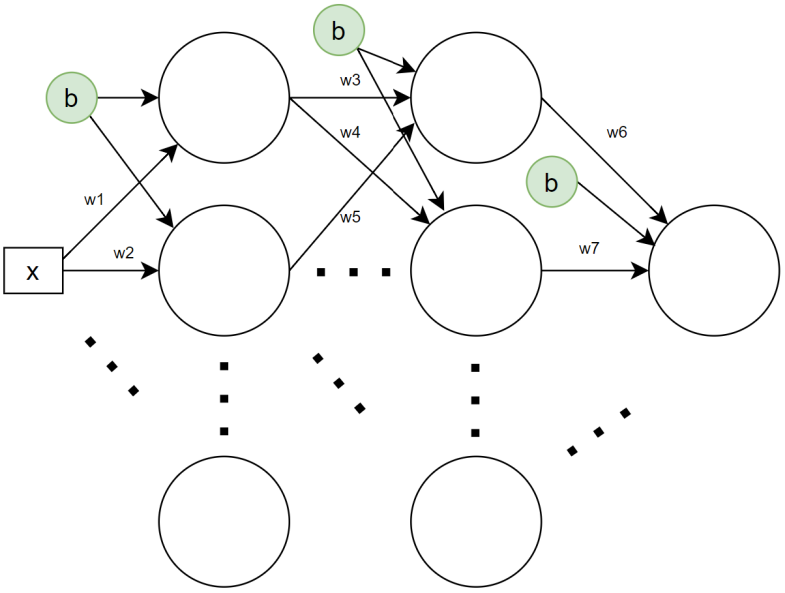



Fig. 2.2 Example diagram of a feed forward neural network showing connections between neurons with weights and biases

Multi-layer perceptrons, commonly referred to as neural networks, are biologically inspired models. In feed-forward neural networks, training typically occurs through back-propagating errors iteratively through the layers, typically using input data $X$. Figure 2.2 shows a simple diagram of an example feed forward network, where the input $X$ is summed with weights and additional bias to the neuron. The weights $W$ of the network are updated by an optimisation strategy, such as gradient descent, which is where the output of the network, given the input $x_i$ is compared to the ground truth label $y_i$. Through connected hidden layers,

the strategy attempts to solve the function that results in the output of the network $\hat{y}$ being as close to the ground truth $y$ as possible. This is shown in further detail in Section 2.3.1. The output of the neuron $z_j$ is shown in Equation 2.4, where $b_i$ and $w_i$ are the bias and weights at the $i^{th}$ iteration of $N$ input features:

$$z_j = \sum_{i=1} w_i x_i + b \tag{2.4}$$

The weights are model parameters, referred to as $\theta$, that transform the input through the layers of the network. The inputs to each neuron are multiplied with the weights and the weights are updated by the training schedule. Biases are added to the weighted sum of the input in order to delay the activation function. An initialisation strategy of the network parameters, such as Xavier initialisation [92], can be used to decrease time to convergence and improve model accuracy. However, in cases where little is known about the search space, the weight parameters are typically selected using random bounded values from a Gaussian distribution [93].

Networks typically consist of multiple feed forward layers. The inner layers are referred to as hidden layers $\mathbf{h}$ where layer $j$ can be described as:

$$\mathbf{h}_j = \sigma(\mathbf{w}_j \mathbf{h}_{j-1} + b) \tag{2.5}$$

where $\mathbf{h}_{j-1}$ denotes the previous layer output, $\mathbf{w}_j$ are the weights, $b$ is the biases and $\sigma$ is the activation function.

### Activation Functions

For each neuron in the network, the inputs are multiplied by the weights and summed to determine the activation. An activation function is used to determine the desired output of the network layer. Non-linear activation functions allow the network to learn more complex

mappings and normalise the output of the neural layers, typically to values between 0 and 1 [94].

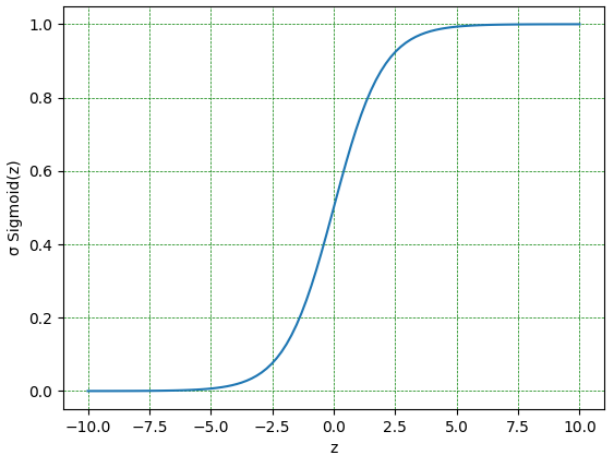



Fig. 2.3 Graph of a sigmoid function

The sigmoid function $\sigma_{sigmoid}$, shown in Figure 2.3, outputs a value between 0 and 1 determined as a weighted sum, $z_j = \sum_{i=0} x_i w_i$ of inputs to a neuron

$$\sigma_{sigmoid}(z_j) = \frac{1}{1+e^{-z_j}} \tag{2.6}$$

The hyperbolic tangent function $\sigma_{tanh}$, shown in Figure 2.4, is similar to the sigmoid function, while it outputs the value between -1 and 1:

$$\sigma_{tanh}(z_j) = \frac{1-e^{-2z_j}}{1+e^{-2z_j}} \tag{2.7}$$

The softmax function $\sigma_{softmax}$, shown in Figure 2.5, computes the relative probability, between 0 and 1, for the outputs of the network over a number of classes. The exponential function is applied to each element of the neural representation vector in the output layer of the network, and is then normalised by dividing by the sum of exponential representations:

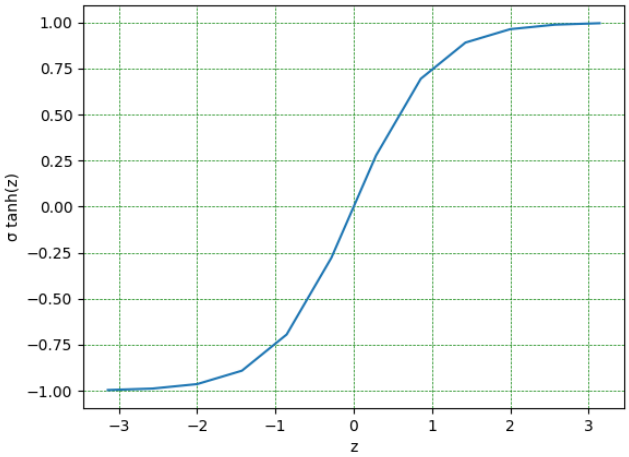



Fig. 2.4 Graph of a hyperbolic tangent function

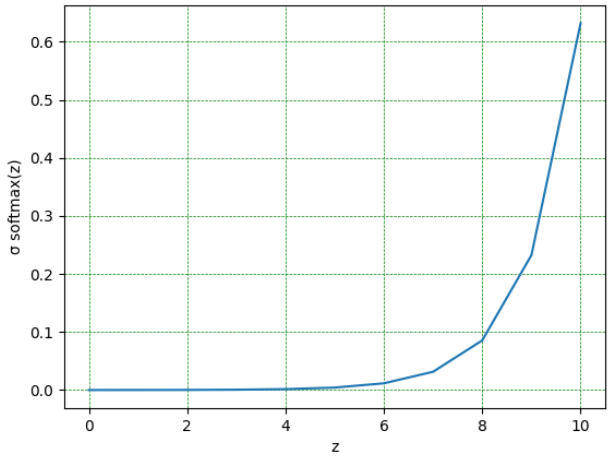



Fig. 2.5 Graph of a softmax function

$$\sigma_{softmax}(z_j) = \frac{e^{z_j}}{\sum_{j=1} e^{z_j}} \tag{2.8}$$

To update the parameters of the neural network with regard to minimising the error, the calculation of the gradient information for back-propagation is critical, and as this gets smaller, training can be inhibited. This is referred to the vanishing gradient problem [95]. Another issue is that the activation function can be computationally expensive to computer,

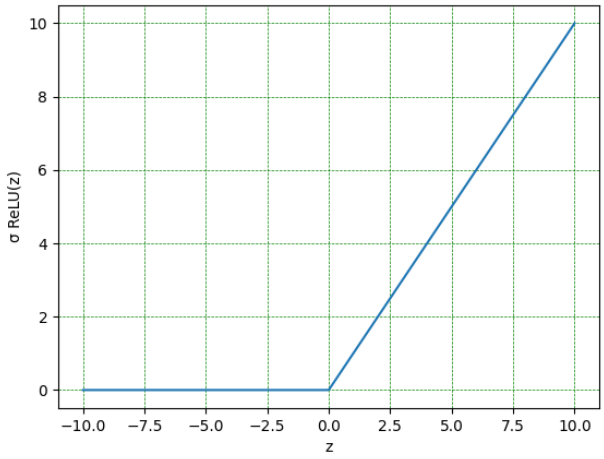



Fig. 2.6 Graph of a ReLU function

especially within deeper networks that consists of many layers. Some activation functions, such as the Rectified Linear Unit (ReLU) function [96] $\sigma_{ReLU}$, attempt to resolve this and avoid saturation. The function, shown in Figure 2.6, pushes the output to 0 when $z_j$ is less than 0 but keeps the value when $z_j$ is greater than or equal to 0:

$$\sigma_{ReLU}(z_j) = \max(0, z_j) \tag{2.9}$$

To increase the range of the ReLU function, the leaky ReLU function [97] is an extension of the original function to allow for small negative activation values and further prevent saturation.

### Training Objectives

Training objectives for neural networks can be defined as optimisation strategies that aim to minimise a defined loss function. Training objectives measure the disparity between the network's predicted output and the expected ground truth value. In order to update the network parameters to train the network to output the target output, the output of the network $\hat{y}_i$ is compared to the target output $y_i$ for the $i^{th}$ label, where $i \in \{1, 2, ..., N\}$. Through

iterative optimisation, cost functions $\mathcal{L}$ are used to tune the network parameters through algorithms, such as backpropogation, where the weights and biases are adjusted to reduce the cost function.

The Mean Squared Error (MSE) function $\mathcal{L}_{MSE}$ measures the loss for linear regression. In order to minimise the total average error, all outputs of the network $\hat{y}_i$ are taken and compared to true labels $y_i$, shown in Equation 2.10:

$$\mathcal{L}_{MSE}(\hat{y},y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{2.10}$$

MSE measures the average squared difference between the network predicted and true values for continuous outputs, therefore it is not commonly used as a training objective for speech recognition tasks, which typically involves discrete outputs, such as characters, phonemes or words.

The Cross-Entropy (CE) loss function is typically used for speech recognition applications as it can be used to measure the dissimilarity between the predicted $\hat{y}_i$ and true probability distributions $y_i$ over speech units. The CE loss function $\mathrm{L}_{CE}$, shown in Equation 2.11, can be defined as:

$$\mathcal{L}_{CE}(\hat{y},y) = -\sum_{i}^{N} y_i \, log(\hat{y}_i) \tag{2.11}$$

By using a criterion such as CE, the parameters of an ASR model are trained to maximise the log probability of the actual output sequence [98].

Scheduled sampling [99] is often used in conjunction with CE optimisation in ASR models to alleviate overfitting issues caused by the subsequent decoding strategies. After the initial few epochs at each decoder step, the prediction from the ASR model is used as the next label with a specified probability.

CE is similar to Kullback-Leibler (KL) divergence $\mathcal{L}_{KL}$, shown in Equation 2.12, which measures the the difference between two distributions over the input $x_i$, where $Q$ is the predicted probability with density $q(x)$ and $P$ is the target value with $p(x)$:

$$\mathcal{L}_{KL}(P|Q) = \sum_{i=1}^{N} p(x_i) log(\frac{p(x_i)}{q(x_i)}) \tag{2.12}$$

The type of objective function selected for a neural network is dependent upon the interpretation of the output either being a probability distribution or a specific value.

### Backpropogation

Back-propagation [100, 101] is used to derive the gradient to adjust the weights with respect to reducing the cost function. As the network is trained, the objective is to drive the cost function as low as possible, so typically gradient descent is used. To find the values of the new network parameters $\theta_i$ that minimise the cost function, gradient descent is defined as:

$$\theta_i = \theta_i - \gamma \frac{\partial}{\partial \theta_i} \mathcal{L}(\theta) \tag{2.13}$$

where $\frac{\partial}{\partial \theta_i} \mathcal{L}(\theta)$ is the gradient scaled by the learning rate $\gamma$. The learning rate is a chosen scalar value that is small enough to avoid overfitting [102]. In many practical approaches, the learning rate is initially set higher and is reduced to an optimal solution using a decay technique, such as [103, 104]. In order to back-propagate from the output layer to the hidden layers, the derivatives of the loss function with regard to the output are computed; the derivatives of the activation function of the output layer are computed and the input of the hidden layer with regard to the weight of the output layer are computed and the gradient can thus be deduced using the chain rule. The new weights are obtained by taking the old weight values within each layer and subtracting the gradient values. Once the gradient becomes smaller than a designated threshold, training is stopped as the network is considered converged.

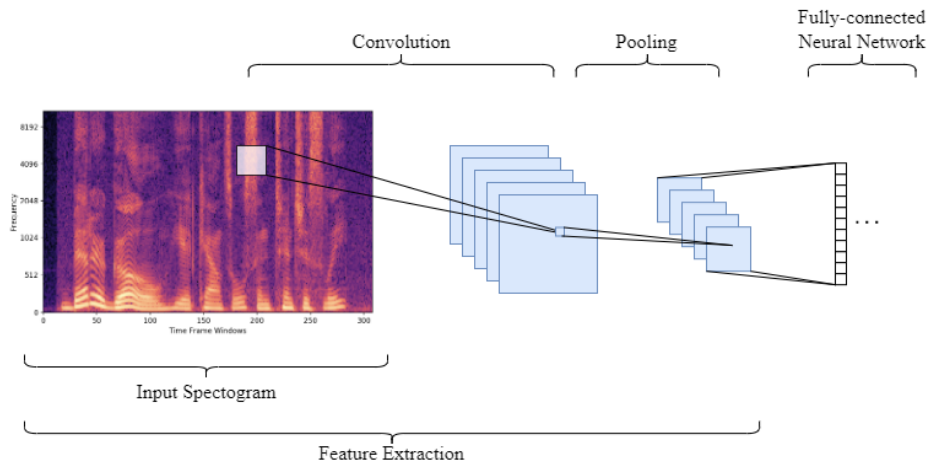



Fig. 2.7 Diagram showing high-level architecture of a convolutional neural network (CNN) for speech recognition

### 2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) first introduced in [105–107] were initially used for image recognition tasks [108]. While feed-forward neural networks connect all neurons between adjacent layers, CNNs employ local receptive fields and shared weight parameters to enable the extraction of spatial and temporal information. CNNs process the input with a grid-like structure, while preserving the spatial relationship. As shown in Figure 2.7, for speech applications, feature representations from windows over the input speech spectogram can be learned. Typical CNN models consist of alternating layers of convolutions and pooling layers with a final fully connected layer. Pooling layers downsample the features to preserve salient information. Batch normalisation and dropouts are added between layers to optimise model performance [109]. Each convolutional layer is comprised of a set of convolutional kernels, which divide the input into windowed sections to aid feature extraction, referred to as the receptive field. As the kernel is moved in steps (strides) across the width and height of the window representation of the input, an activation map is produced. The kernel acts as a filter so when matrix multiplication is computed, fewer parameters are required to represent the “most meaningful” information. For each activation map, the convolution neurons are

constrained to the same weight set, which contributes to the CNN having the property of equivariance to translation. When the input is shifted, the features are preserved but their locations in the activation maps are correspondingly shifted, allowing the CNN to capture local spatial features regardless of their position in the input.

The convolution is derived by two functions $f[x]$ and $g[x]$ over a continuous variable $x$ to produce another function $y[t]$ that describes the amount of overlap that shifts one function to the other. Over an infinite interval, the convolution can be described as:

$$f(x) * gx = \int_{-\infty}^{+\infty} f(\tau) \times g(x-\tau) d\tau \tag{2.14}$$

where $\tau$ refers to the continuous time step, $\times$ is the ordinary multiplication and $*$ is the convolution operator. Where $t$ is discrete time and $f$ and $g$ are functions, the convolution of $g$ over $f$ can be described as:

$$y[x] = f[x] * g[x] = \sum_{t=-\infty}^{+\infty} f[x] \times g[x-t] \tag{2.15}$$

This represents the discrete convolution operation for higher dimensions whereby the input and filters are multi-dimensional arrays. As they are computed with regard to frequency and time, this shows how feature maps are computed to capture relationships between the input and filters.

### 2.3.3 Recurrent Neural Networks

There are several extensions of neural networks such as Recurrent neural networks (RNNs) [110, 100] and Long Short-Term Memory networks (LSTMs) [111]. When combined with adapted ASR modelling techniques, these approaches have been shown to significantly reduce recognition error rates [112, 113, 91]. These ASR specific approaches are described in Section 2.4.

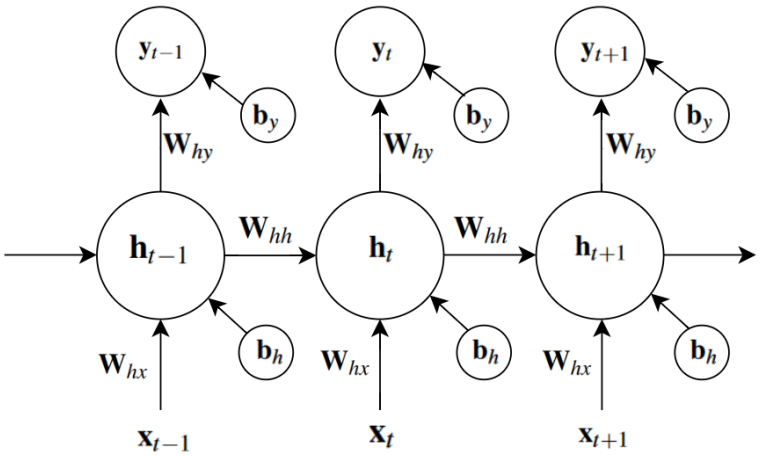



Fig. 2.8 Diagram showing a recurrent neural network (RNN) hidden layer and output layer

RNNs have similar components to a feed forward network and consist of an input layer, output layer and multiple hidden layers. Unlike feed forward networks, the hidden layers of the RNN are calculated from the input values and previous time-step values and weights, shown in Figure 2.8. $\mathbf{x}_t$ represents the input at the current time-step $t$, while $\mathbf{x}_{t-1}$ represents the input at the previous time-step and $\mathbf{x}_{t+1}$ the next time-step. Equation 2.16 describes the hidden layer $\mathbf{h}$:

$$\mathbf{h}_t = \sigma(\mathbf{W}_{hx}\mathbf{x}_t + \mathbf{h}_{t-1}\mathbf{W}_{hh} + b_h) \tag{2.16}$$

where $\mathbf{W}_{hh} \in \mathcal{R}^{F\times F}$ are the weights between neurons within the hidden layer, $\mathbf{W}_{hx} \in \mathcal{R}^{F\times F}$ are the weights for the inputs to the hidden layer, $\mathbf{b}_h \in \mathcal{R}^{1\times F}$ is the bias and $\sigma$ is the activation function. $F$ refers to the feature dimension of the input signal $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}, \mathbf{x}_t \in \mathcal{R}^{1\times F}$.

The output layer $\mathbf{y}_t$ can be defined as:

$$\mathbf{y}_t = \mathbf{h}_t\mathbf{W}_{hy} + b_y \tag{2.17}$$

where $\mathbf{W}_{hy} \in \mathcal{R}^{F\times F}$ are the weights and $b_y$ the biases of the output layer.

The recurrent structure of the network is useful to capture longer-term contextual information from the input signal with respect to time, which is suited for continuous modelling applications. However they can become computationally unstable with longer sequences, which can impact the performance of a model for applications such as speech recognition. This is caused by the "vanishing/exploding gradient problem", discovered in [114], which is due to the propogation of low gradients through time when calculating gradient descent. As the gradient becomes too small or too big, it can become more difficult to train the weights throughout the whole network as the gradient diminishes or explodes exponentially. To address this, it is possible to place an arbitrary threshold on the gradient and to initialise the network to reduce the potential for exploding or vanishing gradients. This led to the development of a related approach to the RNN, the LSTM.

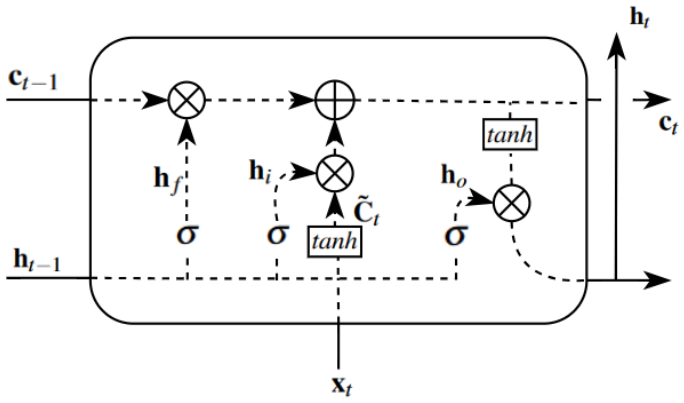



Fig. 2.9 Diagram showing the composition of a long short-term memory network (LSTM)

LSTM modelling aimed to alleviate some of the limitations of RNNs with the introduction of memory cells. The memory cells capture information about the training data by separating the hidden states into long-term and short-term. This is referred to as 3 gates: an input gate; forget gate and output gate, which are constructed with sigmoid functions. The forget gate $\mathbf{h}_f$, shown on the left side of Figure 2.9, similar to the RNN hidden layer, is described by the hidden state at the previous time-step $\mathbf{h}_{t-1}$, the input $\mathbf{x}_t$ at time-step $t$, the weight vectors for

the current layer $\mathbf{W}$, and the sigmoid activation function $\sigma$:

$$\mathbf{h}_f = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{2.18}$$

Parallel to this, the input gate $\mathbf{h}_i$, shown in the middle section of Figure 2.9, determines the information to store in the cell state, which can be described by:

$$\mathbf{h}_i = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{2.19}$$

The memory cell candidates $\tilde{\mathbf{C}}_t$ to add to the cell state can be described by:

$$\tilde{\mathbf{C}}_t = \sigma_{tanh}(\mathbf{W}\mathbf{x}_t + \mathbf{W}h_{t-1}) \tag{2.20}$$

where the activation function $\sigma$ is typically the *tanh* function. To identify information to forget from the cell state, the previous state $\mathbf{c}_{t-1}$ updates the new state $\mathbf{c}_t$ by multiplying the previous state with the forget gate $\mathbf{h}_f$ and the input gate $\mathbf{h}_i$ with the potential candidate values:

$$\mathbf{c}_t = \mathbf{h}_f * \mathbf{c}_{t-1} + \mathbf{h}_i * \tilde{\mathbf{C}}_t \tag{2.21}$$

The output gate $\mathbf{h}_o$, shown on the right side of Figure 2.9, is described by:

$$\mathbf{h}_o = \sigma(\mathbf{W}x_t + \mathbf{W}h_{t-1}) \tag{2.22}$$

which is then multiplied to the candidate values $\mathbf{c}_t$ to determine the output of the neural layer $\mathbf{h_t}$:

$$\mathbf{h}_t = \mathbf{h_o} \cdot \sigma_{tanh}(\mathbf{c}_t) \tag{2.23}$$

LSTM networks can be compiled to be uni-directional or bi-directional. Bi-directional LSTMs (BLSTMs) [113] allow the input to flow from both directions by adding an additional

layer to handle the reverse information flow. The outputs of the layers can be combined by concatenation or summation. Recognition performance of the BLSTM model improves when using future context frames, which is commonly done frame-wise, taking the output from a lower layer as the input to the current frame. The BSLTM model for speech recognition is described with diagrams in Chapter 3 Section 3.4.

Hardware acceleration for training recurrent-based systems can become challenging as as the sequence needs to be input sequentially, given that the previous state input is required to compute the current state. This can lead to incompatibilities with memory bandwidth and training parallelisation. Therefore LSTM-based models have become less popular for state-of-the-art modelling research in recent years due to the development of convolutional models [108, 11] and transformer models [1], discussed further in the next sections.

### 2.3.4 Transformer Networks

Published in [1] and previously formulised in [115], the transformer model does not use the time-steps for sequential data in order. The diagram 2.10 shows the published overview of the transformer, which introduces the concept of an encoder-decoder network. The following section will define and explain the components of this model and the attention mechanism, which are the current state-of-the-art modelling approaches for speech modelling. The encoder-decoder network is explained in further detail in Chapter 3 Section 3.4.

The input signal is converted into latent space representations and a positional encoder is used to provide the sequential context:

$$PE_{k,2i} = sin\frac{k}{10000^{\frac{2i}{d}}} \tag{2.24}$$

$$PE_{k,2i+1} = cos\frac{k}{10000^{\frac{2i}{d}}} \tag{2.25}$$

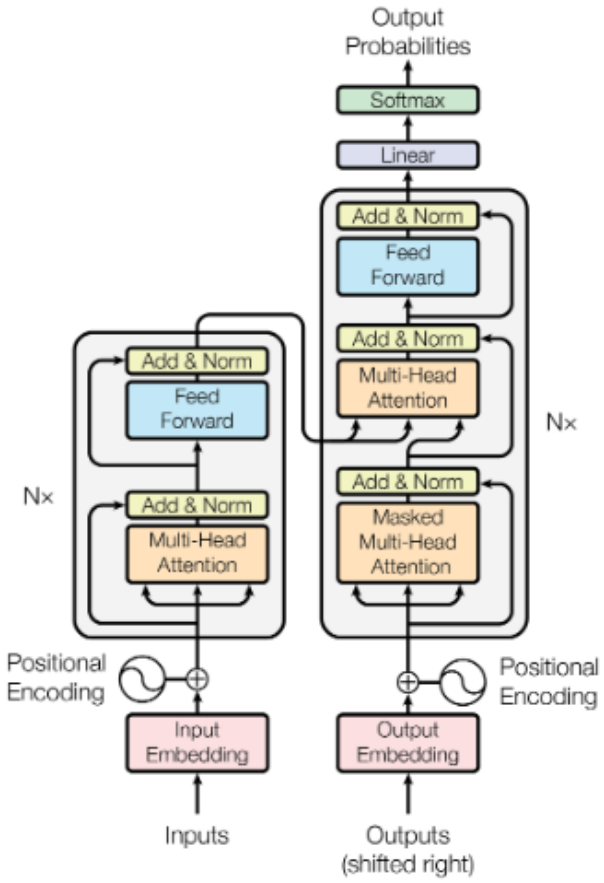



Fig. 2.10 Diagram showing the transformer model architecture from [1]

where the positional encoding function $PE$ for an input at the $k^{th}$ position in the input sequence and the $i^{th}$ index vector to map to the sine ($sin$) and cosine ($cos$) functions. The positional encoding, proposed in [1], is based upon a decaying sinusoidal function and allows the network to learn where each element of the sequence came from despite not having any recurrent connections.

A combination of the input embedding and the positional vector encoding passes to an encoder block, the left side of Figure 2.10, where it is fed to a self-attention layer. The input to this attention layer is comprised of 3 inputs referred to as query **Q**, key **K** and value **V**. The input is passed through 3 separate linear layers with independent weights to form **Q**, **K** and **V**. The attention is performed in parallel with each attention head $\mathbf{c}_{i,j}$ described by:

$$\mathbf{c}_{i,j} = softmax(\frac{\mathbf{Q},\mathbf{K}}{\sqrt{d_k}})\mathbf{V} \tag{2.26}$$

where $\sqrt{d_k}$ is a scaling factor and denotes the embedding size. A scaling factor is used to prevent the dot-product operation becoming too large in magnitude. Figure 2.11 illustrates the attention mechanism whereby elements of the input $\mathbf{x}_t$ are selectively weighted to adjust their influence upon the hidden states of the next layers. The attention weights are determined by the dot-product operation of the keys **K** and the queries **Q**. The softmax function ensures the attention weights do not exceed the value of 1. The values **V** are summed according to the attention weights.

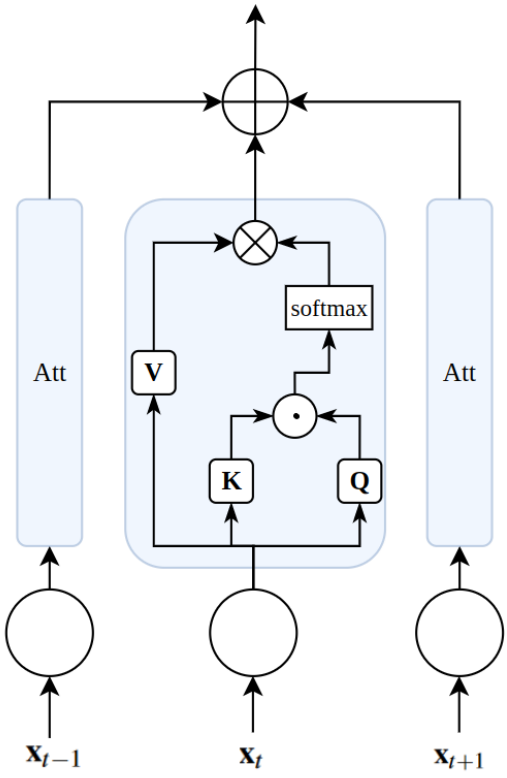



Fig. 2.11 The composition of the transformer attention layer from [1]

The self-attention output is passed to a feed-forward layer which propagates the output to the next encoder layer. Both the attention and feed-forward layers have skip-connections followed by layer normalisation. The output of the encoder is an encoded representation for

each point of the input sequence with attention scores. For each encoder layer, the attention scores are accumulated into the representation. The self-attention mechanism is aiming to solve the relevance of the $i_{th}$ window of the input sequence compared to the rest of the sequence and provide context information. The produced attention vector is aiming to encode the relationship between the windows and the sequence. Each attention process is referred to as an attention head, which occurs in parallel. Transformers are able to capture global dependencies from the input as the entire context is visible to the attention mechanism.

The output of the encoder passes to the each decoder layer in the decoder block, shown in the right side of Figure 2.10. The decoder layer shares a similar structure to the encoder, however the initial self-attention layer receives input from the previous decoder and is forced to attend to only the previous positions in the sequence by attention masking. This masked attention hides the future states of the output sequence from the model to allow learning to occur. A second attention layer, referred to as an encoder-decoder attention layer concatenates the output of the encoder and the previous self-attention layer, only using the values and keys from the encoder and the queries from the previous layer. This is followed by a final feed-forward layer. Layer normalisation and skip connections are between every stage as similar to the encoder. To train the transformer, CE loss is typically used to compute the gradients and update the parameter weights via back-propagation. The encoder-decoder transformer model for speech recognition is described with further diagrams in Chapter 3 Section 3.4.

For speech applications, the self-attention mechanism enables localisation of phonetic and linguistic information. However the main limitations of utilising the transformer model for speech modelling is that the self-attention calculations scale quadratically with the length of the sequence. Some subsequent research, such as [116], has modified the computation of the attention mechanism by assuming the information in the attention matrix can be

approximated without a significant effect on the recognition performance, allowing this approach to be more widely adopted for speech modelling.

## 2.4 End-to-End ASR Modelling Approaches

The term, "End-to-End", in the domain of speech-to-text ASR, typically describes models that have the ability to transcribe words or characters directly from an utterance of speech in one model. This approach aims to remove the previous complex ASR system decomposition where the framework is split into separately trained modules. In traditional ASR models, modules are trained independently due to their complexity, computational constraints, inflexibility, performance dependencies and time constraints, however, optimising in this strategy may not guarantee the global optimum. One of the main advantages of developing an End-to-End system is to model the label posterior probabilities and thereby the output sequence, directly from input speech using a single objective. The objective function is used to optimise the entire system, without having to train each module individually. In more recent years, End-to-End approaches have achieved performance parity with traditional hybrid approaches [117, 118]. These approaches are of particular use for on-device and streaming models in industrial applications due to their improved latency performance, reduced model size and that they do not rely on domain expertise to compile [119, 120]. Traditional modular approaches for ASR are still advantageous in applications where model adaption, robustness and recognition performance are critical. There are currently three main types of End-to-End approach that are widely used and customised based on the application: CTC-based models Figure 2.12, attention-based encoder-decoder models Figure 2.13, and recurrent transducers Figure 2.15. These approaches are described in further details in the following sections.

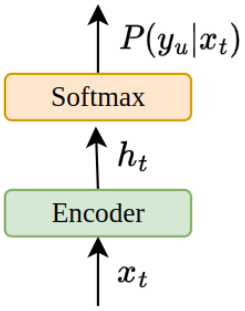



Fig. 2.12 Overview of CTC-based model for End-to-End ASR

### 2.4.1 CTC Models

[121] proposed the Connectionist Temporal Classification (CTC) algorithm for ASR decoding with RNNs, which is a discriminative solution to map the input sequence to output label sequence. The network outputs are transformed into a conditional probability distribution over possible label sequences, which removes the need to explicitly define the alignment between the acoustic features and label boundaries. The number of units in the output layer of the network is equal to the number of labels plus one extra for the the probability of observing no label, referred to as a blank token. The conditionally independent outputs are determined into an alignment of network output probabilities for a set of possible labels. The alignments are the same length as the input sequence, where repeats of labels between blank tokens are removed before removing the blank token.

As shown in Figure 2.12, the output of the CTC model is a probability distribution of observing a label $y_u$ given the input $\mathbf{x}$ at time-step $t$. The alignment between an RNN encoder output $\mathbf{h}_t$ and $y_u$, referred to as $\pi$, is defined as a sequence of labels of length $T$. The alignment probability is $p(\pi|\mathbf{x}) = \sum_{t=1}^{T} y_u^{\pi_t}$. There is an assumption that the length of the target output is equal to or less than the length of the input. Given an input $\mathbf{x}_t$ and the ground truth label $y_u$, the objective function can be described as:

$$\mathcal{L}(\mathbf{x}_t, Y') = \sum_{\pi \in Y'} p(\pi|\mathbf{x}_t) \tag{2.27}$$

where $Y'$ is the set of labels including the blank token. This is computed using the forward-backward algorithm as described in [121]. Once the most likely encoded sequence has been predicted by the decoding search algorithm, the duplicate characters and blank tokens are then removed, leaving the system output.

Computing the cost function with the blank token allows for alignment where the input and output sequence lengths vary, however this is limited by the length of the input and is inherently conditionally independent. Alignments in purely CTC models are monotonic and cannot utilise contextual information, however some recent strategies have attempted to overcome this [122, 123] by using a wider range of context dependent symbols and masking methods. The recognition performance of models that only decode with CTC is limited when compared with other techniques. However the performance of CTC-based models can also be improved when used in conjunction with an attention mechanism, known as a hybrid approach, which is described in more technical detail in Section 2.4.2. Work in [67] added contextual dependencies to an LSTM CTC-based model for an English to French translation task. This approach introduced LSTMs in an encoder-decoder structure. The encoder provides a representation of embeddings from the speech input and then the decoder produces outputs per time-step.

To compare which strategies for ASR encoding performed best for acoustic and language modelling in CTC-based approaches, [124] extended the encoder-decoder approach for conversational speech data. By combining the LSTM from [67] with an RNN model the vanishing gradient issue is alleviated whilst retaining model performance and allowing for increased model sizes, without causing overfitting. This modelling strategy suited recognition of conversational, noisy, open-domain tasks due to the ability to increase model resolution; however the evaluation method lacked consistency and was measured by attempting to

determine the quality of simulated conversations. The Deep Speech approach from [125] uses a combination of CNN, bi-directional RNN with CTC decoding and an n-gram LM. The aim of this approach was to show that it was possible to have a scalable solution that is trainable "End-to-End" although it was noted that this type of model requires large amounts of training data and pre-trained LMs in order to improve performance. The approach taken in [126] aimed to move away from modelling approaches similar to Deep Speech, and remove the requirement for pretraining and forcing alignments. The approach in [126] uses lattice-free maximum mutual information [57], which is derived and modelled by full-left-biphones in a dependency tree. Using a CNN for front-end processing, trained with lattice-free maximum mutual information and rescored with an RNN LM during decoding; the network is able to extract phoneme, semantic and syntax information from various layers. By identifying which structures of the network contain dependent representations, it is possible to integrate additional techniques to exploit specific contextual information, such as attention smoothing [127] or local normalisation [122].

The stand-alone CTC algorithm is used to eliminate traditional ASR model complexity regarding alignments between acoustic modelling and decoding. The neural network doesn't need to have the defined alignment mapping between the output sequence and the target sequence. However this approach does not allow for simple integration of language modelling techniques due to the conditional independence assumptions between the output labels. Some strategies have attempted to use different networks to exploit the contextual information from neural representations within the neural network layers to improve recognition performance without changing the CTC objective function, which has inherently added some complexity into the network that it may not fully benefit from, shown by limited performance improvements.

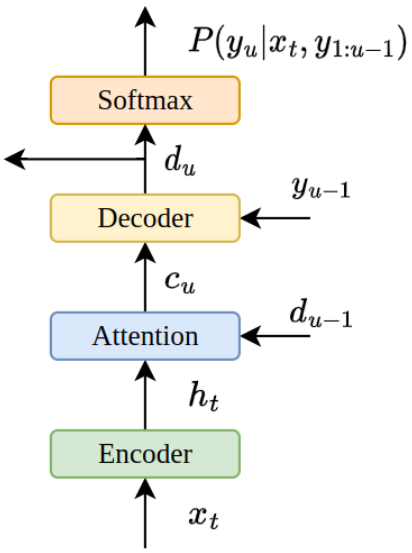



Fig. 2.13 Architecture of attention-based encoder-decoder model for End-to-End ASR

### 2.4.2 Attention-Based Encoder-Decoder Models

The encoder-decoder approach to sequence modelling attempts to provide the mapping between an input sequence $\mathbf{x}_t$ and output label sequence $\mathbf{y}_{1:u-1}$ in three distinct stages. As shown in Figure 2.13, the first stage is the encoder which encodes the input sequence $\mathbf{x}_t$ and propagates it into hidden states $\mathbf{h}_t$, often with residual connections. The attention mechanism computes the frame-wise attention weights $\mathbf{c}_u$ between the encoder hidden state and the previous decoder output $\mathbf{d}_{u-1}$, where $u$ is the output label index, to generate the context vector. The context vector is utilised as the initial hidden state for the decoder. The final stage is the decoder, which functions to predict an output label for each time-step $P(y_u|\mathbf{x}_t, \mathbf{y}_{1:u-1})$, by taking the previous output labels $\mathbf{y}_{1:u-1}$ with the context vector, described by Equation 2.28.

$$P(Y|X) = \prod_u P(y_u|\mathbf{x}_t, \mathbf{y}_{1:u-1}) \tag{2.28}$$

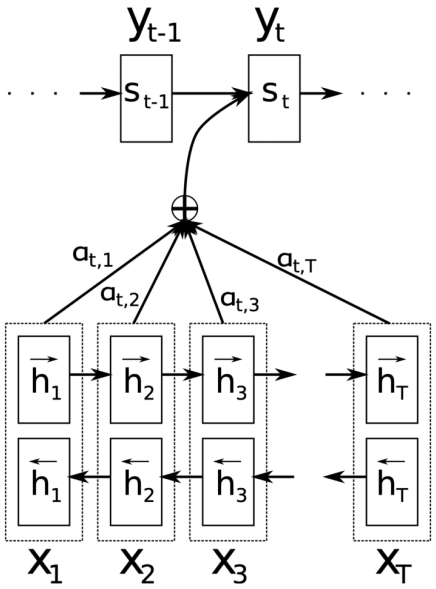



Fig. 2.14 The Bahdanau attention mechanism from [2]

This technique allows for unrestricted differences in length of the input and output sequence, however it can suffer from "bottle-neck" problems due to the attempt to capture a large amount of information in the embedding vector [2]. 'Bottle-necks" can also occur due to changes in dimensionality between network layers. [2] proposed that attention alleviates this and the vanishing gradient issue by finding the weighted sum of the hidden states in the encoder and feeding this to the decoder. As shown in Figure 2.14, the weighted sum of the hidden states $\mathbf{h}_t$ are described by attention weights $\alpha_{t,1:T}$, which are fed to the decoder state $s_t$. This was initially done for translation tasks as the attention mechanism inherently takes advantage of more relevant words that appear at similar places for each language. To further improve the alignment between the input speech sequence and the output labels for the ASR domain, the attention-based encoder-decoder model can be optimised with a CTC model by sharing the encoder [117]. This is referred to as a hybrid model as it also allows the scoring to be combined for decoding [128].

Modelling attention mechanisms for recurrent-based End-to-End system solutions is a challenging task as the mechanism needs to wait for the entire sequence to be encoded

before alignment and decoding can begin. Empirical results from [4] showed that attention mechanisms can cause a model to perform poorly when data is corrupted by noise, causing attention-based models to struggle in real-world environment recognition tasks. [129] showed that an attention-based ASR model could be separated into two sub-modules whilst still being trained End-to-End. This approach consists of an acoustic module referred to as the "Listener". This is a BLSTM that takes filter banks as inputs. The pyramidal structure of the BLSTM is a stack on three BLSTMs that successively decrease in resolution in an attempt to provide higher resolution embeddings and reduce the computational complexity of the model. The "Speller" section of the model is an attention-based RNN and decodes the output of the "Listener" to output characters.

The transformer, described in Section 2.3.4, introduced by [1], uses a different attention-based strategy with an encoder-decoder structure. A query (the previous hidden state) and key-value pairs (encoder hidden states) are mapped to outputs, which is referred to as "Scaled Dot-Product attention". This approach attempted to resolve the "bottleneck" problem as the query is computed from the previous decoder layer and the keys come from the output of the encoder, similar to the mechanism proposed in [2]. Self-attention layers were introduced into the encoder architecture where all the keys, queries and outputs provide a path to all positions in the encoder and previous layers of the encoder. Finally, is the combination, referred to as "Multi-Head attention", where the query and key value pairs are projected into lower dimensions, each with projected attention computation as a "head". Each "head" is then concatenated and transformed then added and normalised to the next layer.

Different network structures for the attention layers were explored in [130] such as Multi-Head Scaled Dot-Product and Position-Wise feed forward layers. It was claimed that stacks of self-attention layers are capable of automatically learning the gradual increase of the receptive field that CTC models are hard-coded with, and using fewer "heads" with restricted context can enable faster network convergence. The transformer model was first compiled for

ASR in [131] by integrating CNN layers to view the entire input sequence at once and model the interactions between speech features at further distances in time before propagating to the transformer layers. As CNNs require less complex front-end processing [132, 133] than transformers and BLSTMs, they are able to mitigate the mismatch in length of the generated feature vector and the target output sequence. This also enables the integration of deeper encoder layers to extract context dependencies using an attention mechanism that attends to the time and frequency axes. Using larger models with increased parameter size has been shown to reduce training time when combined with less computationally complex front-end processing techniques [131, 134], which is advantageous in the development of End-to-End approaches. Also while the transformer-based approaches for ASR are capable of modelling global context across longer ranges, it is difficult to capture local features. The Conformer, proposed in [133], attempts to improve the local context modelling capabilities of the system by using convolutional layers to exploit local information, replacing the encoder blocks of the transformer network. The Conformer blocks are able to learn shared kernel representations over a local window and when this is combined with multi-headed self-attention, position-wise local features and global context are able to be learned. The main drawback of this approach is scalability as recognition performance reduces with smaller models that still have over 10 million parameters and 16 encoder layers.

Typically, models that are to be used in a commercial setting, require streaming capabilities with lower latency, in order to provide recognition at roughly the same time as the speaker [135]. In order to enable streaming with attention-based encoder-decoder models, chunk-wise attention can be applied to the input speech [136, 137], where the encoder is adapted to receive the input sequence in blocks whilst retaining global feature information. However this method still requires a defined activation threshold for streaming applications. In order to stream using a transformer-based approach, the model needs to be adapted to process the input sequence sequentially. [138] developed an extension to the Transducer

approach, described in further detail in Chapter 2.4.3, in conjunction with the transformer, with time-restricted self-attention to limit the context window. The *transformer-transducer* approach also facilitated output probabilities based on both the sequential audio input and also the predicted labels. This approach essentially replaced the recurrent blocks with transformer blocks and reached the current state-of-the-art ASR performance across several datasets. The Emformer, proposed in [139], aimed to reduce the need for limiting context for improved computation efficiency, caused by the self-attention computation that scales quadratically with left-context size. This is done by utilising augmented memory, whereby the keys and values are cached from previous layers for the left-context in combination with GPU parallelisation.

Transformer-based approaches have also been developed for unsupervised ASR, whereby the audio representations are trained with unlabeled data [140, 141]. The approach by [140] attempts to degrade the input spectogram by using time-masking and then train the model to restore the input. While [141] instead explored learning latent representations of the speech input with the convolutional layers, then masking frames at the transformer layer. The approach from [141] used a context network to generate representations to solve a self-supervised prediction task and train the model whilst evaluating on itself.

### 2.4.3 Recurrent Neural Network-Transducers

The RNN-Transducer (RNN-T), published by [142], was initially developed as an attempt to move away from the traditional modelling approaches of the time. Considering the limitations of modelling with only CTC, as discussed in 2.4.1, the RNN-T attempts to model aligned acoustic and linguistic information within an End-to-End framework but also by adding contextual dependencies between outputs through time. This can be described by Equation 2.29 and Figure 2.15, where $\mathbf{z}_{u,t}$ is the alignment path that maps to the output sequence $\mathbf{y}_{1:u-1}$, $\mathbf{h}_t$ is the encoding from input $\mathbf{x}_t$, and $\mathbf{h}_u$ is the encoding from the previous time-step

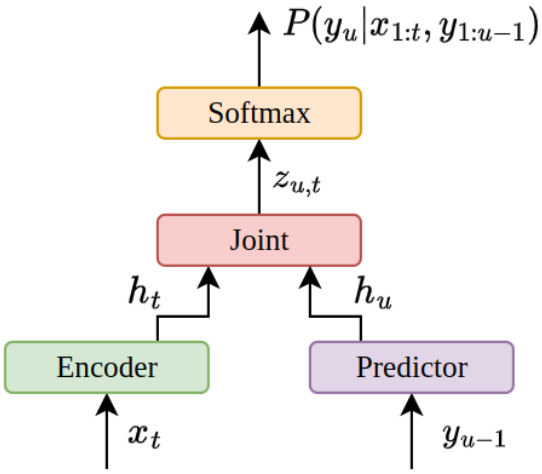



Fig. 2.15 Architecture of Recurrent Neural Network-Transducer model for End-to-End ASR

prediction $y_{u-1}$. $\mathbf{z}_{u,t}$ is calculated by an additional network, named the joint network, and decoded with a beam search algorithm.

$$p(Y|X) = \sum_{\mathbf{z}_{u,t} \in \mathrm{Y}^{-1}(y)} \prod_{t=1}^{T'} p(\mathbf{z}_{u,t}|\mathbf{h}_t \mathbf{h}_u, y_{u-1}) \tag{2.29}$$

This approach conditions the output onto the previous output tokens and input sequence, removing the conditional independence inherent with the CTC algorithm. The RNN-T can be distinguished in three parts; an encoder, prediction network and a joint network. The predictor network takes the previous outputs and produces alignments used to predict the subsequent output. The joint network is typically a feed-forward neural network that combines the vectors from the encoder and the predictor to output probability distribution over all output labels. Equation 2.30 describes the output of the joint network $\mathbf{z}_{u,t}$, which combines the outputs of the encoder $\mathbf{h}_t$ and predictor $\mathbf{h}_u$ with weight matrices $\mathbf{W}$, bias $b$ and non-linear function $\sigma$.

$$\mathbf{z}_{u,t} = \sigma(\mathbf{W}h_t + \mathbf{W}h_u + b) \tag{2.30}$$

This approach is recently being developed for on-device and streaming applications for ASR [135, 119, 143] partly due to the more compact model size of the RNN-T but also because the transcript can be produced whilst receiving the input audio, enabling low latency, real-time transcription. However, there are some remaining challenges of using RNN-Ts for on-device ASR solutions as this modelling approach delays the prediction of the output label until it is more confident, which it achieves by using more future context. Recently, there have been developments of the RNN-T model to improve training efficiency despite the large context requirements: single loop recursions [144], function merging techniques [145] and context limiting approaches [146]. The method developed in [144] enables single loop recursion for training RNN-T models with the aim of improving the computational efficiency with hardware acceleration. By vectorising the forward-backward probabilities, the hardware is able to perform multiple floating point calculations simultaneously, allowing the loop skewing transformation and recursion to be computed in a single loop. This approach is limited whereby the computational efficiency improvements are not a scalable solution and does not address modifications to the training process of the RNN-T to reduce the extensive context requirements.

Limiting the context for streaming End-to-End ASR models, [146] showed that it is not necessary to condition the entire output of the ASR model on the full history of predicted labels, that the same recognition performance can be achieved by using at least the previous 4 predicted labels. A modification to the beam search decoding was also proposed by path merging to form lattices. This was done by merging paths during decoding by approximation, for 2 paths that shared the same local history and thereby freeing space.

However, efficiency improvements and reduction in the memory constraints when training RNN-T models for ASR, the recognition performance was less competitive than the transformer-based models that were being published at the same time. Work by [145] explored improving RNN-T ASR performance benchmarks by using an auxiliary task, in this

case transcriptions for social media video transcription. One of the middle layers of the encoder is branched into an auxiliary and primary branch to provide posterior distributions over the output labels. In order to do this [145] utilised KL divergence [147] between the distributions of the primary and auxiliary branches, in combination with the RNN-T loss, with the aim of dampening the gradients of the branches and balancing the optimisation. This approach aims to distill knowledge from both the ASR and auxiliary tasks in order to improve generalisation and improve recognition performance. RNN-T models have also shown improved recognition performance from pre-training using the CTC algorithm or CE criterion [148, 149].

## 2.5 Summary

This Chapter provided a literature review regarding an overview of speech technology modelling applications in Section 2.2 and neural networks in Section 2.3. Sections 2.4.1, 2.4.2 and 2.4.3 presented and discussed End-to-End ASR and the main approaches. The CTC approach, described in Section 2.4.1, involves aligning the acoustic information of speech to phone states, then compresses the search space for conditionally independent hypothesised labels. This approach does not use any pretrained modules and is the least computationally expensive but the conditional independence potentially loses context information that improves recognition performance for ASR tasks.

The attention-based encoder-decoder approach, described in Section 2.4.2, first encodes the acoustic information into hidden states, then utilises a decoder to predict the output at each time-step. An attention mechanism computes the weights between the encoder hidden state and previous decoder output, allowing the model to capture contextual dependencies and improve recognition performance. However, the complexity of the attention-based encoder decoder approaches scales as the length of the input sequence increases and have a higher latency, making them less suitable for applications such as streaming.

The RNN-T approach, described in Section 2.4.3, also uses an encoder to encode acoustic information to hidden states but also uses a predictor to hypothesise the next output sequence. A joint network combines the encoder and predictor to output the probability distribution over the labels. The RNN-T approach has a lower latency so is more suited to streaming applications, although the recognition performance is lower compared to the attention-based encoder-decoder models.

As it is unclear exactly how the choice in modelling approach affects the recognition performance for the task of End-to-End ASR, Chapter 3 explores the performance of state-of-the-art frameworks using the evaluation metrics described in Section 2.2.5. Chapter 3 Section 3.5 attempts to assess the recognition outputs of a framework and highlight the limitations of the evaluation metric with regard to the interpretability of different performance results. Subsequently, Chapter 3, Section 3.6 describes a developed pipeline that attempts to provide some analysis regarding the relationship between the structures discussed in Sections 2.3.2, 2.3.3 and 2.3.4 and the recognition performance of the model.

# Chapter 3

# End-to-End ASR Modelling Analysis

## Contents

## 3.1 Introduction

The End-to-End approaches for ASR attempt to simplify the system and model the input features to characters, phonemes or words [150]. This approach allows the development of a complete ASR system without the requirement of expert domain knowledge, while attempting to globally optimise the training process. As the development and integration of End-to-End approaches have become increasingly popular, many different software frameworks have been developed [151–153, 62], each comprising of variations of End-to-End approaches, such as attention-based encoder-decoders, CTC models and RNN-T's, as described in Chapter 2.

End-to-End ASR frameworks use different strategies for optimising parameters, different architecture setups and training regimes, with the aim to improve performance on specific datasets or domains. It is not clear which strategy should be utilised for specific applications or how the parameters or architecture can be adapted to domains and improve recognition performance. End-to-End architectures have inherently complex internal dynamics and it is imperative that the model learns to generalise from the training process, in order to yield recognition performance improvements [154, 155]. Research from [156] showed that neural layer depth can attribute to a richer neural representational capacity, but it is still unclear whether the models were generalising or memorising the training data [157]. The hypothesis, that increasing neural layer depth increases the richness of representations, does not always translate to performance improvements in all cases [158] and has had little exploration in the End-to-End ASR domain.

The aims of this Chapter are to explore current state-of-the-art frameworks and to evaluate their performance on recognition of conversational speech, which is a non-trivial task. Several methods of empirical analysis are introduced in an attempt to assess the performance of End-to-End ASR frameworks and observe any identifiable patterns in the produced errors. The identification of error patterns in the modelling approach can aid the development and adaptation of improved models. As little research has been conducted regarding the interactions between the training of End-to-End models and speech data, it is unclear how the dependencies within the model contributes to optimal representations to improve recognition performance. Statistical analysis techniques are also utilised to observe these relationships and provide information regarding the interpretability of End-to-End models.

There are many factors within the data that can impact ASR model performance, including pronunciation, physiology of the speaker [159], infrequent words, word length [160] and faster or slower speech [161]. It is possible to compare the model's hypothesised output with the ground-truth, to observe the performance of an End-to-End ASR model. Potentially there may be indicators, such as high error rates across shorter words, which could signify the model is unable to interpret specific cases. If these cases could be identified by analysis of the recognition outputs, modifications to the architecture could be made to target these errors. The subsequent experiments were developed to assess the performance of state-of-the-art End-to-End ASR approaches for conversational speech recognition, as this is a challenging task with high error rates. Analysing the recognition outputs of the models using several different metrics may aid the interpretation of indicators that could be used to develop and improve modelling approaches.

## 3.2 Data

The Switchboard corpus [6] consists of approximately 260 hours of conversational telephone speech between 2 speakers. There are around 2400 casual conversations between 543

speakers, of which 302 are male and 241 are female. An automated handling system provided recorded prompts, introduced roughly 70 topics for discussion, dialed the speakers and recorded the conversations.

The test sets, referred to as *Swbd* and *Callhome*, are derived from the HUB5'00 corpus [5] and contain 20 unreleased telephone conversations from *Swbd* consisting of 2.1 hours of speech and 20 telephone unscripted conversations from *Callhome* consisting of 1.6 hours of speech. conversation on a topic that was selected by a robot operator (Switchboard data), and a set of unscripted telephone conversations between family members.

The Fisher Corpus transcripts [9] consist of time-aligned transcripts from 2000 hours of English conversational telephone speech, similar in format to Switchboard [6]. Speakers made up to 3 10 minute telephone calls, where they were paired with another unknown speaker to discuss an assigned topic. This was in an attempt to maximise inter-speaker variability and vocabulary while retaining a high level of formality. The collection contains various dialects and accents, primarily US-English but also a small amount of non-US English, Canadian English and foreign-accented English.

## 3.3 End-to-End ASR Frameworks Performance

Table 3.1 WER % Comparison of End-to-End Frameworks on HUB5'00 Corpus [5]

| Method | Framework | Swbd | Callhome |
|---|---|---|---|
| CTC | RNN + CTC [162] | 14.0 | 25.3 |
| CTC+AED | CTC+Attention LSTM [163] | 13.3 | 24.4 |
| CTC | Deep Speech: DNNs+RNN+CTC [125] | 12.6 | 19.3 |
| AED | Espresso: LSTM [3] | 9.2 | 19.1 |
| Other | LF-MMI [126] | 9.3 | 18.9 |
| Other | Hybrid HMM/DNN [164] | 8.3 | 17.3 |
| RNN-T | Neural Transducer: RNN-T [165] | 8.1 | 16.4 |
| CTC+AED | ESPNet [62]: Transformer + augmentation [166] | 6.8 | 14.1 |
| Other | Humans from [167] | **5.1** | **6.8** |

This Section introduces the state-of-the-art modelling approaches and a description of the methods utilised. The methods build upon the elements outlined in Chapter 2, Section 2.4, and are specific to End-to-End ASR modelling for recognising conversational speech. As methods and their results are compared, this provides the background to motivations for selecting specific methodologies to analyse in further detail in Section 3.4.

Table 3.1 shows a comparison of the published WER performance for several End-to-End frameworks and human transcribers [167] evaluated on the HUB5'00 corpus [5]. The column method refers to Chapter 2, Section 2.4 where the main identified methods were CTC-based, attention-based encoder-decoders (AED) and RNN Transducers (RNN-T). As can be seen from the results on the Switchboard test sets, the human transcribers from the experiments in [167] were conducted by utilising 3 independent transcribers, which were quality checked by a 4th transcriber to provide the "best" human result. This human result still significantly outperforms the current state-of-the-art End-to-End ASR frameworks for recognition of conversational speech although the gap has incrementally reduced.

The Deep Speech approach, outlined in [125], is a multi-layer DNN model with recurrent layer and trained with CTC decoding. The model was trained with both the Switchboard training set [6] and Fisher dataset [9] but the approach still has higher WER on both test sets, 12.6% WER on the *Swbd* set and 19.3% WER on the *Callhome* set, especially compared to the approaches that do not use larger amounts of data for training. The framework developed in [162] used a multi-layer RNN trained with CTC decoding while replacing the blank token with some additional character dependent symbols to model output tokens by frame and compress the computation. While this approach yielded high WERs on both the *Swbd* and *Callhome* test sets, of 14.0% and 25.3% respectively, the work was motivated to establish the modelling power of a neural network without a separate decoding process or additional dictionary.

The Lattice-Free Maximum Mutual Information (LF-MMI) model [126] had a related motivation and is a similar method to the CTC-based approaches as it uses sentence-level posteriors for training a DNN while also using state-tying decision trees to model the context dependencies without external language models or dictionaries. The LF-MMI approach reaches a much better performance on both test sets of 9.3% and 18.9% WER.

The Neural Transducer developed in [165] uses the RNN-T model as described in [142], incorporating beam search decoding and rescoring with a pretrained LM to achieve improved performance. The performance of an RNN-T model was also compared without integrating a language model, which had slightly worse results of 8.5% WER compared to 8.1% on the *Swbd* test set and 17.5% WER compared to 16.4% on the *Callhome* test set. In this case, as the LM is also trained on the same training set, rescoring during decoding degrades performance slightly on the *Callhome* set and has very little impact on performance on the *Swbd* set. Compared to the recurrent modelling and CTC approaches, the RNN-T model reaches a better recognition performance overall, even without rescoring with a LM.

Several of the frameworks listed in Table 3.1 are comprised of attention-based Encoder-Decoders, such as [163, 3] and [62]. The Espresso framework, links research from FAIRSEQ [168] with Kaldi software [169], allowing extensions of PyTorch-based modelling [170]. The LSTM attention-based encoder decoder structure used for this model is from [171] which utilises stacked convolutional layers with BLSTM layers on the output channels. The decoder consists of LSTM layers with Bahdanau attention [2].

As the LSTM attention-based encoder-decoder approach is computationally expensive and can take a long time to train, [163] attempted to modify CTC loss for multitask training within the attention-based encoder-decoder model. This approach combined the attention loss with CTC loss to enable joint training, while also exploring variations of loss functions evaluated on the same task. They found that training with additional CTC loss on the encoder is able to slightly improve the acoustic model.

The ESPNet [62] framework includes a similar hybrid approach that is able to achieve much lower error rates on both conversational speech test sets. The framework combines several modular strategies for End-to-End speech recognition to enable the development of open research tasks. The approach in ESPNet uses a hybrid approach of attention combined with CTC, where the attention mechanism learns the alignments of the speech frames and output characters, while the CTC algorithm handles the sequential problems.

Comparing the performance of the End-to-End frameworks with a non-End-to-End ASR system [164], which implemented a HMM/DNN trained with frame-wise CE and beam search decoding, only the approach in ESPNet achieves the lowest error rates on both test sets. The DNN is used to predict the probabilities of the given speech frame, while the HMM minimises the CE loss between the ground-truth label and the model prediction. The HMM/DNN approach was able to outperform the other frameworks across the Switchboard and Callhome test sets, however, the implementation in ESPNet in combination with the augmentation technique from [166] reaches the lowest error rate on both test sets. The augmentation technique combines time and frequency masking to augment the training data and improve recognition performance.

Furthermore, recent work from [172] uses the ESPNet model in combination with a multi-level LM technique during the decoding process, which yields the current state-of-the-art performance on the Switchboard and Callhome test sets of 5.1% and 9.5% respectively. The multi-level LM combines sub-word and word-based modelling techniques trained using the Fisher dataset transcriptions. The performance results of End-to-End ASR frameworks indicate that joint decoding with externally trained LMs can significantly improve recognition performance of ASR models for conversational speech, however this strategy moves away from the objective of End-to-End modelling. Using externally trained LMs could be considered as using external knowledge to enhance the training optimisation.

## 3.4 Experiments

The performance of different strategies for End-to-End ASR varies significantly, even within similar modelling approaches, such as the frameworks using attention-based encoder-decoders. The WER or CER metrics used to evaluate their performance does not give any indication of the causes of the higher or lower error rates, which is crucial to understand how to detect and modify approaches to improve accuracy. The following experiments analyse and compare the performance of attention-based encoder-decoder approaches as the transformer model achieves the best WER performance for conversational speech, while the LSTM has slightly worse performance. It is unclear how the variation in mechanisms affects the dependencies of the modelling approaches and thereby their resulting effect upon recognition performance. Initial experimental analysis is conducted across the outputs of an LSTM encoder-decoder model and a transformer encoder-decoder model from [3], trained with the Switchboard training set but without any LM rescoring techniques during decoding. Removing the LM rescoring techniques degrades WER from the results shown in Table 3.1. These models were chosen for analysis as they reach a low WER and the Espresso framework is modular and controllable to ensure any modifications to the software would not break the training regime.

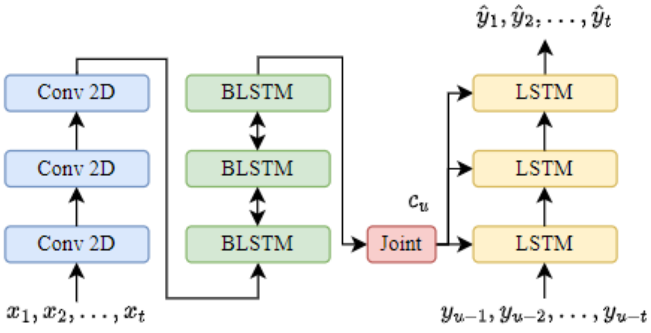



Fig. 3.1 The Espresso [3] attention-based BLSTM encoder-decoder architecture, which uses the attention mechanism from [4]

Figure 3.1 shows an overview of the LSTM encoder-decoder approach where the final layer of the BLSTM encoder layers is projected to an LSTM decoder from [2] with a context vector $c_u$ generated at each time-step with a Bahdanau attention mechanism [4]. For the following experiments, 3 encoder layers and 3 decoder layers were used for the LSTM model. A 3-layer stacked 2-dimensional CNN is utilised on top of the encoder, with a kernel size (3,3) on both the feature and time axis from [173] to provide acoustic features from the input $x_1, x_2, ..., x_t$. The output of the the decoder is the hypothesised output word sequence of the model $\hat{y}_1, \hat{y}_2, ..., \hat{y}_t$.

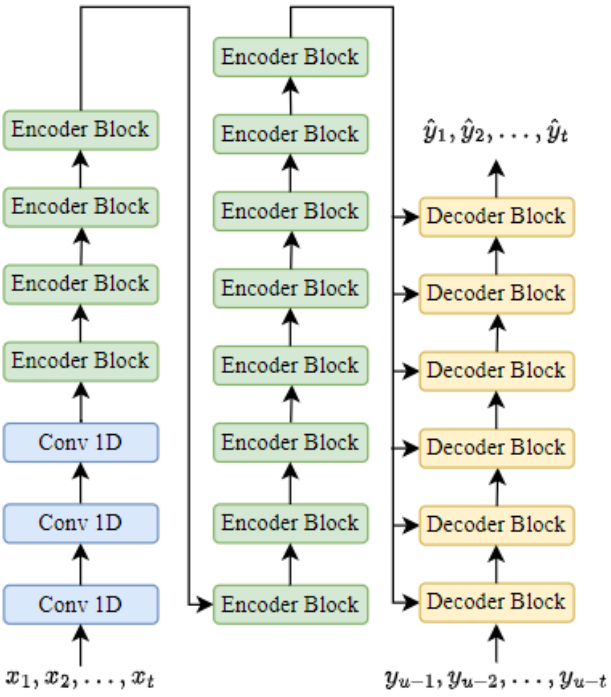



Fig. 3.2 Transformer attention-based encoder-decoder architecture from [1] compiled in Espresso [3]

With reference to Figure 2.10 and Chapter 2, Section 2.3.4, a transformer model from [1], is complied for the subsequent experiments. Figure 3.2 shows an overview of the connections between 3 convolutional layers, 12 encoder blocks and 6 decoder blocks containing identical neural dimensions. Each encoder block has a multi-head self-attention layer and a feed

forward layer, while each decoder block contains multi-head self-attention layers using the previous decoder output and also the encoder outputs, as detailed in Chapter 3, Section 2.3.4.

## 3.5 Empirical Analysis of Recognition Outputs

The following experiments attempt to observe the outputs of both the LSTM and transformer encoder-decoder models to compare and analyse the performance. It was hypothesised that by analysing the output errors of the models empirically, it would be possible to observe patterns between the approaches and thus provide evidence for potential model development. Firstly, the outputs of both models were categorised into confusion pairs, substitutions, insertions, and deletions in Section 3.5.1. Errors across speakers were also compared in Section 3.5.2 to observe whether there existed potential bias. To attempt to expand the categorical classification of errors, model outputs were also categorised across linguistic errors, such as homophones and minimal pairs in Section 3.5.3. Finally, the length of each output error was compared across models to attempt to understand whether errors increase for shorter or longer words in Section 3.5.4.

### 3.5.1 Results

Table 3.2 LSTM [2] and Transformer [1] End-to-End ASR model recognition performance on the HUB5'00 test sets [5]

| | Switchboard | | | | Callhome | | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **WER** | **Sub** | **Del** | **Ins** | **WER** | **Sub** | **Del** | **Ins** |
| LSTM | 13.3% | 9.0% | 2.7% | 1.9% | 25.2% | 17.4% | 4.3% | 3.6% |
| Transformer | 9.5% | 6.5% | 2.5% | 1.6% | 20.6% | 13.7% | 3.8% | 3.0% |

The performance results of both models are shown in 3.2. The LSTM model reached 13.3% WER on the Switchboard test set, and 25.2% WER on the Callhome test set. While the transformer model results are 9.5% WER on the Switchboard test set, and a WER of 20.6%

on the Callhome test set. The transformer model performs slightly better than the LSTM model on both test sets with lower substitution, insertion, and deletion errors. The WER for both models are considerably higher than when using a LM, which are the published results in [2] and [1], which may indicate that the LM is correcting a large proportion of model hypotheses. This hypothesis is explored further in Chapter 5.

Table 3.3 LSTM End-to-End ASR models [2] most commonly produced substitution confusion pairs on the HUB5'00 test sets [5]

| **Switchboard** | | | **Callhome** | | |
|---|---|---|---|---|---|
| **True Label** | **Hypothesis** | **Count** | **True Label** | **Hypothesis** | **Count** |
| gonna | to | 21 | in | and | 30 |
| in | and | 15 | was | is | 20 |
| was | is | 12 | him | then | 18 |
| and | in | 10 | the | a | 17 |
| then | and | 8 | he | you | 17 |
| it | that | 8 | and | in | 15 |
| that | it | 8 | it | that | 14 |
| the | a | 7 | a | the | 14 |
| a | the | 7 | i | they | 12 |
| him | them | 6 | to | a | 11 |

As shown in Table 3.3, the most common confusion pairs for both test sets using the LSTM model are predominantly short determiners, such as "it, the, a, him". Confusion pairs are defined as the hypothesised word that was erroneously predicted by the model and the target word that it was substituted for. The LSTM model returned 1228 total substitution errors for the Switchboard test set and a total of 2663 errors for the Callhome test set.

As shown in Table 3.4 the most common confusion pairs for both test sets using the transformer model are similar to the LSTM model. Errors across the Switchboard and Callhome test sets were very similar words and confusion pairs, such as where the model has confused 14 "the"s for "a" and likewise 12 "a"s for "the" on the Callhome test set.

Table 3.5 shows the most commonly substituted words errors, which are very similar for both test sets, despite the majority of the substitution errors being on the Callhome test set. The errors across models are also similar words with proportions of errors.

Table 3.4 Transformer End-to-End ASR models [1] most commonly produced substitution confusion pairs on the HUB5'00 test sets [5]

| Switchboard | | | Callhome | | |
|---|---|---|---|---|---|
| **True Label** | **Hypothesis** | **Count** | **True Label** | **Hypothesis** | **Count** |
| gonna | to | 13 | him | them | 22 |
| in | and | 11 | in | and | 19 |
| the | a | 11 | was | is | 18 |
| was | is | 10 | he | you | 15 |
| and | in | 9 | the | a | 14 |
| that | it | 8 | a | the | 12 |
| a | the | 7 | to | the | 12 |
| him | them | 5 | to | a | 11 |
| then | and | 5 | and | in | 10 |
| on | in | 4 | it | that | 10 |

Table 3.5 Most common substituted words for both End-to-End ASR models [2, 1] on the HUB5'00 test sets [5]

| LSTM | | | | Transformer | | | |
|---|---|---|---|---|---|---|---|
| Switchboard | | Callhome | | Switchboard | | Callhome | |
| **Word** | **Count** | **Word** | **Count** | **Word** | **Count** | **Word** | **Count** |
| and | 48 | is | 92 | the | 39 | is | 80 |
| that | 40 | and | 85 | is | 31 | to | 71 |
| the | 38 | to | 77 | it | 29 | a | 60 |
| gonna | 35 | in | 73 | that | 29 | and | 58 |
| in | 35 | the | 70 | and | 27 | the | 55 |
| it | 29 | i | 65 | to | 24 | in | 54 |
| to | 26 | it | 61 | a | 23 | it | 53 |
| a | 26 | he | 53 | in | 23 | i | 49 |
| is | 25 | a | 48 | yeah | 23 | he | 47 |
| was | 24 | that | 38 | gonna | 19 | you | 38 |

Table 3.6 shows the most common insertion errors, which are similar to the most commonly substituted words; predominantly short determiners. Both models had the most insertion errors on the word "i" across both test sets and have high error rates for words that are monosyllabic.

The most common deleted words for both models across the test sets are shown in Table 3.7, where there are also some instances of deleted interjections, such as the word "oh".

Table 3.6 Most common inserted words for both End-to-End ASR models [2, 1] on the HUB5'00 test sets [5]

| LSTM | | | | Transformer | | | |
|---|---|---|---|---|---|---|---|
| Switchboard | | Callhome | | Switchboard | | Callhome | |
| Word | Count | Word | Count | Word | Count | Word | Count |
| i | 20 | i | 40 | i | 26 | i | 40 |
| going | 18 | do | 38 | a | 19 | a | 27 |
| it | 17 | you | 36 | going | 16 | you | 23 |
| you | 15 | is | 32 | it | 15 | it | 22 |
| a | 14 | a | 29 | and | 14 | and | 21 |
| and | 14 | not | 24 | you | 13 | is | 20 |
| in | 13 | have | 23 | are | 10 | the | 20 |
| is | 13 | the | 23 | is | 9 | do | 19 |
| do | 12 | are | 18 | to | 9 | to | 19 |
| they | 11 | they | 17 | do | 8 | have | 18 |

Table 3.7 Most common deleted words by both End-to-End ASR models [2, 1] on the HUB5'00 test sets [5]

| LSTM | | | | Transformer | | | |
|---|---|---|---|---|---|---|---|
| Switchboard | | Callhome | | Switchboard | | Callhome | |
| Word | Count | Word | Count | Word | Count | Word | Count |
| i | 39 | i | 57 | i | 37 | i | 41 |
| a | 32 | and | 43 | it | 31 | and | 36 |
| oh | 31 | a | 32 | a | 28 | is | 31 |
| you | 22 | it | 32 | and | 22 | it | 30 |
| it | 20 | oh | 32 | is | 18 | a | 27 |
| and | 20 | is | 30 | the | 18 | to | 24 |
| the | 19 | the | 27 | you | 18 | the | 23 |
| that | 17 | to | 24 | that | 16 | oh | 22 |
| to | 15 | he | 24 | oh | 13 | you | 20 |
| is | 13 | are | 18 | are | 12 | he | 19 |

Similar to the insertion errors, the most commonly deleted word for the models is the word "i" and other monosyllabic words.

Assessing these errors in isolation, out of the context that they were predicted, it is difficult to intuitively determine the reasoning behind the errors. It would also be non-trivial to design an automatic analysis strategy to provide further context behind the errors. However, this could be done manually using external linguistic knowledge to determine an initial baseline

analysis and determine whether there are any identifiable patterns within the output errors that could be attributed to a structure within the ASR framework.

### 3.5.2 Analysis of Speaker Errors

Research from [174] found that ASR systems can bias improved recognition performance to certain speakers or groups of speakers. To alleviate potential bias, it is common practise to attempt to create a dataset with speakers of evenly distributed features, such as gender and amount of speech per speaker. The Switchboard training set contains 543 speakers, of which 302 are male and 241 are female. The Callhome test set contains 32 female and 10 male speakers, whereas the Switchboard test set contains 19 female and 21 male speakers. Due to this gender imbalance being biased towards male speakers in the training set but biased towards female speakers in the test set for Callhome, the models could also be more likely to recognise the wrong output word for a female speaker.

Figure 3.3 and Figure 3.4 compares the WER across female (F) and male (M) speakers on the Switchboard and Callhome test sets with the LSTM and transformer encoder-decoder models respectively. Overall, the WER is higher for female speakers on the Callhome test set for both models, but it can also be observed that the average error rate for male speakers on the Switchboard test set is slightly higher. These results are inconclusive to ascertain whether either model is significantly biasing to recognise a particular gender's speech more consistently.

### 3.5.3 Categorical Analysis of Recognition Outputs

Using a similar root-cause analysis strategy to [175], the output recognition errors from the LSTM encoder-decoder model [2] were categorised in order to assess whether particular types of error were more prominent. By identifying errors that the model was more susceptible

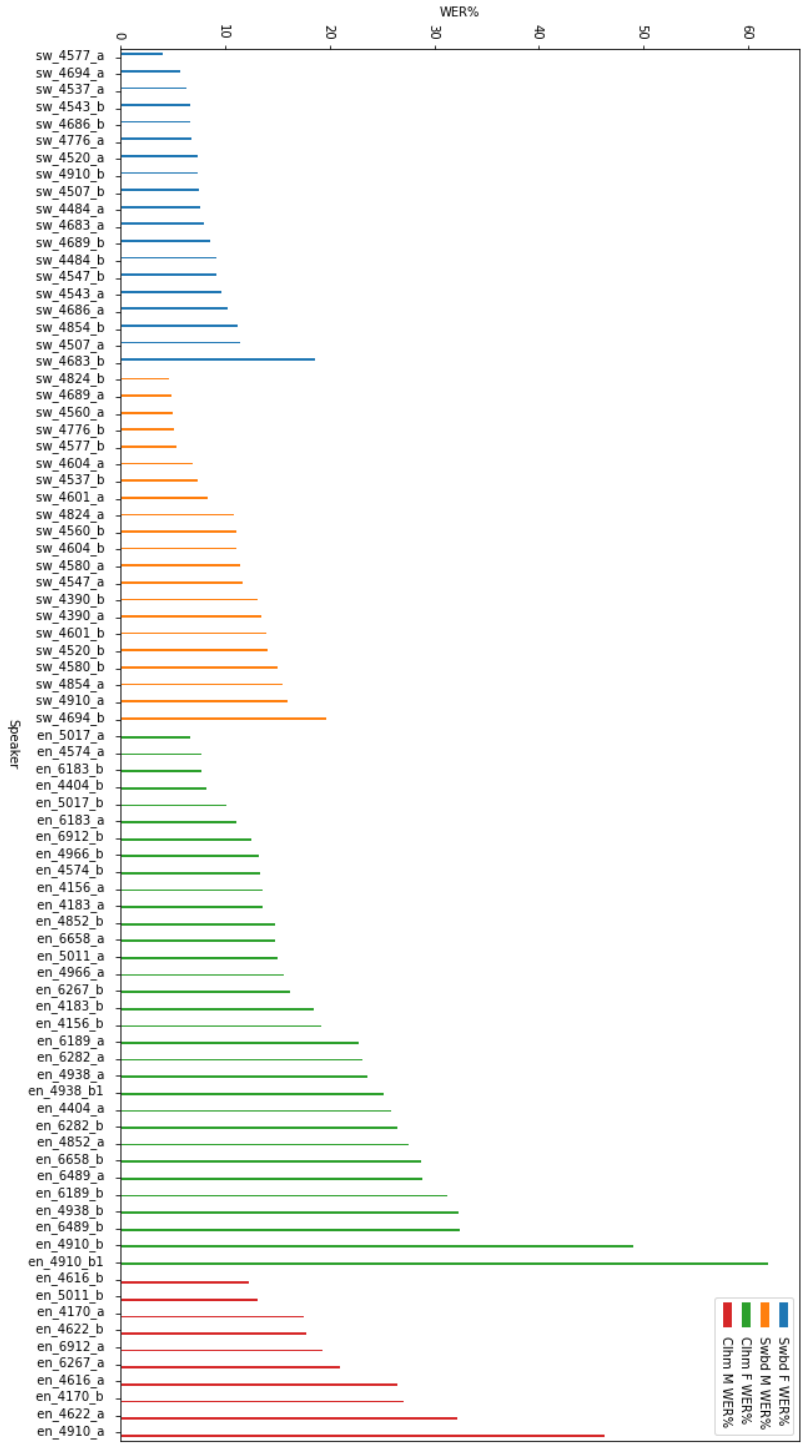



Fig. 3.3 WER across female and male speakers in the Switchboard and Callhome test sets with the LSTM End-to-End ASR model [2]

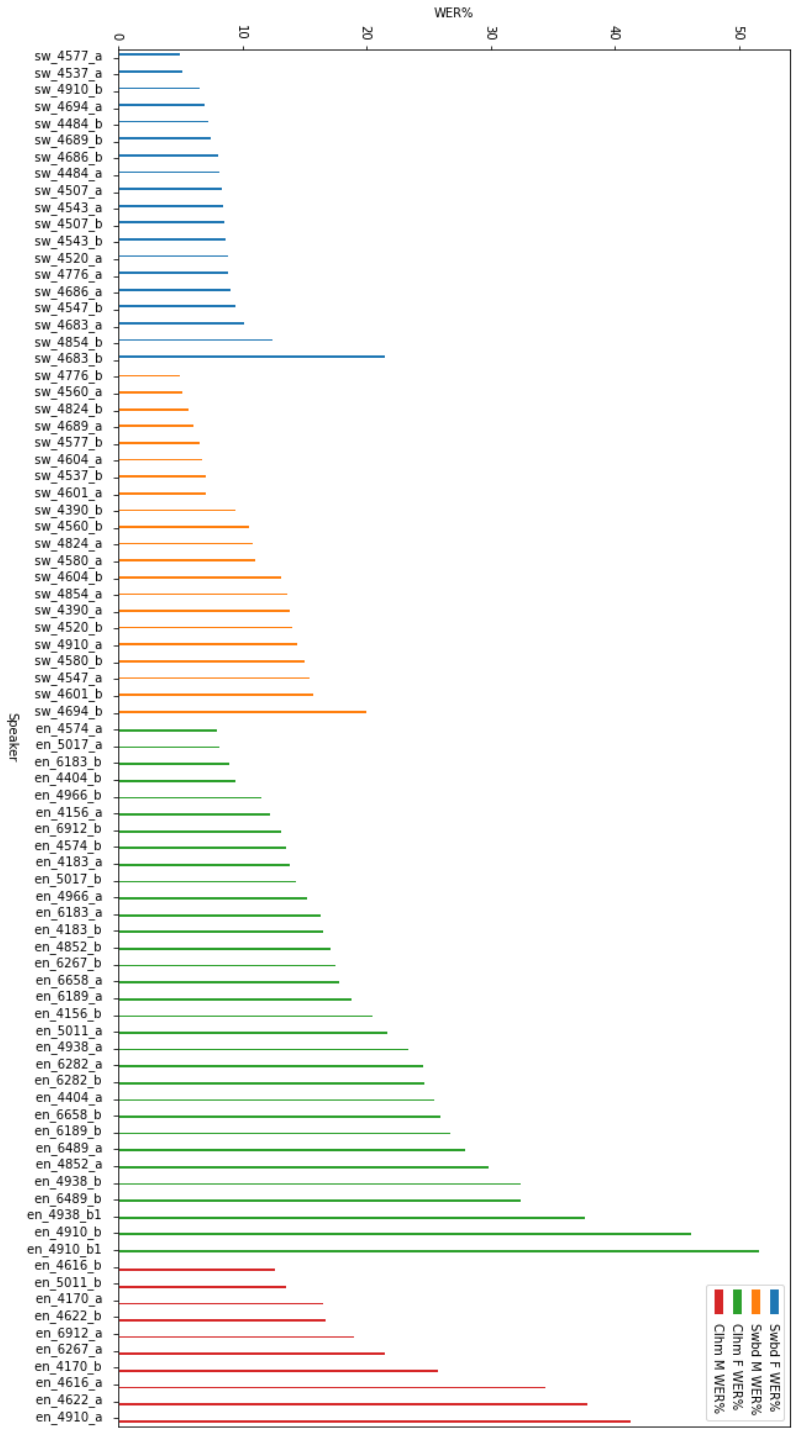



Fig. 3.4 WER across female and male speakers in the Switchboard and Callhome test sets for the transformer End-to-End ASR model [1]

to produce, it was hypothesised that it could be possible to modify the model to target the factors causing these errors. The error categories were defined as follows:

- **Homophone errors:** e.g. “see/sea”, words that typically have the same pronunciation but have different spellings and contextual meaning
- **Minimal pair errors:** e.g. “dog/dug”, a word that has a single phonological difference to the target word
- **Negative errors:** e.g. “can/can’t”, words that have the incorrect prefix or suffix to change the negative particle
- **Breached boundary errors:** e.g. “very ability/variability”, the boundaries of the word or words are diverging or converging from the target word
- **Verb infection errors:** e.g. “laugh/laughed”, modification of the verb conjugation from the target verb
- **Noun inflection errors:** e.g. “grape/grapes”, the plural form of the predicted noun is incorrect
- **Determiner errors:** prediction of the wrong quantifier e.g. “many/any”, article e.g. “he/the”, interrogative e.g. “who/whose”, possessive e.g. “her/his” or demonstrative e.g. “this/that”
- **Interjection errors:** e.g. “uh/oh” words that represents expressions, exclamations or fillers
- **Derivational suffix errors:** e.g. “beauty/beautiful”, where the target word has a different suffix but the same root
- **Unknown errors:** these are errors that do not belong to any of the above defined categories

Table 3.8 Categorised confusion pairs across Switchboard and Callhome test sets [5] from the LSTM End-to-End model [2]

| Key | Swbd # | Swbd % | Ch # | Ch % |
|---|---|---|---|---|
| Homophones | 10 | 4.07% | 21 | 3.30% |
| Minimal pairs | 26 | 10.57% | 68 | 10.69% |
| Negatives | 1 | 0.41% | 0 | 0% |
| Breached boundaries | 6 | 2.44% | 30 | 4.72% |
| Verb inflections | 21 | 8.54% | 51 | 8.02% |
| Noun inflections | 0 | 0% | 7 | 1.1% |
| Determiners | 42 | 17.07% | 177 | 27.83% |
| Interjections | 10 | 4.07% | 27 | 4.25% |
| Derivational suffixes | 4 | 1.63% | 7 | 1.1% |
| Unknown | 126 | 51.22% | 248 | 38.99% |

As can be seen from Table 3.8, 51% of the errors on the Switchboard test set (*Swbd*) and 39% of the errors on the Callhome set (*Ch*) were unable to be categorised into the defined specifications. Speculatively, an increase in homophone errors would indicate that the system is struggling to model a large enough context to determine the correct spelling of the target word, whereas an increase in minimal pair errors could indicate that there is a deficiency in the language modelling strategy. However, the model has produced the majority of errors with determiners (42 for *Swbd* and 177 for *Ch*), minimal pairs (26 for *Swbd* and 68 for *Ch*) and verb inflections (21 for *Swbd* and 51 for *Ch*). These results correlate with findings in [161] where turn-initial words, such as determiners and interjections can be more difficult to recognise, especially in the conversational speech domain where there are inconsistent pauses and speaking rates through turn-taking. It was also hypothesised that similar lexical terms, have similar feature values, although the high rate of minimal pair errors could also be a factor of various other causes such as low word probabilities for the target word or pronunciation disfluencies.

### 3.5.4 Analysis of Word Lengths

The model output errors were extremely high for shorter words, which could indicate that the models struggle to recognise words that are monosyllabic, words where speech rate is varying or is confusing the boundaries between phonemes. Shorter words typically result in a deceleration in speech rate [28], which is also affected by the position of the word in the speech segment.

However, as the total amount of shorter words spoken during typical conversational speech is much higher, the frequency of the word errors should be normalised by the frequency of the occurrence of the word in the dataset. Figure 3.5 shows the frequency of the words as a function of the number of letters that they contain. The most frequent word length is between 4 and 7 letters for the Switchboard test set [5]. This directly correlates to Figure 3.6 which displays the substitution errors on the Callhome test set by length of the word. Upon normalising the frequency of the substitution errors by the total number of occurrences of the word in the dataset, the error frequency by length of the word changes. Figure 3.7 shows that the percentage of substituted words is relatively evenly distributed across word length, aside from an outlier that is 2 letters long, which in this case is the word “is”.

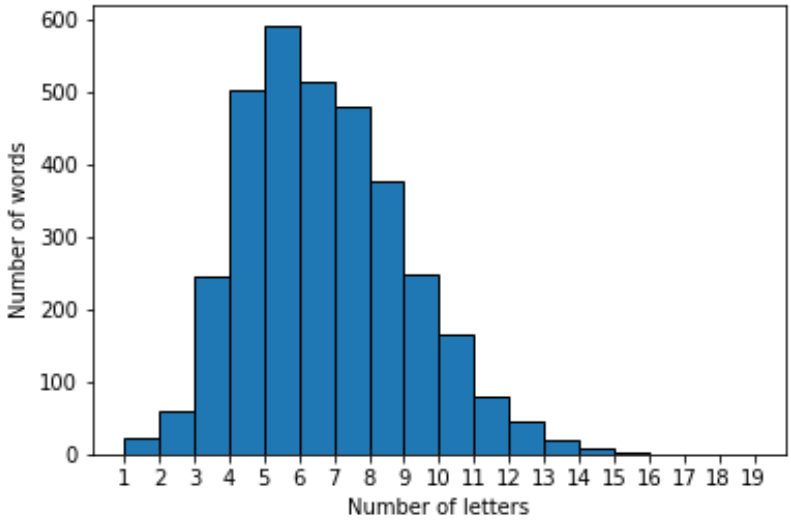



Fig. 3.5 Frequency of words by number of letters in the Switchboard test set from [5]

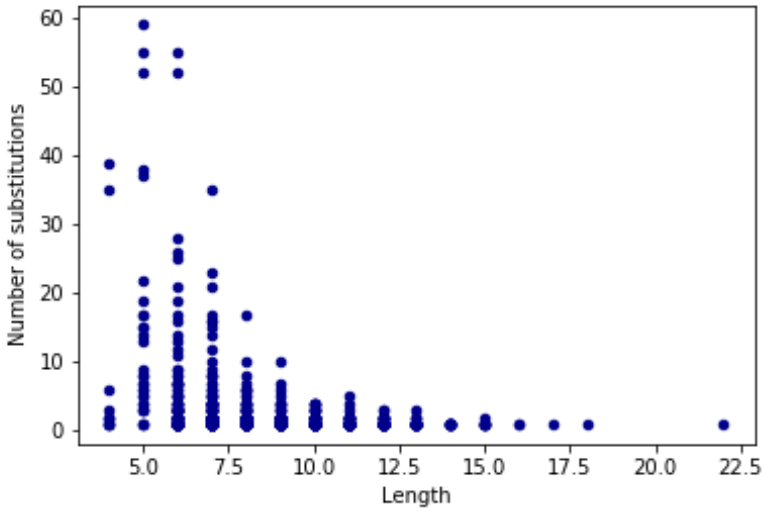



Fig. 3.6 Substitution errors by the LSTM End-to-End ASR model [2] on the Callhome test set [5] by word length

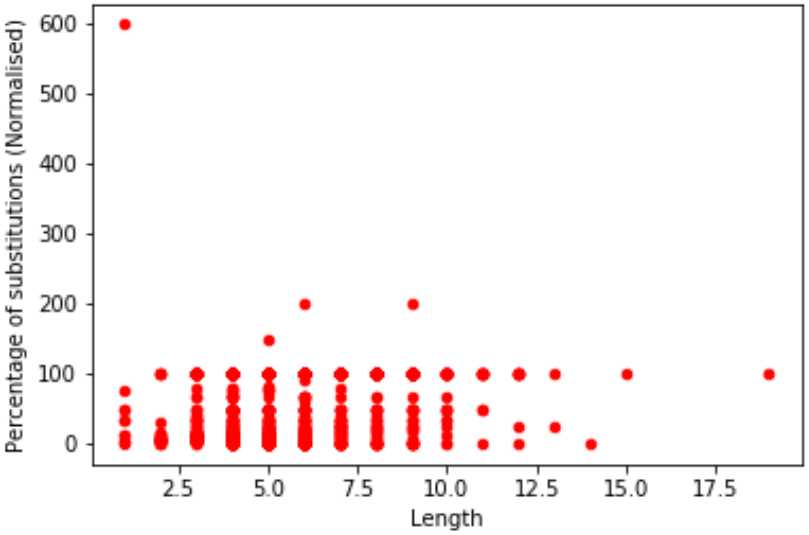



Fig. 3.7 Normalised Callhome [5] substitution errors by word length from the LSTM End-to-End ASR model [2]

Upon normalising the errors by the frequency of the words in the entire training and test set, it seems that there is little correlation between frequency of error and the length of the word.

### 3.5.5 Discussion

As the categorisation of errors is binary, analysing model outputs does not necessarily discriminate between errors that could be attributed to a set of errors. It may be more informative from an interpretative perspective to classify errors into sets of categories. However, the major limitation of these methods of analysis is that they simply highlight patterns of errors but do not give any further causal information to allow interpretation. Trial-and-error improvements to the modelling approach would still need to be performed based on these results, which can be inefficient and computationally expensive. These factors attributed to the development of the analysis framework described in Section 3.6. It is also unclear if one aspect of the modelling approach is altered to target a specific error, whether the modification will have an unknown effect on another downstream process and adversely affect the performance. This hypothesis is investigated further in Chapters 4 and 5.

## 3.6 Experimental Framework for Representation Analysis

As it is unclear how the model architecture or residual connections within the models, described in Sections 2.3.2, 2.3.3 and 2.3.4, contribute to more optimal representations, the neural representations can be analysed with regard to the recognition performance. By generating a method to view a window into the behaviour of neural representations within the network, this was hypothesised to provide information on the interaction between the modelling approaches and the training data.

The following analysis using similarity indexes focuses on interpreting the similarity of the neural representations and providing answers to questions such as: given the same speech input, how similar are the learned representations throughout training epochs? Do specific architectures have a higher representation similarity? Is neural representation similarity correlated with model performance?

Using correlation analysis techniques from [176] and [177], the key contributions of this Section are summarised as follows:

- The development of a framework to enable representation correlation analysis with multiple indexes using state-of-the-art End-to-End architectures, Section 3.6.
- Similarity index comparison on an ASR task, Section 3.6.4.
- Verification that neural representation analysis of End-to-End network structures can be used to visualise pathological aspects of adaptations for conversational telephone speech recognition models.
- Discussion regarding the interpretation of representations of End-to-End architectures and potential future work that would aid and develop these findings, Section 3.6.4.

This work was published as [24].

### 3.6.1 Related Work

Current correlation analysis techniques from [177] and [176] have been used to compare representations within DNNs for various applications such as image recognition and language modelling. Populations of neural representations have been compared and evaluated using several methodologies with the aim of interpreting or explaining relationships between and across neural layers [177, 176, 178]. Correlation analysis provides information regarding how different or similar neural representations are throughout a model layer. When correlation has been conducted through model training iterations, correlations between neural layers and models has allowed interpretation of the dependent neurons upon specific tasks. For example, where correlation is high between neural layers of 2 models trained on a divergent tasks, the neural representation space is similar and potentially independent of the particular task. Interpretation is non-trivial where multiple context dependencies are enveloped in the

same representations, such as End-to-End ASR, therefore it was not clear which aspects of representations should be compared or which similarity index would be a suitable measure.

In order to compare network representations of DNNs, Canonical Correlation Analysis (CCA) [179] and Centered-Kernel Alignment (CKA) [176] have been used as statistical correlation indexes, which are described in more detail in Section 3.6.2. CCA and CKA enabled the identification of shared structures across representations. For SVCCA [177], Singular value decomposition (SVD) is computed before the CCA index to reduce the dimensionality of the neural representations, in order to compare representations across networks. Using SVCCA for an image classification task [180] established that network solutions diverged predominantly in the intermediate neural layers. SVCCA has also been used as an index in order to observe the development of linguistic features during encoding in language models [178]. As neural layer depth was increased, the correlations within the encoded network representations decreased, which indicated the representation spaces of deeper layers were more dependent on the linguistic features investigated. Additionally, CKA was used to observe that task-specific trained neural layers develop more similar neural representations, shown by higher correlations of neural dependencies.

Correlation indexes have not yet been used to analyse neural representations of End-to-End architectures with speech data for speech recognition tasks. The following experiments present a comparative study of neural representation indexes with state-of-the-art End-to-End ASR networks.

### 3.6.2 Similarity Indexes for End-to-End ASR Modelling

Due to the undefined separation of modules within End-to-End ASR networks, it is relatively unclear which, what and where the originally separate ASR system context information is learned, such as acoustic or language modelling. The internal parameter dependencies upon

the structures of the model, and their effect upon the resulting performance, are ambiguous and inherently complex.

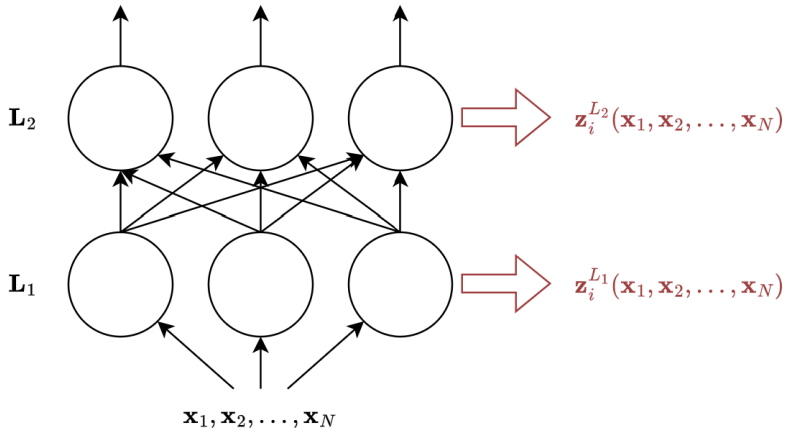



Fig. 3.8 Overview of neural layers $L_1$ and $L_2$ showing the activation output vector for each layer $\mathbf{z}_i^L$

Using statistical correlation analysis methods, it is possible to relate two sets of observations within a network to find their correlation relationship. As shown in Figure 3.8, for the dataset $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and neuron $i$ in layer $L$, the activation output vector is $\mathbf{z}_i^L = (z_i^L(\mathbf{x}_1), ..., (z_i^L(\mathbf{x}_N))$. By conducting correlation analysis techniques that are invariant to affine transforms, this enables comparisons between different neural networks and observations on the dynamic behaviour.

### Singular-Value Decomposition with Canonical Correlation Analysis (SVCCA)

CCA [179] is used to find bases $\mathbf{w}, \mathbf{s}$ for two matrices such that, when the original matrices are projected onto these bases, their correlation $\rho$ is maximised:

$$\rho_{CCA}(L_1, L_2) = \frac{\mathbf{w}^T \sum_{L_1 L_2} \mathbf{s}}{\sqrt{\mathbf{w}^T \sum_{L_1 L_1} \mathbf{w}} \sqrt{\mathbf{s}^T \sum_{L_2 L_2} \mathbf{s}}} \tag{3.1}$$

where $\sum_{L_1L_1}, \sum_{L_1L_2}, \sum_{L_2L_2}$ are the covariance and cross-covariance of layer 1 $L_1$ and layer 2 $L_2$. In the case of ASR neural networks, this is between the neural layers for $N$ data points where $L_1 = \{\mathbf{z}_1^{L1}, ..., \mathbf{z}_N^{L1}\}$ and $L_2 = \{\mathbf{z}_1^{L2}, ..., \mathbf{z}_N^{L2}\}$. The covariance matrix of the layers is a measurement of the variation within the layer activation outputs, while the cross-covariance matrix is a measurement of the variation between the activation outputs of different layers. These are calculated by finding $\frac{1}{N} \cdot (L_1 - \tilde{L}_1)^T \cdot (L_1 - \tilde{L}_1)$ and $\frac{1}{N} \cdot (L_1 - \tilde{L}_1)^T \cdot (L_2 - \tilde{L}_2)$ respectively, where $\tilde{L}$ denotes the mean vector of the layer.

The projected views of $L_1$ and $L_2$ are then obtained to be the top 99% representative dimensions, using SVD, in an attempt to reduce potential noise in the representations, to form subspaces $L_1' \subset L_1, L_2' \subset L_2$ [177]. CCA [181] is used to maximise the correlation of the projections of the linear transform of $L_1', L_2'$ by identifying vectors $\mathbf{w}, \mathbf{s}$ to maximise:

$$\rho_{SVCCA}(L_1, L_2) = \frac{\langle \mathbf{w}^T L_1', \mathbf{s}^T L_2' \rangle}{||\mathbf{w}^T L_1'||\ ||\mathbf{s}^T L_2'||} \tag{3.2}$$

The correlations $\rho$ are higher when the representations have encoded more similar information.

### Centred Kernel Alignment

CKA, first introduced in [182], is another similarity metric that has been used to measure the similarity of neural representations. CKA resembles CCA but is weighted by the eigenvalues of the corresponding eigenvectors. It is also similar in effect to SVCCA but incorporates the weighting symmetrically and doesn't require matrix decomposition. Instead of comparing multivariate features of the neural layers, the coefficiency between every pair of examples in each representation is measured, then the correlation computation is conducted. This is based on the Hilbert-Schmidt Independence Criterion (HSIC), which is defined as:

$$HSIC(F, G) = \frac{1}{(n-1)^2} tr(FHGH) \tag{3.3}$$

where $F_{i,j} = f(\mathbf{z}_i^{L_1}, \mathbf{z}_j^{L_1})$ and $G_{i,j} = g(\mathbf{z}_i^{L_2}, \mathbf{z}_j^{L_2})$ are two kernels of the $i^{th}$ and $j^{th}$ layerwise neural representations $\mathbf{z}^{L_1}$ and $\mathbf{z}^{L_2}$, $n$ is the dimension of the representations, and $H$ is the centering matrix. The centering matrix is defined as $H = I_n - \frac{1}{n}\mathbb{1}\mathbb{1}^T$ [182]. The HSIC [183] is used to measure the statistical independence between two distributions of activations. This attempts to measure the pairwise similarity between examples and between features by taking the sum of the squared dot products between every pair.

To measure the similarity index between the internal representations, the correlation $\rho_{CKA}$ is determined to be a normalisation of HSIC [176]:

$$\rho_{CKA}(K,L) = \frac{HSIC(F,G)}{\sqrt{HSIC(F,F)HSIC(G,G)}} \tag{3.4}$$

which enables CKA to be invariant to the orthogonal permutation of neurons and invariant to scaling, which can allow comparison between layers of neural representations [176].

### Projected Weighted Canonical Correlation Analysis

Another method to reduce the sensitivity of the CCA, is to replace the SVD operation within SVCCA by a weighted mean with CCA, referred to as PWCCA [184]. This allows the canonical correlations to have a higher weight if they are more influential to the underlying representation, based on the hypothesis that CCA vectors that account for a greater proportion of the original outputs are likely to be more important to the underlying representation.

Where $L_1$ has output activation vectors $(\mathbf{z}_1^{L_1}, \mathbf{z}_2^{L_1}, ..., \mathbf{z}_N^{L_1})$ and CCA vector $\mathbf{a} = (\mathbf{w})^T \sum_{L_1,L_1}^{-1/2} L_1$, the approximate weight $\tilde{\alpha}_i$ can be determined by the measure of the output that is accounted for by each $a_i$:

$$\tilde{\alpha}_i = \sum_j |\langle a_i, z_j \rangle| \tag{3.5}$$

By normalising with $\sum_i \alpha_i = 1$, the PWCCA can be determined as a distance:

$$\rho_{PWCCA}(L_1, L_2) = 1 - \sum_{i=1} \alpha_i \rho_{CCA} \tag{3.6}$$

with reference to Equation 3.6.2.

PWCCA alleviates potential accrued errors when pruning with SVD, by taking the weighted average of the vectors to reduce the sensitivity of the analysis, although does not represent the irregularities within the internal representations. This approach is also outperformed by CKA in [176] and is not invariant to networks with different initialisations and architectures either.

### 3.6.3 Experimental Setup

Both the LSTM and transformer encoder-decoder models described in Section 3.4 were used for the following statistical analysis experiments. While the error outputs can be observed for each model after training, there currently exists no strategy or tools to attempt to understand the internal representations of End-to-End models. By understanding the dependencies of optimising parameters, different modelling approaches, and training regimes it is hypothesised that these insights could be used to improve model performance for specific datasets or domains.

To investigate the dependencies of the neural representations within End-to-End models, SVCCA and CKA, described in Section 3.6.2, were applied to the activation outputs of each model layer across training epochs. In order to conduct a consistent analysis for each model and index, several further steps were necessary: firstly, the model parameters were preserved at each epoch of the ASR task; and then they were fed to a separate pipeline for the extraction of the activation embeddings for each neuron. To ensure consistency, this was done by feeding in a controlled input of 100 speech frames to all architectures and extracting

the activation output at each neuron, enabling the representation analysis methods to be conducted concurrently. The input features were 80-dimensional filter banks. Where neural layers did not have the same dimensionality, such as between the 2D convolutions and the 1st encoder layer, linear interpolation of the narrower layer to the same dimensionality as the wider layer was conducted due to the different spatial dimensions of the neural layers and thereby data-points, as both SVCCA and CKA methods require representation vectors to be the same dimensions. To compare the correlation coefficiency across the number of layers in the network, the spatial dimensions of the activation outputs were flattened into the number of data-points, in order to provide a spatial representation of each data-point.

### 3.6.4 Results and Analysis

The proposed analysis framework allowed the observation of the layer-wise representation analysis methods across scaled convolutional layers within an ASR task. Comparison across layers allows the observation of the converged layer correlations, while comparison across epochs shows the hierarchical representations within the layers as the models are trained. In this case, convergence occurs when further training does not improve the recognition performance of the model. Upon evaluation with the Hub5'00 set [185], the WER performance of this architecture is displayed in Table 3.9. Increasing the neural depth of the CNN layers improved accuracy slightly up to 3 layers but results varying the spatial dimensions of each layer showed little improvement. The performance was observed to be limited when varying the dimensionality of each layer across the variable sized CNN architectures, shown in Table 3.9, with the best WER performance achieved with a 3 layer CNN.

Figure 3.9 shows the CCA correlation coefficiency through each layer for all models with differing numbers of network layers (from 1 to 6) once each model has been trained. Upon comparing the correlation coefficiency across the layers within each model, the models with more layers have more variation in the neural representations than the other models. The

Table 3.9 Variable sized CNN layers for LSTM End-to-End ASR model [2] evaluated on the Hub5'00 test sets [5]

| CNN Architecture | SWBD WER% | Clhm WER% |
|---|---|---|
| 6 layers | 11.4 | 22.4 |
| 5 layers | 10.7 | 21.3 |
| 4 layers | 10.9 | 21.2 |
| 3 layers | **10.5** | **20.8** |
| 2 layers | 10.6 | 20.9 |
| 1 layer | 11.6 | 22.5 |

models with 2 and 3 layers also had slightly better performance results, which corroborates with results in [176]. However the 6 layer CNN model had similar coefficiency but worse performance than the other models; this instigated further investigation to understand the representation space relationship with the training regime.

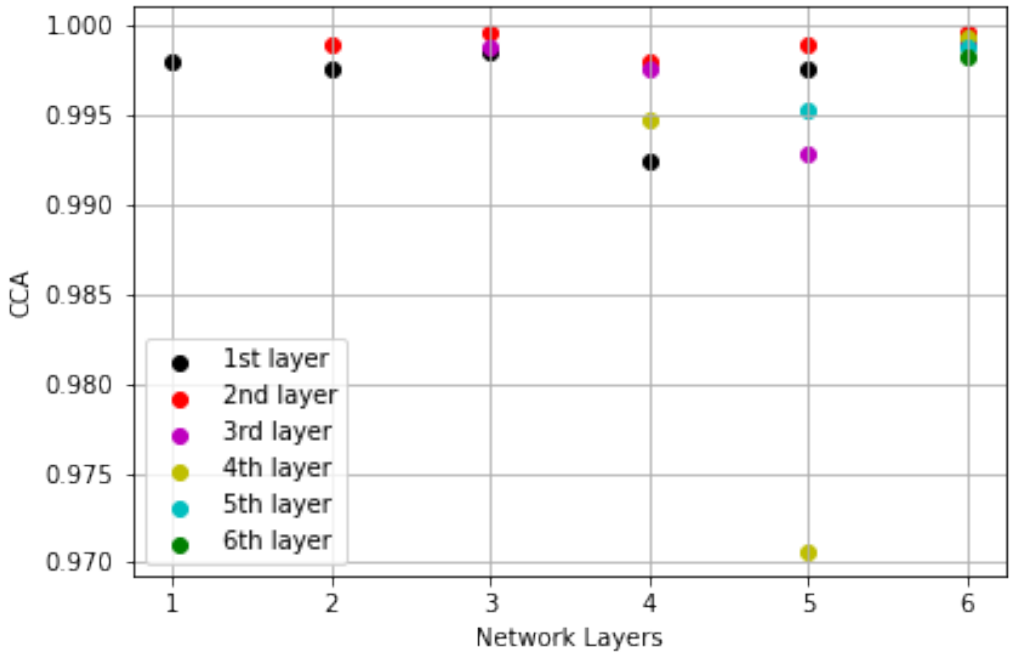



Fig. 3.9 The converged model CCA coefficiency across all the End-to-End ASR models with increasing amounts of layers

During the training process, Figure 3.10 shows that as the number of the layers increased, the coefficiency of each layer approached 1 at different rates but in parallel. Using SVCCA

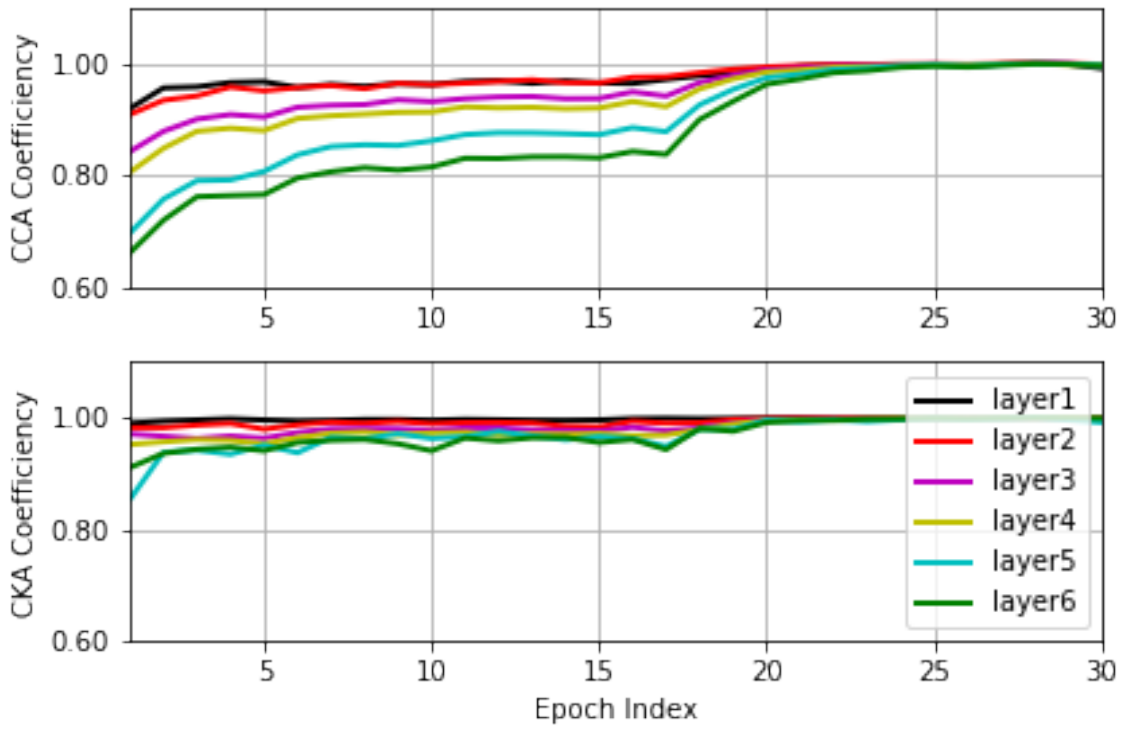



Fig. 3.10 CNN neural representations of the LSTM encoder-decoder model [2] evaluated with SVCCA (**top**) and CKA (**bottom**) through time as performance converges

analysis, described in Section 3.6.2, to correlate the activations across the training epochs, it was observed that layers 1, 2 and 3 converge together at epoch 17, whereas deeper layers (layers closer to the output) converged slightly later. Where the coefficiency rate is closer to 1, this indicates that the representations within that layer between each epoch are relatively consistent, highlighting the relationship between the neural representation space and the training regime.

In order to understand whether different insights could be uncovered using other statistical indexes, Figure 3.10 also shows the CKA coefficiency, described in 3.6.2, of the CNN architecture. It was observed that the SVCCA analysis is more sensitive to the initialisation parameters of the model than CKA as there is a much greater distribution of coefficiency at the first epoch across all the layers. With both strategies, a hierarchical correlation within the layers across training can be observed, although the CKA results suggest that there is some

unstable behaviour present in deeper layers; for example the small spikes in coefficiency across layer 6. The CKA results potentially differ from the SVCCA results, due to the pruning of the SVD component of SVCCA while also assuming that all the coefficiency vectors are equally important to the representation of the ASR task.

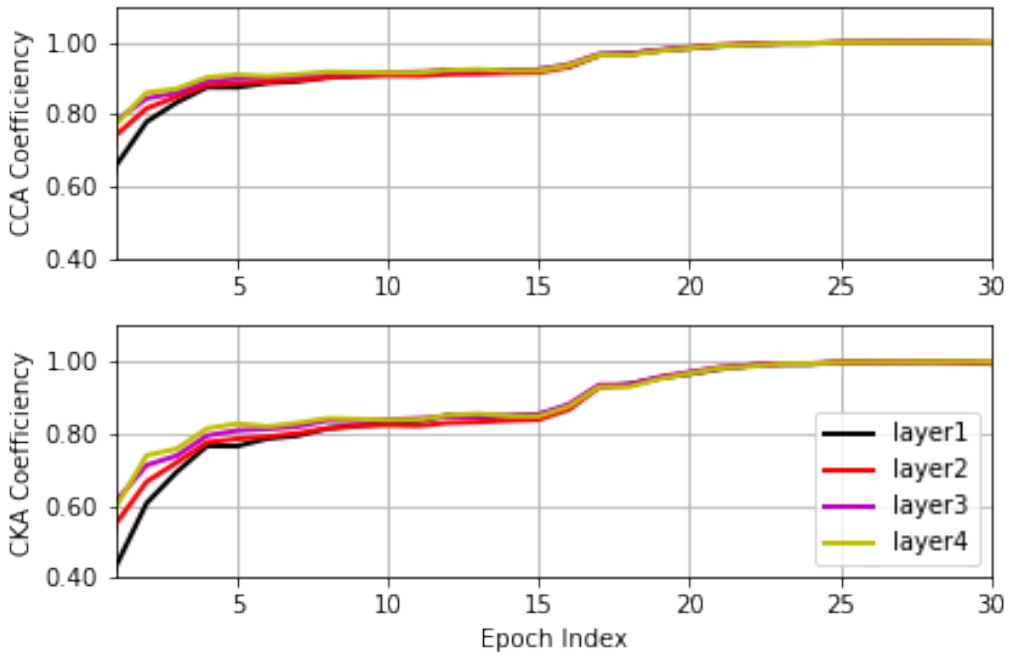



Fig. 3.11 LSTM [2] correlation coefficients of neural representations evaluated with SVCCA (**top**) and CKA (**bottom**) through time as performance converges

The LSTM neural representations, within the encoder-decoder model [2], are displayed in Figure 3.11. Comparing the SVCCA correlation results with the CKA results, it can be observed that the correlation is slightly under-estimated by the SVCCA implementation, although both techniques display similar attributes. By comparing the internal representations with SVCCA and CKA, the behaviour of the internal dynamics of the neural representations within a model can be observed to be invariant to transformations, in a robust method. The coefficiency across epochs suggests that there is an observable bottom-up behaviour within the LSTM representations due to recurrence, with convergence across all layers occurring around epoch 22.

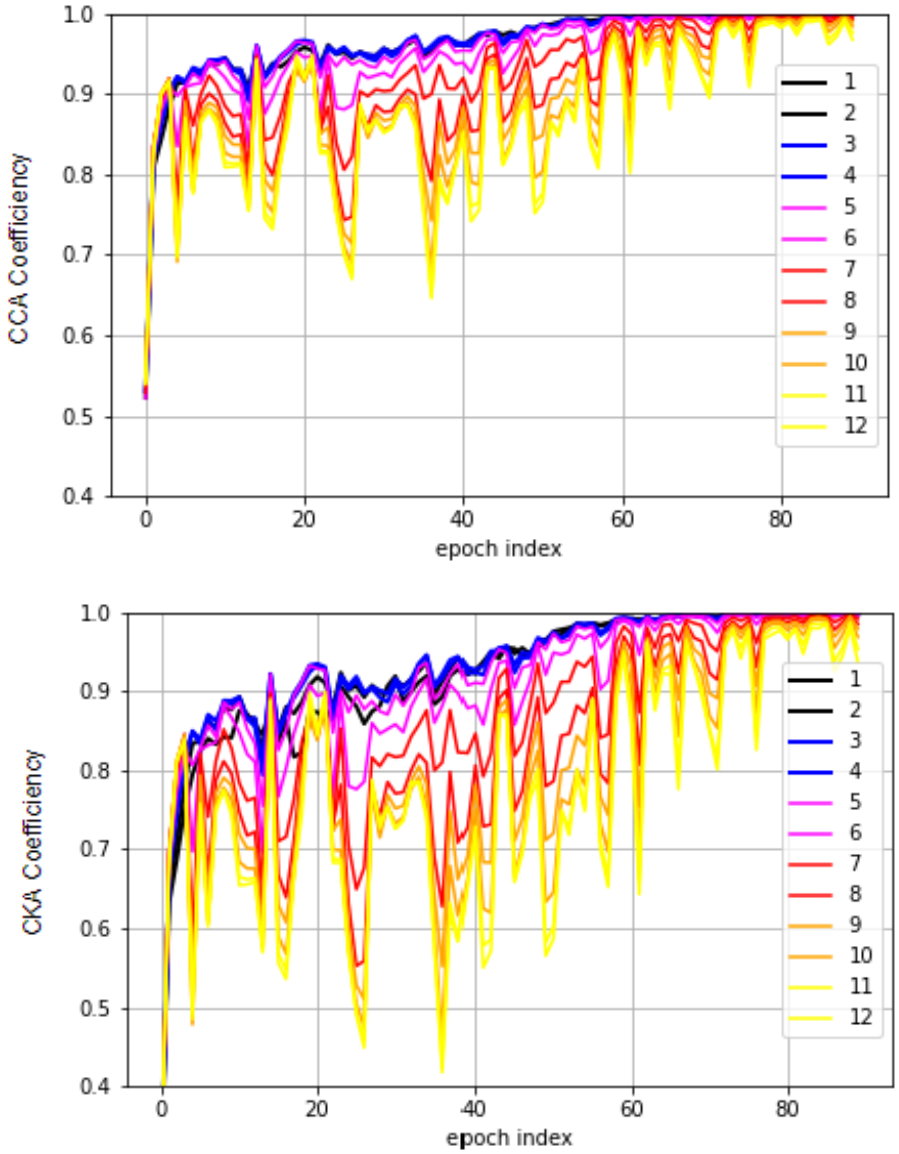



Fig. 3.12 Transformer [1] correlation coefficients through epochs as performance converges produced with SVCCA (**top**) and CKA (**bottom**)

The transformer model [1] layer encoder output representations, shown in Figure 3.12, emphasise the attending mechanism pathology present after the self-attention and linear operator. It can be observed that the higher layers of the Transformer encoder are less susceptible to the attention pathology than the deeper layers, which don't converge completely even after 80 epochs. There is a more noticeable distinction that can be ascertained from

the CKA analysis, which retained more distributed results, that there is similar overall hierarchical learning dynamics as was observed in the CNN layers in Figure 3.10.

### 3.6.5 Discussion

By using SVCCA as a method of analysing the internal representations for an End-to-End ASR framework, it is possible to observe the dynamics of the neural network representation behaviour, although the pruning operation appears to under-represent the neural representations compared to using CKA. This is partly due to the assumption that all of the CCA vectors are equally important to the neural representation but also the SVD component of the SVCCA technique in Section 3.6.2 relies on the reflection of class information, which, for End-to-End speech recognition, is a potential limitation. By implementing the CKA analysis method in Section 3.6.2 it is possible to visualise the pathology of neural representations during training, particularly in the Transformer model, Figure 3.12, which is indicative of the dependencies upon context dependencies, particularly for wider layers.

The techniques described in Section 3.6.2, allow the observation of hierarchical behaviour of CNN and Transformer neural representations across training, Figures 3.10 and 3.12, whilst also providing insight on the bottom-up representation behaviour within the LSTM layers (without residual connections), shown in Figure 3.11. The dependencies between layers across epochs exhibit similar learning dynamics as language models [186]. The similarity indexes could also be used in future work to compare the correlation of the trained neural layers of various modelling approaches across different speech datasets, such as noisy or augmented data, to observe how the neural layers respond dynamically during the training process. These experiments could then be directly correlated with the performance results.

Additionally, it has been noticed that scaling the depth of the convolutional layers had a limited effect upon model performance in the case of the LSTM approach, as shown in Table 3.9. Expanding the results from [184], Figure 3.9 provides some evidence that better

performing models converged to similar solutions, however the 6 layer CNN suggests this is not always the case. The poorer performance of the 6 layer CNN could be attributed to an over-fitting issue and to investigate this further, the potential memorisation within the neural representations would be need to be undertaken. These results can be expanded to further develop and explore better architecture solutions for End-to-End ASR performance, whilst gaining some insight of the effect architecture changes have upon network dynamics.

Further investigation of the attributes for the dependencies would be required, for instance, do the unstable deeper layer neural representation correlations correspond to noisy components within ASR task? Furthermore, an extension to this work could be the analysis of neural representations on out of domain data, as the structural properties of the different layers could be beneficial to building models for few-shot-learning in ASR.

## 3.7 Summary

This Chapter aimed to investigate state-of-the-art End-to-End ASR frameworks, where Section 3.3 discussed the current state-of-the-art End-to-End frameworks and compared the recognition performance. It was found that attention-based encoder decoder models are able to reach the best recognition performance on a conversational test set Hub5'00 [5].

Experiments conducted in Sections 3.5.2, 3.5.3, and 3.5.4 attempted to understand whether there were any underlying patterns within the model output errors using current evaluation metrics. This was to investigate whether different attention encoder-decoder models produce outputs with specific types of errors. These experiments highlighted the fundamental flaws of using the current evaluation metrics when attempting to interpret the relationship between model approach and underlying causes for predicted errors. It is difficult to observe model output errors and therefore difficult to adapt an approach to target those errors.

Instead, Section 3.6 attempted to understand how the modelling approach can contribute to more optimal representations for End-to-End ASR performance. It is argued that it is important to be able to interpret internal dependencies of modelling approaches to develop more informed improvements and have confidence in the behaviour of a chosen modelling approach.

A framework was developed in order to enable neural representation analysis with correlation indexes, described in Section 3.6.2. This framework extracted activation outputs of LSTM and transformer encoder-decoder models during training. The activation outputs were compared through layers, shown in Figures 3.9, 3.11 and 3.12 and it was found that models exhibit a layerwise hierarchy when trained with conversational speech data. The correlation of the hierarchies was observed to be an indicator of recognition performance, where the results of Section 3.6.4 showed that model with highest coefficiency of converged neural layers performed the best.

These insights aim to progress interpretive analysis of modelling for End-to-End ASR and potentially provide techniques that could be utilised in multiple domains. In order to further understand the relationship between neural representations, Chapter 4 extends the analysis with acoustic modelling techniques using different approaches and domains of data. Chapter 5 extends the analysis using different LM regimes to attempt to explain the dependencies of neural representations and highlight dependencies of specific layers to particular contextual information.

# Chapter 4

# Acoustic Context Analysis for End-to-End ASR

## Contents

## 4.1 Introduction

Different acoustic modelling techniques are often required for different domains of data, thereby orchestrating research to explore new approaches. This chapter explores the implementation of scalable multi-band CNN models to capture longer-term acoustic dependencies, inspired by a mixture of experts (MoE), which has been shown to be effective in natural language processing [187] [188] and vision domains [189]. This approach involves partitioning the modelling (experts) to create more specialised sub-modelling, where the outputs of each expert is combined by taking the weighted average. Using this approach aims to improve acoustic representation modelling while keeping the inference cost constant by applying a subset of parameters to each sample.

As it is unclear how the these models adapt to the learned features, the neural representations of the proposed models are compared to observe the interaction between the developed techniques and the data, building upon previous work from [24]. SVCCA has been used previously [177] to compare DNN representations, while this work aims to provide further insights on the similarity of the learned representations across training and provide a discussion on the distinct representations that occur within convolutional-transformer models and the adaptive memorisation capability of the transformers. The following work is comprised of experiments published in [25].

## 4.2 Related Work

Particular focus of current developments with attention-based models have involved data augmentation techniques [166], [190], [191] and vastly increasing model depths [192], [156] in an attempt to provide richer representations. However, it is not always clear whether these

models are generalising to the data or memorising the data and the performance improvements are not attributed to the structures within the actual model architecture. Generalisation refers to the ability of the model to perform well on unseen or new data that was not part of the training data, where the model has developed representations that represent the relationship between the data domain and output. While memorisation refers to the concept of model overfitting, where the model memorises the input-output mapping of the training set and struggles to generalise to new or unseen data. It was argued in [178] that by comparing the similarity of LM representations, it was possible to observe when a LM was memorising data by training models with specific and randomised topics, then comparing the layerwise representations.

The current state-of-the-art approach, described in [133], utilises the combination of CNNs and a transformer to provide further improved ASR performance. It is hypothesised that this is due to the ability of CNNs to capture richer local feature representations while the transformer is better able to capture global context. The approach from [116] attempted to approximate the information in the attention matrix of the original transformer model. This was done by linearly scaling the attention by projecting the embedding matrix into lower dimension space then computing the inner product. A further attempt from [193] aimed to remove the independence assumptions during modelling to capture long-term context dependencies for End-to-End models. This approach used a knowledge distillation technique, where a model trained with different data distills knowledge into another model. In this case, a hierarchical transformer model handles utterance level contextual information and discourse level information independently, while sharing the learned dependencies with another model.

Regarding statistical indexes to analyse models for speech applications, [194] investigated contextual word representation similarity in order to understand how representations from different models and layers could capture different properties. Several similarity metrics

were considered for this task, while CCA was utilised for the assessment of representation-level similarity and was able to provide insights on layerwise representation dependencies. SVCCA was used in a comparative study [195] on state-of-the-art self-supervised algorithms. The similarity index was used to interpret the similarity between representations learned by different models and also to estimate how training loss correlated with model performance for downstream tasks. It was found that the objective of the self-supervision task had a greater impact upon the similarity of learned representations than the model architecture and the authors were able to discern a relationship between the correlation between self-supervised loss and speaker classification performance.

A comparative study of statistical correlation indexes for End-to-End ASR representation analysis was shown in Chapter 3 and published in [24]. SVCCA and CKA indexes showed similar results for correlation analysis of neural representations. This method of representation analysis has not been conducted to assess acoustic context for End-to-End ASR models and could provide insight on context dependencies within these models.

## 4.3 Acoustic Modelling for End-to-End ASR Transformers

When CNNs are combined with models for ASR, the spectogram of the audio signal can be processed in segments. Combing with recurrent-based models allows local feature information to be captured progressively [196], while the transformer model is more able to capture longer range global contexts [1]. Similar approaches, as in [133], combine the self-attention mechanism of the transformer with convolutions to improve recognition performance for ASR tasks. However, it is unclear how the neural representations in models such as the LSTM [2] or transformer [1] adapt to the features learned by the CNN.

As CNNs process the entire spectogram of the audio signal with the same time-frequency resolutions, number of filters, and dimensionality reduction, previous work [197] has shown that higher resolution features can be extracted if the lower frequency bands are processed

with high frequency resolution filters and high frequency bands with high time resolution filters. This is determined by the theory that there is more "voice information" in the lower frequency bands than the higher bands. Furthermore, [198] found that deeper transformer layers dilute audio features, and that the distinction is more profound with spontaneous conversational speech. It is hypothesised that by adapting the CNN layers to learn different representations of the same feature space, that this would cause the dependent representations within the transformer layers to adapt and potentially improve recognition performance for conversation speech.

### 4.3.1 Proposed Acoustic Modelling Approaches

#### Multi-Band (mband) Modelling Approach

A multi-band CNN model (mband) is proposed to learn different representations of features for ASR. The multi-band features $\mathbf{m}_i$ are defined as having $C$ sub-bands. The $i_{th}$ filter bank of the $j_{th}$ band of the frame of speech can be described by:

$$\mathbf{m}_i^{(j)} = f_C^T \mathbf{x}_C^{(j)} \tag{4.1}$$

where $f_C$ is the discrete cosine transform function:

$$f_C = \sqrt{\frac{2}{C}} \cos\left[(k-0.5)\frac{i\pi}{C}\right] \tag{4.2}$$

and where $k$ is the channel energy amplitude.

By modifying the CNN layers with separate filters, features can be extracted separately at multiple levels of the frequency spectrum. The output layers can then be concatenated together. The proposed architecture is shown in Figure 4.1, where the input is passed to 3 band-pass filters that evenly split over the spectral frequency. The output of the filters then passes to CNN layers which are then concatenated and output to the encoder layer.

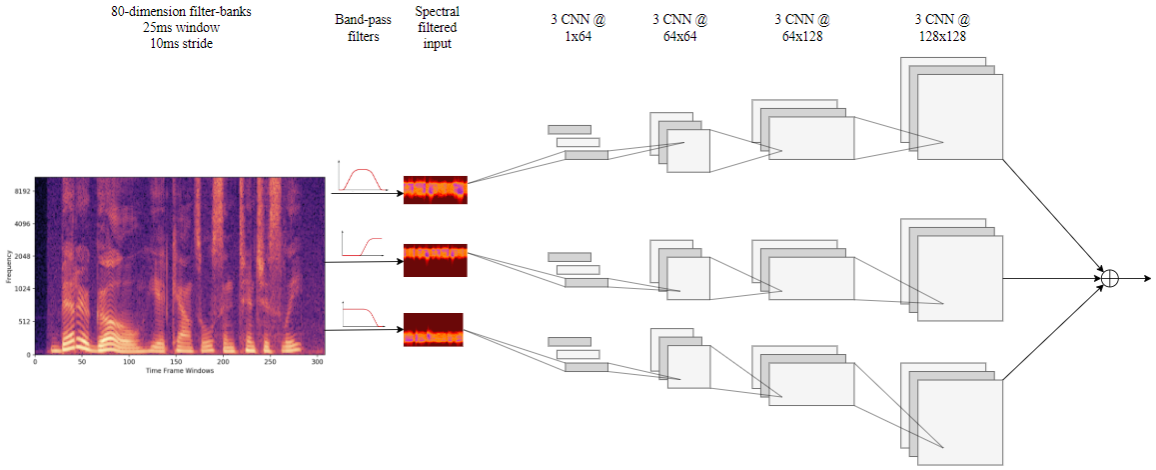



Fig. 4.1 Multi-band CNN architecture: frequency filters applied in parallel through multiple CNN layers

### Multi-Channel Modelling Approach (mchan)

Along a similar methodology, a multi-channel (mchan) approach takes the entire input into parallel convolutional blocks, without band-pass filters, in an attempt to learn different representations of the same acoustic signal. This will also allow a comparison between models with multiple channels but without the additional filtering methodology. The representations are then aggregated using MoE in the same method as the mband approach. As shown in Figure 4.2, instead of taking the frequency bands as different streams, the whole input is taken in multiple streams.

### Transformer End-to-End ASR Model

Transformer models are currently the predominant choice for a multitude of domains, such as image recognition and speech recognition due to their state-of-the-art performance [133, 141, 199]. The model published in [1] has especially been utilised for End-to-End speech recognition tasks due to its ability to create a more accessible parallel training method which has allowed End-to-End solutions to make use of larger amounts of data. As transformer

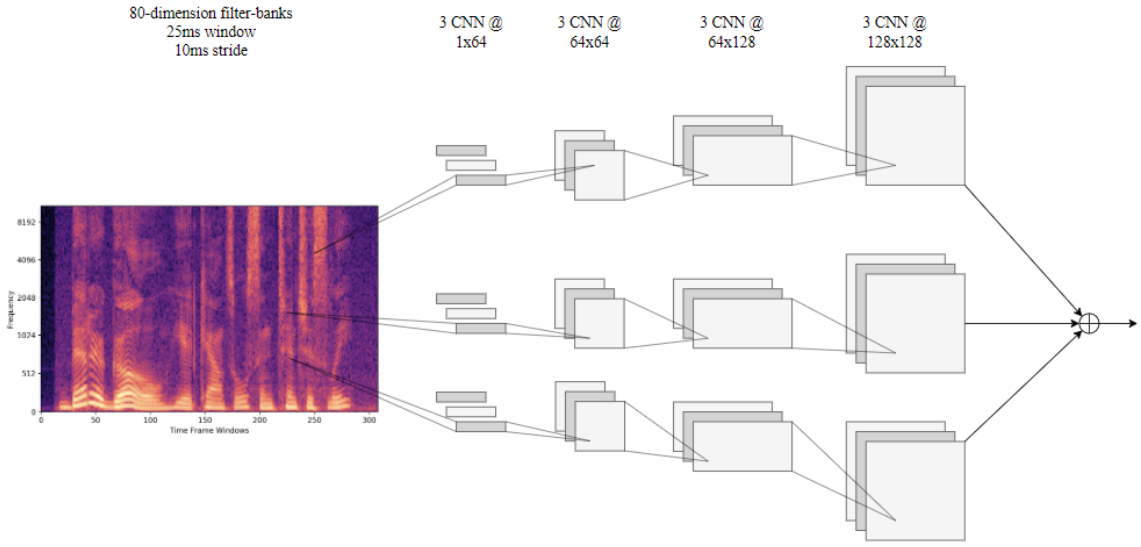



Fig. 4.2 Multi-channel CNN architecture: model with 3 separate streams

End-to-End ASR models are the current state-of-the-art, the implementation in [3] was used for the following representation analysis.

### 4.3.2 Representation Analysis Method

The hypothesis is that the representations learned by the mband and mchan approaches will diverge from representations learned by a standard CNN model, as in Chapter 3, Section 3.6.4. When the output is passed to the transformer layers, it is unclear whether the transformer layers generalise to the different representations. Therefore, the SVCCA analysis index used in Chapter 3, Section 3.6 can be used to directly compare the similarity of the learned representations throughout the transformer layers. The activation embeddings of each neuron, at each epoch were extracted using a separately developed pipeline. To ensure consistency, this was done by passing a controlled input of 100 speech frames through each trained model, and extracting the activation output at each neuron. To aggregate the correlation coefficiency across layers, the spatial dimensions of the activation output vectors were flattened, which provided spatial representation of each neuron.

### 4.3.3 Experiments

The mband and mchan models were trained with the Switchboard dataset [6] with 300 hours of transcribed conversational telephone speech and evaluated on the Hub5'00 [5] and English RT03 [200] test sets. The Switchboard and Hub5'00 data are described in Chapter 3 Section 3.2. The English RT03 set consists of approximately 6 hours of transcripts from conversational telephone speech from the Switchboard and Fisher collection [6, 9].

80-dimension feature filter banks were extracted, using the Kaldi [169] data preparation software, from 25ms windows with a stride of 10ms.

#### Acoustic Modelling Results and Analysis

The mband and mchan models were compiled with the ESPRESSO framework [3]. 80-dimension filter banks were extracted with a 25ms window and 10ms stride to gain more detailed spectral information. The baseline CNN-transformer model has a multi-layer stacked 2-dimensional CNN with dimensionality from 1 to 128 dimensions, with kernel size 3 x 3 and stride 1 and batch normalisation between each CNN layer [173]. The final convolutional layer is projected to a transformer encoder, described in Chapter 3, Section 3.3. The transformer model has 12 stacked encoder layers with embedding dimensions of 512 x 2048 and 6 decoder layers.

Table 4.1 CNN-transformer [1] architectures performance on Hub5'00 Switchboard and Callhome test sets [5]

| Model | Swbd | Clhm |
|---|---|---|
| CNN + transformer | 10.7 | 20.2 |
| Mchan CNN + transformer | 10.4 | 20.4 |
| Mband CNN + transformer | 10.5 | 20.5 |
| Mband CNN + dropout + transformer | 10.6 | 20.2 |

As can be observed in Table 4.1, the mband and mchan models perform comparatively well to the baseline CNN-transformer model on the Hub5'00 test sets [5]. The mband

Table 4.2 CNN-transformer architectures performance on RT03 Switchboard and Fisher test sets

| Model | RT03 S | RT03 F |
|---|---|---|
| CNN + transformer [1] | 21.2 | 13.3 |
| Mchan CNN + transformer | 23.5 | 15.2 |
| Mband CNN + transformer | 23.3 | 14.9 |
| Mband CNN + dropout + transformer | 23.5 | 15.5 |

model achieves a lower WER on the Switchboard test set but slightly worse on the other test sets, while the mchan model performs similarly on both test sets. A mband model with dropout regularisation of 0.1 for each band was also included, in an attempt to improve the generalisation of the network, based on work from [166]. While this showed to improve the performance on the Callhome test set, the Switchboard set showed no improvement.

Furthermore, as can be observed in Table 4.2, the performance of the mband and mchan models are both also worse using the RT03 test sets [200]. The mband model with dropout had the highest WER on the RT03 Fisher (RT03 F) set and on the RT03 Switchboard set (RT03 S). The mchan model had slightly better WER results than the mband models but the results were very similar, while the basline CNN transformer model reached a much lower WER for both RT03 test sets.

Figure 4.3 displays the WER over the validation set across epochs. It can be observed that, initially there is a large spike in WER on all models, although this is significantly higher on the baseline CNN-transformer model. The mband with dropout and mchan models had the smallest spikes in WER during training, which could be partly attributed to the regularisation effect of the dropout parameter with the mband model. All models converged to roughly the same error rate on the validation set a similar epoch.

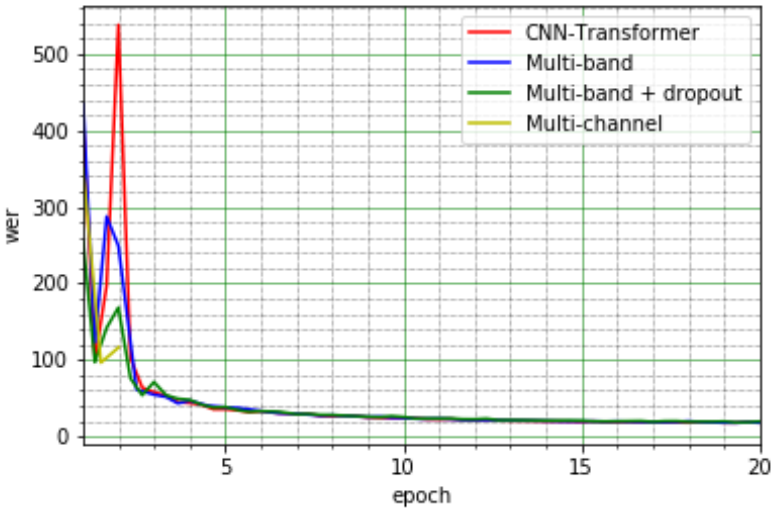
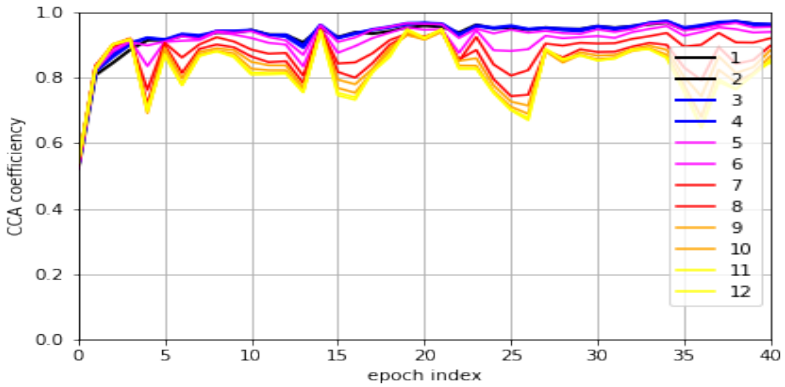


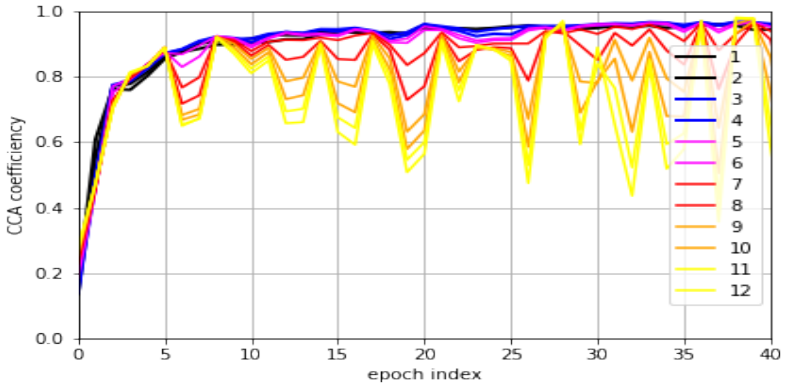

Fig. 4.3 Validation set WER across all models during training on Switchboard data [6]

**Analysis**

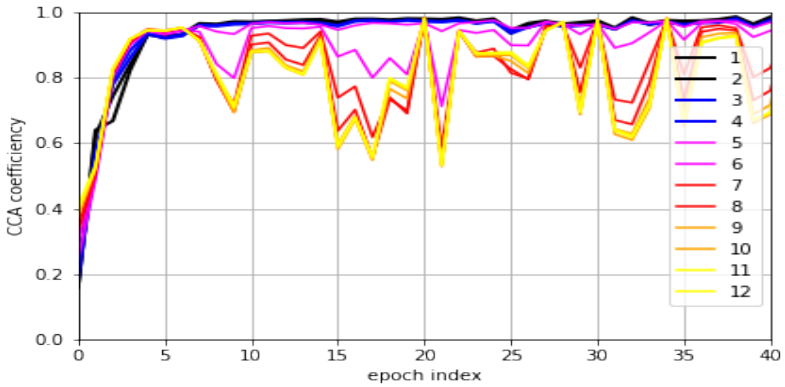
The graph in Figure 4.4a shows the neural representation correlation coefficiency across the layers of the baseline CNN-transformer model through training. There is a hierarchical behaviour to the representation correlations observed without clear convergence even in the earlier layers of the network. The unstable curve suggests that parameter re-weighting for the context information is occurring in the deeper layers, represented by the yellow gradient lines. Very similar patterns can be observed through the mband and mchan models, Figures 4.4b and 4.4c respectively, as both present the same instability of the deeper layers throughout the epochs during training. However, one of the only distinctions is that the convergence of the earlier layers appears to occur earlier, at epoch 5, with the mchan model. Despite these small differences in the neural representations across the earlier layers of each model, the models performed similarly on the Hub5'00 test set [5].




(a) Baseline transformer [1] SVCCA correlation coefficients




(b) Multi-band transformer (mband) SVCCA correlation coefficients




(c) Multi-channel transformer (mchan) SVCCA correlation coefficients

Fig. 4.4 Implementations of different transformer models' correlation coefficients as performance converges with Switchboard data [6]

### 4.3.4 Discussion

The scenarios in these experiments are set with CNNs, multi-channel CNNs, and multi-band CNNs to explicitly distinguish the input layers for the following encoder-decoder transformer layers. The mband model explicitly modelled different frequency bands of the acoustic signal separately and aggregated them. The mchan model learned different representations of the same acoustic signal with a MoE approach to aggregate these representations. Although there was a small difference in convergence speed during training, the overall representation learning and performance remain similar on the Switchboard training set [6]. Thus it can be hypothesised the transformer layers adapted different types of input representations to a similar average representational space, which highlights the memorisation capability of the transformers rather than the generalisation. The performance of the CNN, multi-channel CNN and multi-band CNN models varied in the Callhome and RT03 test sets [5, 200]. As the models were trained with Switchboard data, if the transformer layers had generalised the input acoustic signal and the target categorical lexicon distribution mapping, the result patterns on other test sets should have been similar. These performance results, combined with the analysis showing the similar representation correlations through all models, indicate that the transformers do more memorisation than generalisation with the training data.

As the approaches were trained with the same dataset, it is unclear whether the models would learn a similar representation space for different domains of speech data. There has been little research conducted regarding the representation spaces learned across corpora and the representation dependencies upon performance across corpora.

## 4.4 Cross-Corpora Modelling Analysis

End-to-End ASR frameworks are typically data-driven and often fine-tuned to the corpora in order to improve the recognition performance [62, 3, 63]. The ability to use larger amounts of

training data can also significantly improve the recognition performance [201] however, this can be limited by computational resources or data availability. These factors have influenced techniques to improve the representation capacity of the network structures [202, 166]. By learning representations of speech that are robust across various acoustic conditions and variability, inter-speaker and intra-speaker variance, the general recognition performance of a model should improve without the requirements for increased resources.

The following experiments aim to provide some explainability regarding the dependencies of internal acoustic representations. Despite numerous variations of acoustic modelling structures, there has been little exploration of the ambiguous internal dependencies and their relationship to model recognition performance across different speech corpora. The cross-corpora modelling analysis aims to explore the relationship between performance and salient structures of attention-based End-to-End ASR models with cross-corpora training data. These insights are important for the development of models to exploit hierarchical dependencies and improve recognition performance. Using different corpora and recording scenarios, it is possible to develop diverse representations within the same model structures and observe the changes in these representations. It is hypothesised that statistical analysis can also be used to highlight neural representation dependencies and interpret their ability to generalise, by observing the relationship between the correlation coefficients of neural layers and their cross-corpora recognition performance.

### 4.4.1 Experimental Setup

The LSTM encoder-decoder architecture from [2] and transformer encoder-decoder architecture from [1] were used as the ASR models. Further details regarding the architecture of this approach can be found in Chapter 3, Section 3.4. The modelling approaches were kept the same for the experiments in order to directly compare the learned representation spaces for different datasets.

The CNN architecture from [173] was used in combination with the LSTM and transformer architecture for feature extraction. This model was comprised of 4 stacked 2d convolutions, with kernel size (3,3) on both the feature and time axes, stride of 1 and batch normalisation between each layer. For the LSTM model, the convolutional layer was projected to the 3 bi-directional encoder layers, which had $2688 \times 320$ neurons in the first layer while the second and third layers had a dimensionality of $640 \times 320$. The dimensions of the 3 layer decoder consisted of $688 \times 320$ for the first layer with the second and third layers at $960 \times 320$.

The same CNN architecture was used for acoustic feature extraction with the transformer model [1]. The final convolutional layer was then projected to the 12 stacked encoder blocks with embedding dimensions of $512 \times 2048$ and 6 decoder layers with positional embeddings.

### 4.4.2 Data

For the cross-corpora investigations, three of the most frequently published on datasets for ASR were chosen: Librispeech [8]; Switchboard [6] and Wall Street Journal (WSJ) [7]. All datasets are US-English to avoid cross-lingual incompatibilities during recognition. The Switchboard corpus contains conversational telephone speech, Librispeech is a compilation of read audiobooks and WSJ contains read news. This variation of domains is hypothesised to lead to the generation of speaker-specific and domain-specific contextual embedding representation during training. The test sets are the Hub5'00 sets [5], referred to as *Swbd* and *Callhome*. To ensure performance results and network structures were completed by training on proportionate data, the model trained with Switchboard used up-sampled data (to 16kHz).

The Librispeech dataset [8] consists of 1000 hours of read English audiobooks, mostly from Project Gutenberg [203]. The data has been segmented to be aligned between the read speech and the book text. The training data of the full dataset is split into a 360 hour *clean* set, and 100 hour *other* set, where other refers to a predetermined challenging set of data for

automatic systems to recognise. The models in the following experiments that are trained with Librispeech [8] were trained using the *train-clean-360* subset and not the full training set. This will have some impact upon the overall performance of these models, although these steps should make the experiments more comparable for the analysis process, rather than being affected by training resources. The remainder of the dataset is split across development and test sets that are not utilised for the following experiments.

The WSJ dataset is a compilation of read news articles from the WSJ archive from 1987 to 1989, often referred to as WSJ0 and WSJ1. The total number of speakers reading the articles were split evenly across males and females. The article texts are between 5000 and 20000 words with variable perplexities, speaker-dependent sets and speaker independent sets. The speaker dependent distinction was given to the data in order to train systems for speaker adaptation. The training set for the models trained with WSJ [7] is the 71 hour *si284* set (284 speakers), with the *Dev93* set for validation and *Eval92* for testing. Naturally, further datasets could be suitable for cross-corpora training also and could be explored in future research. Training End-to-End ASR models can be computationally expensive and these results candidly aim to provide initial insights into the learned representations within the model structures.

### 4.4.3 Results

Table 4.3 WER of all End-to-End ASR models across the Hub5'00 [5], WSJ [7] and Librispeech [8] test sets

| Model | Trained | Swbd | Callhome | Dev93 | Eval92 | Test-clean | Test-other |
|---|---|---|---|---|---|---|---|
| LSTM | Switchboard | 13.6% | 24.4% | 26.5% | 25.0% | 32.6% | 62.5% |
| LSTM | WSJ | 77.7% | 82.0% | 15.3% | 12.9% | 41.2% | 71.1% |
| LSTM | Librispeech | 69.9% | 76.1% | 29.6% | 25.9% | 14.7% | 38.9% |
| Transformer | Switchboard | 10.7% | 21.1% | 28.0% | 33.3% | 40.7% | 64.2% |
| Transformer | WSJ | 72.0% | 78.6% | 17.6% | 14.2% | 40.2% | 63.0% |
| Transformer | Librispeech | 63.3% | 70.7% | 26.1% | 22.1% | 14.0% | 33.0% |

The focus of this experiment is to explore the variability among cross-corpora speech representations and not to optimise ASR performance over all corpora. Therefore, the same model architecture has been used with same number of parameters to train all the models and test on a cross corpora regime.

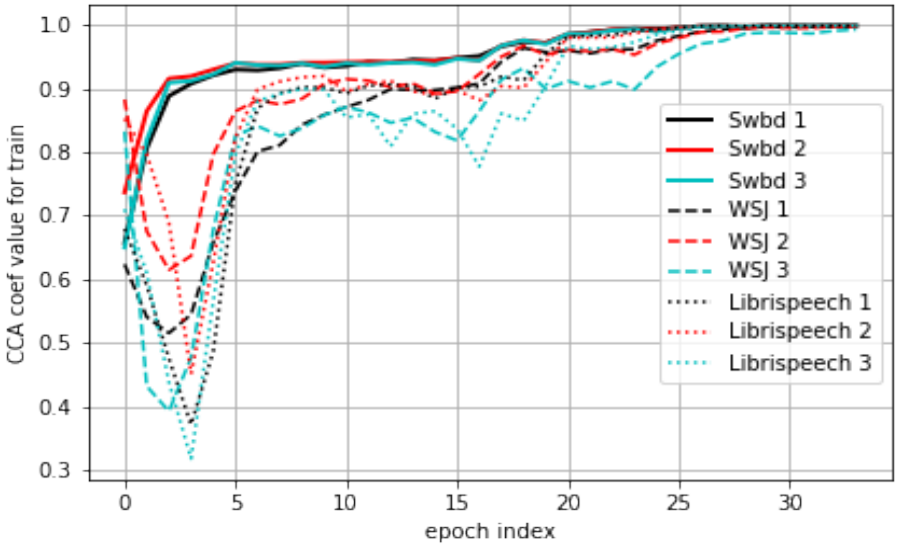
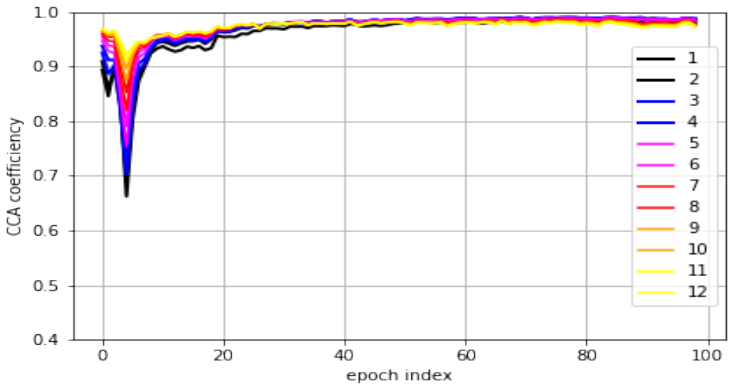


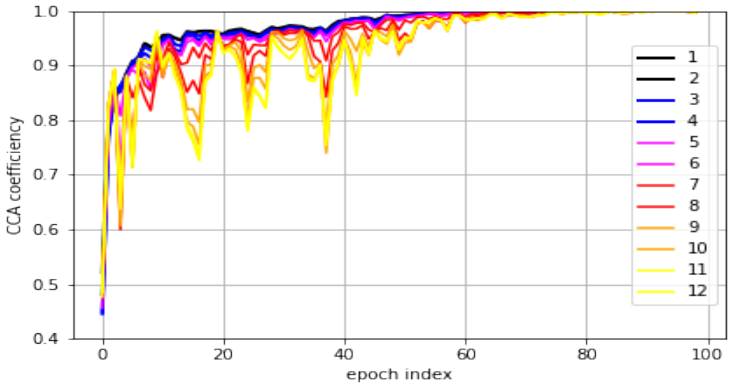

Fig. 4.5 LSTM model [2] correlation coefficients through epochs. The legend depicts the data the model was trained with and the index of the model layer

Table 4.3 shows the performance results, for each LSTM and transformer model trained with the corpora and evaluated across all of the test sets. All of the models performed relatively poorly on the Librispeech *Test-other* set (over 33% WER), likely due to the more challenging speech data not present in any training sets. The models trained with WSJ [7] and Librispeech [8] also resulted in poor recognition performance on the Callhome and Switchboard sets [5]. It can be determined that all of the models struggled to recognise conversational telephone speech unless specifically trained with this type of data, shown by results on the Switchboard and Callhome test sets [5] being over 60% WER.

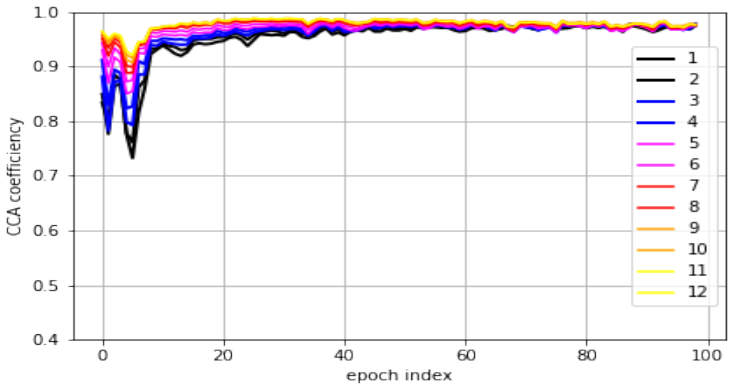
The LSTM models had slightly worse performance than the transformer models when testing with conversational speech test sets [5], however, the models trained with WSJ [7] and Librispeech [8] have such significantly poor results that it's ambiguous to derive whether




(a) Transformer model SVCCA correlation coefficients when trained with WSJ data




(b) Transformer model SVCCA correlation coefficients when trained with Switchboard data




(c) Transformer model SVCCA correlation coefficients when trained with Librispeech data

Fig. 4.6 Transformer model SVCCA correlation coefficients through time, as performance converges, when trained with WSJ [7] (**a**), Switchboard [6] (**b**) and Librispeech [8] (**c**). The legend depicts the index of the neural layers

either model outperformed the other. Across the *Dev93* and *Eval92* test sets the LSTM models trained with Switchboard and WSJ showed slight performance improvement to the

transformer models. Finally, across the Librispeech *Test-clean* set, the LSTM model trained with Switchboard data achieved lower WERs% than the transformer model also trained with the same data, however only the models trained with the Librispeech data managed to reduce the recognition error rate significantly. These results indicate that training with conversational speech and evaluating on cross-domain data using an LSTM model [2] reaches a lower WER than training with read speech and evaluating using cross-domain data using a transformer model [1]. The poor results across all test sets compared to other published models [62, 133] also suggest that the modelling approaches require tuned parameters in order to reach significantly lower error rates. This is explored further in Chapter 5, Section 5.3.3.

Figure 4.5 shows the correlation coefficients of the neural representations through epochs for the LSTM models [2] trained across the datasets. This aims to show a representation of how each layer of the models changes through time and provide a comparison between each model. The solid lines represent the layers of the model trained with Switchboard [6] data, the dashed lines represent the layers of the model trained with WSJ [7] data, and the dotted lines represent the layers of the model trained with Librispeech [8] data. The black lines are the 1st layer of the models, the red lines are the 2nd layer of the models, and the blue lines are the 3rd layer of the models. Despite the random initialisation of the neural layers, there appears to be more similar representations learned within the neural layers of the models trained with WSJ and Librispeech, shown by the parallel pattern of each layer in Figure 4.5. Both of these models' neural representations have parallel correlations through the epochs, while the same model trained with Switchboard [6] data does not exhibit the same pattern, especially during the initial 10 epochs. The LSTM model trained with Switchboard reached a lower WER on the Hub5'00 test sets [5] and the the representations of all neural layers are very similar across epochs. The LSTM model trained with Librispeech data reached the lowest WER on the Librispeech test sets while the representations in layer 1 have a lower correlation coefficiency

in earlier epochs, then become more correlated in later epochs. The representations in layer 3 of the LSTM model trained with Librispeech remain more uncorrelated than the other layers but this model only reached a lower WER for the Librispeech test sets and not the other test sets. There were similar results and representation correlation coefficiency for the LSTM model trained with WSJ and similarly this model only achieved lower WERs on the WSJ test sets. Overall the LSTM model trained with Switchboard data seemed to generalise the best and the representation correlation coefficiency of the layers is higher.

Figure 4.6 displays the correlation coefficients within the transformer model [1] representations. Similar to the LSTM model, the model trained with Switchboard data [6] exhibits contrasting behaviour to the models trained with Librispeech [8] and WSJ [7]. Layers 1-5 of the transformer model trained with Switchboard seem to be more correlated throughout training than layers 6-11. Layers 6-11 also appear to be less consistently correlated, but still hierarchically correlated through epochs than for the other transformer models. All correlations within the layers of the models trained with WSJ and Librispeech seem to be relatively parallel but in mirrored order to the model trained with Switchboard, with the initial layers being less correlated than the deeper layers. The transformer layers in the model trained with Librispeech also display much less variation in the correlation through the first 10 epochs. The transformer models trained with Librispeech and WSJ have similar performance when evaluated with out-of-domain data and the representations within the transformer layers have a higher correlation coefficiency. This indicates that the transformer models are not creating generalised representations, a similar result to the results shown in Section 4.3.3.

### 4.4.4 Discussion

The neural representation analysis in Section 4.4.3 found that in the layers of the LSTM model [2] trained with Switchboard [6] (Figure 4.5), the internal representations among

layers over epochs were more highly correlated than the layers of the models trained with WSJ [7] or Librispeech [8]. [176] suggested that architectures with more significant variation within the higher layer neural representations have poorer recognition performance, which appears to corroborate with the results in Table 4.3. The results also suggest that model parameters would need to be adapted for the particular dataset used for training in order to improve the performance. [166] showed that models with different amounts of augmentation have improved recognition performance with different data domains, such as noisy or conversational speech data. Modelling approaches in [62, 3, 63] tune parameters, such as neuron dimensionality, to datasets. These tuned model parameters may affect the correlation within the layers and contribute to representation spaces that are more optimal when evaluating with specific data.

The analysis observations between the models trained with read-speech show that the neural representations are learned quite differently to the representations analysed in models trained with conversational telephone speech, however using the correlation index SVCCA [177] it was possible to highlight the neural representation similarities and distinctions. The correlation coefficients within the LSTM layers of the models trained with read-speech in Figure 4.5 also appear to show hierarchies of learned representations. The SVCCA correlations shows a pattern of convergence through the network training, which corroborates the theory that the neural layers learn the latent representations by first maximising the mutual information between the input data and the latent representations, then minimising the mutual information between the layer representations and output categorical distribution [178].

The models trained with Switchboard appeared to generate overall more generalised latent representations within the neural layers compared to the other models, as can be seen by the recognition results across all the datasets. This is potentially attributed to features of conversational speech, such as hesitations or incomplete utterances, that can contribute

to model robustness. However, the transformer models seemed to not learn generalised representations, despite the similar correlation coefficiency observed for the models trained with Librispeech and WSJ. This corroborates with findings in [25].

## 4.5 Summary

Chapter 4 aimed to explore how to represent acoustic information for End-to-End ASR by using mixture of experts approaches and parameter augmentation. Multi-band and multi-channel CNN-transformer models, described in Section 4.3, have been implemented for an End-to-End ASR task, with comparable results to a baseline CNN-transformer model, shown in Section 4.3.3. These results showed that current developments in acoustic modelling techniques do not directly translate to improved performance for an End-to-End ASR task. In order to observe whether the chosen modelling approach was learning an enhanced representation space, analysis of neural representations across the model layers was conducted, also in Section 4.3.3. The analysis results provided insight into the potential memorisation behaviour of the transformer model, which was discussed in Section 4.3.4. Future extensions to this work were proposed regarding the analysis of neural representations within End-to-End models on augmented or noisy data to observe the properties of different layers.

Secondly, a cross-domain analysis has been undertaken for an End-to-End ASR task with results and discussion on the performance of LSTM and transformer models when varying training data and model structure. Using SVCCA as a correlation index has also highlighted several aspects of the relationships between the models trained across different corpora and the layerwise neural representations whilst relating the impact these have upon recognition performance. Interpretative analysis is important to develop future modelling approaches with meaningful strategies. Expanding the scope of the investigation into the attributes and potential learned features that could be classified within the layers would provide a deeper understanding of the properties of these architectures.

As the analysis results on acoustic modelling for End-to-End ASR models were able to provide some insights and interpretation regarding the memorising behaviour of transformer models, Chapter 5 introduces methods for modelling language with End-to-End models. A similar approach was undertaken, where the LM approach was varied using the same ASR model, in order to observe modelling dependencies and provide an interpretative analysis.

# Chapter 5

# Language Model Representations for End-to-End ASR

## Contents

## 5.1 Introduction

While Chapter 4 explored acoustic modelling, another technique that potentially impacts ASR model performance is the integration of LMs. However, it is not clear what effect the integration of LM techniques with End-to-End models has upon the performance or internal representations. End-to-End ASR models aim to learn a generalised speech representation to perform recognition and there has been little research done to analyse internal representation dependencies and their relationship to training with LMs. This Chapter investigates cross-domain LM dependencies within transformer encoder-decoder models [1] using SVCCA [177] and uses these insights to exploit the modelling approaches and improve recognition performance. Section 5.2 covers methods of integrating LMs within End-to-End ASR models, while Section 5.2.2 focuses on a technique called LM fusion and provides experiments with reference to the performance dependencies. Section 5.2.3 provides some discussion regarding the results, leading into Section 5.3.2 which highlights the reliance of integrating LMs upon the performance of End-to-End ASR models. Analysis in Section 5.3.3 provide interpretative results to visualise the dependencies used to develop models in Section 5.3.3, with improved performance and how further interpretative analysis may aid development of future models for ASR. The findings within this Chapter have been submitted in [26].

## 5.2 End-to-End ASR with Language Modelling

The typical approach to develop a an ASR system has been to use DNNs to model acoustic features in order to enable recognition of graphemes or phonemes, replacing the requirement for distinctly-optimised modules, such as LMs or pronunciation models. Using End-to-End modelling approaches reduces the need for expert domain knowledge as it aims to train the model jointly while adapting to diversity of speech. However, it has become common practise for End-to-End ASR frameworks to include a pre-trained LM integration technique

in order to improve recognition performance [125, 3, 62, 63]. This somewhat contradicts the intention of End-to-End ASR framework applications and it's unclear whether the LM complexity can limited by the joint modelling approach. Techniques have been developed that are able to improve recognition performance of the ASR system but it remains unclear how dependent End-to-End ASR models are upon the LM integration technique chosen.

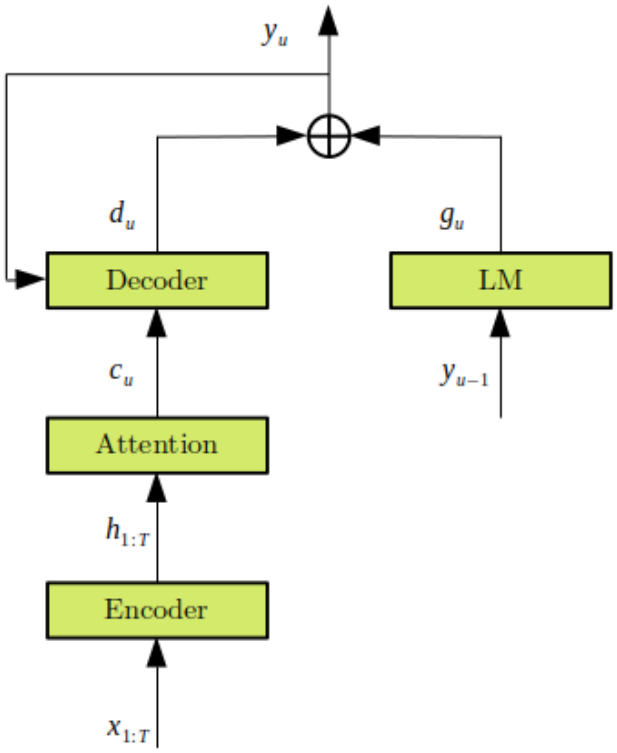



Fig. 5.1 End-to-End ASR encoder-decoder with LM shallow-fusion rescoring

LMs can be integrated and adapted for End-to-End ASR with techniques such as shallow-fusion [204], cold fusion [205] or component fusion [206]. Cold fusion incorporates an external LM using a gating attention-based mechanism. The first stage is to augment the decoder with the gating mechanism, allowing it to selectively utilise the language model at each time-step iteration and enable flexibility. The LM hidden state is then replaced with the LM probability projected into a common embedding space, allowing for the integration of different LMs without state discrepency.

Component fusion [206] integrates an external trained neural network LM with attention-based ASR frameworks by modifying the ASR system to equip an LM component, which can be replaced during decoding. The LM component is incorporated by concatenating the hidden states of the outputs of both the ASR system and the pre-trained LM. Similar to cold fusion, a gate mechanism controls the importance of the contribution of the hidden state of the LM. The main aim of this method was to strengthen the usability of large text-based corpora and allow faster domain adaptation of systems.

Shallow-fusion [204] is a technique to rescore an n-gram LM at decoding to bias the model towards independent context. This was originally presented for neural machine translation where the LM was used to rescore the system's output probabilities at each time-step. In the case of ASR, the ASR system outputs a set of candidate words, then the candidates are scored according to the weighted sum of the scores given by the ASR model and the LM. As shallow-fusion requires lower computational resources, once the LM has been pre-trained, this technique was chosen to be explored in combination with End-to-End ASR models to show how an End-to-End ASR model benefits exactly from LM integration techniques.

With reference to Chapter 2, Section 2.4.2, Figure 5.1 shows how LM rescoring is structured with an encoder-decoder model, where $\mathbf{x}_{1:T}$ are the inputs to the model, $\mathbf{h}_{1:T}$ are the hidden states of the encoder, $\mathbf{c}_u$ are the attention weights, $\mathbf{d}_u$ are the decoder outputs, $\mathbf{g}_u$ is the output prediction of the LM based on the previous output labels $\mathbf{y}_{u-1}$. Shallow-fusion decoding computes the weighted sum of a pair of posterior distributions over sub-words; using one from the ASR model $d_u$ and one from the sub-word LM $g_u$. The sub-word LM is an LSTM-based LM trained with restricted computational complexity, by only keeping the most frequent sub-words and splitting the rest into characters, to enable conversion with low information loss. The LM outputs $\mathbf{g}_u$ is $logP(\mathbf{y}_u)$ and the decoding is rescored by a log-linear

combination:

$$logP(\mathbf{y}_u) = logP(\mathbf{d}_u) + \beta logP(\mathbf{y}_u) \tag{5.1}$$

where $\mathbf{y}_u$ are the output labels, $\mathbf{d}_u$ is the decoder output and $\beta$ is a scalar LM fusion weight.

Another technique called label smoothing [207, 208] has been used across multiple domains, as well as ASR, to calibrate and improve model recognition accuracy. Label smoothing modifies the CE loss with a weighted mixture of one-hot targets from the dataset. There are 3 common types of label smoothing: uniform smoothing is where the mixture is $(1-p)$ of the one-hot targets and the remainder of the probability is distributed uniformly across the vocabulary; unigram smoothing is a mixture using an LM trained on gold transcripts; and temporal smoothing is a mixture using the distribution of neighbouring tokens in the transcript. These approaches are used to aid the beam-search process and to help the model recover from errors.

### 5.2.1 Related Works

A study in [198] probed language embeddings in attention-based approaches [141, 140] with different types of speech: native read speech, native spontaneous speech and non-native speech. One aspect of their study included the analysis of features within different layers of the transformers to interpret semantic and syntactic features. Some layers of the ASR models learned particular features, which were able to be extracted to improve the performance of a speaker recognition task and a phone classification task. This approach extracted embeddings from each layer of the transformer and then used a probing model to derive a predicted feature to compare with features extracted directly from the raw speech. This study showed the value of interpreting linguistic features for specific ASR model layers for downstream applications.

Similar work from [209] attempted to adapt a LM to differing attention-based End-to-End ASR model domains by pretraining the decoder and augmenting the LM. However, none

of these studies have focused on how End-to-End ASR models could adapt to different LM domains using a current fusion integration and smoothing techniques or whether interpretative insights can be derived from the analysis of layerwise representations.

### 5.2.2 Language Model Fusion Experiments

The integration of LMs within End-to-End architectures has been used to supervise the training optimisation and also for decoding rescoring to aid recognition performance. For the following analysis experiments, a sub-word LM is integrated by shallow-fusion decoding [204, 208] and unigram label smoothing [207] techniques.

An LSTM encoder-decoder model with Bahdanau attention is used in combination with a 4-layer stacked CNN. The CNN layers kernel size is (3,3) on both feature and time axes, stride of 1 and batch normalisation between each layer. The final convolutional layer is projected to 3 BLSTM encoder layers, which had 2688 x 320 neurons in the initial layer and 640 x 320 neurons in the subsequent layers. The decoder dimensionality is 688 x 320 in the initial layer and 960 x 320 neurons in the 2 subsequent layers.

The ASR model was trained with the *si284* WSJ training dataset [7], described in further detail in Chapter 4 Section 4.4.2, and decoded using shallow-fusion with an LSTM word-based LM from [15]. The LSTM structure consists of 3 layers of 1200 x 1200 neurons. The perplexity on the *Dev93* validation set is 72.48 and the perplexity on the *Eval92* test set is 59.55.

As can be seen from Table 5.1, the models with LM fusion weighting at 0.7 to 0.9 performed significantly better. The amount of substitution errors reduces for both test sets decreases as the fusion weighting increases, however from a fusion weighting of approximately 0.6 the amount of deletion and insertion errors increases slightly. This could be due to the LM hypothesis not taking into account longer sequence context that the ASR system is better able to handle. The model using an LM fusion weighting of 0.8 achieved the lowest

Table 5.1 LM fusion experiments using an LSTM model trained with WSJ and LM from [15]

| **LM** | **EVAL92** | | | | **DEV93** | | | |
|---|---|---|---|---|---|---|---|---|
| **Fusion $\beta$** | **WER%** | **Sub%** | **Ins%** | **Del%** | **WER%** | **Sub%** | **Ins%** | **Del%** |
| 0.0 | 11.98 | 9.57 | 1.31 | 1.10 | 14.65 | 11.82 | 1.35 | 1.48 |
| 0.1 | 9.27 | 7.11 | 1.15 | 1.01 | 10.97 | 8.68 | 1.03 | 1.25 |
| 0.2 | 7.28 | 5.62 | 0.94 | 0.73 | 9.44 | 7.40 | 0.97 | 1.07 |
| 0.3 | 6.56 | 4.87 | 1.05 | 0.64 | 8.63 | 6.63 | 0.87 | 1.13 |
| 0.4 | 5.60 | 4.22 | 0.85 | 0.53 | 7.68 | 5.89 | 0.78 | 1.01 |
| 0.5 | 5.12 | 3.81 | 0.82 | 0.50 | 7.06 | 5.37 | 0.77 | 0.92 |
| 0.6 | 4.66 | 3.54 | 0.73 | 0.39 | 6.45 | 5.02 | **0.69** | 0.74 |
| 0.7 | 4.39 | 3.33 | 0.67 | 0.39 | **6.28** | 4.83 | 0.72 | **0.73** |
| 0.8 | **4.11** | 3.17 | 0.64 | **0.3** | 6.41 | **4.72** | 0.78 | 0.91 |
| 0.9 | 4.22 | **3.15** | **0.6** | 0.46 | 6.55 | **4.72** | 0.90 | 0.92 |

WER on the *Eval92* test set while the model with an LM fusion weighting of 0.7 achieved the lowest WER on the *Dev93* test set.

### 5.2.3 Discussion

The shallow fusion decoding experiments aimed to explore how recognition performance changed when rescoring a model with shallow fusion. The results shown in Section 5.2.2 suggest that the End-to-End ASR model is not powerful enough to model linguistic context alone and recognition performance can be improved by up to 77% relative by incorporating an LM using shallow fusion. The performance is improved when the fusion weight is increased to an average of 0.75, while any higher weight degrades recognition performance, shown by an increase in insertion and deletion errors in Table 5.1. Incorporating a LM with shallow fusion reduces substitution errors, while not as effective at reducing insertion or deletion errors. These results correspond with results from [210] that showed unigram smoothing acts as a regulariser to penalise low entropy predictions.

By simply assessing the performance results, it remains unclear how the LM affects the properties of the ASR model. It is also unclear if End-to-End ASR models can be adapted

to improve performance on specific domains or whether the properties are independent of cross-domain linguistic features.

## 5.3 Cross-Domain Language Modelling

As transformer encoder-decoder modelling approaches achieve state-of-the-art results for End-to-End ASR, and transformer models have been utilised for the previous studies in Chapters 3 and 4, the subsequent experiments aim to understand whether ASR models trained with cross-domain LMs learn a similar representation space. It is hypothesised that it would be possible to observe how representations in transformer models adapt to out-of-domain LMs by analysing correlation of the representation across layers. Put simply, the relationship between End-to-End ASR performance and the LM dependent neural representations is explored. Using unmatched sub-word LMs, it is possible to observe the dependencies of the layerwise representations and observe the impact of variations. Experiments in Section 5.3.3 show that observing the representation dependencies is important to develop intuitive modelling approaches and improve recognition performance.

The implementation of cross-domain LMs in End-to-End ASR models has been defined in Section 5.2 and the representation analysis experiments are shown in Section 5.3.3. Experiments analysing the adaptation of transformer model parameters are conducted in Section 5.3.3 with the results discussed in further detail in Section 5.3.4. Altogether, this chapter provides analysis of the modelling approaches affecting contextual LM dependencies and ASR performance, and can be used to create or adapt better performing End-to-End ASR models and also for downstream applications.

### 5.3.1 Related Works

Despite numerous variations of modelling approaches, there has also been little exploration of the internal model representations, and their relationships, in order to model recognition performance across different LM domains. It is unclear how the internal dependencies of the End-to-End models handle latent LM representations and whether there are similar learned representation spaces that are robust across different domains. By training models with cross-domain LMs, it is possible to observe these dependencies by comparing models with SVCCA analysis.

Layer-wise analysis of models has been used to interpret modelling approaches and relationships between representations in multiple domains [176, 184, 24] and in Chapters 3 and 4. SVCCA analysis techniques have been used to highlight neural representations with respect to their ability to generalise to different acoustic conditions, by observing the relationship between the correlation coefficients of neural layers during training [25].

### 5.3.2 Experimental Setup

The following experiments use a transformer-based End-to-End ASR model from [1], which consists of 12 stacked transformer encoder blocks with embedding dimensions of $512 \times 2048$ and 6 decoder layers with positional embeddings. A CNN front-end is incorporated with the transformer layers for feature extraction. The input features were 80-dimensional Mel-filter banks with a 10ms stride and 25ms window.

The framework developed in [24] was utilised to investigate the relationships between internal dependencies and retain the models during training for analysis. This approach is described in further detail in Chapter 3, Section 3.6. For all the experiments, the transformer models and LMs were trained using the ESPRESSO framework [3]. The analysis was conducted for all models by extracting the activation outputs of each neural layer of the

encoder for each training epoch. A controlled input of 100 frames of unseen speech data was fed through the layers, whilst simultaneously extracting the activation outputs for each layer.

For the following experiments, two common US-English datasets from differing domains for ASR were chosen: Switchboard (conversational) [6, 9] and [8] and WSJ (read news) [7]. This data is described in further detail in Chapter 4 Section 4.4.2.

### 5.3.3 Cross-Domain Language Modelling Analysis

Correlation analysis of the neural representations across the transformer model layers is used to measure and analyse the changes in correlation when cross-domain LMs are integrated with LM fusion. Figure 5.2 shows the difference in SVCCA coefficients, as training converges, between the encoder layers of two transformer models [1]. This involves calculating the SVCCA correlation of each model and then subtracting one from the other to compare the representations through time. The models were both trained with Switchboard data [6] but one model uses an in-domain Fisher [9] sub-word LM, and the other model uses an out-of-domain WSJ [7] sub-word LM. These models are both trained with sub-word units using SentencePiece [211] and integrated during the training process using the label smoothing method [207] and decoded with shallow-fusion [204], as described in Section 5.2. The correlation analysis shows very little difference in coefficiency between layers 1 to 6 (top graph of Figure 5.2), aside from in the initial epochs which could be attributed to the random initialisation of parameters. This suggests that the neural layers of both of these models are converging to similar representation spaces. However, between layers 7 to 12 (bottom graph of Figure 5.2), the differences in coefficiency are much larger throughout training. This suggests that the representations learned in these deeper layers are more dependent upon the LM domain.

Figure 5.3 displays the standard deviations of the coefficiency between the models trained with cross-domain LMs. This aims to show the variation in coefficiency by layer more clearly,

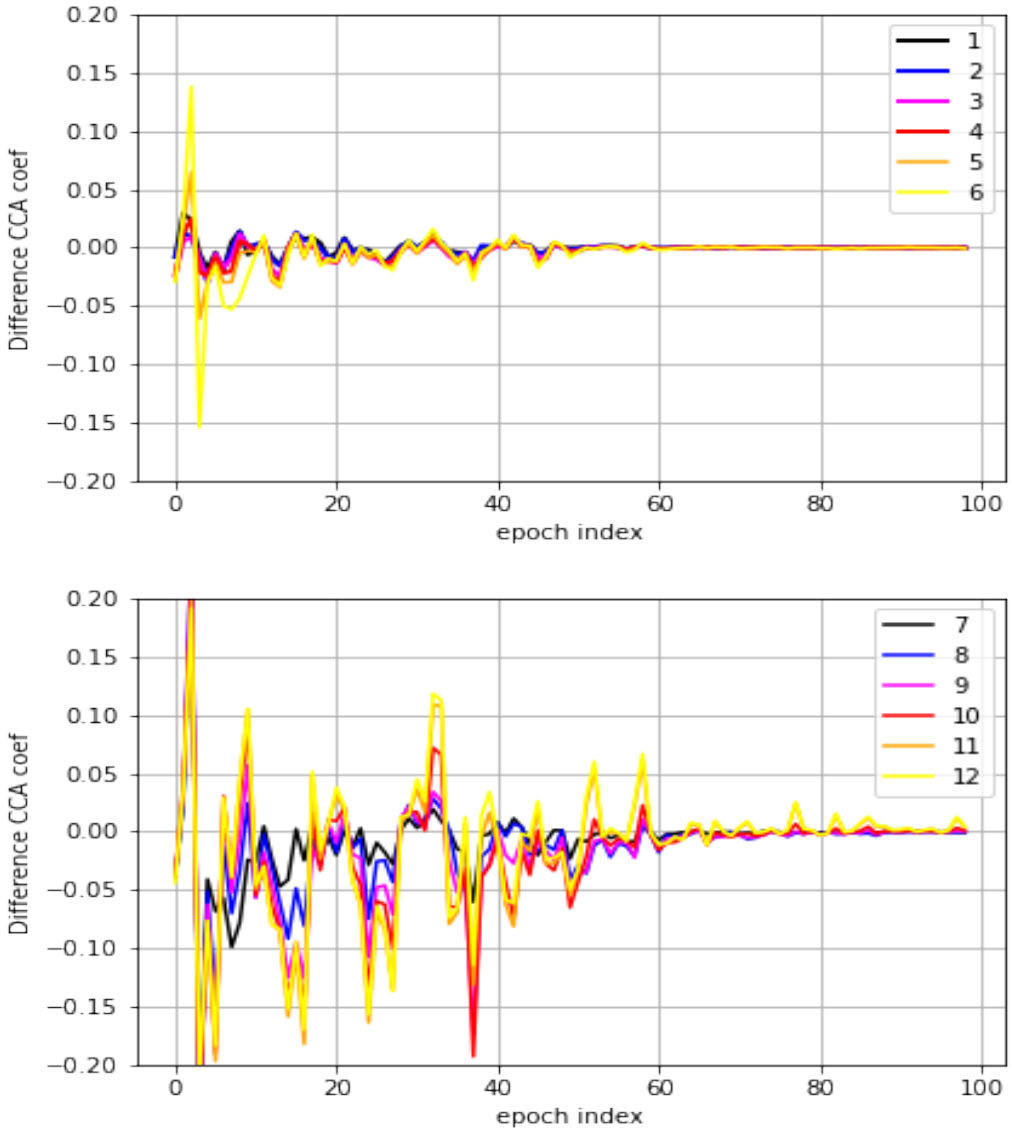



Fig. 5.2 Difference in correlation coefficients as performance converges within transformer layers [1] 1 to 6 (**top**) and layers 6 to 12 (**bottom**), between a model trained with a Fisher [9] LM and a model trained with a WSJ [7] LM

where the standard deviations in layers 10, 11 and 12 are highest. The top graph of Figure 5.3 shows the variance in coefficiency within the neural layers of a model trained without label smoothing [207] or shallow-fusion decoding [204] compared to the model trained with the Fisher [9] sub-word LM. This suggests that a similar observation can be made for LM specific representations, whereby the variance is higher overall and the coefficiency of layers

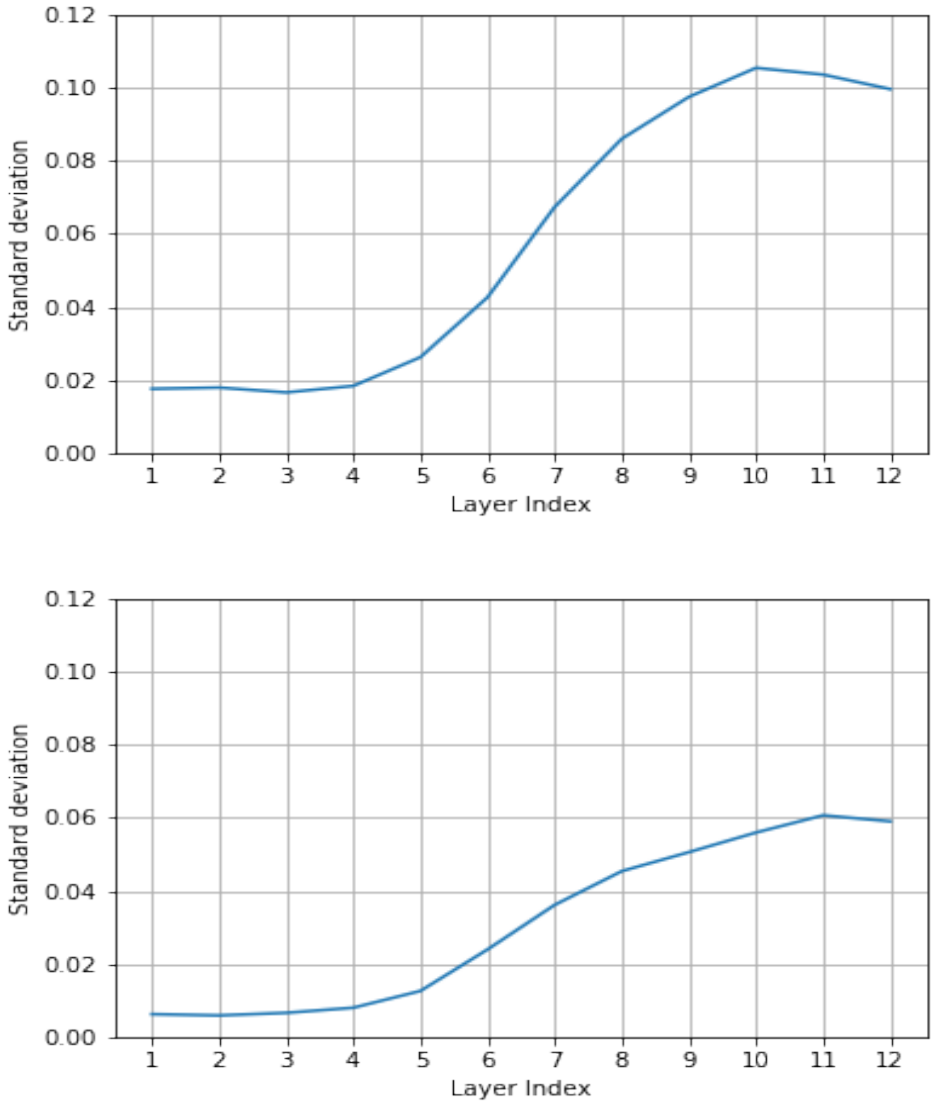



Fig. 5.3 Standard deviation of correlation coefficiency across transformer model [1] layers with and without a LM (**top**) and with unmatched LMs (**bottom**)

8 to 12 deviates the most. The results in Figure 5.3 also imply that layers 1 to 4 have very little dependency on LM representations. These insights suggest that encoder layers 1 to 4 of the transformer model can be frozen when fine-tuning with LMs and the optimisation regime of End-to-End ASR models can be adapted to improve downstream tasks.

Regarding performance, the model that was trained with the Fisher LM reached 9.5% WER on the Switchboard test set and 19.1% on the Callhome test set, while the model

that was trained with the WSJ LM was 10.7% and 21.1% respectively. The differences in recognition performance are attributed to the domains of the LMs and the test sets used for evaluation.

### End-to-End ASR Model Adaptation

Typically, in order to optimise the parameters for state-of-the-art End-to-End ASR models, many iterations are trained with slight parameter modifications. This is partly due to the ambiguity of the optimal dimensionality of layers and parameters required to learn neural representations that contribute to the best performance. Often small changes to the parameters can be made incrementally to observe their effect on model performance. Optimisation of model parameters to specific datasets to achieve the best recognition performance possible, as in several frameworks [3, 62, 63], is referred to here as *tuned*. For example, the dimensionality, number of layers and also the hyperparameters have been observed to impact the recognition performance. As shown in Table 5.2, using a transformer model [1] with the same parameters and composition for several datasets does not achieve the lowest WER across all of the datasets. These *tuned* models are typically reached by extensive hyperparameter optimisation techniques [212], which are computationally expensive and considerably time consuming. For End-to-End ASR, optimisation is typically conducted without providing observational evidence regarding the dependencies of certain parameters upon the recognition performance.

Using cross-corpora correlation analysis while varying the parameters, it is possible to interpret these dependencies in a more meaningful way and provide some observational evidence to reduce the future need for extensive hyperparameter optimisation when developing new models or fine-tuning trained models. By understanding the representation dependencies, this can potentially reduce the computational resources required to improve speech recognition model architectures. Table 5.2 shows the results of 3 transformer models with variations in model parameters that are used in state-of-the-art End-to-End ASR frameworks.

All models are the same transformer-based encoder-decoder architecture with the following variations:

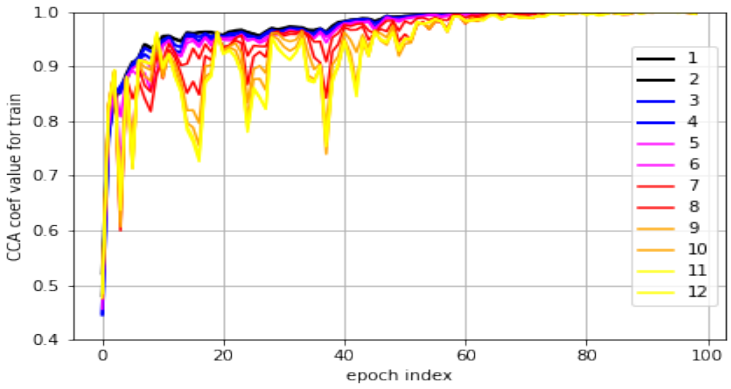
- **Model 1** has an embedding dimension of 512, a feed forward embedding dimension of 2048, 4 attention heads, and an attention dropout of 0.25.
- **Model 2** has an embedding dimension of 256, a feed forward embedding dimension of 1024, 4 attention heads, and attention dropout of 0.25.
- **Model 3** has an embedding dimension of 512, a feed forward embedding dimension of 2048, 8 attention heads, and an attention dropout of 0.1.

To observe the relationship between the learned representations of the adapted models and attribute these adaptations to improved recognition performance with specific data, the model performance was assessed across all test sets, as shown in Table 5.2. For the Switchboard and Callhome test sets (*Swbd*, *Chm*), the recognition performance of model 1 is the best, while model 2 reaches slightly worse performance on the Callhome set and model 3 has the highest WER for both test sets.

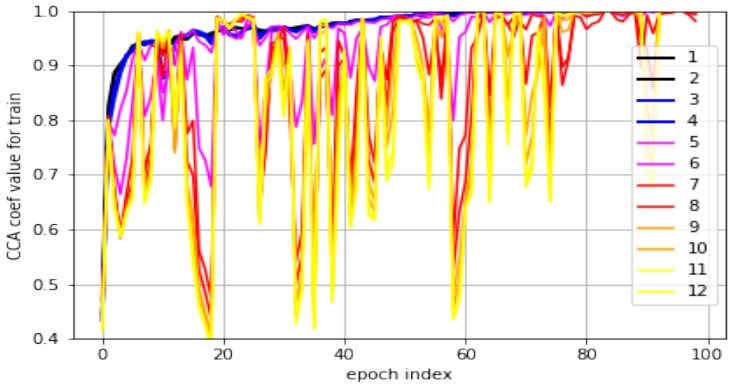
Table 5.2 Transformer model WER on Hub5'00 [5], WSJ [7] and Librispeech [8] test sets with *tuned* parameters

| | **Hub5'00** [5] | | **WSJ** [7] | | **Librispeech** [8] | |
|---|---|---|---|---|---|---|
| **Model** | **Swbd** | **Chm** | **Eval92** | **Dev93** | **Test-cln** | **Test-oth** |
| M1 | **9.5** | **19.1** | 4.59 | 7.54 | 3.5 | 8.51 |
| M2 | 9.6 | 20 | **4.13** | **6.3** | 3.99 | 8.72 |
| M3 | 10.4 | 21.6 | 4.52 | 7.43 | **1.9** | **3.9** |

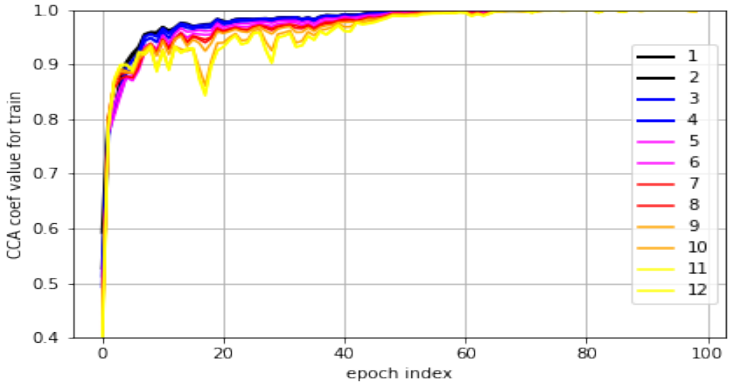
Figure 5.4 displays the SVCCA coefficients for each model trained with the Switchboard dataset. Model 2's mean coefficiency, across layerwise representations, are substantially less correlated than the other models, as shown by Figure 5.4 **b**. The standard deviations of the correlations within these layers also vary significantly higher than Model 1, Figure 5.4 **a**, or Model 3, Figure 5.4 **c**. Model 3's mean coefficiency across layerwise representations is fairly




(a) Transformer Model 1 (M1) SVCCA correlation coefficients when trained with Switchboard data




(b) Transformer Model 2 (M2) SVCCA correlation coefficients when trained with Switchboard data




(c) Transformer Model 3 (M3) SVCCA correlation coefficients when trained with Switchboard data

Fig. 5.4 Transformer layer [1] correlation coefficients as performance converges across all models when trained with Switchboard data [6]. The legend depicts the index of the neural layers

similar for all layers with very small standard deviation. It is observed that correlations within the layers of Model 2 have lower coefficiency, and the recognition performance of this model

is lower for the Hub5'00 test sets [5]. Also, there are little hierarchical coefficiency patterns throughout the layers of model 3, and this model also has a slightly worse performance, which corroborates with results from [24]. Model 1 has lower coefficiency within layers 8-12 and has the best recognition performance.

### 5.3.4 Discussion

The findings in Section 5.3.3 correlate with findings from [198] where semantic and syntax level features of speech are predominantly dependent upon deeper layers of transformer-based models, while acoustic and fluency features are predominantly dependent on the shallower layers. In the case of End-to-End transformers for ASR tasks, the LM-dependent representations are shown to be primarily dependent within layers 7-12. The cross-domain LM-dependent representations are observed within layers 10-12. Further experiments training with the WSJ dataset with Fisher and WSJ sub-word LMs showed very similar observed behaviours across layer coefficiency. These observations can be used to check for possible biases in the modelling process that affect recognition performance, without the need for extensive training requirements, to improve joint optimisation. The analysis also aids in the interpretability of the impact of representation dependencies within End-to-End ASR models.

The experiments in Section 5.3.3, attempt to show these internal dependencies with regard to the model parameters within the same modelling approaches. As shown in Figure 5.4 **b**, Model 2 used shallower embedding dimensions than model 1, shown in Figure 5.4 **a**, which has caused the coefficiency of many of the layers to become highly uncorrelated. Model 3, shown in Figure 5.4 **c**, is observed to have very highly correlated layers, however there are little distinct hierarchies in the neural representations when the attention heads are increased to 8 and the attention dropout is reduced. By adapting the parameters of transformer models, the layers with the most dependency for representing domain-specific information are altered. These changes in hierarchical representations have been observed

to impact recognition performance, and further suggests a relationship between correlated hierarchical representations and the ability for the model to generalise, particularly for cross-domain speech recognition. Increasing the attention dropout is theorised to improve model robustness [166], where typical features of conversational speech are boundary uncertainties and hesitations. In the case of End-to-End conversational speech recognition, the results show that using substantial attention dropout in transformer models is important to produce correlated hierarchies in dependent layers but also utilise a model with sufficient embedding dimensionality that the coefficiency of representations within context-critical layers don't become too uncorrelated.

## 5.4 Summary

This Chapter aimed to investigate how ASR model performance is related to rescoring weight with LM fusion and also to assess whether models trained with different LMs learn a similar representation space. Section 5.2 detailed the methods used to integrate LMs with End-to-End ASR models, while Section 5.3.3 detailed the impact of weighted integration of an LM for an LSTM encoder-decoder ASR model. It was found that increasing the LM fusion weighting significantly improves the recognition performance of the model and thus argued that that ASR model is not powerful enough to model linguistic context alone.

Using the SVCCA analysis framework developed in Chapter 3 Section 3.6, has highlighted several aspects of the relationships between the neural representations, transformer-based modelling parameters and the impact these have upon recognition performance, in Section 5.3.3. The analysis indicated that ASR models trained with different domain LMs learn different representation space. Using the insights on the dependencies of representations across data from different domains, Section 5.3.3 adapted model parameters to improve recognition performance for specific data.

It is argued that interpretative analysis is important to develop future modelling approaches for meaningful improvement strategies. Expanding the scope of the investigation into the attributes and potential learned features that could be classified within the layers would provide a deeper understanding of the properties of these dependencies and how these could be further exploited. The insights into the dependencies of the neural representations can be used for the development of models for few-shot learning and downstream tasks for End-to-End ASR.

# Chapter 6

# Approximating Context Information for Speaker Verification

## Contents

## 6.1 Introduction

Subsequent to exploratory analysis in Chapters 3, 4 and 5, techniques for modelling context within speech, could be applied to speaker recognition models to improve recognition performance. The following Chapter presents an approach to model context information for speaker recognition applications. The first Section 6.2 provides a summarised literature review of speaker recognition systems and the main approaches that have directed this domain. This section also outlines the specific scope that the proposed approach is attempting to address within a speaker verification task. The current advances in context modelling and state-of-the-art speaker verification approaches are compared in Section 6.3. Section 6.4 introduces the proposed modelling approach to improve capturing context information for speaker verification with dynamic convolutions, described in Section 6.4.1. Experiments are described in Section 6.5 that explore the results of the proposed model topology compared to other state-of-the-art approaches for a speaker verification task, whilst also providing results regarding the computation requirements. These results are discussed further in Section 6.6. Finally Section 6.7 provides a summary of the chapter and the observed results. The findings of this Chapter have been submitted in [27].

## 6.2 Background

Prior approaches for speaker identification and verification modelling have involved using Gaussian mixture models (GMMs) [213–215] to map between spoken utterances and speakers. GMMs are probabilistic models comprised of Gaussian density functions, which can be used to model the speaker-independent distributions of features. GMMs were used in approaches, such as [214], in combination with a Universal Background Model (UBM), which

is trained with speech from many speakers in order to provide more robust acoustic references to adapt the target model. Due to the sensitivity of GMM-based models to variance and noise in the data, subsequent research focused on extracting speaker features [216, 217]. [217] introduced i-vectors, which are intermediate representation vectors representing speaker and channel variabilities extracted from a projection matrix. These i-vector speaker features are fixed-length and can be compared directly using a similarity metric such as probabilistic linear discriminant analysis (PLDA) [218]. PLDA scoring also reduces the dimensionality of the vector using linear discriminant analysis (typically to 25% of the original size), in order to further distinct the embedding subspace. The class can then be determined by the maximum probability of the utterance coming from the same class as a known set. For speaker verification tasks, the score is further determined using a metric such as equal error rate (EER), described in [86], which measures the similarity between the FAR and FRR of the classifications. FAR is the percentage of speakers recognised that are incorrectly accepted as the target, while FAR is the percentage of recognition instances where the target speaker is incorrectly rejected. EER is the point where FAR is equal to FRR.

More recently, DNNs have been incorporated with modelling approaches to enable to construction of deeper models, trained with larger amounts of data, to improve speaker recognition [219, 10, 173, 69]. The X-vector approach, shown in Figure 6.1, from [10] uses a time-delay neural network (TDNN) to extract segment-level speaker embeddings. The TDNN layers capture temporal information from the MFCC input features, while the pooling layer aggregates the frame-level information into a single vector. The recording layers provide dimensionality reduction and contain information that represents the entire utterance. Typically the final layer of the recording layers is taken as the x-vector speaker embedding. The output layer provides the probability distribution of each training speaker given the input. The probabilities are then used to optimise the weights of the layers during model training.

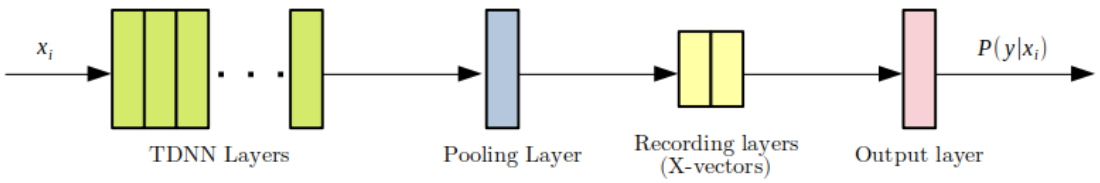



Fig. 6.1 X-vector model architecture from [10]

The x-vector DNN approach was shown to outperform the i-vector approach as it is capable of exploiting large amounts of data [10].

In more recent research, CNN-based approaches have been introduced for image classification tasks [11] and adapted for speaker recognition. ResNets [11], composed of ResNet blocks shown in Figure 6.2, use skip connections over convolutional layers in an to attempt to learn more complex context information from the input data. ResNets were proposed for training deeper neural networks, which previously suffered from the "vanishing gradient" problem and degraded network performance. By attempting to learn a residual mapping (the difference between the input and the target output), using residual connections, information from different neural layers is directly passed to deeper layers, reducing the "vanishing gradient". Skip connections also reduces the number of parameters required by the model.

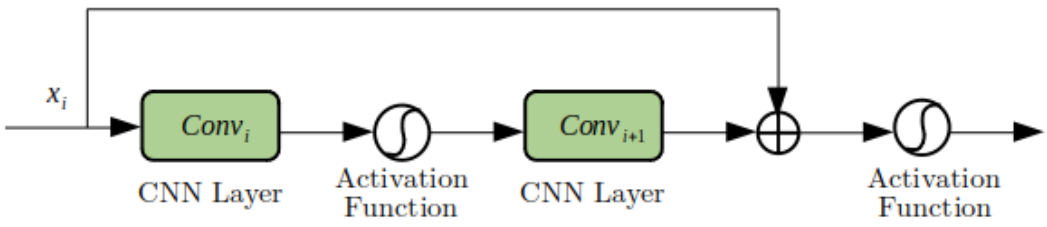



Fig. 6.2 Residual block of ResNet architecture from [11]

Developments in ResNet modelling approaches such as [220] and [17] contain 4 residual blocks between a frame-level representation extraction module and an utterance-level aggre-

gator. The outputs of the residual blocks are fed to attention modules which attempt to learn the channel dependencies prior to the skip connections. Frame-level speaker representations are encoded into fixed-length utterance level representations by the aggregator and followed by softmax layer. For speaker recognition models, this can potentially capture more abstract and intricate features from the speech data, allowing relevant information to be preserved and propagated more effectively.

## 6.3 Related Works

There have been significant developments in modelling speech with respect to expanding the ability to capture contextual information [10, 221, 222]. Adding context information so that models can learn the intricacies of these dependencies, has been shown to improve recognition and verification performance of the system [223, 173]. CNNs, RNNs, LSTMs and Transformers are utilised to capture various context dependencies as DNNs are capable of passing history from previous states to improve the recognition performance of a model. It was hypothesised that simply using deeper architectures to improve modelling is unnecessary, and that simpler introductions of context dependent phoneme models using duration modelling have shown more promising results [224]. Modifications of modelling approaches to add limited contextual information has also been shown to further improve recognition results [162, 225]. [162] modelled local and global context information for an ASR system by replacing the blank symbol in CTC with symbolic characters, such as letter units with apostrophes and capitalisation. This approach was able to model pronunciation and language specific context information and reduce the model output errors where traditionally a LM or pronunciation dictionary would have been used to correct the errors. [225] use triphone embeddings with a DNN for context dependent modelling, which aims to capture latent representations from the input speech frames.

A study on the internal representations were compared across two languages [226], by taking frame-level features generated by a model, and classifying them into phonemes. The technique implemented gave some insights as to how the phonetic information across languages was captured by models of increasing complexity and parameter size. The findings showed that within the different models, the initial CNN layers improved the quality of the phoneme representations, but there is a drop in quality of these particular representations within subsequent layers. These findings suggest that the earlier layers of the network are capturing specific contextual information, which could be due to later layers ignoring phonetic information when trained to output characters, thereby representative of globalised information. Certain linguistic units such as vowels were also more likely to be classified correctly by all network approaches explored in [226] and features extracted from vowels were proposed to improve speaker recognition during noisy conditions in [227]. However, the approaches only regarded phonemes and didn't consider other potential types of context representations that could be derived for speaker recognition tasks.

State-of-the-art speaker recognition systems, such as [10, 228–230], have typically focused on speech enhancement techniques with more training resources, whilst implementing increasingly deeper (more layers) and wider (number of channels) models to improve their verification performance. Speech enhancement techniques aim to improve the quality of the speech, and can be used for speaker recognition modelling to improve performance with more challenging data [231, 232]. The topology of these models and the embedding hierarchies are hypothesised to represent different speaker characteristics. ResNet based models in [11, 233, 17] attempted to learn stronger representations with residual skip connections as this enables the composition of deeper models, with multiple ResNet blocks, and compensating for vanishing gradients. These speaker embeddings can be further distilled by learning the salient regions with an attention mechanism. The DNN models using attention and skip connections [234, 228], such as the Emphasized Channel Attention, Propogation

and Aggregation TDNN (ECAPA-TDNN) model, proved a considerable improvement over the traditional x-vectors and i-vector embeddings [10]. Both RNNs and CNNs have been used to learn temporal dependencies for speaker representations, noting that the CNN-based models have typically produced better performance with fewer number of parameters than RNNs [235].

The current state-of-the-art speaker recognition approaches use TDNNs and attention mechanisms in the convolutional channel outputs, which further improved the performance results [228]. However, verification tasks are still a challenging and computationally demanding task, especially in poor acoustic conditions. Large models, with a large amount of parameters that have been pre-trained using huge datasets perform well [236]; however, training and serving these models is becoming increasingly computationally demanding. Work by [237] introduced the CNN-ECAPA-TDNN, which builds upon the approach in ECAPA-TDNN from [228], where the convolutional front-end allows the network to construct local, frequency invariant features to integrate frequency positional information. In order to enable the network to be invariant to small shifts in the frequency domain and to compensate for the potential intra-speaker variability, 2D convolutions are used to model at a higher resolution. However, this approach also uses large amounts of training data (VoxCeleb1 [16], VoxCeleb2 [17], Librispeech [8], Common Voice [238] and DeepMine [239]), where typically first a pretrained large-scale model is used to then be fine-tuned for state-of-the-art results. The ResNet-based models [11, 17, 220] can suffer from overfitting due to the increases in layer dimensionality and it can also take an excessive amount of time and computational resources (over 30 hours per epoch) to fine-tune the hyperparameters to improve the performance.

As discussed, the general trend for CNN based architectures has been to increase the depth and complexity of the network, while simultaneously increasing training data size for improved accuracy [240, 241]. However, considering the challenges for modelling speech

data, it is becoming necessary to make systems that are more efficient with regard to size and training speed. The main contribution of the following approach is to integrate attention-based dynamic kernels within convolutions for a speaker verification task, which has not previously been explored. The proposed approach uses parallel dynamic convolutional kernels described in 6.4.1, which are able to adjust parameters dependent upon the input attention. Dynamic kernels have shown promising potential for boosting the model representation capabilities without increasing the computational cost [242].

## 6.4 Proposed Modelling Approach

The proposed model builds upon the original ResNet model [11], which uses a 2D CNN based approach. This method can also be integrated into other CNN-based approaches, such as the CNN-ECAPA-TDNN [237] for potentially further improved speaker verification performance without the requirement for larger or pretrained models. The main motivation for using this approach is to improve representation capacity, which is shown in the following experiments to improve verification performance without increasing the computation. This is possible with the dynamic convolution approach as the kernels share the output channels, and it is observed to outperform similar models with increased layers, parameters and training data. Section 6.6 discusses the results of the experimental models with additional details regarding the average computation time of each epoch.

### 6.4.1 Dynamic Convolutions

The dynamic convolution approach proposed in [12], is a technique developed to increase the model resolution capability without requirements for increasing the model depth or width to improve accuracy. *Dynamic* in this case refers to the combination of kernels for the changing input sequence by the application of input dependent attention weighting. This

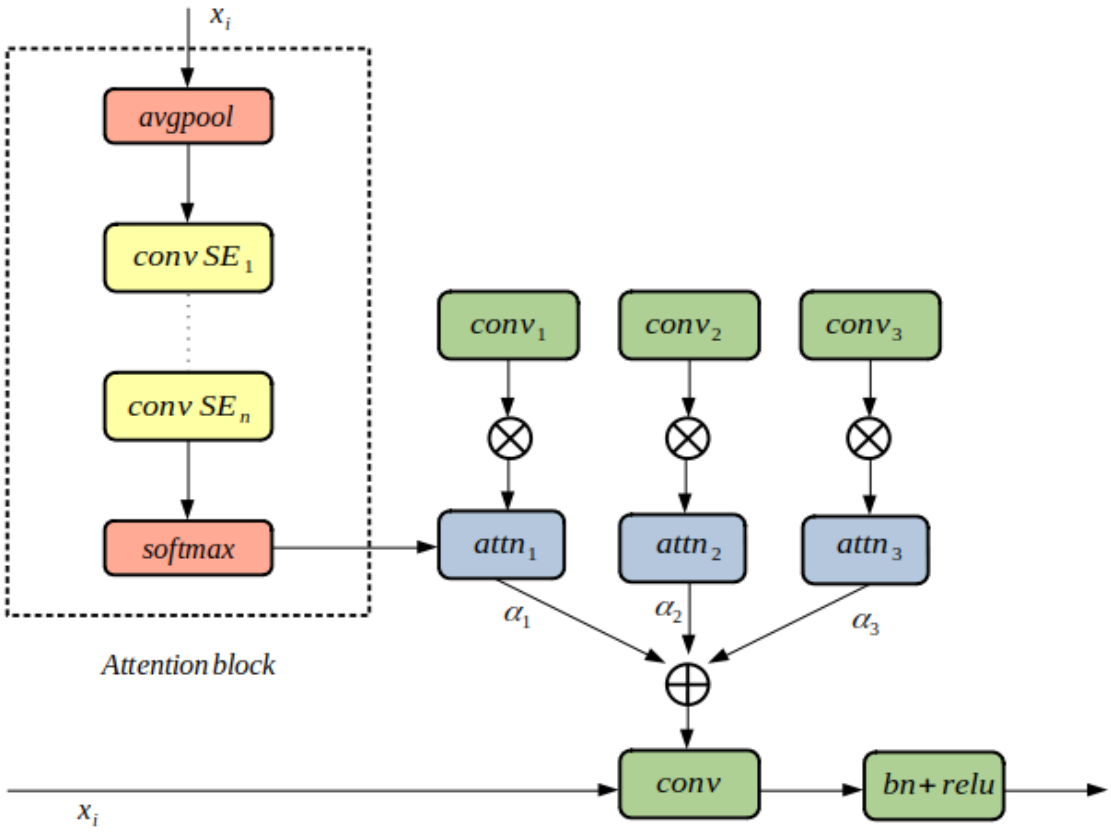



Fig. 6.3 Dynamic kernel convolution block [12], where $\alpha_k$ refers to attention weights for the $k^{th}$ linear function

is integrated by aggregating parallel convolution kernels based on their attention weights, shown in the model diagram in Figure 6.3. The residual dynamic convolution architecture is shown in Figure 6.4. Using a dynamic approach, models from other domains, such as image recognition, have been shown to have greater feature representation capacity for image classification and human pose estimation, while also being more computationally efficient due to the kernels sharing the output channels compared to the typical static convolutional models.

Figure 6.3 describes the dynamic kernel convolution block to show how the input sequence $\mathbf{x}_i$ is propagated to compute the attention weights $\alpha_i$ from each convolution. The computation of the input sequence $\mathbf{x}_i$ to the target label $y_i$ is evaluated during training as a

mapping function $f$ that minimises the loss between the input and target label:

$$f(x) = \mathbf{W}(\mathbf{x})x + b \tag{6.1}$$

where $\mathbf{W}$ is the weight during linear transformation and $b$ is the bias. By adding the attention module, the convolution weights become dynamic and a weighted sum of the convolution kernel information:

$$\mathbf{W}(\mathbf{x}) = \sum_{i=0}^{k-1} (\alpha_i(\mathbf{x}) kernel_i) \tag{6.2}$$

and:

$$b(\mathbf{x}) = \sum_{i=0}^{k-1} (\alpha_i(\mathbf{x}) b_i) \tag{6.3}$$

where $kernel_i$ is the input information in the ith convolution kernel and $\sum_{k=1}^{K} \alpha_k(\mathbf{x}) = 1$. The output is derived after the ReLU activation function and batch normalisation as joint optimisation is required for all kernels. Figure 6.3 shows the squeeze and excitation [243] that is applied to compute the kernel attentions, where the global information is squeezed by attention pooling. The dynamic convolution block is denoted by the term *dconv* in this approach and the *dconv* architecture is shown in Figure 6.3.

### 6.4.2 Proposed Model Topology

Building upon the original x-vector model, the ECAPA-TDNN [228] used hierarchically grouped convolutions rather than 1-dimensional convolutions. In this approach, embeddings from multi-layer residual dynamic kernel convolutions are concatenated, as shown in Figure 6.4. The input features $\mathbf{x}_i$ are separately chunked and fed into hierarchical layers connected with skip connections before passing to the next layer's squeeze and excitation block. The main motivation for using a skip connection is that is a method of compensation to collect information from previous layers and thereby the features learned by the current layer. This

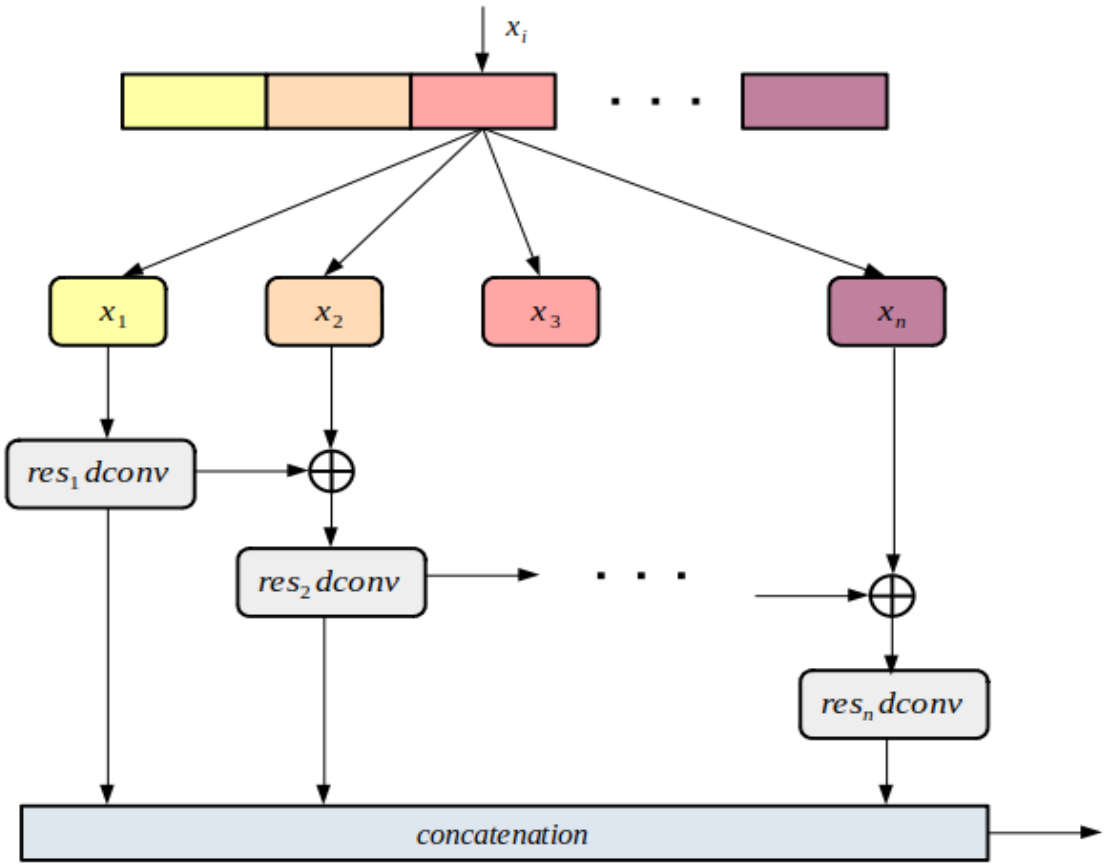



Fig. 6.4 Residual structure of dynamic convolution (**dconv**) blocks

enables the construction of models without the need to increase the model size (number of layers, dimensionality of layers) as the model should be able to learn saliency regions.

After the residual *dconv* layers are concatenated the output passes to a squeeze and excitation layer, shown in Figure 6.5. Squeeze and excitation blocks were used in [12] which adjust the context bound frame-level features per channel over time according to the global utterance properties. A subsequent pooling layer uses channel-wise self-attention to attend to different speaker characteristics at different time steps for each feature map. The weighted standard deviation of channel $\tilde{\mathbf{C}}$ is shown in Equation 6.4. The output of the attention pooling layer is a concatenation of the weights $\mathbf{W}$ and the standard deviation:

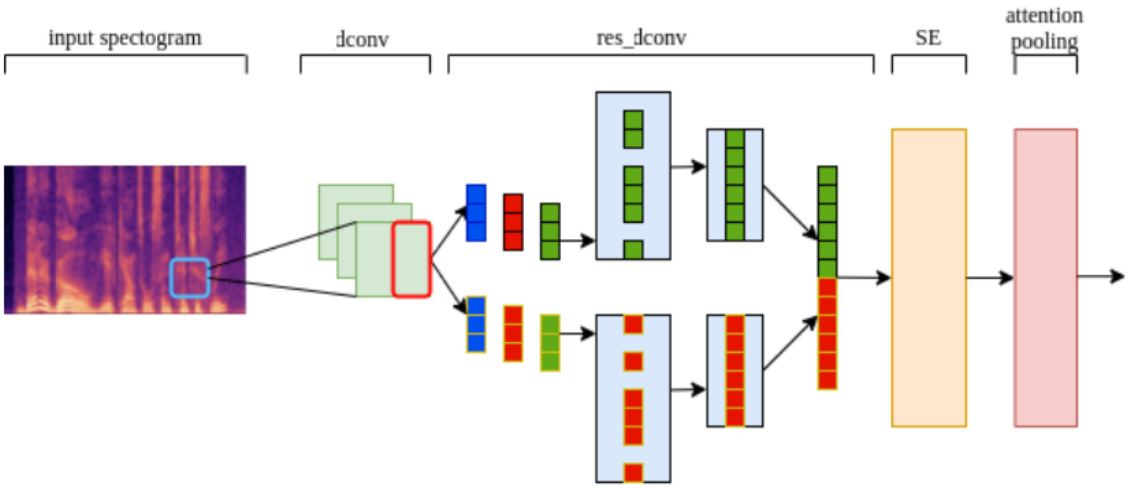



Fig. 6.5 Overall model pipeline of the proposed *dconv* network

$$\tilde{\mathbf{C}} = \sqrt{\sum_{k=1}^{K} \alpha_k(\mathbf{x}) b_k^2 - \mathbf{W}(\mathbf{x})^2} \tag{6.4}$$

The proposed implementation of the architecture is described in Section 6.5.

## 6.5 Speaker Verification Experiment

Speaker verification models aim to determine if a given speaker matches the claimed identity of that speaker. To do this, normalised 80-dimensional log Mel filter bank coefficients were obtained with a 25ms window and 10ms frame shift. The spectograms were normalised by mean and variance on the frequency axis. Random 3 second segments were taken as mini-batches to form an input dimension of $80 \times 300$. In the enrollment phase, a unique embedding is created for each speaker so that the model is able to discriminate between speakers. During verification, the model receives the input speech signal and compares it to the stored embeddings of enrolled speakers. The cosine distance is used to measure the similarity scores, whereby a threshold determines whether the identity is a match or not. Model performance is evaluated using EER, described in Chapter 2, Section 2.2.5.

In order to map the voice spectograms into compact embeddings for computation, relatively shallow *dconv* models were constructed. Models were built with varying layer dimensions and depth, shown in Table 6.1, to control the parameter computation and observe the impact upon verification performance, shown in Table 6.2. The training data for each model was augmented, as described in Section 6.5.1, as this has been shown to create sparsity and attempts to improve generalisation.

The models were implemented using the PyTorch-based Speechbrain framework [63] and run on 1 NVIDIA RTX3060 GPU over 10 epochs. The main model pipeline is shown in Figure 6.5. The details of the model compositions are described in Table 6.1, where variations of the *dconv* models were built with two size dimensions of layers; 1024 and 512 dimensions, at depths of 3, 4 and a version with 5 layers. The number of attention channels for all models remained at 128 and 192 linear neurons. Adam optimisation was used to initialise the network parameters. Each setup's learning rate was set at $0.01 \times 10^{-6}$ with a learning decay of $2 \times 10^{-6}$ up to a value of 0.001. The similarity scores were measured using the cosine distance.

Table 6.1 Experimental architecture setup of *dconv* model implementations

| **Layer Name** | **Channels** | **Kernel** | **Dilation** |
|---|---|---|---|
| Dconv-3 (small) | 512 | 5,3,1 | 1,2,1 |
| Dconv-3 | 1024 | 5,3,1 | 1,2,1 |
| Dconv-4 (small) | 512 (1024) | 5,3,3,1 | 1,2,3,1 |
| Dconv-4 | 1024 (2048) | 5,3,3,1 | 1,2,3,1 |
| Dconv-5 | 1024 (3072) | 5,3,3,3,1 | 1,2,3,4,1 |

### 6.5.1 Data

VoxCeleb1 and 2 from [16] and [17] were used for training and testing the proposed approaches as these datasets have widely reported baselines, and thus have multiple models to compare to. Both datasets are audio-visual clips of human speech taken from interview

videos. The training and development set is split into 1,211 speakers and the testing set is split into 40 speakers for VoxCeleb1-O. The total number of utterances is 153,516 with 116 per speaker on average. The development set and evaluation set for the VoxCeleb datasets contain both monoaural multi-speaker recordings taken from professionally edited Youtube videos, and general conversation. There are numerous challenging aspects that affect recognition within the dataset such as overlapping speech, background noise, music, laughter, applause and singing. Comparative cited research for speaker verification tasks also use the considerably larger VoxCeleb2 dataset [17] for training, which contains 6112 speakers and a total of 1,128,246 utterances, however these training and serving models with this data is computationally expensive.

After showing promising results in [244], frequency and time domain data augmentation was performed for all models using Specaugment [166]. This was used to attempt to increase the amount of diversity in the training data and improve model generalisation. Reverberation was convoluted with the original speech using the RIRs from [245]. The augmentations were randomly chosen between babble, music, noise and reverberation.

## 6.6 Results and Discussion

Contrary to the typical scenario where the models are trained using large amounts of data, commonly with both VoxCeleb1 and 2, and extensive computational resources, here all except one of the *dconv* models were trained only using VoxCeleb1 training data and 1 GPU. The motivation for training models using on VoxCeleb1 is to constrain the computation required, and assess the performance of a model trained with less resources compared to the state-of-the-art approaches which typically use more resources. The *dconv* model trained with both datasets computed one epoch in over 33 hours, while the *dconv* model with the same parameter size trained with only VoxCeleb1 took just under 4 hours.

The *dconv* models trained with only VoxCeleb1 achieved better performance compared to several models trained with both datasets, as shown in Table 6.2. The training results are also visualised in the graph in Figure 6.6. The x-vector model from [10] and ECAPA-TDNN from [228] were compiled in the same pipeline as observational baselines to the developed models and comprised of 4 layers at 512 dimensions. The ECAPA-TDNN is a combination of convolutional and residual blocks whereby the speaker embeddings are extracted using attentive statistical pooling [246], which is the current state-of-the-art approach.

The VGG-M [16] is a deep CNN model with an aggregation layer, which aggregates the features produced by the CNN in the time dimension to produce a fixed length representation for each input. The ResNet-34 model from [247], consists of multiple layers of residually connected CNNs and a dictionary-based NetVLAD layer [248], which is discriminatively trained to aggregate speaker information into a fixed-size descriptor, to improve model robustness. While the ResNet-34 model from [17] does not use a dictionary-based layer and instead just contains wider CNN layers with added residual connections.

All the dynamic kernel based convolution models outperformed the x-vector and cited ResNet models despite not having hyperparameters tuned for optimum performance, as is commonplace for training ResNet-style architectures. The results also suggest that reducing the depth of the convolutions but widening the dimensionality of the layers improves the verification performance of the convolutional network as the 3 layer model with 1024 dimension layers achieved an EER of 2.89% with miniDCF 0.275. The *dconv* model trained with both VoxCeleb1 and 2 achieves an EER of 1.62% with miniDCF 0.18 which is a 17% relative improvement to the ECAPA-TDNN model.

Figure 6.6 displays the error rate per epoch across models with varying layers. The first observation that can be made is that the x-vector model has a worse performance despite attaining lower validation loss across epochs, suggesting there is poor generalisation capability within this model. Another key observation from the Figures, is that the 3 layer

Table 6.2 Experimental results of models trained using VoxCeleb1 [16] and VoxCeleb2 [17], evaluated on the VoxCeleb1-O test set

| Model | Vox1-O EER% | Training set Vox 1 | Training set Vox 2 | Params |
|---|---|---|---|---|
| ResNet-34 [247] | 10.48 | ✓ | ✓ | 10m |
| VGG-M [16] | 10.2 | ✓ | x | 67m |
| ResNet-34 [17] | 5.04 | ✓ | ✓ | 63.5m |
| X-vector [10] | 4.33 | ✓ | x | 8.2m |
| ECAPA-TDNN [228] | 1.95 | ✓ | ✓ | 22.2m |
| Dconv-4 (1024) | **1.62** | ✓ | ✓ | 21m |
| Dconv-3 (1024) | 1.64 | ✓ | ✓ | 12.1m |
| Dconv-5 (1024) | 2.946 | ✓ | x | 32m |
| Dconv-4 (1024) | 2.926 | ✓ | x | 21m |
| Dconv-4 (512) | 2.935 | ✓ | x | 6.4m |
| Dconv-3 (1024) | 2.89 | ✓ | x | 12.1m |
| Dconv-3 (512) | 2.941 | ✓ | x | 3.9m |

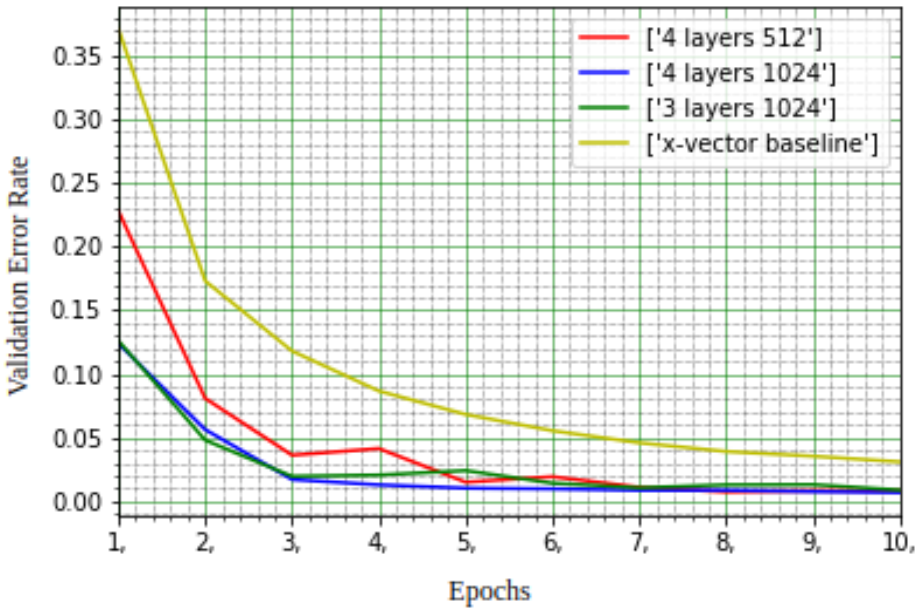



Fig. 6.6 Comparison of error rates on validation set for models of varying dimension parameters

and 4 layer *dconv* models that have a dimension of 1024 perform similarly across validation loss and error rate, which suggests that within the structure of the embeddings compiled

by the dynamic kernels, critical context is learned and contained across the dimensionality of the layers rather than across the depth (number of layers) of the models. Despite the reduced parameters of the 3 layer (12 million parameters) model compared to the 4 layer (21 million parameters) model, it is possible to retain the modelling accuracy using the proposed approach by improving the embedding representation capabilities.

Table 6.3 displays the average computation across the *dconv* models and the X-vector baseline using an NVIDIA RTX3060 GPU. The average computation time for an epoch using the X-vector approach took approximately 2 hours, which is slightly faster than an epoch for the *dconv* models, however to achieve the EER performance of 4.33%, the number of epochs was increased to 25. The number of epochs for all *dconv* models was 10 to attain the results listed in Table 6.2, therefore while each epoch with a 3 or 4 layer *dconv* model may take longer to compute, the models will finish training with less overall time and with a slightly improved performance. To train the *dconv* model on both VoxCeleb1 and 2 datasets took an average of 33 hours per epoch, while the ECAPA-TDNN model took an average of 24 hours per epoch.

Table 6.3 Average epoch computation time of models training on VoxCeleb1 using an NVIDIA RTX3060 GPU

| **Layer Name** | **Average computation time per epoch** |
|---|---|
| X-vector [10] | 02:05:12 |
| ECAPA-TDNN [228] | 03:04:20 |
| Dconv-3 (512) | 02:46:43 |
| Dconv-3 (1024) | 03:45:46 |
| Dconv-4 (512) | 03:23:27 |
| Dconv-4 (1024) | 05:47:35 |
| Dconv-5 (1024) | 08:33:18 |

## 6.7 Summary

This Chapter provided some background regarding speaker recognition modelling in Section 6.2 and then investigated the hypothesis that it is possible to capture acoustic context information using CNNs. This work was inspired by the acoustic modelling work with transformers, to attempt to approximate context without vastly increasing the computation. A novel approach for speaker verification has been proposed in Section 6.4 that provides improved representational capabilities while controlling network dimensionality, allowing the use of lower resources for training and computation. This approach is able to assimilate hierarchical global features for speaker embeddings. The *dconv* model, described in Section 6.4.1 can be trained to extract high resolution features while being computationally inexpensive. Several iterations of the *dconv* model were evaluated on VoxCeleb 1 and compared to a baseline x-vector model, which demonstrated the proposed approach's effectiveness at lowering the EER with low resources.

Section 6.6 shows that for the task of speaker verification, dynamic convolutional spatial dimensions (width) contribute to a slightly increased performance improvement than increasing model depth (layerwise). This corroborates with experiments in Chapters 3, 4 and 5 where increasing model parameters does not necessarily lead to improved performance for ASR tasks and speaker verification tasks. Section 6.6 also explored how modelling approaches and parameter size affects computation time, which is argued to be an important factor to consider when serving and training modelling approaches. The proposed modelling approach could be further extended across different variations of convolutional architectures and also for other domains such as ASR or diarisation.

# Chapter 7

# Using Context Representations to Model Speech Emotion

## Contents

## 7.1 Introduction

The following Chapter aims to explore the role of acoustic context for SER. Section 7.2 introduces the research domain of SER and the particular research goal of the work, along with some background information. Section 7.3.1 discusses context modelling in SER and introduces the idea of overlapping context regions and phone units. Section 7.3.2 presents the consonant-vowel (CV) boundaries and phonemic overlapped regions and their significance in speech emotion perception cues and recognition. Section 7.4 explains the underlying SER model for the interpretation framework and attention. Section 7.5 describes the cross-corpus data, features, experimental framework, and presents the results and graphs. Section 7.7 discusses the interpretation of the presented results in Section 7.6 and suggested directions for the development of future work. Finally a summary of the work in this Chapter in presented in Section 7.8. The findings of this Chapter have been submitted in [249].

## 7.2 Background

Typically, when a speech emotion corpora is created, each audio segment is labelled as a specific emotion category by the annotators, and it is assumed that the whole audio segment signifies that single emotion label or static emotion [250, 14, 13]. Phone boundaries are the boundaries between individual speech sounds and play a critical role in shaping the acoustic context. The position of the phone boundary influences the acoustic characteristics of neighbouring phones and therefore how they are perceived. Coarticulation, where phonetic features are assimilated between neighbouring sounds, can affect the acoustic context

depending on the timing and duration of the boundary. At word-initial boundaries, there may be acoustic effects, such as aspiration in plosives */t/ /p/ /k/*, which results in a brief spike of noise. Phone boundaries are also important for perceptual segmentation, allowing a listener to segment the speech into discrete phonetic units. The clarity and distinctiveness of the boundary is shown to affect the accuracy of perception [251].

It is theorised that the perceptual cues for phone boundaries and acoustic context are ambiguous as they share information for various emotion states [252, 253]. The acoustic stimuli change in speech segments are distributed events and can therefore overlap and present as non-static. From a psycholinguistic perspective, these distributed, continuous stimuli transitions constitute theories of human perception of SER [252, 253]. The context cues can be of different lengths, and the perceptual acoustic context can be modelled with different length acoustic cues. Work from [254] shows that speech emotion can be modelled with small acoustic cues (200 ms). Therefore, the assumption that each acoustic speech segment that is attributed to only one emotion state likely negatively impacts recognition performance. Multiple sub-emotions can be present depending on the contextual variation between different segment regions, as shown in Figure 7.1. This work focuses on acoustic perceptual cues and the implication of the length and the distribution of these cues over speech audio segments for SER.

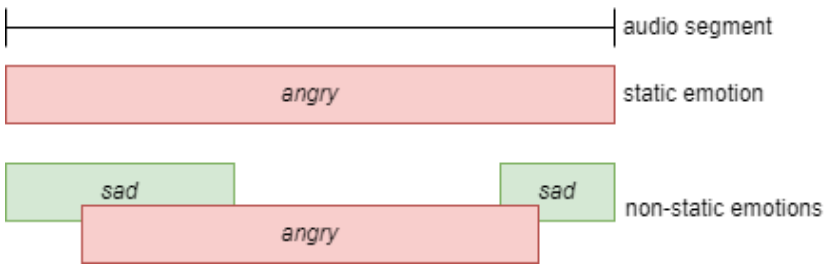



Fig. 7.1 An example of distributed emotions where labelling an utterance as a single discrete category could be overlooking other perceived emotions

Previous research for SER mainly focuses on modelling generalised emotion with different neural network architectures while adapting to speech variability and reducing redundancy

for speaker invariance to improve SER capability [255–259]. As the focus of current research has shifted towards embedding modelling and left-right context cues, work by [260] proposed a spatial representation learning method with CNNs, to model mid to long-term sequence dependencies. After the advent of the transformer architecture, the SER models focused more on transformer-based and multi-head fusion-based modelling approaches [261, 262].

There remains a gap between the psycholinguistic and cognitive theories regarding speech emotion perceptual cues and the currently developed computational modelling methods. Research focusing on interpretability is still underdeveloped for SER models, particularly where the model's internal intricacies and representations with the corresponding acoustic segments can be explained. This work attempts to find a mutual accord with the theories of speech emotion perception cues across multiple disciplines and bridge the gap to speech emotion models. By projecting model attention weights across different time frames (based on various acoustic cues) of the acoustic segment, the emotion classification is observed to shift. Several corpora have been considered to demonstrate the task across various types of speech emotion data (acted, natural, elicited).

## 7.3 Related Work

### 7.3.1 Context Modelling

Context cues for speech emotion can be described as linguistic and paralinguistic. The linguistic aspects consist of semantic structure of the speech segment and the textual meaning. The nonverbal or paralinguistic aspects provide a rich source of perceptual context cues that facilitates projecting expressiveness in social discourse in both intra-cultural and cross-cultural scenarios [263, 264]. Although verbal comprehension mainly dictates social discourse, perceptual context cues can deliver meaning and emotion independent of the verbal comprehension using the acoustic changes that influence the speech delivery [265, 266].

Work in [252] used psychoacoustic features (such as tempo, prosodic contour, loudness etc.) for modelling emotion and concluded that different emotional states have different perceptual cues and that they are subjective to individual contexts despite having a universal representation of emotion states. Furthermore, the acoustic contexts are not orthogonal, and the shared information/dimensions represents the redundant acoustic stimuli which provide context [252, 253]. Naturally, if the acoustic stimuli changes, the perceptual context cue will also change accordingly. If the acoustic stimuli are redundant for the cues that define emotion states, these stimuli share overlapping regions. Typically, a "phone" is regarded as one of the smallest units of an acoustic speech sound. To explore the implication of the various stimuli regions, the phone boundaries should be explored. The CV boundaries for context cues are discussed in Section 7.3.2.

The authors in [267] have presented left context (referred to as "forward effects" by the authors), right context (referred to as "backward effects" by the authors), proximal context and distal acoustic context cues as in acoustic events over time. The sensory attention emphasises the change among these acoustic stimuli, which maximises the potential information for facilitating speech perception [268]. The stimuli changes at a particular time over left-right time frames to reflect the emotion state and speech perception cue at that given point of time. Therefore, it can be assumed that emotion is a distributed event in acoustic segments, not a single discrete emotion category. To investigate this hypothesis, a simple computational model of left-right modelling with attention has been applied in Section 7.4.

### 7.3.2 Linguistic Boundaries

Contextual cues, consisting of phonetic characteristics of speech, can be used to aid the determination of the emotional state at a given time. These characteristics of speech include phonemes, articulation, vowels, consonants, suprasegmental features and other phonetic aspects. Suprasegmental features are aspects of pitch, loudness and timing that affects larger

units and contribute to the prosody of the speech. The phonetic forms can have similarities and dissimilarities among the phone boundaries. A clear distinction has been found between the clusters of vowel and consonant phone data points by work from [269, 270]. The consonant phones play a decisive role in word meaning comprehension, such that removing initial prosodic variations in vowel phones (referred to as acoustic reduction) has been shown to enhance the word intelligibility [269]. However, contrasting studies showed that replacing intermittent consonants with noises or a change in the emphasis on the vowels, increases the perceived intelligibility of words and sentences to human listeners [271, 272]. It was argued that vowel phones are potentially more responsible for defining the emotional state of the speech acoustics, and intelligibility due to stressed vowel regions and wide harmonic variations [270, 273, 274].

Furthermore, the harmonic variations and variations in the pitch within vowels, change the CV boundaries over time and thereby the contextual cues related to acoustic perception. These continuous perceptual context cues are distributed over CV boundaries in acoustic segments [273]. Thus it may be possible that at different left-right time-frames, different regions from the same acoustic segment may be labelled differently. This can be described as the relationship of perceptual CV cues with the acoustics, which has previously been referred to as acoustic-phonetic context for speech perception [275, 276]. The aim of this work is to understand the distributed nature of these perceptual acoustic cues which form an intra-linguistic determinism between the acoustic structure and meaning that humans perceive as emotion. Here, meaning and intelligibility are explored only from acoustic segments as no LM or external multi-modal data has been used.

## 7.4 Model architecture: *BLSTMATT*

The focus of this work is to explore perceptual acoustic cues and their relationship to current speech emotion recognition modelling. Developing and training large-scale SER models is

out of the scope of this work as this approach is to determine the concept of this relationship. As the previously discussed related theories, regarding speech emotion perception, take into account past and future context, this can be modelled as a form of left and right acoustic cues.

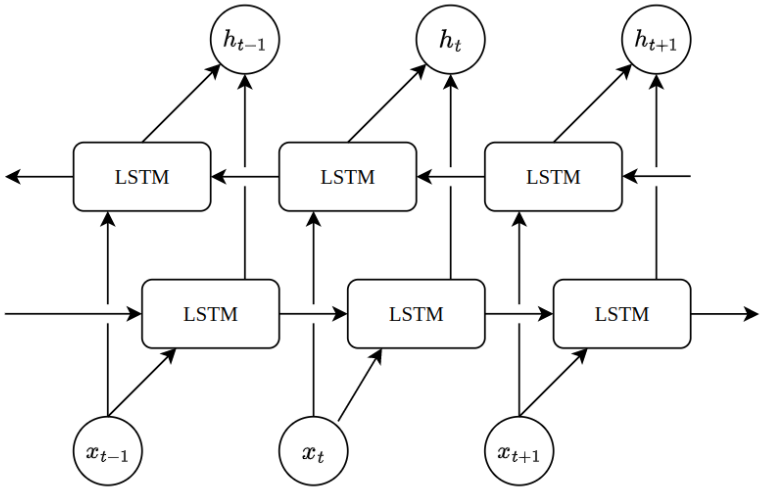



Fig. 7.2 BLSTM architecture overview with forward and backward LSTM layers

LSTM networks are unable to exploit the future context and instead they solely focus on the temporal order of the sequence, whereas BLSTMs [113] comprise of an additional layer of hidden connections, which allows temporal information to pass in the opposite direction to exploit future and past contextual information [277]. With reference to Chapter 2, Section 2.3.3, the BLSTM is comprised of a forward and backward LSTM layer, shown in Figure 7.2. The hidden connections between $N$ layers $\mathbf{h}^n$ are iteratively compiled from $n = 1$ to $N$ and $t = 1$ to $T$:

$$\mathbf{h}_t^n = \sigma(\mathbf{W}_{\mathbf{h}^{n-1}\mathbf{h}^n}\mathbf{h}_t^{n-1} + \mathbf{W}_{\mathbf{h}^n\mathbf{h}^n}\mathbf{h}_{t-1}^n + b_h^n) \tag{7.1}$$

where $\mathbf{W}$ defines the weight matrices, $\sigma$ represents the activation function and $b$ refers to the bias vector. Using this approach, a temporal feature distribution over the sequence can be obtained, which is more effective for SER tasks [278].

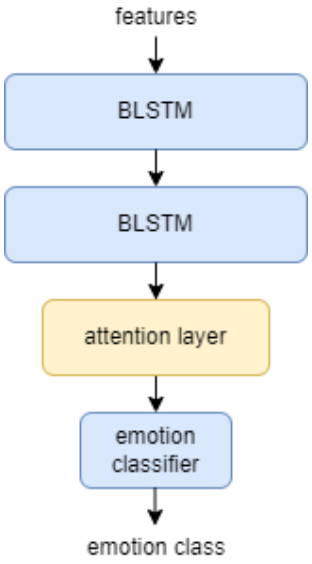



Fig. 7.3 The *BLSTMATT* model pipeline consists of 2 BLSTM layers, with an attention layer and linear classifier

The chosen modelling approach utilises a BLSTM neural network with a subsequent attention layer, referred to as *BLSTMATT*. An overview of the model structure is displayed in Figure 7.3. The BLSTM layers consist of 2 x 512 dimension hidden layers feeding into an attention layer, which computes a 128 dimension context vector. For classification, the network uses a fully-connected linear layer which projects the attention output. In order to classify over the number of emotions in the target, the output is normalised with a $softmax$ layer before the loss is computed.

An attention mechanism in the attention layer enables computation of longer-term inter-sequence dependencies. The additive method for computing attention from [279] is applied for this approach, also referred to as globally contextualised attention (GBA). GBA is a view-specific attention-based weighting that computes a context embedding over the view using a memory network. The memory network combines features across all views to provide a prediction over the current view. Utilising the global mean, the attention mechanism enables the network to attend to specific parts of itself which in turn captures global information. The non-linearity function *tanh* is used to multiply the global mean over the whole temporal

vector which computes the positional dependency of each element. **H** denotes the matrix of output vectors from the LSTM layer, by summing the average time of **H** across contextual modalities, the shared memory matrix **M** can be formed by repetition until it matches the dimension of $\mathbf{H}_s$, where **s** refers to the view for the context. Where $T$ refers to iterations, **V** denotes the parameters controlling the influence within the view and from the shared memory, the attending mechanism **a** can be described by:

$$\mathbf{a}^{(\tau)} = tanh(\mathbf{V}_{s1}^{(\tau)} tanh(\mathbf{H}_{s1})) \cdot tanh(\mathbf{V}_{s2}^{(\tau)} \mathbf{M}^{(\tau)}) \tag{7.2}$$

$$\alpha_s^{(\tau)} = \mathbf{V}_{s3}^{(\tau)T} \mathbf{a}^{(\tau)} \tag{7.3}$$

$\mathbf{V}_{s1}$, $\mathbf{V}_{s2}$ and $\mathbf{V}_{s3}$ are parameters used to compute the attention weight $\alpha$.

## 7.5 Experiments

### 7.5.1 Data

The scope for these experiments regards English speaking adult datasets across three emotion types: one acted dataset, eNTERFACE [250], one natural dataset, MOSEI [13], and one elicited dataset, IEMOCAP [14]. An overview of the emotion classifications represented in each dataset are each described briefly below. For each dataset, the big-six emotions [78] are considered in training and testing: *happy*, *sad*, *anger*, *surprise*, *disgust* and *fear*.

eNTERFACE (ENT) consists of roughly 1 hour of acted English utterances [250]. The training set is comprised of 38 speakers and the testing set contains the remaining 5 speakers. The data is split by 8 female speakers and 35 male speakers from 14 different nations.

IEMOCAP (IEM6) comprises of over 12 hours of US-English utterances from 10 speakers (5 female and 5 male) [14]. There are five dyadic sessions (between two speakers) which are specifically scripted or contrived to elicit certain emotions. The training data consists of

the first 4 sessions (4 speakers) and the last session is split for the test set (2 speakers). It is common for IEMOCAP to be evaluated as four classes: *happy*, *sad*, *anger* and *neutral* (where *excitement* is combined with *happy*). This test set will be referred to as IEM4.

MOSEI (MOS) is the largest sentiment and emotion dataset with approximately 65 hours of data and more than 1000 speakers [13]. Data is collected from YouTube and the videos are not specifically designed as an emotion dataset so the emotional speech is seen as natural. The official training, validation and test splits for the ACL 2018 conference have been considered, where the training and validation sets are combined for training. These can be found at https://github.com/A2Zadeh/CMU-MultimodalSDK/blob/master/mmsdk/mmdatasdk/dataset/standard_datasets/CMU_MOSEI/cmu_mosei_std_folds.py.

### 7.5.2 Implementation

For SER, research suggests that log-Mel filter bank acoustic features have yielded better performance over MFCCs [280]. Further experiments from [281] showed how sequence-based SER systems performed best in terms of unweighted and weighted accuracy with 23-dimensional log-Mel filter bank features.

The *BLSTMATT* contains two hidden layers of 512 nodes each. The output layer (size 1024) is passed into the attention mechanism computing a context vector (size 128), which is projected to 1024 nodes. This is then fed into the emotion classifier which linearly projects to the 6 classes. The cross-entropy loss function is applied, which is preceded by a $softmax$ layer. The *BLSTMATT* produces a variable length attention vector based on the input segment length, as mentioned in section 7.4. The attention vectors have been extracted and mapped with the phones and words in the input segments to be able to interpret the acoustic attention.

### 7.5.3 Evaluation

Unweighted accuracy (UA) and the weighted accuracy (WA) are the metrics typically applied for SER evaluation. The UA calculates accuracy in terms of the total correct predictions divided by total samples, which gives the same weight to each class:

$$UA = \frac{TP+TN}{P+N} \tag{7.4}$$

where, $TP$ is the number of correct positive correct instances, $TN$ is the number of correct negative instances, $P$ is the number of positive instances (equivalent to $TP+FN$) and $N$ is the number of negative instances (equivalent to $TN+FP$). As some of the datasets are imbalanced across the emotion classes, see Tables 7.1, 7.2 and 7.3, the WA is calculated which weighs each class according to the number of samples in that class:

$$WA = \frac{1}{2}(\frac{TP}{P} + \frac{TN}{N}) \tag{7.5}$$

Further details regarding the implementation of the scoring scripts can be found in [279].

### 7.5.4 Acoustic Context

As discussed in Section 7.3.1 and 7.3.2, the recognition of speech emotion is hypothesised to be influenced by overlapping perceptual acoustic cues consisting of variation in the phone boundaries. So, in theory, if the phone boundaries are shifted, the emotion classification may differ from the previous predicted emotion state that considered the whole segment. To further explore this hypothesis, the acoustic context is changed in the following series of experiments.

Experiments are performed removing frames from the end and beginning of the original, whole test segments. In Tables 7.1, 7.2 and 7.3, this is listed in the first column labelled ‘skip

frames (left-right)' where a number of frames are skipped, or removed, from the left and right (left and right context) of each test segment. For example, 20-200 means 20 frames have been removed from the left context of each test segment and 200 frames have been removed from the right context of each test segment. Table 7.1 shows the results where right frames are skipped, Table 7.2 shows results where only left frames are skipped and Table 7.3 shows results where both left and right frames are skipped. If the length of a test utterance is less than the length of context frames, the test utterance remains unchanged. Therefore, when the skip context frames become longer, such as 200-100 (that means a total of 300 frames to be removed), only the test segments with more frames than 300 are used. The percentage of test corpora that is modified with the context is also reported. For example, in the SEGS% column, 91.3% means that 8.7% of the test segments from the corresponding corpora remains the same due to shorter segment length and 91.3% of the test segments are modified with the corresponding context. The weighted and unweighted accuracy are reported along with the change in the context length.

As the experiments consider context length variations, the baseline for this work is the result when no left or right context is removed. This is the first line in all Tables with context 0-0. It is the emotion modelling baseline where one emotion is given for each complete test utterance. For further details about the validity of the *BLSTMATT* model, please see work in [281] and [254].

## 7.6 Results

The experimental results in Tables 7.1, 7.2 and 7.3 suggest that the SER results would change when either the left, right or both contexts are changed. For example, the model tested on MOS has a UA of 73.3% without changing the context length but upon skipping the context right 100 frames, skip frames 0-100 in Table 7.1, the UA degrades. The same observation of UA degradation occurs when skipping left frames or both left and right frames, while there is

| Skip Frames (left-right) | Unweighted Accuracy (UA %) | | | | Weighted Accuracy (WA %) | | | | Percentage of segments (SEGS %) with modified context | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Context | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS |
| 0-0 | 93.33 | 69.06 | 88.79 | 73.30 | 88.00 | 64.57 | 63.81 | 54.29 | - | - | - | - |
| 0-30 | 86.89 | 68.73 | 88.28 | 73.55 | 76.40 | 63.82 | 64.44 | 54.76 | 100.0 | 100.0 | 100.0 | 100.0 |
| 0-100 | 82.22 | 68.73 | 87.94 | 72.81 | 68.00 | 63.45 | 61.21 | 54.70 | 91.3 | 99.3 | 99.5 | 98.0 |
| 0-200 | 86.22 | 68.09 | 86.63 | 71.53 | 75.20 | 62.08 | 61.32 | 53.63 | 40.0 | 77.1 | 80.2 | 89.4 |

Table 7.1 Cross-corpora emotion recognition results with variable context length, where right frames are skipped.

| Skip Frames (left-right) | Unweighted Accuracy (UA %) | | | | Weighted Accuracy (WA %) | | | | Percentage of segments (SEGS %) with modified context | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Context | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS |
| 0-0 | 93.33 | 69.06 | 88.79 | 73.30 | 88.00 | 64.57 | 63.81 | 54.29 | - | - | - | - |
| 20-0 | 91.56 | 69.22 | 88.74 | 73.01 | 84.80 | 64.76 | 63.83 | 54.13 | 100.0 | 100.0 | 100.0 | 100.0 |
| 30-0 | 89.33 | 69.66 | 88.68 | 72.97 | 80.80 | 65.03 | 63.83 | 54.18 | 100.0 | 100.0 | 100.0 | 100.0 |
| 100-0 | 84.44 | 69.90 | 87.66 | 72.50 | 72.00 | 64.60 | 61.00 | 54.16 | 91.3 | 99.3 | 99.5 | 98.0 |
| 200-0 | 87.78 | 69.38 | 87.14 | 71.98 | 78.00 | 63.47 | 61.97 | 54.07 | 40.0 | 77.1 | 80.2 | 89.4 |
| 300-0 | 92.89 | 68.61 | 86.86 | 71.11 | 87.20 | 62.89 | 61.66 | 53.45 | 8.7 | 54.5 | 57.8 | 80.2 |

Table 7.2 Cross-corpora emotion recognition results with variable context length, where left frames are skipped.

| Skip Frames (left-right) | Unweighted Accuracy (UA %) | | | | Weighted Accuracy (WA %) | | | | Percentage of segments (SEGS %) with modified context | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Context | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS | ENT | IEM6 | IEM4 | MOS |
| 0-0 | 93.33 | 69.06 | 88.79 | 73.30 | 88.00 | 64.57 | 63.81 | 54.29 | - | - | - | - |
| 20-200 | 86.89 | 68.73 | 87.49 | 71.39 | 76.40 | 63.11 | 62.44 | 53.81 | 38.7 | 71.6 | 73.4 | 87.6 |
| 200-100 | 92.67 | 68.25 | 86.58 | 71.30 | 86.80 | 61.83 | 61.42 | 54.00 | 8.7 | 54.5 | 57.8 | 80.2 |

Table 7.3 Cross-corpora emotion recognition results with variable context length, where left and right frames are skipped.

a slight improvement in UA when skipping 30 right frames. In the case of skip frames 0-30, removing 30 ms from the end of the segment modifies 100% of the segments across all the test sets. The results for ENT and IEM4 are worse for both UA and WA, but for MOS the performance improves. For IEM6, the UA degrades whereas the WA improves. The majority of the results across all the datasets degrade upon varying the context length due to the target label, supplied with those segments, being a fixed discrete emotion category. This finding corroborates the initial hypotheses that speech emotion is not a fixed entity that remains the

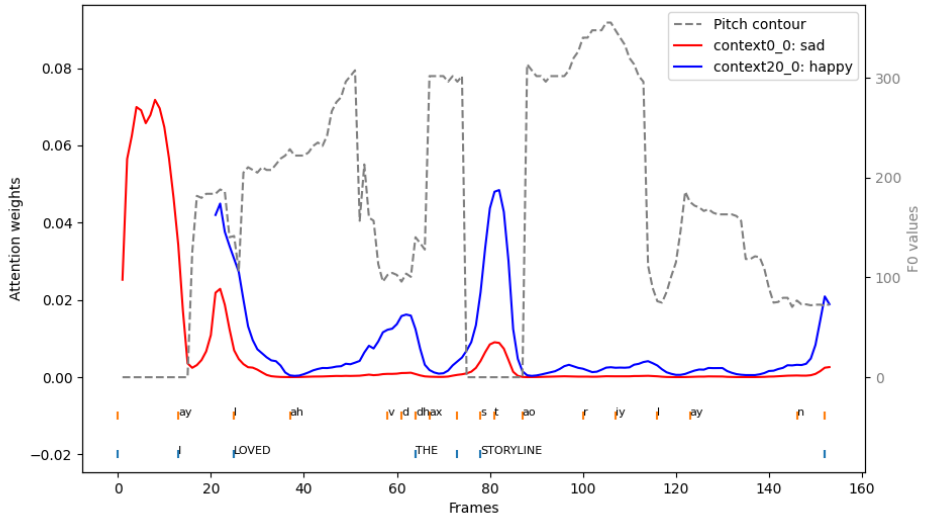



Fig. 7.4 A *happy* MOS [13] utterance with no context removed mislabelled as *sad* compared to 20 left frames removed correctly labelled as *happy*, along with the pitch contour

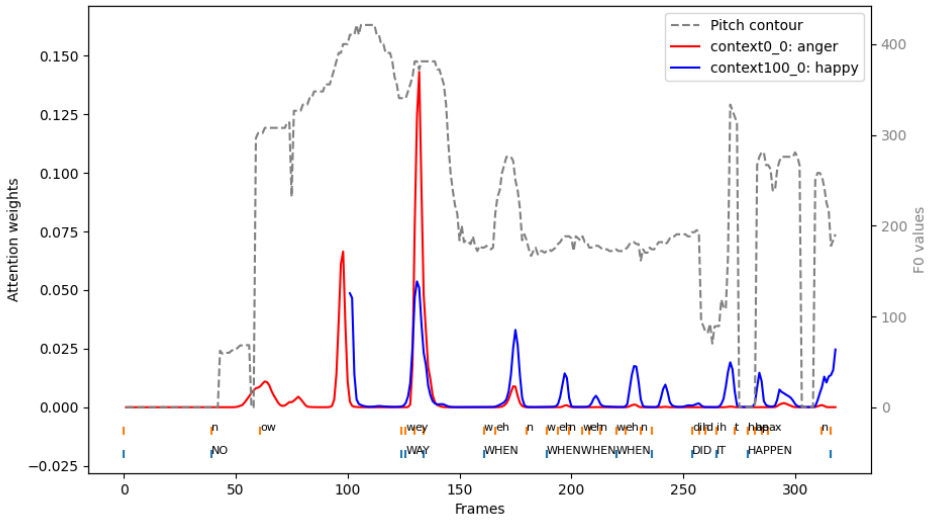



Fig. 7.5 A *happy* IEM4 [14] utterance with no context removed mislabelled as *anger* compared to 100 right frames removed correctly labelled as *happy*, along with the pitch contour

same over the whole audio segment, and that it is subject to be distributed over different overlapping shorter context queues.

To observe the relationship between the SER results and the hypotheses regarding the acoustic segments in more detail, the attention weights were extracted for each test utterance and mapped to the aligned words and phones. Additionally, the pitch contour was calculated to understand the pitch correlation with respect to the prosodic utterance using the algorithm

found at https://github.com/google/REAPER. The attention maps for a sample of the test utterances are presented in Figures 7.4 and 7.5: the former from the MOSEI corpus and the latter from the IEMOCAP corpus. Figure 7.4 shows that the attention projection drifts while changing the phone boundaries from the same audio segment and therefore the emotion state also changes. With context 0-0, the model incorrectly predicts the emotion *sad* (attention weights in Figure 7.4 indicated by red line) whereas removing 20 left frames helps the model correctly predict the emotion *happy* (attention weights in Figure 7.5 indicated by blue line). The attention weights focus more strongly on different portions of the test utterance. Similar behaviour can be seen in Figure 7.5 where skipping 100 left frames allows the model to make the correct prediction.

## 7.7 Discussion

In some datasets, such as ENT and IEM4, the SER change is not very prominent, as can be seen from the results listed for WA and UA. This is potentially due to several reasons. The first is that some segments in particular datasets have a length shorter than 200 frames, and these segments remained unchanged during the context modification. So to properly interpret the results, the percentage of data that is modified with context at each experiment should be taken into account. Secondly, the acoustic *BLSTMATT* model perception is a direct result of the relationship between the training data and the corresponding labels given by the annotators, which could have added bias factors and add to recognition uncertainty. To attempt to mitigate the inherent biases and to attempt to generalise the model perception cues for these experiments, the model is trained with four different corpora consisting of acted, natural and elicited emotions. Consequently, it is argued that the results corroborate the argument that an continuous approach to emotion recognition is the optimal strategy based on observed acoustic stimuli shift. This work is an attempt to bridge the gap that current SER

models have, by explaining the SER model's internal intricacies and how the representations correspond with acoustic segments.

Figures 7.4 and 7.5 show the attention weight propensity towards the vowel based regions. This corroborates with the claims from the linguistic and cognitive theories about speech emotion recognition and CV boundaries, as proposed in Section 7.3.2, that consonants play a decisive role in word meaning but vowels are more responsible for the emotion perception cues as a result of harmonic variations and stressed regions. The vowels are observed to change the CV boundaries and the context cues for emotion perception causing many hard boundaries to be redundant. This suggests the cues for phone boundaries and acoustic context can share information relative to the perceived emotion state.

For the IEM6 dataset, when the context lengths were skipped left frames, there was a slight improvement in UA or WA, while recognition with the MOS and IEM4 datasets improved slightly when context lengths were skipped right frames. These results highlight that where context cues vary in length, it is possible for the acoustic segments to contain more than one distinct emotion state. As the UA and WA vary positively and negatively according to context lengths, this suggests overlapping regions where the acoustic stimuli are more or less informative regarding the emotion state. As future speech emotion datasets are compiled and annotated, if the labels for emotion classes were adjusted to allow for overlapping categories, this could potentially aid the recognition performance of current and future developed models. These results and insights can also be used to modify computational models and mechanisms that are able to adapt and recognise emotion from various speech domains to be more in-line with the psycholinguistic theories.

## 7.8 Summary

In the current trend of SER models, discussed in Section 7.2, emotion labels are treated as discrete labels attributed over a whole segment. Sections 7.3.1 and 7.3.2 present the published

related theories regarding context cues and linguistic cues for SER. The problem explored in this Chapter is regarding the state-of-the-art approaches that assume that an utterance's global attributes correlate with the local characteristics over different time frames in the same segment for learning one discrete emotion category. This is observed to not be the case most of the time with results in Section 7.6. Vowel-consonant envelopes rapidly change over time, attributing to different acoustic context. Hence the paralinguistic cue also changes with acoustic context. The results demonstrate this argument. Moreover, by treating acoustic segments and emotion correspondence as a context-oriented continuous relationship, this should aid emotion recognition models across languages and dialects due to the distribution of acoustic boundaries across models trained on various emotion data. As a result, it could be possible to learn the variability of acoustic context in speech emotions rather than the variability of acoustic segments in speech emotions.

Future development of this framework will enable improved emotion modelling by understanding the intermediate representations and relating audio data with the computational models. Furthermore, it will help create more accurate annotations for emotion labels, improving SER corpora generation.

This Chapter argues that discrete categorical emotion classification should not be the preferred approach to develop future SER models as it has been observed that emotion cues present as a distributed event, corroborating directly with cognitive linguistic theory that it is also continuous to recognise. Finding a suitable approach for accurate modelling of emotion states should be the aim of future research.

# Chapter 8

# Conclusion

Speech technology aims to aid communication between humans and machines by attempting to listen, understand and learn speech. Deep learning networks have facilitated the development of speech technology to levels reaching human parity. The focus of recent research has been to increase recognition performance, build systems for multiple domains, and to reduce some of the modelling requirements for domain knowledge. As research continues, there has been a corresponding demand to be able to explain deep learning models in order to predict how networks will behave in real-world scenarios, whether there is any bias within the model, and to further scientific knowledge. Much like the human brain, the relationship between how contextual dependencies, such as acoustic and linguistic context dependencies, are represented within models was unclear.

One of the initial goals of this research was to analyse the latent space of neural representations within ASR models and to provide some interpretation of how those representations relate to recognition performance. It was hypothesised that these insights could be used to understand whether analysis of the dependencies of neural representations can be used to interpret and improve modelling across speech technology, such as speaker and emotion recognition. Furthermore, it was unclear whether it was possible to exploit acoustic or linguistic context dependencies with attention-based models across domains to improve model

performance without increasing computational requirements when training. The final goal was also to assess how the choice of context modelling technique affects the performance of models for speech technology.

Chapter 2 provides a literature review regarding current speech, speaker, and emotion recognition modelling approaches, properties of neural networks and the foundations for End-to-End speech recognition. As reviewed in Section 2.4, the current End-to-End speech recognition approaches are CTC-based, attention-based encoder-decoders and RNN-Ts. CTC approaches do not rely on any domain-specific knowledge, however the approach assumes conditional independence between the output labels. Recognition performance with purely CTC-based modelling approaches is limited due to the loss of context information. The RNN-T approach attempts to model context dependencies between outputs through time using an encoder network and predictor network. RNN-Ts are typically memory intensive for many applications as they need to condition on the previous predicted labels. Attention-based encoder-decoders model acoustic information into hidden states, then use a decoder to predict the output label for each time-step. An attention mechanism is used to compute the weights between the hidden states of the encoder and the previous output in order to capture context dependencies. Transformer models are the state-of-the-art encoder-decoder approach and are able to capture global context information, however they are difficult to scale due to the self-attention mechanism, which needs to condition on the previous layer output.

The developed frameworks that utilise the approaches described in Chapter 2 have variations regarding the training optimisation and setup, but it was not clear how these contributed to performance within specific domains. Chapter 3, Section 3.3 described several state-of-the-art frameworks and evaluation results using a conversation telephone speech task. The results showed that attention-based encoder-decoders reach the lowest WER for conversational speech recognition. In order to attempt to understand what factors could be contributing to the modelling errors, empirical analysis, in Section 3.5, attempted to observe

patterns within the model outputs to potentially target those errors. Despite using several metrics of empirical analysis, it was not clear whether there were any indicators that could be used to adapt or improve the modelling approaches. Subsequently, an experimental analysis framework was proposed in Section 3.6 to attempt to interpret and understand the neural representation dependencies across approaches. The contributions and findings of Chapter 3 are:

- The development of a framework to enable representation correlation analysis using state-of-the-art End-to-End models.
- A detailed comparison of similarity indexes where it was found that CKA and SVCCA produce similar correlation results for an ASR task.
- Neural representation analysis of attention-based encoder-decoder approaches to visualise more optimal learned dependencies for conversational telephone speech modelling. The correlation of the learned representation hierarchies were shown to be an indicator of improved recognition performance.

Capturing longer-term acoustic dependencies to improve speech modelling recognition performance using challenging data, such as conversational speech, was explored in Chapter 4. Using a mixture of experts and augmentation approaches, had been shown to improve model performance across different tasks and domains. However, the results of the proposed approaches, presented in Section 4.3.3 showed very similar performance results when evaluated with the in-domain Hub5'00 test sets, an average of 10.5% WER compared to 10.7% on the Switchboard set and an average of 20.4% WER compared to 20.2% on the Callhome test set. The mixture of experts models produced higher rates of output errors when evaluated with the RT03 test set, an average of 23.4% WER compared to 21.2% on the RT03 Switchboard set and an average of 15.2% WER compared to 13.3% on the RT03 Fisher test set. These results showed that the current developments in modelling acoustic context do no

directly translate to improved performance for an End-to-End ASR task. Upon conducting representation analysis using the framework developed in Section 3.6, the results indicated that the transformer approaches were learning the same representation space, indicating potential memorisation behaviour. Section 4.4 presented cross-domain analysis of LSTM and transformer models to interpret the relationship between neural representations and the dependencies upon recognition performance. The results in Section 4.4.3 corroborated with the results in Section 3.6.4 where more significant variations within the higher layer neural representations was attributed to poorer recognition performance. The contributions and findings of Chapter 4 are:

- Incorporation of established mixture of experts approaches for an End-to-End ASR task, where the disparity in performance was discussed and analysed to observe the learned representation spaces of transformer models.
  - The performance results on the Hub5'00 and RT03 test sets showed that current developments in acoustic modelling techniques cannot be directly incorporated into the current state-of-the-art End-to-End ASR modelling approaches.
  - SVCCA analysis indicated that the transformer modelling approaches learn the same representation space and attempt to memorise the data, leading to poorer recognition performance on out-of-domain testing scenarios.
- Cross-domain SVCCA analysis showed that models with high variations in neural representations within the higher layers have poorer recognition performance.
- SVCCA analysis between models trained with different data domains is able to observe that the representations learn different hierarchical latent spaces. The results evaluating models trained across corpora indicate that model parameters would need to be adapted for each dataset in order to improve the recognition performance.

Chapter 5 introduced fusion techniques for incorporating language modelling with End-to-End ASR models, described in Section 5.2, which were shown to improve recognition performance of state-of-the-art approaches. However, results from LM fusion rescoring experiments indicated that improved performance for End-to-End ASR models is highly dependent upon higher weighting on the LM output. Using the SVCCA analysis framework to analyse transformer models with cross-domain LMs, it was possible to observe that the LM-dependent representations are predominantly dependent upon the deeper layers. Using the insights from Section 5.3.3 and in Section 4.4 the model parameters were tuned across different domain datasets to improve recognition performance. It was argued that interpretative analysis is important to develop modelling approaches with knowledge of the dependencies that contribute to improved performance. The main contributions of Chapter 5 are:

- The LSTM End-to-End ASR model improves recognition performance of 77% relative to 4.11% WER on the EVAL92 test set and 6.28% WER on the DEV93 test set with an LM fusion weighting of between 0.7 and 0.8, indicating that the model is not powerful enough to model linguistic context alone.

- Using SVCCA analysis highlighted that LM-dependent representations could be observed within the deeper half of the transformer model layers. The cross-domain LM representations could be observed in the deeper final layers of the transformer model. These results were able to be used to observe representation dependencies that affect recognition performance.

- By tuning the model parameters of transformer models using evidence from the analysis experiments, the recognition performance was improved:

  - On the Hub5'00 test sets to 9.5% and 19.1% WER.

  - On the WSJ test sets to 4.13% and 6.3% WER.

  - On the Librispeech test sets to 1.9% and 3.9% WER.

The hypothesis that modelling context within speech could be applied to speaker recognition models to improve recognition performance was derived using the insights provided by Chapters 3, 4, and 5. Initially, the prior approaches for speaker recognition were discussed where recent developments focused upon capturing context information to improve speaker recognition and verification performance. Research showed that the current state-of-the-art models struggle to recognise speakers in poor acoustic conditions and recent developments rely on large amounts of training data and model parameters to improve performance. Based on the gathered research, a dynamic convolution approach is proposed in Section 6.4.1 which improved speaker verification performance on the VoxCeleb1-O test set to 1.62% by improving representation capacity without increasing the computational complexity. The results in Section 6.6 corroborated with the results from Section 4.4.3 where models with increased parameters do not necessarily contribute to improved recognition performance. The main contributions of Chapter 6 are:

- A novel approach for speaker verification using dynamic convolutions was proposed that improves EER on the VoxCeleb1-O test set to 1.62% when training with both VoxCeleb1 and 2 training sets.
- The proposed dynamic convolution approach is able to reach a lower EER on the VoxCeleb1-O test set when training using only the VoxCeleb1 training set to 2.89%.
- The best performing model has a smaller parameter size of 21 million compared to other state-of-the-art approaches. The average computation time per epoch was increased by approximately 30% compared to the X-vector model, however the performance was also improved using a smaller amount of training data.
- For the task of speaker verification, increasing dynamic convolution spatial dimensions (width) contributes to a slight reduction in EER than increasing model depth (layerwise)

from 2.946% to 2.89%. Despite the reduced parameters of the 3 layer (12 million parameters) model compared to the 4 layer (21 million parameters) model, it was possible to retain the modelling accuracy using the proposed approach by improving the embedding representation capacity.

The role of acoustic context was also explored for emotion recognition as this research domain uses similar modelling techniques to ASR and speaker recognition. Chapter 7, Section 7.2 outlined that emotion labels used for classification are treated as discrete events over a whole speech segment, similar to how ASR and speaker recognition models output discrete labels. This was shown to be in contradiction to published psycholinguistic theories regarding context and linguistic cues for emotion perception. Results in Section 7.5 showed that when the left or right contexts were changed across multiple corpora, the predicted emotion target label also changed. The attention weight of the model had a propensity towards vowel-based regions which corroborated with the theories on consonant-vowel boundaries. The main findings of Chapter 7 are:

- State-of-the-art SER approaches assume that an utterance's global attributes directly correspond with local characteristics across different time frames in order to learn a discrete emotion label. The analysis results varying context length show that this is a naive approach and that acoustic segments and emotion classification should be treated as a continuous relationship to improve recognition.

- The results and hypotheses derived from the analysis provide a previously unexplored link between cognitive theories of emotion perception and SER, which could direct future modelling approaches to grounded in more psycholinguistic theory.

- Analysis regarding the distribution of context cues over acoustic segments argues for future speech emotion datasets to allow for overlapping categories which could aid recognition performance for SER models.

Altogether this work aims to explore context modelling in speech technology using novel and established methods of analysis. This analysis has been used to consider modelling techniques as well as acoustic and linguistic context to provide insights on representations and dependencies related to performance.

## 8.1 Future Work

Potential future works from the research conducted in Chapter 4 could be the further analysis of neural representations for ASR models trained on augmented or noisy data, in order to observe the properties of different layers across models. This would be able to direct the adaptation of models that are more robust to noise.

The scopes of Chapter 4 and 5 could also be expanded to investigate the attributes and possible learned features that could be classified within the layers or across modelling approaches. A classification model could be trained alongside the ASR model to model distributions of features and provide more empirical interpretation of modelling approaches across different domains. This would contribute to a deeper understanding of the properties of the representation dependencies which could be used to aid the development of models few-shot learning or downstream tasks.

The research presented in Chapter 6 could be extended across different variations of convolutional models. These models could also be evaluated with more challenging data, such as overlapping speakers or higher levels of noise, which would provide evidence as to how the proposed modelling approach performs with different acoustic scenarios. The proposed approach is also not limited to the task of speaker recognition and could be used for modelling approaches across extended domains such as ASR.

Finally the analysis work in Chapter 7 could be used to develop continuous modelling approaches that incorporate the cognitive theories of emotion perception to improve state-of-the-art speech emotion recognition models. These modelling approaches could also aim

to contribute to networks that more accurately simulate biological systems and behave in a more human-like manner.

# References


[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Proc. ICLR*, 2014.

[3] Yiming Wang, Tongfei Chen, Hainan Xu, Shuoyang Ding, Hang Lv, Yiwen Shao, Nanyun Peng, Lei Xie, Shinji Watanabe, and Sanjeev Khudanpur. Espresso: A fast end-to-end neural speech recognition toolkit. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 136–143. IEEE, 2019.

[4] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28, 2015.

[5] LD Consortium et al. 2000 hub5 english evaluation transcripts ldc2002t43. *Philadelphia: Linguistic Data Consortium*, 2002.

[6] John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

[7] Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.

[8] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[9] Christopher Cieri, David Miller, and Kevin Walker. The fisher corpus: A resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71, 2004.

[10] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020.

[13] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018.

[14] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

[15] Takaaki Hori, Jaejin Cho, and Shinji Watanabe. End-to-end speech recognition with word-based rnn language models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 389–396. IEEE, 2018.

[16] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. *Proc. Interspeech 2017*, pages 2616–2620, 2017.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pages 1086–1090, 2018.

[18] Mehmet Ali Tuğtekin Turan, Emmanuel Vincent, and Denis Jouvet. Achieving multi-accent asr via unsupervised acoustic model adaptation. In *INTERSPEECH 2020*, 2020.

[19] Qin Jin and Tanja Schultz. Speaker segmentation and clustering in meetings. In *INTERSPEECH*, volume 4, pages 597–600, 2004.

[20] Yuanchao Li, Peter Bell, and Catherine Lai. Fusing asr outputs in joint training for speech emotion recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7362–7366. IEEE, 2022.

[21] Haytham M Fayek, Margaret Lech, and Lawrence Cavedon. On the correlation and transferability of features between automatic speech recognition and speech emotion recognition. In *Interspeech*, pages 3618–3622, 2016.

[22] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. Technical Report MSR-TR-2016-71 (revised), February 2017.

[23] Courtney Mansfield, Sara Ng, Gina-Anne Levow, Richard A Wright, and Mari Ostendorf. Revisiting parity of human vs. machine conversational speech transcription. *Proc. Interspeech 2021*, pages 1997–2001, 2021.

[24] Anna Ollerenshaw, Md Asif Jalal, and Thomas Hain. Insights on neural representations for end-to-end speech recognition. *Proc. Interspeech 2021*, pages 4079–4083, 2021.

[25] A Ollerenshaw, MA Jalal, and T Hain. Insights of neural representations in multi-banded and multi-channel convolutional transformers for end-to-end asr. In *IEEE 30th European Signal Processing Conference*. Institute of Electrical and Electronics Engineers (IEEE), 2022.

[26] Anna Ollerenshaw, Md Asif Jalal, and Thomas Hain. Probing statistical representations for end-to-end asr. *In IEEE 31st European Signal Processing Conference. Institute of Electrical and Electronics Engineers (IEEE), 2023*, 2023.

[27] Anna Ollerenshaw, Md Asif Jalal, and Thomas Hain. Dynamic kernels and channel attention with multi-layer embedding aggregation for speaker verification. 2022.

[28] Kouichirou Yamauchi, Megumu Fukuda, and Kunihiko Fukushima. Speed invariant speech recognition using variable velocity delay lines. *Neural Networks*, 8(2):167–177, 1995.

[29] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011.

[30] Victor W Zue and James R Glass. Conversational interfaces: Advances and challenges. *Proceedings of the IEEE*, 88(8):1166–1180, 2000.

[31] Alessandro Vinciarelli, Anna Esposito, Elisabeth André, Francesca Bonin, Mohamed Chetouani, Jeffrey F Cohn, Marco Cristani, Ferdinand Fuhrmann, Elmer Gilmartin, Zakia Hammal, et al. Open challenges in modelling, analysis and synthesis of human behaviour in human–human and human–machine interactions. *Cognitive Computation*, 7(4):397–413, 2015.

[32] Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953, 2019.

[33] Torsten Zeppenfeld, Michael Finke, Klaus Ries, Martin Westphal, and Alex Waibel. Recognition of conversational telephone speech using the janus speech engine. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1815–1818. IEEE, 1997.

[34] Andreas Stolcke and Jasha Droppo. Comparing human and machine errors in conversational speech transcription. *Proc. Interspeech 2017*, pages 137–141, 2017.

[35] Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Modeling bilingual conversational characteristics for neural chat translation. In *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)*, pages 5711–5724, 2021.

[36] MA Anusuya and Shriniwas K Katti. Speech recognition by machine, a review. *arXiv preprint arXiv:1001.2267*, 2010.

[37] B Yegnanarayana and Raymond NJ Veldhuis. Extraction of vocal-tract system characteristics from speech signals. *IEEE transactions on Speech and Audio Processing*, 6(4):313–327, 1998.

[38] Nancy Jean VanDerveer. *Ecological acoustics: Human perception of environmental sounds*. Cornell University, 1979.

[39] Randy L Diehl, Andrew J Lotto, Lori L Holt, et al. Speech perception. *Annual review of psychology*, 55(1):149–179, 2004.

[40] Victor W Zue. The use of speech knowledge in automatic speech recognition. *Proceedings of the IEEE*, 73(11):1602–1615, 1985.

[41] Sunit Sivasankaran, Aditya Arie Nugraha, Emmanuel Vincent, Juan A Morales-Cordovilla, Siddharth Dalmia, Irina Illina, and Antoine Liutkus. Robust asr using neural network based speech enhancement and feature simulation. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 482–489. IEEE, 2015.

[42] I Ding. Enhancement of speech recognition using a variable-length frame overlapping method. In *Proceedings of International Symposium on Computer, Communication, Control and Automation (3CA)*, pages 375–377, 2010.

[43] Muhammad Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. 2017.

[44] Sandeep Kumar Pandey, Hanumant Singh Shekhawat, and SR Mahadeva Prasanna. Deep learning techniques for speech emotion recognition: A review. In *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 1–6. IEEE, 2019.

[45] Jared J Wolf. Efficient acoustic parameters for speaker recognition. *The Journal of the Acoustical Society of America*, 51(6B):2044–2056, 1972.

[46] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

[47] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[48] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, 8(3):185–190, 1937.

[49] John S Bridle and Michael D Brown. An experimental automatic word recognition system. *JSRU report*, 1003(5):33, 1974.

[50] Md Rakibul Hasan, Md Mahbub Hasan, and Md Zakir Hossain. How many mel-frequency cepstral coefficients to be utilized in speech recognition? a study with the bengali language. *The Journal of Engineering*, 2021(12):817–827, 2021.

[51] Lawrence Rabiner. Fundamentals of speech recognition. *Fundamentals of speech recognition*, 1993.

[52] Hector NB Pinheiro, Fernando MP Neto, Adriano LI Oliveira, Tsang Ing Ren, George DC Cavalcanti, and André G Adami. Optimizing speaker-specific filter banks for speaker verification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5350–5354. IEEE, 2017.

[53] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

[54] Stephan Kanthak and Hermann Ney. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–845. IEEE, 2002.

[55] Jason J Humphries and Philip C Woodland. Using accent-specific pronunciation modelling for improved large vocabulary continuous speech recognition. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[56] Alexander Rudnicky. Language modeling with limited domain data. 1995.

[57] Claude E Shannon and Warren Weaver. The mathematical theory of information. *Urbana: University of Illinois Press*, 97, 1949.

[58] Lalit R Bahl, Frederick Jelinek, and Robert L Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2(2):179–190, 1983.

[59] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. 2016.

[60] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

[61] Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994.

[62] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211, 2018.

[63] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. Speechbrain: A general-purpose speech toolkit. 2021.

[64] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[65] Jørgen Bang-Jensen, Gregory Gutin, and Anders Yeo. When the greedy algorithm fails. *Discrete optimization*, 1(2):121–127, 2004.

[66] Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng. First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns. 2014.

[67] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[68] J Markel, B Oshika, and A Gray. Long-term feature averaging for speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):330–337, 1977.

[69] Arnab Poddar, Md Sahidullah, and Goutam Saha. Speaker verification with short utterances: a review of challenges, trends and opportunities. *IET Biometrics*, 7(2):91–101, 2018.

[70] John HL Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, 32(6):74–99, 2015.

[71] Joseph P Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[72] Matthieu Hébert. Text-dependent speaker recognition. In *Springer handbook of speech processing*, pages 743–762. Springer, 2008.

[73] Douglas A Reynolds and William M Campbell. Text-independent speaker recognition. In *Springer Handbook of Speech Processing*, pages 763–782. Springer, 2008.

[74] Keith Oatley, Dacher Keltner, and Jennifer M Jenkins. *Understanding emotions*. Blackwell publishing, 2006.

[75] Lisa Feldman Barrett, Batja Mesquita, and Maria Gendron. Context in emotion perception. *Current directions in psychological science*, 20(5):286–290, 2011.

[76] Szabolcs Levente Tóth, David Sztahó, and Klára Vicsi. Speech emotion perception by human and machine. In *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction: COST Action 2102 International Conference, Patras, Greece, October 29-31, 2007. Revised Papers*, pages 213–224. Springer, 2008.

[77] Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar, Tariqullah Jan, Mohammad Haseeb Zafar, and Thamer Alhussain. Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, 7:117327–117345, 2019.

[78] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[79] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[80] Ye-Yi Wang, Alex Acero, and Ciprian Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *2003 IEEE workshop on automatic speech recognition and understanding (IEEE Cat. No. 03EX721)*, pages 577–582. IEEE, 2003.

[81] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. Automatic speech recognition errors detection and correction: A review. *Procedia Computer Science*, 128:32–37, 2018.

[82] Eric D Sandness and I Lee Hetherington. Keyword-based discriminative training of acoustic models. In *Sixth International Conference on Spoken Language Processing*, 2000.

[83] Youngja Park, Siddharth Patwardhan, Karthik Visweswariah, and Stephen C Gates. An empirical analysis of word error rate and keyword error rate. In *INTERSPEECH*, volume 2008, pages 2070–2073, 2008.

[84] Hiroaki Nanjo and Tatsuya Kawahara. A new asr evaluation measure and minimum bayes-risk decoding for open-domain speech understanding. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–1053. IEEE, 2005.

[85] Andrew Cameron Morris, Viktoria Maier, and Phil Green. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Eighth International Conference on Spoken Language Processing*, 2004.

[86] John Oglesby. What's in a number? moving beyond the equal error rate. *Speech communication*, 17(1-2):193–208, 1995.

[87] Roddy Cowie, Cate Cox, Jean-Claude Martin, Anton Batliner, Dirk Heylen, and Kostas Karpouzis. Issues in data labelling. *Emotion-oriented systems: The Humaine handbook*, pages 213–241, 2011.

[88] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[89] Frank Rosenblatt. Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309, 1960.

[90] Marvin L Minsky and Seymour A Papert. Perceptrons: expanded edition. *MIT press*, 1988.

[91] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[92] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[93] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[94] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

[95] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[96] Abien Fred Agarap. Deep learning using rectified linear units (relu). 2018.

[97] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA, 2013.

[98] Michael D Richard and Richard P Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991.

[99] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

[100] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[101] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[102] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[103] Jinia Konar, Prerit Khandelwal, and Rishabh Tripathi. Comparison of various learning rate scheduling techniques on convolutional neural network. In *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–5. IEEE, 2020.

[104] Matthew D Zeiler. Adadelta: an adaptive learning rate method. 2012.

[105] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959.

[106] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[107] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[108] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[109] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[110] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.

[111] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[112] Albert Zeyer, Ralf Schlüter, and Hermann Ney. Towards online-recognition with deep bidirectional lstm acoustic models. In *Interspeech*, pages 3424–3428, 2016.

[113] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[114] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91(1), 1991.

[115] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[116] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. 2020.

[117] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017.

[118] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end ASR: from supervised to semi-supervised learning with modern architectures. In *ICML 2020 Workshop on Self-supervision in Audio and Speech*, 2020.

[119] Bo Li, Shuo-yiin Chang, Tara N Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu. Towards fast and accurate streaming end-to-end asr. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6069–6073. IEEE, 2020.

[120] Mahaveer Jain, Kjell Schubert, Jay Mahadeokar, Ching-Feng Yeh, Kaustubh Kalgaonkar, Anuroop Sriram, Christian Fuegen, and Michael L Seltzer. Rnn-t for latency controlled asr with improved beam search. 2019.

[121] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[122] Jan Chorowski, Adrian Lancucki, Bartosz Kostka, and Michal Zapotoczny. Towards using context-dependent symbols in ctc without state-tying decision trees. *Interspeech*, 2019.

[123] Yosuke Higuchi, Hirofumi Inaguma, Shinji Watanabe, Tetsuji Ogawa, and Tetsunori Kobayashi. Improved mask-ctc for non-autoregressive end-to-end asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8363–8367. IEEE, 2021.

[124] Oriol Vinyals and Quoc Le. A neural conversational model. *ICML Deep Learning Workshop, 2015*, 2015.

[125] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. 2014.

[126] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. End-to-end speech recognition using lattice-free mmi. *Proc. Interspeech 2018*, pages 12–16, 2018.

[127] Zhe Yuan, Zhuoran Lyu, Jiwei Li, and Xi Zhou. An improved hybrid ctc-attention model for speech recognition. 2018.

[128] Takaaki Hori, Shinji Watanabe, and John R Hershey. Joint ctc/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–529, 2017.

[129] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 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)*, pages 4960–4964. IEEE, 2016.

[130] Julian Salazar, Katrin Kirchhoff, and Zhiheng Huang. Self-attention networks for connectionist temporal classification in speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7115–7119. IEEE, 2019.

[131] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

[132] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

[133] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech 2020*, pages 5036–5040, 2020.

[134] Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alexander H. Waibel. Self-attentional acoustic models. *Interspeech*, 2018.

[135] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6381–6385. IEEE, 2019.

[136] Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. In *International Conference on Learning Representations*, 2018.

[137] Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, 2019.

[138] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7829–7833. IEEE, 2020.

[139] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer. Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6783–6787. IEEE, 2021.

[140] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE, 2020.

[141] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[142] Alex Graves. Sequence transduction with recurrent neural networks. 2012.

[143] Xiaohui Zhang, Frank Zhang, Chunxi Liu, Kjell Schubert, Julian Chan, Pradyot Prakash, Jun Liu, Ching-Feng Yeh, Fuchun Peng, Yatharth Saraf, et al. Benchmarking lf-mmi, ctc and rnn-t criteria for streaming asr. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 46–51. IEEE, 2021.

[144] Tom Bagby, Kanishka Rao, and Khe Chai Sim. Efficient implementation of recurrent neural network transducer in tensorflow. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 506–512. IEEE, 2018.

[145] Chunxi Liu, Frank Zhang, Duc Le, Suyoun Kim, Yatharth Saraf, and Geoffrey Zweig. Improving rnn transducer based asr with auxiliary tasks. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 172–179. IEEE, 2021.

[146] Rohit Prabhavalkar, Yanzhang He, David Rybach, Sean Campbell, Arun Narayanan, Trevor Strohman, and Tara N Sainath. Less is more: Improved rnn-t decoding using limited label context and path merging. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5659–5663. IEEE, 2021.

[147] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[148] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. 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)*, pages 193–199. IEEE, 2017.

[149] Hu Hu, Rui Zhao, Jinyu Li, Liang Lu, and Yifan Gong. Exploring pre-training with alignments for rnn transducer based end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7079–7083. IEEE, 2020.

[150] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR, 2014.

[151] Yajie Miao, Mohammad Gowayyed, and Florian Metze. 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)*, pages 167–174. IEEE, 2015.

[152] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[153] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE, 2016.

[154] Sadaoki Furui. Generalization problem in asr acoustic model training and adaptation. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 1–10. IEEE, 2009.

[155] Atsunori Ogawa and Takaaki Hori. Asr error detection and recognition rate estimation using deep bidirectional recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4370–4374. IEEE, 2015.

[156] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

[157] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[158] SH Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119, 2020.

[159] Ioana Vasilescu, Martine Adda-Decker, and Lori Lamel. Cross-lingual studies of asr errors: paradigms for perceptual evaluations. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3511–3518, 2012.

[160] Takahiro Shinozaki and Sadaoki Furui. Error analysis using decision trees in spontaneous presentation speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.*, pages 198–201. IEEE, 2001.

[161] Sharon Goldwater, Dan Jurafsky, and Christopher D Manning. Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication*, 52(3):181–200, 2010.

[162] Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke. Advances in all-neural speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4805–4809. IEEE, 2017.

[163] Jia Cui, Chao Weng, Guangsen Wang, Jun Wang, Peidong Wang, Chengzhu Yu, Dan Su, and Dong Yu. Improving attention-based end-to-end asr systems with sequence-based loss functions. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 353–360. IEEE, 2018.

[164] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. Improved training of end-to-end attention models for speech recognition. *Interspeech*, 2018.

[165] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu. Exploring neural transducers for end-to-end speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 206–213. IEEE, 2017.

[166] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech*, 2019.

[167] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al. English conversational telephone speech recognition by humans and machines. *Proc. Interspeech 2017*, pages 132–136, 2017.

[168] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[169] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF in CONF. IEEE Signal Processing Society, 2011.

[170] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[171] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016.

[172] Weiran Wang, Yingbo Zhou, Caiming Xiong, and Richard Socher. An investigation of phone-based subword units for end-to-end speech recognition. *Interspeech*, 2020.

[173] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849. IEEE, 2017.

[174] Siyuan Feng, Olya Kudina, Bence Mark Halpern, and Odette Scharenborg. Quantifying bias in automatic speech recognition. 2021.

[175] Maryam Sadat Mirzaei, Kourosh Meshgi, and Tatsuya Kawahara. Leveraging automatic speech recognition errors to detect challenging speech segments in ted talks. *CALL communities and culture–short papers from EUROCALL*, pages 313–318, 2016.

[176] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[177] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Narain Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *NIPS*, 2017.

[178] Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with svcca. *North American Chapter of the Association for Computational Linguistics*, 2018.

[179] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[180] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[181] Bruce Thompson. Canonical correlation analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[182] N Shawe-Taylor and A Kandola. On kernel target alignment. *Advances in neural information processing systems*, 14:367, 2002.

[183] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[184] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

[185] Ram Sundaram, Aravind Ganapathiraju, Jonathan Hamaker, Joseph Picone, et al. Isip 2000 conversational speech evaluation system. In *Speech Transcription Workshop, College Park, Maryland, USA*, 2000.

[186] Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *International Conference on Machine Learning*, pages 6972–6986. PMLR, 2020.

[187] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[188] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations*, 2017.

[189] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 2021.

[190] Aleksandr Laptev, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey Rybin. You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation. In *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 439–444. IEEE, 2020.

[191] Matthew Wiesner, Adithya Renduchintala, Shinji Watanabe, Chunxi Liu, Najim Dehak, and Sanjeev Khudanpur. Pretraining by backtranslation for end-to-end asr in low-resource settings. *Interspeech 2019*, pages 4375–4379, 2019.

[192] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

[193] Ryo Masumura, Naoki Makishima, Mana Ihori, Akihiko Takashima, Tomohiro Tanaka, and Shota Orihashi. Hierarchical transformer-based large-context end-to-end asr with large-context knowledge distillation. In *ICASSP 2021-2021 IEEE International*

*Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5879–5883. IEEE, 2021.

[194] John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, 2020.

[195] Yu-An Chung, Yonatan Belinkov, and James Glass. Similarity analysis of self-supervised speech representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3040–3044. IEEE, 2021.

[196] Jiarui Zhang, Yingxiang Li, Juan Tian, and Tongyan Li. Lstm-cnn hybrid model for text classification. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 1675–1680. IEEE, 2018.

[197] Emad M Grais, Fei Zhao, and Mark D Plumbley. Multi-band multi-resolution fully convolutional neural networks for singing voice separation. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 261–265. IEEE, 2021.

[198] Jui Shah, Yaman Kumar Singla, Changyou Chen, and Rajiv Ratn Shah. What all do audio transformer models hear? probing acoustic representations for language delivery and its structure. 2021.

[199] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[200] Jonathan Fiscus, George Doddington, Audrey Le, Greg Sanders, Mark Przybocki, and David Pallett. Nist rich transcription evaluation data. *Linguistic Data Consortium*, 2003.

[201] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Rwth asr systems for librispeech: Hybrid vs attention. *Proc. Interspeech 2019*, pages 231–235, 2019.

[202] Chanwoo Kim, Kean K Chin, Michiel Bacchiani, and Richard M Stern. Robust speech recognition using temporal masking and thresholding algorithm. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[203] Bryan Stroube. Literary freedom: Project gutenberg. *XRDS: Crossroads, The ACM Magazine for Students*, 10(1):3–3, 2003.

[204] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. 2015.

[205] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. Cold fusion: Training seq2seq models together with language models. *Proc. Interspeech 2018*, pages 387–391, 2018.

[206] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie. Component fusion: Learning replaceable language model component for end-to-end speech recognition system. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5361–5635. IEEE, 2019.

[207] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[208] Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *Proc. Interspeech 2017*, pages 523–527, 2017.

[209] Zhuo Gong, Daisuke Saito, Sheng Li, Hisashi Kawai, and Nobuaki Minematsu. Can we train a language model inside an end-to-end asr model?-investigating effective implicit language modeling. In *Proceedings of the Second Workshop on When Creative AI Meets Conversational AI*, pages 42–47, 2022.

[210] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *Proc. ICLR*, 2017.

[211] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, 2018.

[212] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019.

[213] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, 3(1):72–83, 1995.

[214] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[215] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak. Language recognition via i-vectors and dimensionality reduction. In *Twelfth annual conference of the international speech communication association*, 2011.

[216] Patrick Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal,(Report) CRIM-06/08-13*, 14(28-29):2, 2005.

[217] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

[218] Sergey Ioffe. Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542. Springer, 2006.

[219] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4052–4056. IEEE, 2014.

[220] Li Zhang, Qing Wang, and Lei Xie. Duality temporal-channel-frequency attention enhanced speaker representation learning. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 206–213. IEEE, 2021.

[221] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. pages 1021–1028. IEEE, 2018.

[222] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *Proc. Interspeech 2020*, pages 3610–3614, 2020.

[223] Dong Yu and Jinyu Li. Recent progresses in deep learning based acoustic models. *IEEE/CAA Journal of Automatica Sinica*, 4(3):396–409, 2017.

[224] Andrew Senior, Haşim Sak, and Izhak Shafran. Context dependent phone models for lstm rnn acoustic modelling. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4585–4589. IEEE, 2015.

[225] Mohit Yadav and Vivek Tyagi. Deep triphone embedding improves phoneme recognition. 2017.

[226] Yonatan Belinkov, Ahmed Ali, and James Glass. Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. *Proc. Interspeech 2019*, pages 81–85, 2019.

[227] Khaled Daqrouq and Tarek A Tutunji. Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers. *Applied Soft Computing*, 27:231–239, 2015.

[228] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *Proc. Interspeech 2020*, pages 3830–3834, 2020.

[229] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

[230] Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99, 2021.

[231] Navneet Upadhyay and Abhijit Karmakar. An improved multi-band spectral subtraction algorithm for enhancing speech in various noise environments. *Procedia Engineering*, 64:312–321, 2013.

[232] Maxim Tkachenko, Alexander Yamshinin, Nikolay Lyubimov, Mikhail Kotov, and Marina Nastasenko. Speech enhancement for speaker recognition using deep recurrent neural networks. In *International Conference on Speech and Computer*, pages 690–699. Springer, 2017.

[233] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot. But system description to voxceleb speaker recognition challenge 2019. *The VoxCeleb Speaker Recognition Challenge 2019*, 2019.

[234] Yun Tang, Guohong Ding, Jing Huang, Xiaodong He, and Bowen Zhou. Deep speaker embedding learning with multi-level pooling for text-independent speaker verification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6116–6120. IEEE, 2019.

[235] Yong Zhao, Tianyan Zhou, Zhuo Chen, and Jian Wu. Improving deep cnn networks with long temporal context for text-independent speaker verification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6834–6838. IEEE, 2020.

[236] Zhengyang Chen, Sanyuan Chen, Yu Wu, Yao Qian, Chengyi Wang, Shujie Liu, Yanmin Qian, and Michael Zeng. Large-scale self-supervised speech representation learning for automatic speaker verification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6147–6151. IEEE, 2022.

[237] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck. Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification. *Interspeech 2021*, pages 2302–2306, 2021.

[238] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *International Conference on Language Resources and Evaluation*, 2019.

[239] Hossein Zeinali, Lukáš Burget, and Jan Honza Černockỳ. A multi purpose and large scale speech corpus in persian and english for speaker and speech recognition: the deepmine database. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 397–402. IEEE, 2019.

[240] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[241] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[242] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[243] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[244] Hitoshi Yamamoto, Kong Aik Lee, Koji Okabe, and Takafumi Koshinaka. Speaker augmentation and bandwidth extension for deep speaker embedding. In *Interspeech*, pages 406–410, 2019.

[245] Emanuel AP Habets. Room impulse response generator. *Technische Universiteit Eindhoven, Tech. Rep*, 2(2.4):1, 2006.

[246] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. *Proc. Interspeech 2018*, pages 2252–2256, 2018.

[247] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Utterance-level aggregation for speaker recognition in the wild. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5791–5795. IEEE, 2019.

[248] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[249] Anna Ollerenshaw, Md Asif Jalal, Rosanna Milner, and Thomas Hain. Empirical interpretation of the relationship between speech acoustic context and emotion recognition. *Frontiers in Artificial Intelligence*, 2023.

[250] Olivier Martin, Irene Kotsia, Benoit Macq, and Ioannis Pitas. The enterface'05 audio-visual emotion database. In *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pages 8–8. IEEE, 2006.

[251] James L McClelland and Jeffrey L Elman. The trace model of speech perception. *Cognitive psychology*, 18(1):1–86, 1986.

[252] Eduardo Coutinho and Nicola Dibben. Psychoacoustic cues to emotion in speech prosody and music. *Cognition & emotion*, 27(4):658–684, 2013.

[253] Gabriela Ilie and William Forde Thompson. Experiential and cognitive changes following seven minutes exposure to music and speech. *Music Perception*, 28(3):247–264, 2011.

[254] Md Asif Jalal, Rosanna Milner, and Thomas Hain. Empirical interpretation of speech emotion perception with attention based model for speech emotion recognition. In *Proceedings of Interspeech 2020*, pages 4113–4117. International Speech Communication Association (ISCA), 2020.

[255] Björn Schuller, Gerhard Rigoll, and Manfred Lang. Hidden markov model-based speech emotion recognition. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–1. Ieee, 2003.

[256] Shiqing Zhang, Shiliang Zhang, Tiejun Huang, and Wen Gao. Multimodal deep convolutional neural network for audio-visual emotion recognition. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 281–284, 2016.

[257] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 2227–2231. IEEE, 2017.

[258] Che-Wei Huang and Shrikanth S Narayanan. Attention assisted discovery of sub-utterance structure in speech emotion recognition. In *Interspeech*, pages 1387–1391, 2016.

[259] Zhengwei Huang, Ming Dong, Qirong Mao, and Yongzhao Zhan. Speech emotion recognition using cnn. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 801–804, 2014.

[260] Jaebok Kim, Gwenn Englebienne, Khiet P Truong, and Vanessa Evers. Deep temporal models using identity skip-connections for speech emotion recognition. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1006–1013, 2017.

[261] Anish Nediyanchath, Periyasamy Paramasivam, and Promod Yenigalla. Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7179–7183. IEEE, 2020.

[262] Zheng Lian, Bin Liu, and Jianhua Tao. Ctnet: Conversational transformer network for emotion recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:985–1000, 2021.

[263] Deirdre Wilson and Tim Wharton. Relevance and prosody. *Journal of pragmatics*, 38(10):1559–1579, 2006.

[264] Katie Hoemann, Alyssa N Crittenden, Shani Msafiri, Qiang Liu, Chaojie Li, Debi Roberson, Gregory A Ruark, Maria Gendron, and Lisa Feldman Barrett. Context facilitates performance on a classic cross-cultural emotion perception task. *Emotion*, 19(7):1292, 2019.

[265] Klaus R Scherer, Rainer Banse, and Harald G Wallbott. Emotion inferences from vocal expression correlate across languages and cultures. *Journal of Cross-cultural psychology*, 32(1):76–92, 2001.

[266] William Forde Thompson and LL Balkwill. Decoding speech prosody in five languages. *Semiotica*, 158(1/4):407–424, 2006.

[267] Christian Stilp. Acoustic context effects in speech perception. *Wiley Interdisciplinary Reviews: Cognitive Science*, 11(1):e1517, 2020.

[268] Keith R Kluender, Christian E Stilp, and Fernando Llanos Lucas. Long-standing problems in speech perception dissolve within an information-theoretic perspective. *Attention, Perception, & Psychophysics*, 81(4):861–883, 2019.

[269] Marco Van de Ven and Mirjam Ernestus. The role of segmental and durational cues in the processing of reduced words. *Language and Speech*, 61(3):358–383, 2018.

[270] Daniel Fogerty and Diane Kewley-Port. Perceptual contributions of the consonant-vowel boundary to sentence intelligibility. *The Journal of the Acoustical Society of America*, 2009.

[271] Michael J Owren and Gina C Cardillo. The relative roles of vowels and consonants in discriminating talker identity versus word meaning. *The Journal of the Acoustical Society of America*, 2006.

[272] Ronald A Cole, Yonghong Yan, Brian Mak, Mark Fanty, and Troy Bailey. The contribution of consonants versus vowels to word recognition in fluent speech. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 2, pages 853–856. IEEE, 1996.

[273] Peter Ladefoged and Donald Eric Broadbent. Information conveyed by vowels. *The Journal of the acoustical society of America*, 29(1):98–104, 1957.

[274] Danying Xu, Fei Chen, Fan Pan, and Dingchang Zheng. Factors affecting the intelligibility of high-intensity-level-based speech. *The Journal of the Acoustical Society of America*, 146(2):EL151–EL157, 2019.

[275] Franklin S Cooper, Pierre C Delattre, Alvin M Liberman, John M Borst, and Louis J Gerstman. Some experiments on the perception of synthetic speech sounds. *The Journal of the Acoustical Society of America*, 1952.

[276] Joanne L Miller. On the internal structure of phonetic categories: A progress report. *Cognition*, 50(1-3):271–285, 1994.

[277] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[278] T Mani Kumar, Enrique Sanchez, Georgios Tzimiropoulos, Timo Giesbrecht, and Michel Valstar. Stochastic process regression for cross-cultural speech emotion recognition. *Proc. Interspeech 2021*, pages 3390–3394, 2021.

[279] Rory Beard, Ritwik Das, Raymond WM Ng, PG Keerthana Gopalakrishnan, Luka Eerens, Pawel Swietojanski, and Ondrej Miksik. Multi-modal sequence fusion via recursive attention for emotion recognition. In *Proceedings of the 22nd conference on computational natural language learning*, pages 251–259, 2018.

[280] Hynek Hermansky and Sangita Sharma. Traps-classifiers of temporal patterns. In *Fifth International Conference on Spoken Language Processing*, 1998.

[281] Rosanna Milner, Md Asif Jalal, Raymond WM Ng, and Thomas Hain. A cross-corpus study on speech emotion recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 304–311. IEEE, 2019.