Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

Department of Electronic and Electrical Engineering

Deep Cross-Modal Alignment in Audio-Visual Speech Recognition

George Sterpu

July 5, 2021

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: _____ Date: _____

Abstract

Modern studies in cognitive psychology have demonstrated that speech perception is a multimodal process, as opposed to a purely auditory one with visual carryover as in the classic view. This led researchers to investigate the nature of the audio-visual speech integration process in the brain. The ability to combine the two sources of information delivering uncertain predictions improves the recognition of speech. In this thesis we aim to develop efficient machine learning algorithms and computational models of audio-visual speech recognition (AVSR) that learn to capitalise on the visual modality from examples.

My original contribution to knowledge is an efficient strategy for the multimodal alignment and fusion of audio-visual speech on the task of large vocabulary continuous speech recognition. This strategy, termed AV Align, makes limited use of domain knowledge, but exploits the hypothesis that there is an underlying alignment between the higher order representations of the audio and visual modalities of speech. To achieve a controllable decoding latency, we develop a speech segmentation strategy termed Taris. This strategy aims to segment a spoken utterance by learning to count the number of words from speech data.

Our multimodal systems are presented with audio and video recordings of speech from two large vocabulary audio-visual speech datasets, TCD-TIMIT and LRS2. We corrupt the audio channel with noise taken from a cafeteria environment at three signal to noise ratios. For each noise condition, we evaluate the character error rate of the multimodal system, and compare it to an equivalent audio-only system trained on the same data to assess the added benefit of the visual modality to speech recognition.

We show empirically that AV Align discovers a monotonic trend in the alignment between the audio and visual modalities. This monotonicity is achieved while AV Align is allowed to search for a soft alignment across full speech utterances, without any supervision or constraints placed on the alignment pattern. On LRS2, the most challenging audio-visual speech dataset used in this work, AV Align obtains improvements over an audio-only system ranging from 6.4% under clean speech conditions up to around 31% at the highest level of audio noise. These improvements were made possible after an exploration of the learning difficulties specific to the audio-visual speech recognition task, which led us propose a multitask learning approach based on estimating the intensities of two facial action units from video.

We also show that the word counting objective of Taris favours the segmentation

of speech into units following a similar length distribution as the one of word units estimated with forced aligner. The correlation between our segments and the word units remains only speculative. Since we design the decoding process of Taris to be robust to segmentation imperfections, we achieve a comparable level of accuracy with equivalent systems that make full use of the utterance-level context and are indifferent to latency.

Our findings reflect that we have discovered two well informed modelling assumptions contributing to the domain knowledge of audio-visual speech. The first one is the underlying higher order fusion of cross-modally aligned audio and visual speech representations. The second one is the possibility to learn the word count in a spoken utterance from either audio and audio-visual cues as a mechanism to segment transcribed speech lacking intermediate alignments. Both AV Align and Taris have objectives expressed as fully differentiable functions of the parameters. We believe these will be key ingredients to the adoption of audio-visual speech recognition technology into real products in the years to come.

Acknowledgements

I would like to thank my supervisor, Prof Naomi Harte, for her continued guidance and support over the last four years. I am incredibly grateful for having the chance to work you, and I owe my development as a researcher to your outstanding supervision. It has equally been an honour to work with Christian Saam. Thank you both for your valuable advice, feedback, and interesting discussions.

I am extremely grateful to Jenny Kirkwood for helping me settle in Ireland during a time of soaring rents. It has been a wonderful experience to live with the family of Bridget and Gary in Glenageary, and at Brigid's home in Bray.

Thank you João for your incredible collegiality. I am thankful to the rest of my colleagues in Trinity College and from Stack B: Wissam, Matt, Marco, Hugh, Luke, Matej, Jing, Chao, Ylva, Adriana, Marie-Caroline, Anna, Daniel, Sebastian, Sebastien, Ali, Anil, Francois, Rodrigo, and John Sloan from UCD. I wish the best of luck to Ayushi, Mark, and Edward, who are at the start of their doctoral journeys. Thanks John, Conor, and Graziano for all your technical support with the computing infrastructure. It has been an honour to be part of the ADAPT Centre, and attend the frequent scientific and social events they organised.

I am grateful to have met many colleagues in academia along my journey: Subhadeep, Vinayak, Weipeng, Skanda, Alexis, Dimitra, Francois, Bastien, Pavan, Athenais, Romain, Nam, Lesly, Tatjana, Suraj, Prof Francois Fleuret, Prof Phil Garner, Alexandre. I would like to thank my previous advisors Prof Laurent Girin, Georgios, Prof Radu Horaud, and Grigore Stamatescu, for indirectly guiding me towards a PhD. I am thankful to my examiners, Prof Mark Gales and Prof Gerasimos Potamianos, for providing valuable feedback during my thesis defence.

I greatly enjoyed all the fun evenings practising with the tennis club in Trinity College, with coach Kristine, and privately with Fernando, Abhik, and Darren. Spending time with you on the courts has been the most brilliant way to disconnect from work. I am also grateful to all the dedicated staff in Trinity College for the great support they provide to students.

This thesis would not have been possible without the effort of the open-source contributors to knowledge. Thanks for every useful response posted online, for every great open source software tools.

Last but not least, I would like to thank my family, who understood my total dedication to my degree in the last four years, and accepted the almost total disconnection. I dedicate this thesis to the memory of my mom.

Contents

List of Figures

List of Tables

Notation

The following notation has been used throughout this thesis.

α_{ji}	Normalised attention score between query timestep *j* and the key timestep *i*
η	Alphabet size for the character-based models
A	An audio input sequence
a_i	Timestep *i* of the *A* audio sequence
b_k	Bias vector
c_i	Attention context vector corresponding to query timestep *i*
d_{FF}	Transformer feed-forward state size
d_{LA}	Taris decoder look-ahead segments
d_{LB}	Taris decoder look-back segments
d_{model}	Transformer state size
e_{LA}	Taris encoder look-ahead frames
e_{LB}	Taris encoder look-back frames
F	Taris decoder-encoder attention mask
L	Label sequence length
M	Video sequence length
N	Audio sequence length
o_A	Audio encoder output
o_V	Video encoder output
o_{AV}	Fused audio-visual speech representations
V	A video input sequence

v_i Timestep i of the V video sequence

W_k Weight matrix

X A generic input sequence

x_i Timestep i of the X input sequence

Y A label output sequence

y_i Timestep i of the Y label sequence

Acronyms

AAM Active Appearance Model

ASR Automatic Speech Recognition

AU Action Unit

AVSR Audio-Visual Speech Recognition

BN Batch Normalisation

CE Cross-Entropy

CER Character Error Rate

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

DCT Discrete Cosine Transform

DNN Deep Neural Network

EOS End of Sentence

FFN Feed-Forward Network

GMM Gaussian Mixture Model

HMM Hidden Markov Model

LM Language Model

LN Layer Normalisation

LSTM Long Short-Term Memory

PE Positional Encoding

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

RNN-T Recurrent Neural Network Transducer

ROI Region of Interest

SNR Signal-to-Noise Ratio

STFT Short-Time Fourier Transform

WER Word Error Rate

1 Introduction

1.1 Speech and language

The human language is arguably one of the most important tools of our time. While its origins remain the subject of an ongoing debate due to limited direct evidence, in modern times language has become an integral part for internal thought (Berwick and Chomsky, 2015). The externalisation of language through articulation enables a highly efficient form of human communication, namely speech. Being able to recognise speech reliably using a computer enables a large set of applications such as voice assistants, voice-based search engines, voice command and dictation, spoken content retrieval, customer support, education, health care and others. Such applications allow us to interact more naturally with computers.

1.1.1 Speech perception is inherently multimodal

Until the early 1950s, it was widely believed that speech perception is primarily an auditory process, with a secondary visual component available for lipreading. It was only with the discovery of a perceptual illusion by McGurk and MacDonald (1976) that speech perception started to be thought of as being inherently multimodal. In this illusory effect, which now bears the name of the first author of the study, McGurk and MacDonald (1976) combine conflicting audio-visual stimuli of a voice saying the syllables *ba-ba* dubbed onto the video of a face pronouncing *ga-ga*. They show that human subjects are very likely to perceive the syllables *da-da* instead. Dodd (1977) reached similar findings on full words rather than isolated syllables. An example from that study dubbing incoherent audio and visual stimuli for the words *tough* and *hole* respectively shows that some subjects recognise the word *towel*, which was not included in the list of words presented to the participants. This led to the development of an entirely new theory of audio-visual speech integration in the brain.

The visual modality of speech is believed to provide multiple sources of informa-

tion that complement the auditory one. Summerfield (1987) explains this complementarity on the basis of place and manner of articulation. More precisely, the sounds that are most difficult to comprehend in noise are also the ones that are the easiest to recognise on the lips. Conversely, the sounds made with the less visible speech articulators are more robust to noise masking. Massaro and Stork (1998) give an example for the nasal sounds /m/ and /n/, which may sound very similar in noisy conditions, but can be visually distinguished since the lips are closed at onset for /m/, whereas they are open for /n/. Likewise, the sounds /f/ and /v/ look the same on the lips, whereas they can be distinguished acoustically by voicedness (vibration of the vocal folds). When a word begins with one of the visually distinguishable consonants, Summerfield (1987) argues that there are fewer alternative explanations for the completion of the word. This aspect suggests that, even under clean speech conditions, there may be a lower cognitive load for perceiving audio-visual speech than from hearing alone. Summerfield (1987) adds that the visual modality contributes to the spatial localisation of the speaker and to the detection of voice activity. We see that such advantages have the potential to improve the task of an automatic speech recognition system in the realistic setting of loud ambient noises or multi-party conversations.

Two main research questions arise from the study of Summerfield (1987). First, at what stage are the auditory and visual representations integrated with respect to the phonetic categorisation ? An eventual integration before categorisation implies the existence of a classifier working directly on a multimodal representation. On the other hand, a late integration allows the fusion of decisions from two modality-specific classifiers. Second, how are the two modalities represented prior to integration ? To date, this integration phenomenon in the brain remains an open challenge. The computational model proposed in this thesis provides an answer to both of these questions. As it will become clear in Chapter 4, our motivation is founded in the learning theory and resorts to well-informed prior knowledge of speech to accomplish efficiency.

Audio-visual systems can potentially make speech recognition technology more valuable in acoustically noisy environments. Some examples include noisy streets downtown, restaurants, reverberant rooms, cars, or public transport. It has been estimated that humans can tolerate on average up to a 15dB decrease of the speech to noise ratio when they leverage their sight while listening in such less ideal conditions (Macleod and Summerfield, 1987; Sumby and Pollack, 1954). This represents the main benefit of incorporating vision into speech recognition. In most of the environments listed above, we expect the technology to work interactively. For example, we may be at a museum asking for directions, place

an order at a restaurant, or check-in at a hotel lobby. In these situations, it is important to guarantee a short response time to ensure an interactive communication. In Chapter 5 of this thesis, we will follow up on the proposed audio-visual integration model and will update its computational requirements to control the overall latency.

1.2 Deep learning in speech recognition

Given the high complexity of human language, speech modelling arguably becomes unmanageable when defined by a set of rules derived from expert knowledge. This led to the development of machine learning approaches that can learn from examples and partly substitute human expertise. In particular, the area of neural networks has seen remarkable advancements in the last four decades. Falling under the umbrella of *deep learning*, LeCun et al. (2015) describe these advancements as having accomplished breakthroughs in the areas of image, video, and speech processing. The backpropagation algorithm allowed researchers to learn meaningful internal representations with neural networks, as opposed to designing engineered features (Rumelhart et al., 1986a). This reduced the importance of searching for robust features in speech. Convolutional Neural Networks (LeCun et al., 1989) with residual connections (He et al., 2016b) proved to be a good inductive bias for the efficient learning of representations from images. Recurrent neural networks with gated cells (Gers et al., 1999; Hochreiter and Schmidhuber, 1997) enabled the efficient modelling of time series with long term dependencies, and sidestepped the necessity to annotate a speech corpus at the phonetic level (Bahdanau et al., 2015; Graves et al., 2006). As opposed to the traditional approach, these networks no longer required design constraints from experts. Instead, the concept of *end-to-end* training gained popularity in many areas including speech processing. Better optimisation strategies (Kingma and Ba, 2015) and well tuned initialisation and regularisation methods (Pascanu et al., 2013) helped the methods based on stochastic gradient descent achieve better convergence to optimal solutions.

1.2.1 Modelling hypotheses

In spite of these major breakthroughs, deep learning has not fully provided an answer to the main research questions in audio-visual speech integration. The mere fact that the internal representations are transformations achieved by neural networks does not explain *when* the modality representations are integrated and *what* their form is right before integration. (Goodfellow et al., 2016, Section

6.4) explain that, according to the *universal approximation theorem*, a sufficiently large fully connected neural network *can* represent most functions, but learning may be unsuccessful for generalisation. (Goodfellow et al., 2016, Section 5.11) add that, to achieve good generalisation, we need to incorporate good prior beliefs into machine learning algorithms, which limit the family of functions (or the hypothesis space) that can be learnt. LeCun (1989) characterises the likelihood of correct generalisation based on three factors: *(i)* the size of the hypothesis space, *(ii)* the size of the solution space, and *(iii)* the number of training observations. While the second factor is innate to our task, there is a close interplay between the first and the third ones. Therefore, it is important to search for appropriate network architectures in order to adequately control the representation capacity of the model and increase the chance of correct generalisation given sufficient data points. Currently, there is no consensus on which prior assumptions are useful in automatic audio-visual speech recognition. We will review them in Section 2.6.

1.3 Objectives

The aim of this thesis is to investigate computational strategies of audio-visual speech integration on the task of automatic speech recognition in adverse auditory conditions. To this end, in the first part of the thesis, we propose a new computational model that takes better advantage of the auditory and visual modalities of speech. This model, coined *AV Align*, learns an explicit cross-modal alignment between the representations of the two input streams. A major design goal in this thesis is simplicity on multiple levels. We only use differentiable objective functions with respect to every model parameter to facilitate a straightforward training pipeline. The second part of this thesis addresses a limitation of the multimodal system proposed in the first part. This limitation is not specific to the model we proposed, but is an underlying issue of the generic sequence to sequence neural network architecture that our model extends. As a result, the finding we report in Chapter 5 has a much broader impact outside multimodal fusion.

1.4 Thesis outline

Chapter 2 reviews the current status of audio-based speech recognition technology and its main challenges. This will motivate the need for multimodal speech recognition systems. Then, we investigate in more depth the advantages offered by the visual modality. We will refer to multiple experiments in cognitive psychology that demonstrate the primacy of multimodality for speech perception and

offer specific insights into the nature of audio-visual integration. We will then analyse the major neural network architectures that have been used in speech recognition, as they represent the foundations of the multimodal extension proposed in this thesis. The review in this chapter illustrates that while neural networks have surpassed the performance of traditional systems in speech recognition on certain benchmarks, there are several latency considerations that need to be addressed. We then introduce and motivate the two audio-visual speech datasets used in this work. Finally, we describe the main evaluation metric used in speech recognition and discuss some of its limitations.

Chapter 3 investigates the role of the modelled linguistic unit on the task of continuous visual speech recognition. Multiple studies have shown that we can only perceive a limited set of distinctive units from vision. We aim to explore the feasibility of lip-reading characters or phonetic units as opposed to visual ones. This will motivate a design choice in the following chapter where the system has both modalities available for decoding.

Chapter 4 is the central part of our proposed strategy for audio-visual speech integration, *AV Align*. Here we offer an answer to the question of how should the two speech modalities be represented right before integration. In AV Align, an audio representation is merged with a soft-aligned visual representation that relates to the amount of correlation with the respective audio representation. We explore in greater detail the strengths and limitations of AV Align, and propose a multi-task approach based on regressing facial action units to circumvent the observed convergence problem. We then compare AV Align with a simpler fusion strategy based on feature concatenation, and also with the frequently cited method *Watch, Listen, Attend, and Spell* (WLAS) of Chung et al. (2017).

Chapter 5 addresses a major limitation of the models used in Chapter 4 to enable the online decoding of speech. Here we propose a model coined *Taris*, that segments a spoken utterance by learning to count the number of words within the utterance. We show this results in an end-to-end differentiable solution to speech recognition, as opposed to alternative strategies that rely on dynamic programming. We evaluate Taris both on English and Mandarin speech. Finally, we evaluate an audio-visual extension of Taris that learns to count words from both auditory and visual cues.

Chapter 6 discusses the significance and impact of the original contributions presented in this thesis. We also provide a deep insight into the remaining challenges and outline six major directions of future work.

1.5 Original contributions

This thesis develops the *AV Align* strategy for cross-modal alignment and fusion of audio and visual speech modalities, and the *Taris* strategy for learning the segmentation of a speech utterance into word-approximating units. Taken together, we contribute an end-to-end trainable audio-visual speech recognition model based on neural networks that is capable of online decoding.

The original contributions to knowledge can be summarised as follows:

Chapter 3

- A direct comparison between traditional models and sequence to sequence neural networks on the task of large vocabulary continuous visual speech recognition

Chapter 4

- AV Align, an audio-visual speech integration strategy based on the explicit cross-modal alignment of higher order audio representations with higher order visual representations
- An investigation into the convergence issues of the machine learning algorithm on the task of audio-visual speech recognition
- An auxiliary loss function based on the regression of three Facial Action Units
- Showing that AV Align can discover plausible alignments with a monotonic trend between the audio and visual speech modalities
- A comparison between alternative implementations of AV Align using LSTM and Transformer networks, which investigates a hypothesis regarding the learning difficulties faced by the former architecture
- An empirical demonstration that the Transformer network achieves an implicit alignment between modalities, in the absence of the cross-modal alignment module

Chapter 5

- Taris, a speech modelling strategy that aims to segment an utterance by learning to count the number of words spoken
- An empirical finding that the segments estimated by Taris follow a segment length distribution similar to the word length distribution estimated with a

forced alignment tool

- A variant of Taris that learns to count Chinese characters instead of words in spoken Mandarin
- An audio-visual extension of Taris that applies cross-modal alignment constrained to a fixed temporal window

Chapter 6

- An outline of a learning algorithm for audio-visual speech recognition systems that aims to make the auxiliary Action Unit loss obsolete
- An outline of two training strategies for audio-visual speech recognition systems that promote the learning of an audio-visual system which is superior or at least on a par with an audio-only system for the measured accuracy

1.5.1 Publications

The work in this thesis has been in part disseminated in the following publications: Sterpu and Harte (2017); Sterpu et al. (2018a,b); Sterpu et al. (2020a,b); Sterpu et al. (2021)

1. **Sterpu, G**., Saam, C., and Harte, N. (2021). Learning to Count Words in Fluent Speech Enables Online Speech Recognition. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), 2021, pp. 38-45, DOI: 10.1109/SLT48900.2021.9383563.
2. **Sterpu, G**., Saam, C., and Harte, N. (2020b). Should we hard-code the recurrence concept or learn it instead ? Exploring the Transformer architecture for Audio-Visual Speech Recognition. In Proceedings of Interspeech 2020, pp. 3506-3509, DOI: 10.21437/Interspeech.2020-2480.
3. **Sterpu, G**., Saam, C., and Harte, N. (2020a). How to Teach DNNs to Pay Attention to the Visual Modality in Speech Recognition. In IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1052-1064, 2020, DOI: 10.1109/TASLP.2020.2980436.
4. **Sterpu, G**., Saam, C., and Harte, N. (2018a) Attention-based Audio-Visual Fusion for Robust Automatic Speech Recognition. In Proceedings of the 20th ACM International Conference on Multimodal Interaction (ICMI '18). Association for Computing Machinery, New York, NY, USA, pp. 111–115, DOI: 10.1145/3242969.3243014
5. **Sterpu, G**., Saam, C., and Harte, N. (2018b). Can DNNs Learn to Lipread

Full Sentences? In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 16-20, DOI: 10.1109/ICIP.2018.8451388.

6. **Sterpu, G**. and Harte, N. (2017). Towards Lipreading Sentences with Active Appearance Models. In Proceedings of the 14th International Conference on Auditory-Visual Speech Processing, pp. 70-75, DOI: 10.21437/AVSP.2017-14.

All articles have been made publicly available on my personal website and on the arXiv pre-print server. They were initially disseminated after the peer review process. To accelerate dissemination, we have made available the more recent publications before peer review.

1.5.2 Open-source software projects

A major goal of this thesis was to promote the reproducibility of the experiments. To achieve it, we applied the following principles. First, we only relied on publicly available software and datasets that are free to use for academic research. Second, the software written for the experiments in each chapter was documented and made publicly available on the GitHub platform together with the associated publication. We released the following projects.

- https://github.com/georgesterpu/**Taris**
  A Transformer-based online speech recognition system implemented in TensorFlow 2. It has been used for all Transformer-related experiments in Chapters 4 and 5

- https://github.com/georgesterpu/**avsr-tf1**
  Implements the AV Align strategy in TensorFlow 1.x using sequence to sequence LSTM networks. It has been used for all LSTM-related experiments in Chapters 3 and 4

- https://github.com/georgesterpu/**pyVSR**
  A Python toolkit for lipreading using a traditional pipeline. It offers support for the extraction of visual features (DCT, AAM) and implements a wrapper for the HTK toolkit for the modelling of the visual features using HMMs. It has been used for the lipreading experiments in Section 3.5.

2 Approaches to Speech modelling

The purpose of this literature review is to provide a structured foundation for the original contributions of this work in the space of audio-visual speech recognition. We will first highlight the importance of this topic in relation to audio-only speech recognition, to have a clearer big picture of the potential offered by the visual modality of speech. Next, we will introduce the terminology needed to discuss the prior work in automatic speech recognition, and then analyse the most established methods for the modelling of auditory speech. This analysis will be extended to multimodal approaches. Here we will pinpoint the major gaps in the AVSR literature into which our multimodal alignment and fusion strategy *AV Align* discussed in Chapter 4 is situated. We will then discuss the structural shortcomings of speech recognition models that are currently limiting their use in practical settings. This gap will be filled with our system *Taris* in Chapter 5, another original contribution to the speech recognition domain. Finally, we will describe and motivate our choice for the two audio-visual datasets used in this work.

In this literature review, we include a detailed technical presentation of the system architectures that our contributions are based on. Older systems based on traditional architectures will be analysed and interpreted with respect to the significance of their findings to the progress in ASR and AVSR research. However, as some of their limitations are now well understood by the research community, their specific details will be omitted from the discussion. We believe this will be necessary to protect the integrity of the chapter and guide the readers seamlessly through the different sections and topics.

2.1 Current status of audio speech recognition

Over the last 10 years, there has been a remarkable progress in the space of automatic speech recognition (ASR). Recent experiments show that the technology is rapidly approaching human-level performance on conversational speech. Saon et al. (2017) take a look at two of the most challenging publicly available speech

datasets, namely Switchboard and CallHome, part of the NIST 2000 evaluation of conversational speech (Fiscus et al., 2000). They first measure the performance of human annotators, reporting a 5.1% Word Error Rate (WER) on Switchboard, and 6.8% on CallHome. Their best automatic system scored very close on the former dataset, 5.5%, thought to be easier, but achieved 10.3% WER on the latter. These performance differences led them to the conclusion that human parity has not been achieved yet. Xiong et al. (2017) disagreed with the methodology of Saon et al. (2017) for measuring the human level performance. In particular, Xiong et al. (2017) claimed that pre-exposing the annotators to the data characteristics and introducing a second pass refinement step could have biased the results in favour of the human annotators. Xiong et al. (2017) also perform their own assessment of the human level performance on the same two datasets, reaching figures of 5.9% and 11.3% respectively. With their best automatic system, Xiong et al. (2017) reach WERs of 5.8% and 11.0% respectively, just slightly lower than the human baseline they measured. Looking at these performance figures, it may seem that conversational speech recognition is an almost solved task. On a closer inspection of the evaluation data, Fiscus et al. (2000) describes the recording conditions as "fairly clean", with peak Signal to Noise Ratios (SNR) above 40dB. While the debate on reaching human parity remains unsettled, it is clear that an eventual verdict would only be applicable to clean audio conditions. This remains an open question for many environments naturally exposed to more and stronger noise sources.

To bridge this knowledge gap, researchers started to look into more acoustically challenging settings. Barker et al. (2018) recorded a large scale speech dataset of multi-party conversations, *CHiME-5*, using six different devices of four microphones each. This dataset simulates a natural dinner conversation scenario with no imposed structure, and all the participants are familiar with each other. Saon et al. (2017) previously argued in their study that such casual conversations make the speech recognition task harder because the dialogues become less structured. On CHiME-5, Barker et al. (2018) report an error rate of 67.2% with a similar system used by Saon et al. (2017) and Xiong et al. (2017). This high error rate can be attributed to the relatively small dataset containing 40 hours of training data, whereas Saon et al. (2017) and Xiong et al. (2017) combine multiple sources of data totalling more than 2,000 hours of recordings. To the best of our knowledge, no study to date has attempted to pre-train the speech model on a similar amount of conversational speech data before evaluating in the challenging conditions of CHiME-5. They also report that the error rate increases to 94.7% when recognising the speech from the distant microphones as opposed

to the ones attached to each speaker. It is worth noting that better results were obtained on this dataset with highly specialised systems, although the absolute error rate is several times higher than the 11% conversational speech baseline on CallHome. Renals and Swietojanski (2017) evaluate multiple systems on the AMI speech corpus of multiparty meetings, which contains a larger amount of over 100 hours of recordings, but the dialogues are considered less natural and diverse. They report speech error rates of the same magnitude as in Barker et al. (2018), in spite of using a larger, less realistic corpus. Watanabe et al. (2020) report an absolute 15% error increase on the same dataset when their specialised system needs to additionally perform speaker diarisation as opposed to receiving oracle segmentations. Therefore, the combined factors of a complex dialogue structure, overlapped speech, environmental noise, and the natural degradation of the speech signal captured by distant microphones, are currently making ASR technology impractical for this type of setting.

Making use of multi-microphone arrays to help speaker diarisation poses several technical challenges. Barker et al. (2018) and Watanabe et al. (2020) report difficulties in synchronising the four audio channels of a commercially available device used to record CHiME-5 due to clock drift and frame skipping. To correct for it, they first need to align all the recorded signals by playing a synchronisation tone. Afterwards, they re-estimate every ten seconds a binaural time delay with respect to one reference channel. We can see that guaranteeing proper synchronisation is technologically expensive, while the compensation procedure may still be prone to errors.

2.2 The promise of multimodal integration

Several of the aforementioned challenges can benefit from the addition of vision. Binnie et al. (1974) show that the visual modality provides reliable information about the place of articulation, which is the articulatory feature most severely impacted by noise masking, as opposed to voicing or nasality. Dodd (1977) confirms this finding with an experiment on 25 secondary school children asked to reproduce one word at a time spoken by an instructor, while listening to various levels of white noise played through headphones. Dodd reports that most of the errors made by the subjects come from the same place of articulation, e.g. consonant /m/ is frequently confused with /p/ or /b/. Multiple studies, such as the one of Macleod and Summerfield (1987) or Sumby and Pollack (1954), have found experimentally that the use of the visual modality in speech recognition compensates for about 11dB at critical levels of reception. Summerfield (1987) argues

that the visual modality can provide the maximum benefit to speech intelligibility in the auditory SNR range from 6dB to -5dB, where the amount of correctly recognised words in a spoken sentence decreases from 90% down to 10%. Summerfield (1987) writes that the visual modality facilitates the localisation of the sound source other than from sound, which helps the listener in the task of speaker diarisation. Additionally, Summerfield (1987) mentions the segmental visual cues of speech and voice activity detection. Multimodal systems may be more robust to asynchronies between channels. Campbell and Dodd (1980) show that the human brain can store the visual stimuli for up to 1600ms. This has the potential to alleviate the necessity for a large number of microphone channels and for their regular resynchronisation. All these examples demonstrate the complementarity of audio and visual speech for perception.

The visual modality may also be well suited for the convenient setup of distant speech recognition. Jordan and Sergeant (2000) show that congruent audio-visual stimuli can enhance the perception of isolated syllables even from distances of 30 metres. Previously, Jordan and Sergeant (1998) have also found that subjects can tolerate a reduction of the facial image displayed on a monitor 1 metre away even when the image is reduced to 10% of its original size. This suggests that expensive visual equipment for recording at high resolutions may not be necessary in audio-visual speech recognition. At such reasonable distances, the noises that can naturally occur are then mostly owed to occlusion or a high variability of the head pose with partial visibility of the speech articulators.

McGurk and MacDonald (1976) show that the auditory and visual cues are inseparable in speech perception. They find that 98% of adults perceive the sounds *ba-ba* dubbed onto a face pronouncing *ga-ga* as a different intermediate category *da-da*. Whereas *ba* is a bilabial sound produced at the front of the mouth, and *ga* is velar consonant articulated with the back part of the tongue, *da* is typically articulated with the tip of the tongue at the alveolar ridge found behind the upper teeth, thus an intermediate position in the mouth between *ba* and *ga*. They also report that the effect occurs when even objectively knowing about the incoherence of the two channels. Later, Jordan and Sergeant (2000) show that the McGurk effect only diminishes when the subject is 20 meters away or more from the stimuli, where it becomes more likely for a shift of attention to occur. In a similar experiment using words instead of syllables, Dodd (1977) reports that conflicting audio and visual stimuli for the words *tough* and *hole* respectively led some subjects to recognise the word *towel*, which was not included in the list of words presented to the participants. The studies of McGurk and MacDonald (1976) and Dodd (1977) therefore show that conflicting stimuli result in a com-

petition, which can not be resolved by dismissing one of the modalities, but by providing a compromise between the two. The importance of their studies was to highlight the inadequacy of the common belief that speech perception is a purely auditory process. Finding good modelling approaches for audio-visual speech data is one step forward in the direction of understanding how the human brain integrates multiple sources of information.

To be able to discuss the technological advancements in audio and audio-visual automatic speech recognition, we first need to introduce the terminology regarding speech and its representation in a computer. In the next section we will walk together through the concept of speech as a time series, its partitioning into successive overlapping audio frames, and its transformation into a time-frequency representation.

2.3 Speech terminology

To externalise our thoughts through speech, we chain together one or more words in our mind, and use our vocal tract to produce a vibration in the air which propagates as an acoustic wave. A receptor, such as the microphone, converts the acoustic wave into an electric signal through sampling at uniform intervals. Since much of the energy of human voices is typically below 4,000 Hz, the sampling rates of the speech signal are above 8kHz, commonly of 16 or 48 kHz. At this stage, speech is represented in a computer memory as a time series, encoding the amplitude of the sampled signal in time. One of the most widely used representations is Pulse-code modulation (PCM) (Oliver et al., 1948). The PCM representation, often termed in speech research literature as the *raw waveform*, is considered one of the most general way to represent the audio signal.

Practical experience has led to the observation that the properties of the speech signal change very slowly over short time windows on the order of 10ms to 40ms (Rabiner and Schafer, 2010, Section 6.2). This enables the analysis of speech in short frames, which are commonly shifted by smaller amounts such as 10ms, allowing an overlap between consecutive frames. Using the Short-Time Fourier Transform (STFT), we obtain a two-dimensional time-frequency description of the one-dimensional audio waveform. An advantage of this transformation is the reduction in the number of elements in the time dimension. For example, if one second of audio is recorded as a one dimensional vector of 48,000 scalars, a typical time-frequency transformation (25ms frame length, 10ms frame increment, 512 frequency bins) produces a matrix of shape [100 x 512]. This represents a reduction by a factor of 480 along the time axis. As we will see in the

next section, this reduction has a practical importance for the efficient processing of the audio signal.

In the experiments of this thesis we will use the time-frequency representation of the audio speech signal. This is in line with the state of the art systems proposed in automatic speech recognition. Purwins et al. (2019) provide a discussion on the benefits of this compact representation for speech analysis tasks. More generally, we will consider the auditory input as a *variable length* sequence of time-frequency vectors. We will use N to denote the length of the audio sequence, where each element of the sequence is a vector that encodes the energy in a small window of the original audio signal at different frequency bins:

$$A = [a_1, a_2, ..., a_N] \tag{2.1}$$

A by-product of speaking is the movement of the speech articulators. This results in a visual signal transmitted as a light wave and captured by a recording device such as a video camera, sampling it typically at 30 images per second. We aim to map each image onto a vector representation, using either an engineered or a learnable transformation, in order to reduce the high dimensionality of the input without a considerable loss of information. Since the common sampling rates of the visual signal and the auditory one or its framed variant differ, the corresponding visual track of the same audio-visual event is another variable length sequence of image *representations* V of a different length M:

$$V = [v_1, v_2, ..., v_M] \tag{2.2}$$

The specific visual transformations used in this work will be discussed in the experimental sections of the following chapters. These techniques will make use of an automated pipeline to segment the object of interest in each image, such as the face or the lips region, on which the visual transformation is applied.

The models in this work aim to capture the statistical relationship in speech between the two input modalities and the symbolic transcription (i.e. text) of the spoken message. We denote the label sequence of length L as:

$$Y = [y_1, y_2, ..., y_L] \tag{2.3}$$

Mathematically, a speech recognition system models the following probability distribution:

$$p(Y|A : \theta) \tag{2.4}$$

where θ is the set of model parameters. In this thesis we are mainly interested in the extension to the auditory and visual modalities of speech considered together, leading to the following modelling problem:

$$p(Y|A, V : \theta) \tag{2.5}$$

Now that we have a fundamental understanding of the speech signal, we will proceed to review the major approaches to the modelling of this signal in computers. In the next section we study the motivation behind neural networks in speech, and introduce the main neural architectures that allow the translation of audio and visual speech sequences into words. These architectures stand at the foundation of the original contributions of this thesis.

2.4 A state-space approach to speech recognition

The digital revolution in the second half of the 20th century enabled scientists to approach various tasks related to speech recognition using a computer. Most of the early attempts were greatly limited by the available computation power, and approached simplified tasks such as isolated word recognition, and/or small vocabularies (Huang et al., 2014; Reddy, 1976). Some of the most notable systems were developed at Carnegie Mellon University and IBM Watson Research Centre (Baker, 1975; Jelinek, 1976). They were based on the assumption that speech can be modelled as a hidden stochastic process, particularly a Markov process. This concept later became widely known as the Hidden Markov Model (HMM), although the term is considered a misnomer since it is not the model that is hidden, but the underlying stochastic process (Huang et al., 2014). The HMM allowed researchers to consider speech as a piecewise stationary signal, and design a discrete set of abstract hidden states, such as phones, that defined the sub-word constituents of every spoken sentence. The state transition process is separated from a state output process that generates speech observations.

An HMM uses Bayes' Rule to transform Equation (2.4) into the equivalent problem of finding:

$$p_{HMM}(Y|A) \propto p(A|Y)p(Y) \tag{2.6}$$

The first part of Equation (2.6), $p(A|Y)$, is known as the *acoustic model*, and denotes the likelihood of observing the sequence of audio vectors A given the sequence of linguistic units Y. The second part, $P(Y)$, is known as the *language model*, and denotes the likelihood of the sequence Y.

Given a word w in Y, the HMM defines its *pronunciation* as a sequence of phone states $\mathbf{q}^w$. The same word may have more than one valid pronunciation. By concatenating the pronunciations of multiple words in a spoken utterance, this forms a composite HMM sequence $Q = [q^{w_1}, q^{w_2}, \cdots, q^{w_L}]$. The transitions between consecutive phones i and j within a word w are modelled with a probability p_{ij}. Likewise, the distribution of the audio observations a_t associated with each phone, denoted with b_j, is usually modelled with a mixture of Gaussian density functions:

$$b_j(a_t) = \sum_m \phi_m \mathcal{N}(a_t; \mu_m^{(j)}, \Sigma_m^{(j)}) \tag{2.7}$$

where m is the density function index, ϕ_m represents the mixture weight, while $\mu^{(j)}$ and $\Sigma^{(j)}$ denote the mean and the covariance matrix associated with a particular phone state s_j from the defined state space.

Consequently, the HMM acoustic model calculates the probability of the audio observations A given the word sequence Y as:

$$p(A|Y) = \sum_Q p(A|Q)p(Q|Y) \tag{2.8}$$

$$p(A|Q) = \sum_\theta p(\theta, A|Q) \tag{2.9}$$

$$p(\theta, A|Q) = p_{\theta_0\theta_1} \prod_{t=1}^{N} b_{\theta_t}(a_t) p_{\theta_t\theta_{t+1}} \tag{2.10}$$

$$p(Q|Y) = \prod_{l=1}^{L} p(q^{w_l}|w_l) \tag{2.11}$$

where $\theta = \theta_0, \theta_1, \cdots, \theta_{N+1}$ is the sequence of state indices associated with Q.

The second component in Equation (2.6), $p(Y)$, is determined by a *language model*. As Gales and Young (2008) explain, $p(Y)$ can be typically represented by a N-gram language model, which assumes an N-th order Markov chain:

$$p(Y) = \prod_{l=1}^{L} p(y_l|y_{l-1}, y_{l-2}, \cdots, y_{l-N+1}) \tag{2.12}$$

Gales and Young (2008) explain that basic HMMs need several structural refinements in order to perform adequately on challenging tasks involving unconstrained and complex vocabularies. These refinements require a considerable amount of domain expertise, and their detailed presentation is out of the scope

of this thesis.

There are three fundamental problems associated with an HMM, according to Rabiner (1989):

1. *(Scoring)* Assessing the likelihood of a particular sequence of features

2. *(Decoding)* Searching for the most likely sequence of speech units given a sequence of speech features

3. *(Training)* Optimising the model parameters to best describe a feature sequence

Three important inherent limitations of HMMs in speech are the assumptions that successive speech observations are conditionally independent, that the state transitions abide the Markov property, and that the observations can be modelled well by mixtures of density functions (Rabiner, 1989). In parallel with the HMM refinements developed over the years, this led to an increase of interest in neural networks as a possible alternative to HMMs for speech modelling. We will discuss the neural network based approaches in Section 2.5.

2.5 Representation learning with Deep Neural Networks

A major breakthrough in the efficient training of neural networks was the inception of the backpropagation algorithm (Le Cun, 1986; Parker, 1985; Rumelhart et al., 1986a,b; Werbos, 1982), which applies the chain rule to calculate the gradients of the loss function with respect to the network parameters. These parameters are then updated using a gradient descent algorithm, which seeks a local minimum of the loss function. The potential of neural networks to overcome the limitations of HMMs in speech was recognised soon after (Lippmann, 1989). Whereas some systems proposed a hybrid framework combining HMMs with multilayer perceptron networks (Bourlard and Morgan, 1993; Renals et al., 1994) or recurrent neural networks (Robinson et al., 1996) for continuous speech, others relied on an entirely neural framework to accomplish the task of phoneme classification (Waibel et al., 1989). In the next section we will introduce the recurrent neural network, which currently represents one of the most widely used approaches to speech processing.

2.5.1 Recurrent processing of sequences

RNN

A recurrent neural network (RNN) is a class of neural networks designed for the modelling of time sequences, such as the speech signal in our case. Whereas HMMs are bound by the Markov property, the rich internal state of the RNN allows it to model longer time dependencies. The RNN maintains the separation between an observation and the internal state that exists in the HMM. However, the observations are now regarded as inputs rather than outputs, and can directly influence the evolution of the internal state. Furthermore, the concept of discrete transitions between states disappears, and the state becomes a purely abstract vector representation defining a continuous trajectory in a high dimensional space. Correspondingly, RNNs no longer require task specific knowledge, and may be used as general purpose sequence processors.

An RNN processes the input sequence step by step, maintaining and updating an internal state with each step. In the most general case, for an input sequence $X = [x_1, x_2, \cdots, x_T]$, the RNN computes an output sequence $Y = [y_1, y_2, \cdots, y_T]$ of the same length T as follows:

$$h_t = \sigma_h(W_i x_t + W_h h_{t-1} + b_h) \tag{2.13}$$

$$y_t = \sigma_o(W_o h_t + b_o) \tag{2.14}$$

where h_t is the internal state of the model at timestep t, W_i, W_h, W_o, b_h, b_o are parameter matrices and vectors, and σ_h, σ_o are non-linear activation functions. The indices i, h, o denote transformations applied to the input, state, and output respectively. By convention, the time index of the first sequence element is 1, and the RNN requires an initialisation of its internal state h_0 prior to the application on the sequence. Functionally, the RNN is mapping each timestep of the input sequence onto an abstract representation, aiming to model the causal relationships between timesteps.

Vanishing gradients and LSTM

Hochreiter and Schmidhuber (1997) show that a fundamental problem associated with training RNNs is gradient vanishing. Due to the potentially large number of timesteps T in a sequence, and the saturating activation functions σ_h, σ_o, the error signal propagating from the last timestep of the sequence all the way to the first one reaches very small values making the learning task difficult. Their proposed remedy is the introduction of multiplicative gating units into the RNN

architecture which allow the network to dynamically scale the flow of information and potentially allow gradients to flow unchanged across multiple timesteps. The most commonly used LSTM variant was proposed by Gers et al. (1999), which adds a forget gate.

$$i_t = \text{sigmoid}(W_i x_t + V_i h_{t-1} + b_i) \tag{2.15}$$

$$f_t = \text{sigmoid}(W_f x_t + V_f h_{t-1} + b_f) \tag{2.16}$$

$$\tilde{c}_t = \tanh(W_c x_t + V_c h_{t-1} + b_c) \tag{2.17}$$

$$o_t = \text{sigmoid}(W_o x_t + V_o h_{t-1} + b_o) \tag{2.18}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.19}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.20}$$

where $\text{sigmoid}(x) = 1/(1 + \exp(-x))$. Here, W_i, W_f, W_c, W_o are parameter matrices corresponding to the input, forget, cell, and output gates respectively (i_t, f_t, c_t, o_t). The i_t, f_t, o_t gates are vector representations scaled between 0 and 1, whereas the candidate cell gate $\tilde{c}_t$ is a vector of real values between -1 and 1.

At every timestep t, the LSTM cell chooses how much to retain from the previous cell state c_{t-1} and how much information to incorporate from the candidate cell state $\tilde{c}_t$ at the current timestep. Similar to the RNN, the LSTM state needs to be initialised, however the initial cell state is now represented by the (c_0, h_0) pair.

Since the RNN and LSTM networks apply the same computation block to every input timestep, they can theoretically process sequences or arbitrary lengths. This computation block is commonly referred to as *cell* in machine learning frameworks (e.g. *LSTM Cell*, *RNN Cell*). Both RNN and LSTM networks can learn a mapping between input and output sequences of identical lengths. In speech, the RNN or LSTM outputs were initially interpreted as a probability distribution of the phone classes estimated for every audio frame (Graves and Schmidhuber, 2005; Robinson, 1994). This method poses the disadvantage of requiring a dataset annotated at the acoustic unit level. Furthermore, the RNN in Robinson et al. (1996) is not explicitly optimised for word recognition, but requires a decoding algorithm with a lexicon to efficiently search through the estimated phone likelihoods.

Connectionist Temporal Classification

Graves et al. (2006) approach for the first time the task of predicting a sequence of labels from an *unsegmented* sequence of inputs of a different length, termed *sequence transduction*, using an RNN. Their new training method for RNNs,

Connectionist Temporal Classification (CTC), removes the need to pre-define an alignment between the inputs and the target outputs. Specifically, CTC only requires pairs of speech inputs and their corresponding word-level transcription for training, without intermediate phonetic annotations.

A CTC model first uses a recurrent neural network to make frame-wise predictions over a finite alphabet augmented with a special *blank* token, which denotes the absence of any other token for a particular frame. We can denote these predictions as follows:

$$h_t = RNN(A_{1:t}) \tag{2.21}$$

$$p_t(h_t) = \text{softmax}\left(W_\eta h_t\right) \tag{2.22}$$

$$q_t = \arg\max p_t(h_t) \tag{2.23}$$

where the output linear layer W_η projects the RNN output h to the dimension of the task alphabet plus one, i.e. $p_t \in \mathbb{R}^{\eta+1}$. The sequence of frame-level label predictions $Q = [q_1, q_2, \cdots, q_N]$ is referred to as *alignment*, since it helps establish a correspondence between each label symbol and a unique segment in the input. More than one valid alignment is possible between Q and Y after eliminating from Q all the blank symbols and the consecutive repetitions of the same label. For this reason, CTC defines an objective function that predicts the conditional probability of the output sequence Y by marginalising over the set of valid alignments $Q^* \in Q$:

$$p_{CTC}(Y|A) = \sum_{Q \in Q^*} p(Q|h) \tag{2.24}$$

$$= \sum_{Q \in Q^*} \prod_{t=1}^{N} p_t(h_t) \tag{2.25}$$

The objective function of a CTC model can be expressed as minimising the log probabilities of the correct alignments:

$$CTC\ Loss = -\log(p_{CTC}(Y|A)) \tag{2.26}$$

To make the computation tractable, Graves et al. (2006) use a dynamic programming algorithm for Equations (2.25) and (2.26).

One distinctive feature of CTC is the lack of explicit modelling of the inter-label dependencies. This can sometimes result in the under-exploitation of the patterns in the text, as we will see in the following sections. On the other hand, this creates the opportunity to specialise a language model on a domain without

requiring a transcribed speech dataset.

A CTC model can be viewed as a special case of a linear/left-right HMM that includes an optional blank state between any two consecutive non-blank nodes in the graph. Moreover, because it models $p(Y|A)$ directly, this makes CTC a discriminative model, in contrast with the generative approach of the HMM. A more detailed analysis of the similarities and differences between CTC and HMMs can be found in Zeyer et al. (2017). In more recent developments for CTC-based speech recognition, such as Amodei et al. (2016); Kim et al. (2017); Kriman et al. (2020), the state space of a CTC model is typically made of the letters in the alphabet, punctuation tokens, and blank, instead of the traditional context dependent or independent phones. This choice removes the necessity to design an appropriate state space and manage a pronunciation dictionary, unlike in HMMs. Concurrently, Prabhavalkar et al. (2017); Pundak and Sainath (2016); Zeyer et al. (2018) find that a well tuned conventional HMM still outperforms CTC on several speech recognition tasks. Nonetheless, these examples suggest the potential of CTC to provide an alternative approach to speech modelling, comparable in performance with the more established HMM one.

Recurrent Neural Network Transducer

CTC is improved in Graves (2012) to additionally model the inter-dependencies between the output labels with the *RNN Transducer* (RNN-T) architecture. As later shown in (Battenberg et al., 2017; Prabhavalkar et al., 2017), modelling the conditional dependence between predictions at successive timesteps is essential to improving the speech recognition accuracy when no external language model is used.

The additional inter-label conditioning takes the form of another RNN, termed *the prediction network*, receiving as inputs the history of predicted labels. The acoustic encoder is termed *the transcription network*, and has the same role as in the CTC model. We will denote the internal states of the prediction and transcription networks with g and h respectively:

$$g_{l_t} = RNN(Y_{1:l_t}) \tag{2.27}$$

$$h_t = RNN(A_{1:t}) \tag{2.28}$$

where l_t represents an index in the output sequence Y that is synchronised with the audio input timestep t. This synchronisation is embedded into the RNN-T architecture as a hard monotonicity constraint, since the prediction network receives the next label token y_{l_t} only when the predicted state at timestep t is dif-

ferent from a *blank* or an identical label at the previous step $t-1$. In other terms, $g_{l_t} = RNN(y_{<l_t})$. The frame-level prediction made by the RNN-T becomes:

$$p_t(h_t, g_{l_t}) = \text{softmax}\, f(h_t, g_{l_t}) \tag{2.29}$$

$$q_t = \arg\max p_t(h_t, g_{l_t}) \tag{2.30}$$

where f can be any parametric or non-parametric function that combines the hidden states of the transcription and the prediction networks h_t and g_{l_t}. Finally, the conditional probability of the output sequence in an RNN-T can be expressed as:

$$p_{RNN-T}(Y|A) = \sum_{Q \in Q^*} p(Q|h) \tag{2.31}$$

$$= \sum_{Q \in Q^*} \prod_{t=1}^{T} p_t(h_t, g_{l_t}) \tag{2.32}$$

The additional conditioning on g_{l_t} in Equation (2.32) can be seen as an intrinsic language model of the RNN-T exploiting the text patterns in the speech dataset jointly with the patterns in the audio signal. An external language model can still be incorporated in order to rescore the predictions of the RNN-T, as exemplified by Battenberg et al. (2017). It is important to note that the grapheme-based RNN-T achieves a different internal modelling of the patterns in text than conventional HMMs, which typically rely on word units. This chapter aimed to unify the notations used to present all models, meaning that Y is a generic target sequence that can specialise to the particularities of each model.

Similar to the CTC model, the computation of $p_{RNN-T}(Y|A)$ is intractable with a naive algorithm. Although Graves (2012) defines an efficient forward-backward algorithm for RNN-T, in practice the implementation often needs to be specialised in a low-level programming language (e.g. making use of CUDA warp-level primitives, or domain specific compilers as in (Bagby et al., 2018)) to avoid a computational bottleneck in the calculation of the loss function. Furthermore, Li et al. (2019b) report a high memory usage of the RNN-T compared to CTC and the encoder-decoder model presented in the next section. For some of the applications targeted by this thesis, such as real-time audio-visual speech recognition, which could have a great value in resource constrained environments, it would be more desirable to investigate an alternative set of tools for sequence transduction with a lower computational footprint.

Sequence to Sequence neural networks

An alternative approach to address the structural limitation of RNNs, different from CTC, is the Encoder-Decoder, or the sequence to sequence architecture (seq2seq) proposed by Cho et al. (2014); Forcada and Ñeco (1997); Kalchbrenner and Blunsom (2013); Sutskever et al. (2014). This model consists of two distinct recurrent networks, one for the input sequence and one for the targets respectively. The main idea consists in mapping the entire input sequence onto a fixed-length vector (e.g. the cell state at the last timestep of the input sequence) using a recurrent network termed the *Encoder*, and using the fixed-length representation to initialise the cell state of the second recurrent network, the *Decoder*, that models the output sequence. Formally, for an LSTM-based sequence to sequence model:

$$c_i^{Enc}, h_i^{Enc} = LSTM^{Enc}(a_i) \quad \text{for } i = 1 \cdots T \tag{2.33}$$

$$[c_0^{Dec}, h_0^{Dec}] = [c_T^{Enc}, h_T^{Enc}] \tag{2.34}$$

$$c_j^{Dec}, h_j^{Dec} = LSTM^{Dec}(y_j) \quad \text{for } j = 1 \cdots L \tag{2.35}$$

The initial state of the Encoder network, $[c_0^{Enc}, h_0^{Enc}]$, is commonly set as vectors of zeros, or small random values.

Equation (2.34) which sets the initial state of the decoder network as the final state of the encoder network, represents the connection point between the two networks, and it is generally assumed that the summary of the input sequence is unfolded in the second network. Bahdanau et al. (2015) conjectured that summarising an entire sequence into a fixed-length vector represents an informational bottleneck. To address it, they introduce an attention mechanism in the decoder network that learns to extract a context vector c_j by soft aligning the decoder state h_j^{Dec} at every output timestep j with every encoder state h_i^{Enc}:

$$\alpha_{ji} = \text{softmax}_j(h_j^{Dec^T} \cdot h_i^{Enc}) \quad \text{where } \text{softmax}_j(X) = \frac{\exp(x_j)}{\sum_i \exp(x_i)} \tag{2.36}$$

$$c_j = \sum_{i=1}^{T} \alpha_{ji} \cdot h_i^{Enc} \tag{2.37}$$

Intuitively, α_{ji} represents the normalised relative importance of frame i in the encoder network for frame j in the decoder network. The softmax operation en-

sures that $\sum_i \alpha_{ji} = 1$. Since the decoder is now able to retrieve the most relevant representations from the encoder side, the fixed length representation is no longer a computational bottleneck. Furthermore, Bahdanau et al. (2015) explain that the encoder, relieved from the burden of having to compress an entire sequence into a single fixed length representation, can adopt new encoding strategies. In speech processing, we may imagine that one possible outcome of attention-based decoding is the learning of frame-based representations that are more related to the acoustic content at each timestep. Whereas a system without attention can only rely on a single representation for encoding and inevitably has to aggregate information from multiple steps, in an attention network the full sentence is available, and it would be more advantageous to maintain the high granularity in the input signal.

In contrast with the state-space models from Section 2.4, a seq2seq neural network models $p(Y|A)$ directly, without breaking it into an acoustic and language model. As a result, we can consider that the decoder part of a seq2seq network is implicitly learning a language model. On a large speech dataset of approximately 12,500 hours of recordings, Chiu et al. (2018) show that there are diminishing returns for incorporating a separate language model to rescore the predictions made by the decoder. This suggests that, given a relatively large amount of annotated speech data, it may only be advantageous for seq2seq architectures to decode with a specialised external language model for known specific domains.

CTC or Sequence to Sequence modelling ?

Prabhavalkar et al. (2017) directly compare the RNN Transducer and the seq2seq model with attention on a large scale speech recognition task using approximately 12,500 hours of speech recordings. They find the two models to be comparable in performance. The seq2seq model allows a considerably simpler training procedure based entirely on backpropagation using gradient descent, whereas the RNN Transducer requires a more complex dynamic programming algorithm (Graves, 2012). For this reason, we will use the seq2seq model with attention as the starting point for the work in this thesis.

2.5.2 Attention mechanisms

An integral part of many modern deep learning architectures is the attention mechanism. Conceptually, an attention mechanism establishes a relationship between two sequences or modalities by computing similarity scores between their constituent elements. The similarity scores are typically normalised to be

non-negative and sum up to 1. As a result, each element of the second sequence will have a corresponding *context vector* computed as a weighted sum of the elements from the first sequence, where the weights are given by the similarity scores. This allows the contextualisation of the elements of the second sequence and enables more informed predictions derived from the fusion of the two sources of information, or modalities.

Attention mechanisms can be classified by the type of the similarity function used to score the compatibility between the representations of different sequences, and by the range of the search space under consideration when computing the alignment scores. We will now discuss this classification below.

Attention types by similarity function

Luong et al. (2015) considered three possibilities for the scoring function, namely dot (Equation (2.38)), general (Equation (2.39)), and concatenated (Equation (2.40)).

$$score(h_t, h_s) = h_t^T h_s \tag{2.38}$$

$$score(h_t, h_s) = h_t^T W_a h_s \tag{2.39}$$

$$score(h_t, h_s) = v_a^T tanh(W_a[h_t; h_s]) \tag{2.40}$$

The common aspect in the three equations above is that the scoring function depends on both the target representation h_t and the source representation h_s. The similarity score is calculated as a function of the *content* of the two representations. Consequently, they are referred to as ***content-based*** scoring functions. Bahdanau et al. (2015) present one of the earliest forms of attention in deep learning architectures, which corresponds to the concatenated variant in Equation (2.40).

Furthermore, instead of calculating a compatibility score between two representations, Luong et al. (2015) also experiment with a scoring function that predicts the mixing weights solely from the target representation h_t, as in Equation (2.41).

$$score(h_t, h_s) = score(h_t) = softmax(W_a h_t) \tag{2.41}$$

This bears the name of ***location-based*** scoring function. Chorowski et al. (2015) present a modified variant of location-aware attention that includes the alignment

scores estimated for the previous timestep:

$$\alpha_t = score(h_t, \alpha_{t-1}) \tag{2.42}$$

However, they argue that location-based attention is not suitable for speech recognition, since it would have to perform the difficult task of predicting the acoustic duration of a target symbol, particularly a phoneme in their case, from only the current representation h_t in the target sequence. For this reason, they propose a third type of scoring function termed ***hybrid***:

$$\alpha_t = score(h_t, h_s, \alpha_{t-1}) \tag{2.43}$$

Global and Local Attention

A **global attention** mechanism, such as the one of Bahdanau et al. (2015), computes contextualised representations for every target state h_t by making use of the entire encoded sequence $h_s, s \in [1, N]$. Two disadvantages of global attention are the quadratic time complexity, and the prohibition of online decoding. Output tokens can only be predicted when the entire sequence of input representations is available. This may be too restrictive for speech recognition.

For such tasks, we can incorporate any available prior knowledge into the structure of the attention mechanism by limiting the attention window to a small subset of the input sequence. This concept is known as **local attention**. Luong et al. (2015) propose one form of local attention that first predicts the centre of an attention window using the target state h_t:

$$p_t = N \cdot sigmoid(v_p^T \tanh(W_p h_t)) \tag{2.44}$$

This allows them to limit the attention range to $[p_t - D; p_t + D]$, where the window length $2D + 1$ is an additional hyper-parameter. Furthermore, they propose to modulate the attention window with a Gaussian distribution centred on p_t with a standard deviation $\sigma = D/2$ in order to favour locality.

$$\alpha_t = score(h_t, \bar{h_s})\ exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right) \tag{2.45}$$

The advantage of local attention over global attention is the possibility to perform the inference step in real time. Each decoding timestep would provide information regarding the necessary encoded information that needs to be made available.

Narrowing the attention span alone still poses the shortcoming of having to store the entire encoded sequence. This is because p_t in Equation (2.44) can take any real value in the range $[0; N]$ at any decoding timestep. By setting additional monotonicity constraints on the shift applied to the attention window from one timestep to another, where appropriate, we would have the possibility to discard the previously visited elements in the attention memory. This is a property naturally suited to the pairs of audio signals and their transcribed text sequences.

Monotonic Attention

Monotonic attention can be thought of as a sliding attention window process over the encoder representations. The goal of monotonic attention is to make the prediction of the output tokens conditionally independent of irrelevant audio frames. Once the attention window no longer covers a prefix of the encoder output, it can be flushed from the memory buffer.

Two main challenges for speech signals consist in finding the appropriate window length and shift. This is owed to the structure of the speech signal, where the modelled units (e.g. graphemes, words) have a variable duration. For maximum efficiency, one would need to obtain an oracle segmentation of a speech stream into linguistic units, such that the contextualised representation of an unit is only computed from the relevant speech frames (e.g. one word plus some context).

Luong et al. (2015) considered a simple approximation for the machine translation task, setting $p_t = t$. Given the variable length duration of the speech units and the complexity of the sound to spelling mappings in speech, this form of monotonicity would not be suitable for speech recognition. Raffel et al. (2017) experiment with an online variant of hard monotonic attention, where the context vector is made of a single encoded frame. This idea is further refined in Chiu and Raffel (2018) to allow attention windows longer than one time frame. However, one limitation of the approach of Chiu and Raffel (2018) is that the window size is fixed. This would not be an optimal design for speech recognition, where the linguistic units have a variable duration. Nevertheless, the approach of Chiu and Raffel (2018) provides one alternative to the problem of decoding speech in real time with a sequence to sequence architecture.

2.5.3 Regularisation methods

In order to improve the generalisation of neural networks trained with limited data, many regularisation methods have been developed. As this thesis makes use of common regularisation methods in order to compensate for the relatively small amount of training data used in the experiments, we will briefly describe them in the sections below.

Dropout

Srivastava et al. (2014) introduce the idea of randomly setting to zero a fraction of the internal activations in the neural network during training. The authors claim that dropout prevents the co-adaptation of neural units by lowering the reliability of any particular unit as a feature detector. Details regarding the neural layers we will apply dropout on, as well as the dropout probability, will be provided where necessary throughout this thesis.

Activity normalisation

Ioffe and Szegedy (2015) identify a problem in deep neural networks where the distribution of the inputs to each layer changes in training, phenomenon they call *internal covariate shift*. To address it, they propose to normalise the inputs to each layer by relying on the statistics of the minibatch of training examples, technique called *batch normalisation*. The method tracks the mean the standard deviation of the activations in the network across the batch dimension, and uses those statistics to obtain a zero mean, unit variance vector representation:

$$\widehat{z^k} = \frac{z^k - \mu(z^k)}{\sigma(z^k)} \tag{2.46}$$

where $\mu(z^k)$ and $\sigma(z^k)$ are the running averages of the batch mean and standard deviation of a layer representation z^k. The batch normalisation operation also introduces a learnable affine transformation using the scale and bias parameter vectors γ^k and β^k respectively, leading to:

$$BN(z) = \gamma^k \widehat{z^k} + \beta^k \tag{2.47}$$

Ba et al. (2016) propose an alternative way of normalising z^k by computing the layer / channel statistics instead of the batch ones. This strategy offers the advantage of being applicable to online learning tasks and in distributed training settings. Later in Section 2.5.4 we will refer to this transformation as $LN(z)$.

Weight norm penalty

One of the earliest forms of regularisation for ill-posed problems is referred to as Tikhonov regularisation (Tikhonov and Arsenin, 1977). In the case of neural networks, this regularisation method takes the form of a penalty term added to the objective function, which consists of the sum of all norms of the weight matrices in the neural network. This penalty term favours the learning of solutions with small weights. In deep learning, the most commonly used matrix norms are L1 and L2.

Scheduled sampling

Bengio et al. (2015) identify a discrepancy between the training and evaluation of sequence to sequence neural networks owed to the feeding of ground-truth tokens to the inputs of the decoder during training, whereas at inference they are replaced with the previous prediction. The authors explain that feeding wrong predictions at inference can cause an accumulation of errors at the sequence level. To bridge this gap, they propose to train the network under more similar conditions with the evaluation stage by allowing the trainable network to use its past predictions with a small probability. Since always feeding the previous prediction in training could lead to slow convergence, Bengio et al. (2015) propose to gradually increase the probability of sampling from the previous output, as opposed to the ground truth. This approach is termed *scheduled sampling*.

Gradient clipping

The gradient descent algorithm updates the parameters of a neural network using the gradients obtained through backpropagation, scaled by the learning rate. Pascanu et al. (2013) investigate the problem of exploding gradients in RNNs, and, in order to mitigate it, they propose a gradient norm clipping strategy. In other words, the algorithm proposed by Pascanu et al. (2013) re-scales all the gradients g above a threshold λ by $\frac{\lambda}{||g||}$, where $||g||$ denotes the gradient norm. All the experiments in this thesis make use of gradient clipping, and details will be provided where necessary.

Data augmentation

A common regularisation approach is the application of transformations on the raw input signals. This can have the effect of simulating additional training data, and explicitly optimising the neural network to become invariant to such transforms. Common transformations for images include translations, rotations, re-

scaling, or mirroring.

In speech, a recent method proposed by Park et al. (2019) applies augmentations on the audio spectrogram, consisting of time and frequency masking, and time warping. The method has quickly gained popularity in 2020. Most of the experiments in this thesis were already consolidated by then, and it was preferable to maintain a coherency of our training strategy through different chapters. As a result, the speech augmentations used in this thesis consist of additive noise applied in the time domain before computing the spectrogram.

Multi-task learning

Multi-task learning is the process of training one system on multiple sub-problems, keeping a shared internal representation across all of them. According to the review of Caruana (1997), multi-task learning helps improve generalisation by placing more constraints on those parameters shared across tasks. One example of multi-task learning in AVSR is the work of Tao and Busso (2021), which combines the tasks of voice activity detection and speech recognition for training an audio-visual system.

Pre-training may be seen as a special case of multi-task learning, where one task is used to bootstrap a fraction of the network parameters. For example, Petridis et al. (2018b) first train the visual front-end of their AVSR system on a separate word classification task, and keep those parameters frozen when later training the rest of the parameters for sequence prediction. However, given that the tasks are learnt sequentially and not jointly, there may be diminishing generalisation returns with pre-training, outside the benefit of providing a better initialisation than a random one.

2.5.4 The Transformer architecture

An inherent computation bottleneck in sequence to sequence or encoder-decoder architectures powered by RNN back-ends is the impossibility to parallelise the processing of both input and output sequences on multiple threads along the time axis. This often leads to an under-utilisation of the hardware accelerator in many practical settings. To overcome this limitation, Vaswani et al. (2017) propose to remove the recurrent connections between successive timesteps within a network layer and rely entirely on the principle of attention mechanisms to model long term dependencies, with the *Transformer* architecture. At the core of the Transformer is the self-attention network, defined on a generic sequence of vec-

tors $X = [x_1, x_2, \cdots, x_T]$ as follows:

$$query_i = W_Q x_i \tag{2.48}$$

$$key_i = W_K x_i \tag{2.49}$$

$$value_i = W_V x_i \tag{2.50}$$

$$\alpha_{ji} = \text{softmax}(\frac{query_j key_i^T}{\sqrt{d_k}}) \tag{2.51}$$

$$c_j = \sum_{i=1}^{T} \alpha_{ji} \cdot value_i \tag{2.52}$$

Here, W_Q, W_K, W_V are learnable weight matrices used to produce internal representations termed *queries*, *keys*, and *values* respectively, which allow a separation between the computation of the compatibility scores α and the computation of the output vector c_j. d_k is a scalar term representing the dimension of the key_j. Vaswani et al. found this scalar beneficial for the normalisation of the dot product ($query_j key_i^T$) for increasing model sizes. Functionally, we will use $SelfAttention(X)$ to denote the sequence of the operations in Equations (2.48)-(2.52) applied to a generic input sequence X.

There are two main blocks defined by the Transformer architecture, namely an Encoder and a Decoder. The Encoder computes a transformation of the input sequence X by applying the following operations:

$$X' = X + PE(X) \tag{2.53}$$

$$Z = X' + SelfAttention(LN(X')) \tag{2.54}$$

$$Z' = Z + FFN(LN(Z)) \tag{2.55}$$

PE stands for the positional encodings defined by Vaswani et al. (2017), *LN* is the layer normalisation operation described in Section 2.5.3, while *FFN* is the notation for a feed-forward network block consisting of two position-wise linear transformations with a ReLU activation between them, i.e.:

$$FFN(x_i) = W_2 ReLU(W_1 x_i + b_1) + b_2 \tag{2.56}$$

with W_1, W_2, b_1, b_2 representing two sets of parameter matrices and vectors respectively, projecting the dimensions of the internal representations to a priori chosen model sizes d_{model} and d_{FF}, making $W_1 \in \mathbb{R}^{d_{FF} x\ d_{model}}$, $b_1 \in \mathbb{R}^{d_{FF}}$, $W_2 \in \mathbb{R}^{d_{model} x\ d_{FF}}$, $b_2 \in \mathbb{R}^{d_{model}}$, while x_i and $FFN(x_i) \in \mathbb{R}^{d_{model}}$.

We denote the Equations (2.54)-(2.55) above as EncodeLayer(X'). We can com-

pose this transformation multiple times to create a stack of encoder layers, as follows:

$$Z'' = \underbrace{\mathit{EncodeLayer} \circ \mathit{EncodeLayer} \circ \cdots \circ \mathit{EncodeLayer}}_{\text{N times}} (X') \tag{2.57}$$

Finally, the Transformer encoded output sequence is obtained after the application of an additional layer normalisation operation:

$$o_X = LN(Z'') \tag{2.58}$$

The Decoder stack of the Transformer is defined in a similar fashion to the Encoder one, but introduces two modifications. First, between the *SelfAttention* and *FFN* blocks (i.e. between Z in Equation (2.54) and Z' in (2.55)) it inserts an *Attention* layer attending to the encoder outputs o_X, as in the conventional sequence to sequence model with attention. The only difference between *SelfAttention* and *Attention* is that the latter uses the encoder outputs o_X as keys and values, while the queries come from the decoder's representations. Second, the decoder is restricted to process the target sequence casually, which is a requirement of the inference step. Such restriction is typically implemented by masking all the future positions in the decoder's *SelfAttention* blocks during training.

The encoded sequence o_X is functionally equivalent to the output of the recurrent encoders discussed in Section 2.5.1. One main advantage of using self-attention over RNNs is the possbility to compute all the operations from Equations (2.48)-(2.52) in a single step. Vaswani et al. (2017) show that the computational complexity of self-attention is lower than of a RNN when the sequence length T is smaller than the model size d. This is often the case when modelling text or short pre-segmented speech utterances. We note, however, that the decoder of a Transformer network still obeys the property of causality, which limits the inference performance of the model where it acts akin to a RNN.

Another advantage of the Transformer block is the constant and unitary path length between any two timesteps in the sequence. This contrasts with the LSTM architecture where the internal state is updated k times between timesteps i and $i + k$, and only a proper gating behaviour enables the retention of gradient information across longer dependency paths. However, this aspect may also suggest a sub-optimality of the Transformer network in speech, as there are typically no acoustic relationships between sounds beyond several words. Indeed, the Transformer network was originally proposed for text modelling where such long range dependencies are necessary to model aspects such as morphology or syntax.

Therefore, making use of speech domain knowledge has the potential to further increase the model efficiency.

RNNs rely on the principle that sequences should be processed sequentially, or eagerly. Instead, Transformers imply that the raw input signal is first stored in a temporary buffer, and then a buffer-level representation is computed. Determining the optimal way to process a sequence remains an open problem. This thesis will explore both the RNN and the Transformer in order to get a better understanding of their strengths and weaknesses.

2.5.5 Convolutional Neural Networks

It is usually accepted that incorporating prior knowledge into a learning system is an effective strategy for improving generalisation (LeCun, 1989). In the case of neural networks with image inputs, one way to apply this principle is by enforcing a local connectivity pattern between successive layers of neurons, and sharing weights acting as shift-invariant feature detectors. This structure is commonly known as a Convolutional Neural Network (CNN). The first example of a CNN architecture was the neocognitron proposed by Fukushima (1980) to simulate a subset of the human vision system. Later, LeCun et al. (1989) presented a CNN architecture trained with the backpropagation algorithm and used to recognise hand-written zip codes. He et al. (2016a) propose ResNet, a deep CNN with shortcut connections shown to be easier to optimise with increased network depth than plain CNNs. In this thesis we use the standard ResNet architecture to extract visual speech representations from image sequences of a speaker's mouth region. In the next section we will analyse the application of the sequential and convolutional neural networks for the modelling of auditory, visual, and audio-visual speech.

2.6 Audio-Visual speech modelling

Multiple perceptual studies have examined the relative visual contribution to oral speech comprehension as a function of the speech signal to noise ratio (Erber, 1969, 1975; O'Neill, 1954; Sumby and Pollack, 1954). These studies find that the combined audio-visual performance is superior to the auditory one alone. In addition, the contribution of the visual cues is found to increase as the listening conditions become more adverse. Replicating these two findings with automatic speech recognisers has been the goal of most AVSR systems proposed in the literature.

There are two fundamental problems associated with the design of an audio-visual automatic speech recogniser. The first one is the extraction of good visual representations. Many studies analysed the visual modality in isolation for the decoding of speech, a task known as lipreading. These systems will be discussed in Section 2.6.1, paying attention to their feature extraction principles. The second problem is the optimal integration of the audio and visual speech modalities to produce a multimodal prediction. We will discuss in Section 2.6.2 the recently proposed multimodal systems in the space of neural networks, and will further break down the integration task into *alignment*, *fusion*, and *co-learning*.

2.6.1 Modelling visual speech

Rosenblum et al. (2007) show in their article "Lip-read me now, hear me better later" that lipreading familiarity with a speaker increases the accuracy of acoustically understanding the speech from the same speaker than from an unfamiliar one. This suggests that visual and auditory speech, beyond some level, are virtually inseparable. Macleod and Summerfield (1987) suggest a possible connection with a simplified perception model. In this model, while the auditory and visual modalities have separate auditory and visual analysis processes, they both share a linguistic process downstream. The model of Macleod and Summerfield (1987) describes the task of lipreading as a succession of visual and linguistic analysis. Likewise, silent speechreading and audio-visual speech perception are likely to share a common process of visual analysis. This motivates a separate investigation of lipreading as the means of understanding audio-visual speech.

Hennecke et al. (1996) present an overview of the earlier visual speech recognition systems (1984-1996). Most of these studies were performed in controlled laboratory conditions (e.g. fixed speaker location) and relied on engineered image processing algorithms to detect the mouth region and extract visual representations. Fernandez-Lopez and Sukno (2018) review the evolution of lipreading systems proposed between 2007 and 2018, observing that until 2016 the HMM was the most prevalent classifier, while neural networks have started to replace all the blocks of the traditional systems in the recent years. Therefore recent lipreading systems aim to substitute the expert knowledge with abstract representations learnt directly from data with generic neural models.

Due to the limited public availability of audio-visual datasets of continuous speech, later discussed in Section 2.8, the newer systems powered by neural networks still approached simplified tasks of isolated speech units or constrained vocabu-

laries. Petridis et al. (2017a); Petridis and Pantic (2016); Petridis et al. (2017b) use a deep autoencoder and a LSTM network to classify the letters of the English alphabet, the digits from 0 to 9, or 10 common phrases. Assael et al. (2016) use a CTC-based recurrent model that decodes visual speech at the character level on the GRID corpus (Cooke et al., 2006), an audio-visual speech dataset with a vocabulary of only 51 words. Their best model achieves a Character Error Rate of only 6.4%, which increases to 6.7% when no language model is used. Wand et al. (2016) use the word-level alignments available in the GRID corpus and train a LSTM-based lipreading network to predict the correct word from the possible 51. Chung and Zisserman (2017) introduce a large visual speech dataset containing 500 different words, and classify one second long utterances using a CNN. Again, this network relies on the pre-segmentation of the training dataset, and does not scale well with the increase of the vocabulary. On the same 500-word dataset, Stafylakis and Tzimiropoulos (2017) proposed an improved neural network achieving 83% word classification accuracy. With the exception of the LipNet model of Assael et al. (2016), none of these examples can be used for large vocabulary sentence-level lipreading, leaving on the table an unused potential to build a strong visual language model that can solve ambiguities specific to visual speech.

The systems of Assael et al. (2016) and Chung et al. (2017) can be considered the first neural architectures capable to decode continuous visual speech. The underlying methods in both cases, CTC and the Seq2seq model, had already been introduced in 2006 and 2013 respectively, and are directly applicable to audio-only speech recognition as we saw in Section 2.5. Therefore, in addition to collecting suitable data, another prerequisite for large vocabulary sentence-level lipreading was the design of a suitable neural network architecture that enabled the sequence transduction task. Our work in Chapter 3 uses the seq2seq neural model to investigate the lipreading of large vocabulary continuous speech.

2.6.2 Audio-Visual speech recognition

Although the previously mentioned experiments in cognitive psychology demonstrate the contribution of the visual modality to speech perception, they do not provide an explanation of *how* the human brain integrates the auditory and visual modalities. Summerfield (1987) started from these early studies and placed them into a broader context, proposing several plausible cognitive theories and models of integration. Robert-Ribes et al. (1996) and Schwartz et al. (1998) continue from these theories and reduce them to four basic architectures, namely direct identification (DI), separate identification (SI), dominant modality recoding

(DR), and motor/amodal recoding (MR). The purpose of the computational models, as they argue, is to provide a set of reasonable constraints that enable the efficient learning of the task from data. Similarly, Stork and Hennecke (1996) distinguish between early, intermediate, and late stages of integration. Stork and Hennecke (1996) also make the point that early integration is the more general model, since it has the possibility to virtually integrate the auditory and visual features at any stage as deemed necessary. Based on the empirical evidence of the last decade in machine learning, such a view may only be applicable to an ideal learning system whose training algorithm leads to a global optimum of the parameters. In practice, searching for well informed inductive biases for the network architecture has been an effective strategy to advance the knowledge in multimodal speech processing.

Currently, there is no clear consensus on the optimal architecture for audio-visual speech integration. Adjoudani and Benoît (1995) compare empirically the DI and SI models for small vocabulary recognition of 54 non-sense words, and find that SI better takes advantage of the visual modality. Furthermore, they report that DI does not meet the basic requirement of improving or at least matching the recognition accuracy from the audio modality alone under all noise conditions. Massaro and Stork (1998) are in favour of SI applied at the syllable level, supporting a Bayesian rule of integration. They argue that very little "crosstalk" between the audio and visual modalities may take place in the brain. In contrast to these findings are the views of Summerfield (1987), who has supported the theory that an early integration of audio-visual cues should take place before any phonetic classification. Rosenblum (2008) has also shared the view of Summerfield (1987), but acknowledges that the specific form of integration remains unclear. The DR and MR models of Robert-Ribes et al. (1996) provide two alternative explanations: either the visual modality is recoded to a representation specific to the dominant audio modality (DR), or both modalities are projected to amodal representations (MR). Summerfield (1987) argues that representing the visual modality in an auditory space by estimating the filter function of the vocal tract is arbitrary and unsupported by experimental evidence. In the recent years, this problem seems to have been settled in the computational models thanks to the adoption of artificial neural networks. The great potential of neural networks for modelling audio-visual speech has been recognised very soon after the popularisation of the backpropagation algorithm in the late 1980s. One such early example is the work of Yuhas et al. (1989, 1990), who proposed to map an image of a person's lips onto a spectral envelope that is fused with a noise degraded audio envelope in order to classify vowel sounds more reliably. This model can be

viewed as an instance of DR. Against this model, Summerfield (1987) exemplifies that inverting the modalities of the *ba-ba* and *ga-ga* stimuli from the McGurk experiment does not yield the same effect, thus the spectral envelopes cannot be averaged.

Deep neural networks trained on large amounts of audio-visual speech data no longer require the specification of features of the two modalities, and mainly fall into the MR class. Nevertheless, helping training algorithms for neural networks converge to good optimums by imposing a set of architectural constraints remains an open problem in audio-visual speech.

Beyond the stage of integration, a key question concerns the representation of the two modalities right before fusion. The model proposed in this thesis, AV Align, provides an answer to fill the knowledge gap. Common with other related models, both modalities in AV Align are represented by the activations from neural layers. The novelty of AV Align consists in re-formatting the visual modality at the point of integration with the audio modality. More precisely, AV Align proposes to obtain a visual representation that is soft aligned with each of the audio representations. In essence, AV Align transfers the underlying principle of attention used between the decoder and the encoder of a seq2seq network, and re-applies it between the two modality encoders. The next section aims to structure the challenges addressed by AV Align and place them into a broader context that allows a more detailed comparison with the related work.

2.6.3 Sub-problems of multimodal integration in AVSR

The recent survey of Baltrušaitis et al. (2019) proposes a new taxonomy for the challenges faced in multimodal machine learning. It goes beyond the traditional pipelines presented in Potamianos et al. (2017, 2003), which were mostly limited to feature *extraction* and modality *fusion*, and introduces the *alignment*, *co-learning* and *translation* of modalities, noting that the latter does not represent a challenge in AVSR due to the uniqueness of the label. We consider the related work in AVSR from the perspective of the main challenges identified by Baltrušaitis et al. (2019), as it allows a clearer separation of the proposed techniques.

Representation

Most of the recent work in AVSR uses variations of Convolutional Neural Networks to learn visual representations as a function of data, bypassing the necessity for feature engineering. Purwins et al. (2019) show in their review that

the acoustic modality is still widely represented as a log mel-scale spectrogram, since learning features directly from time domain signals remains a challenging task with minimal yield over the carefully engineered features. Petridis et al. (2018a) find that learning acoustic features directly from the speech waveform outperforms Mel Frequency Cepstral Coefficients (MFCC) in noisy conditions on the simpler task of word classification. The authors' subsequent work in Petridis et al. (2018b) reverts to MFCC when attempting the more challenging continuous speech recognition, without reporting results with the previously introduced end-to-end architecture, possibly hinting at the difficulties of learning from raw audio. One explanation could be the requirement for a large amount of well balanced audio data in order to learn powerful representations efficiently without any constraints.

Alignment

Identifying direct relationships between (sub)elements of the visual and auditory modalities is a primary step towards learning enhanced representations. Even when the camera and microphone are time synchronised, there is still a natural asynchrony between sounds and mouth movements. Schwartz and Savariaux (2014) show that, for chained syllable sequences, the asynchrony fluctuates with the phonetic context, varying between 20 ms audio lead to 70 ms audio lag, and up to 200 ms audio lag (video lead) for more complex speech structures. For example, the largest video lead is typically achieved during the preparatory gestures for plosive sounds such as */p/*. Karpov et al. (2011) also report a variable delay between viseme and phoneme pairs on continuous speech in Russian, noticing a higher visual lead at the start of a sentence, in line with the experiments of Schwartz and Savariaux on isolated syllables.

The reviews of Potamianos et al. (2017, 2003) suggest that modality alignment has generally not received sufficient attention in AVSR. Katsaggelos et al. (2015) analyse the publications in the space of audio-visual fusion up to 2015, taking a closer look at the audio-visual asynchrony aspect. They conclude that only a limited progress has been made, and that asynchrony modelling remains a difficult problem. Hennecke et al. (1996) note that modelling the natural asyncrhony between the audio and visual modalities of speech is not a major requirement in decision fusion systems. The recent work of Petridis et al. (2018b) or Afouras et al. (2018b) relies on the tight synchronicity assumption between the two speech modalities, and enforce an identical sampling rate so that the learnt representations can be conveniently concatenated. An eventual alignment would only happen implicitly, as we will later discuss in Sections 4.3.10 and 4.4.2. Chung

et al. (2017) propose WLAS, an extension of the sequence to sequence model of Bahdanau et al. (2015) for two modalities using two attention mechanisms on the decoder, one for each modality. Their work represents an instance of explicit alignment modeling in neural-based AVSR systems, although this is the indirect result of aligning the target space representation with each input modality. As an alternative to the dual attention decoding design, in this thesis we propose a cross-modal alignment architecture, where the acoustic representations are explicitly aligned with the visual ones in an unsupervised way using a cross-modal attention mechanism. Nevertheless, both approaches allow arbitrary sampling rates for each modality. Subsequent work in Afouras et al. (2018b), representing an update of WLAS of Chung et al. (2017) according to the authors, proposes an alternative architecture based on the Transformer network of Vaswani et al. (2017). This system is trained on the newer LRS2 dataset instead of the original unreleased LRS1 dataset, and there is no published evaluation of the WLAS network on LRS2, making it impossible to draw a direct comparison between the two models proposed by the same group.

Bengio (2002) presented an asynchronous Hidden Markov Model for AVSR that uses an alignment variable between two coherent observation streams to describe the speech process with a single set of states. When plotting this variable, Bengio (2002) observes that the optimal audio-visual alignment differs from the linear one that would characterise a single stream HMM with concatenated modalities, empirically showing that the optimal audio-visual alignment allows small variations around the actual timestamps. Kolossa et al. (2009) use a coupled HMM which models the two asynchronous speech modalities with a different set of states, and only enforce synchronisation at the word boundaries. This architecture also has the potential to generate an alignment between the two streams of audio and visual observations, although this path is not fully explored in their article. Recently, Tao and Busso (2018a) introduced an audio-visual feature alignment scheme using an attention mechanism. One LSTM network transforms handcrafted visual features into higher order representations, and a second LSTM processes the acoustic features while also extracting a visual context vector at every frame as a linear combination of all visual representations. The representations are optimised to minimise the reconstruction error of the acoustic features from the visual features. In contrast, *AV Align* can be seen as the end-to-end alternative to the work of Tao and Busso (2018a) with a different objective function: it learns the visual representations directly from the raw pixels, and jointly optimises them with the character-level decoder, minimising the character error rate. *AV Align* is described in Chapter 4 of this thesis. Regressing

audio features from video features as in Tao and Busso (2018a) enables learning from unlabelled data. We argue, however, that rather than learning cross-modal correlations, the network can simply learn to copy audio features to the output, which is encouraged by the reconstruction loss.

Fusion

Early integration models typically concatenate the auditory and visual representations. A downside of this fusion strategy is the necessity to ensure an identical sampling rate for the two input streams. For example, Makino et al. (2019) operate on video data harvested from YouTube that is heterogeneous in terms of frame rate or coding standard. In the case of variable frame rate, they report tailoring the audio feature extraction strategy to match the video frame rate by shifting the STFT analysis window with variable increments. On the other hand, late integration models have two independent recognisers for each stream, and only need to synchronise their decisions, making the input asynchrony less crucial.

A frequently seen design in neural multimodal fusion involves concatenating time aligned hidden representations from each modality and applying a stack of neural layers to map the representations onto a shared space (Afouras et al., 2018b; Petridis et al., 2018a,b; Stafylakis and Tzimiropoulos, 2017). Instead, the architecture of Chung et al. (2017) concatenates the visual and auditory context vectors extracted by two independent attention mechanisms. Zhou et al. (2019) propose an update to Chung et al. (2017) by incorporating explicit mixing weights for the two context vectors at each timestep. Similarly, using a hybrid DNN-HMM system, Tao and Busso (2018b) demonstrate the benefit of introducing a gating unit to scale audio-visual features before concatenation. Since the system lacks a modality alignment module, this design may implicitly prefer linguistic units which are already time-synchronised, such as plosives, leading to an under-exploitation of cross-modal correlations, though this hypothesis is not fully explored in that article.

Cangea et al. (2020) present a neural network architecture that allows a flow of information between all the intermediate layers of a convolutional neural network processing the video input and a multilayer perceptron network processing auditory features of speech. To achieve this, they propose cross-connection blocks that convert between 1-dimensional audio representations and 2-dimensional image feature maps. Unlike other approaches concatenating flattened representations from modality-specific encoders, the network of Cangea et al. (2020) allows a low-level intervention between modalities from the earliest stage. Their choice

contrasts with the theory of Massaro and Stork (1998) that limited crosstalk should take place between modalities. In their ablation study, Cangea et al. (2020) find the cross-connections to be rarely beneficial to the tasks of classifying isolated digits and letters, and more often they are detrimental. Instead, they found it more advantageous to add a residual connection from the raw input of one modality to the intermediate representations of the other modality. Their study does not address the correlations between different timesteps of the two modalities, and can be viewed as a more advanced fusion strategy for time-synchronous multimodal features.

Co-learning

When labelled data for a particular task is limited, exploiting unlabelled data in a different modality creates the opportunity to learn more robust representations. Ngiam et al. (2011) explore the cross-modal learning opportunities in greater detail. They first demonstrate how to learn better visual speech representations given unlabelled audio-visual data for pre-training. In addition, they demonstrate the benefit of learning shared representations which allow cross-modal reconstructions. More recently, transfer learning has gained popularity in AVSR, although it is not as expressive as the two strategies of Ngiam et al. (2011). Transfer learning typically implies pre-training the acoustic and language models on a much larger dataset (Chung et al., 2017), or learning visual representations on a word classification task (Afouras et al., 2018b; Petridis et al., 2018b) without fine-tuning for AVSR. None of these pre-training strategies exploit the audio-visual data jointly, and only speed up unimodal representation learning (He et al., 2019) rather than transfer knowledge between modalities. Moreover, building a stronger language model on a large external dataset, as in Chung et al. (2017), poses the risk of obscuring the true benefit of the visual modality when comparing AVSR methods. A fair experiment should be designed using an identical amount of text data. Contrary to leveraging additional external training data at increased cost, a different school of thought seeks to overcome the fundamental problem of vanishing and exploding gradients with architectural innovations such as gated RNNs (Gers et al., 1999) and residual connections (He et al., 2016b). We believe this to be a preferable direction for research in AVSR, and our proposed method is a step in this direction.

Discussion

It is worth noting that there is no single architecture that works best for every multimodal task. Alpaydin (2018) gives one example of the integration of a per-

son's face with their fingerprint for biometric authentication. In this case, they argue that a late integration strategy of decisions is more suitable, since there is no low-level correlation the pixels on a face with the pixels on a fingerprint image.

While many of the architectures discussed above provide reasonable integration strategies, they often overlook the temporal limits of audio-visual integration. Typically, the samples of an audio-visual speech corpus represent either words spoken in isolation or short phrases. In the case of isolated words, the unit boundaries are provided explicitly to the system, which is the case with the architectures of Petridis et al. (2018a), Chung and Zisserman (2017), or Stafylakis and Tzimiropoulos (2017) for audio-visual word classification. For multi-word phrases, integration often takes place at pre-defined timesteps dictated by the sampling rate, as it is the case with direct feature fusion. The strategy of Chung et al. (2017) that fuses aligned audio and visual context vectors performs its search over an entire utterance. For such phrase-level modelling tasks, which is the focus of this thesis, one consequence is the conditioning of all the audio and visual representations on each other. This introduces an artificial delay that prohibits the decoding of the first part of the phrase before it has been entirely encoded, and makes the technology less interactive. Moreover, since memory is a finite resource, both training and inference can only be performed on short segments that were extracted from a longer video clip using external heuristics. In the next section we will study the progress in the space of online decoding with a neural network. This will allow us to position another original contribution of this thesis, a strategy that enables the joint segmentation and recognition of speech from auditory and visual cues.

2.7 Online speech recognition

Having a natural conversation with a computer has fascinated humankind for a long time. A key ingredient of this ambition is granting computers the ability to recognise spoken words with minimum latency. This allows a more interactive communication, where the computer is able to interrupt a speaker to acknowledge or ask for clarifications.

The basic meaningful unit of a language is the *word*, and phrases are constructed by joining together multiple words. A relatively modern development in written languages (although greatly varying among different languages) is the introduction of a space delimiter between words, accelerating reading comprehension. Spoken languages do not share this feature, demanding instead the use of an

extensive set of cues for word segmentation within speech perception. To date, the adequate segmentation of a spoken utterance into words using a neural network has not been fully investigated.

While the sequence to sequence networks discussed in Section 2.5.1 are currently the best performing models at recognising speech from pre-segmented utterances (Chiu et al., 2018), they fall short of the ability to recognise words as soon as they are spoken. Instead, the entire speech utterance needs to be encoded before the first word is transcribed, greatly increasing the latency of the system. Such models do not achieve a hard segmentation of the input. This aspect offers them a greater degree of modelling flexibility during decoding, at the expense of an increased latency and the reliance on external segmentations of utterances. As noted in the comparative study of Prabhavalkar et al. (2017), the inference in the RNN Transducer (RNN-T) has the potential to be performed in a frame-synchronous manner if coupled with a unidirectional encoder, although their work only investigated bidirectional encoders to allow a more fair comparison to the attention model. Indeed, the RNN-T model has already been tested in a practical setting, Sainath et al. (2020) showing that the RNN-T is comparable in latency and accuracy with a conventional model for only a fraction of the size. However, a shortcoming of RNN-T is its inference complexity, where two separate modules, the *Prediction* and *Transcription* networks, dynamically alternate their turns depending on the current output label being either a blank or non-blank token. Wang et al. (2019) analyse the shortcomings of the RNN-T, finding that its dynamic programming training algorithm marginalises over a large number of alignment paths including many unreasonable ones, and report training difficulties. Furthermore, Battenberg et al. (2017) note that bridging the modelling assumptions between the RNN Transducer and attention models, particularly by equipping attention models with the monotonicity constraints of the RNN Transducer, is a promising avenue.

The Segmental RNN (Lu et al., 2016; Tang et al., 2017) adapts CTC with an explicit segmentation model, however it maintains the complex training algorithm that marginalises out all possible input segmentations. Empirically, Tang et al. (2017) find no statistical difference between the Segmental RNN and CTC models on a phoneme recognition task. Beck et al. (2018a,b) find their segmental RNN variants to be inferior to both a hybrid DNN-HMM and an attention model. Recent work from Zeyer et al. (2020) suggests that the class of Transducer models has several potential benefits if provided with a good external alignment at a pre-processing step.

Overall, there is strong evidence showing that the RNN-T model has a non-

negligible training complexity, and warrants an alternative approach to online ASR. Henceforth, we will aim to adapt the sequence to sequence model with attention by making use of speech domain knowledge and insert the local monotonicity constraint into its structure, while preserving the model's property to be fully differentiable and trainable with backpropagation. Achieving an explicit segmentation as in the Segmental RNN is a desired property, however we aim to condition each output label on a dynamic but narrow window of acoustic frames. Finding the right limits of this window will be the key problem to solve.

A major technical challenge in learning to segment speech is the difficulty of formulating the problem in a fully differentiable framework. Some previous attempts include the Recurrent Neural Network Transducer (Battenberg et al., 2017; Graves, 2012; Graves et al., 2013; Rao et al., 2017), Neural Transducer (Jaitly et al., 2016; Sainath et al., 2018), segmental conditional random fields (Beck et al., 2018a; Lu et al., 2016; Tang et al., 2017), hard monotonic attention (Luo et al., 2017; Raffel et al., 2017), segment attention (Chiu and Raffel, 2018; Fan et al., 2019; Hou et al., 2020), or triggered attention (Moritz et al., 2019). However the models made use of dynamic programming, training in expectation, or policy gradients, and the authors report training difficulties. When the objective function is differentiable with respect to the model parameters, we can use highly efficient optimisation methods (Kingma and Ba, 2015). Our work retains the segment attention design, but approaches the problem of speech segmentation from a different angle. By learning to count words through self-supervision, we introduce a mechanism that allows end-to-end training using only backpropagation.

More recent proposals for online speech recognition address this challenge by assuming one sub-word unit per segment (Dong and Xu, 2020; Li et al., 2019), or discover an inventory of sub-word units (Drexler and Glass, 2020), a concept previously explored in machine translation (Kreutzer and Sokolov, 2018). Our focus in this work is on word units. In English, words allow a monotonic and bijective mapping between their acoustic and symbolic representations, however these properties do not hold at the sub-word level due to the highly complex spelling rules in English orthography. Words may be identified as clusters of co-occurring sounds, since the correlations between sounds are relatively stronger within words than at the word boundaries (Saffran et al., 1996). Moreover, words can be counted in a deterministic way, which allows us to introduce a self-supervision word counting task without requiring new annotations. Our contribution to online speech recognition will be described in Chapter 5.

2.8 Audio and visual speech datasets

In this section we describe the two audio-visual speech corpora used in this work. Specific details of how are the datasets partitioned and prepared in terms of feature extraction and noise augmentation are discussed in the respective chapters.

Since we are investigating the topic of audio-visual speech modelling for continuous speech recognition, we are looking for corpora with the following requirements: i) medium to large vocabularies, ii) fluent speech, as opposed to isolated words and sounds, iii) sufficiently large to allow good generalisation, iv) medium to large number of speakers for speaker independent models, and v) freely available for academic research to foster reproducibilty. Harte and Gillen (2015) concluded that no audio-visual dataset released before 2015 meets such criteria, and introduced the TCD-TIMIT corpus. Fernandez-Lopez and Sukno (2018) reach similar conclusions after reviewing the work in visual speech recognition published between 2007 and 2018. Since TCD-TIMIT contains high quality audio-visual recordings and relatively low adverse conditions for AVSR that will be later detailed, we choose it for the faster prototyping of our algorithms. Later, a collaboration between BBC Television and The University of Oxford led to the availability of the considerably larger Lip Reading Sentences (LRS) corpus, initially for an internal work (Chung et al., 2017), followed by the LRS2 corpus publicly available for academic research (BBC and University of Oxford, 2017), and LRS3-TED additionally available for industrial research (Afouras et al., 2018a). Since LRS3-TED has a comparable size with LRS2 but contains less challenging auditory and visual conditions that are specific to the format of the TED talks, we will focus our prototyping and experimentation on LRS2.

2.8.1 TCD-TIMIT

TCD-TIMIT (Harte and Gillen, 2015) is a publicly available audio-visual dataset consisting of high quality audio-visual footage of 62 speakers reading a total of 6,913 examples of both phonetically compact (*sx*) and diverse (*si*) sentences from the prompts of the TIMIT dataset (Garofalo et al., 1993) in laboratory conditions. The videos have an image resolution is 1920x1080 pixels, a rate of 30 frames per second, and the audio is sampled at 48,000 Hz. Three of the 62 speakers are professionally trained to exaggerate their visual articulation of speech and were not included in our study. The remaining 59 speakers recite 98 phonetically balanced sentences each from a vocabulary of 6000 words, totalling around 8 hours of recordings. Sentences vary from 10 to 80 characters

in length. It is important to note, in the context of how the results are later discussed, that there is a difference between the coverage of these two types of sentences. Specifically, 450 *sx* sentences are spoken by seven different speakers on average, whereas 1890 *si* sentences are unique to each speaker. The dialect-dependent sentences (name begins with *sa*) were removed. As in the original TIMIT database, these two sentences were spoken by all the speakers. Consequently, they would provide an overestimate of the true model accuracy, predominantly owed to the relatively higher predictability of the target labels. Three of the 59 regular speakers have different native accents from the others, namely British and Spanish, and were also not included in our study to further control the system exposure to outliers. The remaining 56 speakers are displayed in Figure 2.1. Below are several scripts from TIMIT read by the TCD-TIMIT volunteers.

- *He took his mask from his forehead and threw it unexpectedly across the deck*
- *Civilization is what man has made of himself*
- *The mayan neoclassic scholar disappeared while surveying ancient ruins*

Figure 2.1: *The 56 English speakers of the TCD-TIMIT dataset with Irish accents used to train our models in this work.*

Two partitioning schemes have been originally proposed for TCD-TIMIT (Harte and Gillen, 2015). A speaker dependent partition includes all the speakers in both the training and test sets, and uses 61 and 32 sentences from every speaker

for each of the two sets respectively. Harte and Gillen (2015) also propose a speaker independent partitioning, however it was not used in this work. During our initial experiments on this dataset with deep neural networks we acknowledged an overfitting problem owed to the highly repetitive patterns of the TCD-TIMIT sentence scripts. Consequently, in this work we manually designed a new speaker independent partitioning scheme aiming to maximise the diversity of the sentences seen in training, while also balancing the gender and the level of facial hair. This partitioning will be detailed in Section 4.3.3. We considered gender as more likely to have a higher impact on the recognition of auditory and visual speech. Although it is not fully clear if facial hair plays a significant role in visual speech perception, we considered that neural networks may not generalise well on a dataset of this size when not exposed to a wider range of appearances of the mouth area.

In Table 2.1 we list the typical system error rates achieved on this dataset. The best performing video-only / lipreading model, developed by Thangthai and Harvey (2018), is based on a hybrid DNN-HMM system trained with state-level minimum Bayes risk (sMBR), and extracts visual features from a deep autoencoder network, on top of which is applied a stack of LDA, MLLT, and feature space maximum likelihood linear regression (fMLLR) (Gales, 1998). Thangthai and Harvey find that the strength of language greatly affects the overall performance, with the error rate decreasing from 93.76% without a LM, to 31.55% with a 4gram LM. There are considerable differences between the unigram (89.31%), bigram (46.17%), or trigram (32.31%) LM. Koumparoulis et al. (2020) take a different approach of predicting context-dependent HMM phoneme posterior probabilities with a stack of CNN and time-delay neural network (TDNN), which are then decoded using a weighted finite-state transducer (WFST) incorporating a bigram language model. Apart from the stronger language model used by Thangthai and Harvey (2018), the difference between these two systems could additionally be owed to the relatively small size of TCD-TIMIT, making it more challenging to learn the parameters of the neural front-end of Koumparoulis et al. (2020) without the HMM structure.

On the task of Audio-Visual speech recognition, Abdelaziz (2018) achieves a *phone* error rate (PER) of 18.2% on clean speech with a multi-stream hidden Markov model (MSHMM) (Potamianos et al., 2003), improving from a PER of 21.6% obtained with a single-stream HMM. With the MSHMM, Abdelaziz estimates a PER reduction of 26% over the audio only model when averaging over multiple noise types and signal to noise ratios in the [-5db : 20db] range. It is important to note that Abdelaziz (2018) uses a different partitioning of the TCD-

TIMIT corpus than Harte and Gillen (2015), making their results less comparable.

System	Unit	Error Rate (SD)	Error Rate (SI)
Audio only			
GMM-HMM [3]	Phone	50.16	52.37
DNN-HMM + 2gram LM [5]	Phone		21.6*
Video only			
DNN-HMM [2]	Word	66.94	80.85
DNN-HMM sMBR [1]	Word	42.64	46.17
DNN-HMM sMBR + 4gram LM [1]	Word		31.55
GMM-HMM [3]	Viseme	65.46	
DNN-HMM [2]	Viseme	53.39	55.4
DNN-HMM + 2gram LM [5]	Viseme		51.7
DNN-HMM + 2gram LM [5]	Phone		65.4
CNN+TDNN + 2gram LM [4]	Word		44.86
Audio-Visual			
GMM-HMM clean speech [3]	Phone	62.76	
MSHMM clean speech [5]	Phone		18.2*

Table 2.1: *Previous results obtained on TCD-TIMIT by various studies.* [1] *Thangthai and Harvey (2018),* [2] *Thangthai et al. (2017),* [3] *Harte and Gillen (2015),* [4] *Koumparoulis et al. (2020),* [5] *Abdelaziz (2018).* * *denotes a custom dataset partitioning.*

2.8.2 LRS2

LRS2 (BBC and University of Oxford, 2017) consists of spoken sentences from BBC television, containing a number of 96,318 examples for pre-training, 45,839 for training, and 1,243 for testing. Unlike TCD-TIMIT, it contains more challenging head poses, uncontrolled illumination conditions in outdoor environments, a much lower image resolution of 160x160 pixels, a lower frame rate of 25 images per second, and the vocabulary size is of approximately 15,000 words. The audio track is sampled at 16,000 Hz. A small subset of LRS2 speakers is displayed in Figure 2.2. Several phrases spoken by the LRS2 speakers are listed below.

- *They don't all have to be red hot*
- *And Norwich is actually quite good for that*
- *This being more common than our normal kind of planet*

LRS2 is partitioned into a pretrain set of 96,318 sentences, training set of 45,839 sentences, a validation set of 1082 sentences, and a test set of 1243 sentences.

Figure 2.2: *Several professional presenters, journalists, or reporters from the LRS2 dataset. Unlike the members of the public, these speakers generally look directly into the camera and have a more prepared speech. The dataset license does not allow the explicit display of other categories of speakers, however the models used in this work were trained on the entire main training set of this dataset.*

Since the training set represents a clean subset of the pretrain one, we only make use of the former in all our experiments in this work. In Table 2.2 we list typical error rates obtained on this dataset.

Chung et al. (2017) implement sequence to sequence neural networks for audio-only, video-only, and audio-visual speech recognition. The multimodal network, coined *Watch, Listen, Attend, and Spell* (WLAS) adds a second attention mechanism to align the decoder state with both the audio and the video encoders respectively. This architecture will be discussed in greater detail in Chapter 4. Because the data used by Chung et al. (2017) was not made publicly available, their results are not entirely comparable with other approaches in this table. Afouras et al. (2018b) introduce two architectural modifications, first replacing the LSTM-based encoders and decoders with equivalent Transformer networks (TM in the table), and second experimenting with CTC-based decoding and objectives. Petridis et al. (2018b) combine the CTC and the Attention-based sequence to sequence networks, obtaining a 8.3% WER with an audio-only model, and 7.0% WER with an audio-visual model on a clean speech recognition task.

System	Modalities	Error rate [%]				Unit
		clean	10db	0db	-5db	
WAS[1]$*$	V	42.1	-	-	-	character
LAS[1]$*$	A	16.2	33.7	59.0	-	character
WLAS[1]$*$	A + V	13.3	22.8	35.8	-	character
WAS[2]	V	70.4	-	-	-	word
TM-seq2seq[2]	V	49.8	-	-	-	word
TM-CTC[2]	A	15.3	-	64.7	-	word
TM-CTC[2]	A + V	13.7	-	33.5	-	word
TM-seq2seq[2]	A	10.5	-	58.0	-	word
TM-seq2seq[2]	A + V	9.4	-	35.9	-	word
CTC-Attention[3]	V	42.1	-	-	-	character
CTC-Attention[3]	A	4.4	-	-	-	character
CTC-Attention[3]	A	8.3	$\approx$16	$\approx$61	$\approx$94	word
CTC-Attention[3]	A + V	3.6	-	-	-	character
CTC-Attention[3]	A + V	7.0	$\approx$10	$\approx$33	$\approx$63	word
TM-CTC + DFN[4]	V	64.08				word
TM-CTC + DFN[4]	A	4.2		6.3		word
TM-CTC + DFN[4]	A + V	2.4		4.8		word

Table 2.2: *Previous results obtained on LRS2 by various studies.* [1] *Chung et al. (2017),* [2] *Afouras et al. (2018b),* [3] *Petridis et al. (2018b),* [4] *Yu et al. (2021). The results labelled with a* $*$ *are reported on the unreleased dataset LRS which used similar source material from BBC. The results labelled with* $\approx$ *were approximated from line plots.*

Very recently, Yu et al. (2021) develops an improved model combining both the CTC and the cross-entropy objectives in a Transformer-based neural architecture, making the audio-visual fusion operation aware of signal reliability in each stream. They use a face detector and facial action units for annotating the confidence in each video frame, and also estimate the voice pitch and signal to noise ratio from audio.

It is important to note that the aims of this thesis go beyond achieving state of the art results on LRS2. From Table 2.2 it is difficult to draw a conclusion regarding the optimal audio-visual fusion strategy. The work of Petridis et al. (2018b) shows that early fusion is superior to late fusion in the attention-based seq2seq framework regularised with the CTC training objective, but only for clean speech conditions, without investigating late fusion at lower signal to noise ratios. Furthermore, since Petridis et al. (2018b) and Afouras et al. (2018b) use different amounts of data and strategies for pre-training the visual front-end, and different language models, it is difficult to directly compare early fusion in the former work with the dual attention fusion strategy in the latter. We can see that both Petridis et al. (2018b) and Afouras et al. (2018b) require a separate task of 500-class visual word classification using the visual front-end alone in order to help the

gradient descent algorithm converge on the audio-visual task. Whether the correlations in speech between audio and video could be learnt without resorting to training recipes and additional tasks remains an open question. Another example illustrating the difficulty of comparing two methods is the audio-only model of Yu et al. (2021) evaluated on noisy speech (6.3% WER at 0db SNR) outperforming the audio-visual model of Petridis et al. (2018b) evaluated on clean speech (7.0% WER). Understanding the fusion problem in more depth would allow us to train audio-visual speech models with fewer prior assumptions, while also reducing the associated time, effort, and supplementary data costs. As a result, we do not aim to achieve the most accurate results on the relatively small datasets used in this thesis. Merely increasing the amount of video, audio, or text data alone outside the main $\approx 30h$ train partition of LRS2 could push the absolute baselines of both audio-only and audio-visual models, but would incur higher costs and slow down the rate of testing new hypotheses. To achieve a fair comparison of the fusion method developed in this thesis with both WLAS and early fusion, we will re-implement them in our own framework and ensure an identical training process for all methods.

Note that we use two additional speech datasets in Chapter 5 for audio-only experiments, namely LibriSpeech (Panayotov et al., 2015) and AISHELL-1 (Bu et al., 2017). They will be introduced in Section 5.5.

Returning to the limits of audio-based speech recognition discussed in Section 2.1, we notice that both LRS2 and TCD-TIMIT contain less challenging speech material than the conversational speech of CHiME-5, Switchboard, or CallHome. Given the prior work discussed in Section 2.6, there are several aspects that we need to solve before increasing the task difficulty. The evaluation in this thesis will be specific to read or prepared speech from the broadcast material of BBC news. The multimodal recognition of conversational speech remains a challenge unaddressed in this work. Nevertheless, the proposed strategy in this work is by no means limited by the speech content type.

2.9 Evaluation

In this thesis, automatic speech recognition will be seen as the task of mapping a spoken utterance onto a sequence of graphemes, unconstrained by the vocabulary size or the requirement of intermediate annotations at the sub-sequence level. We will use the terms *continuous / connected / fluent speech* to refer to the act of speaking phrases of multiple words not separated by unnatural pauses. The most commonly used metric to evaluate the ASR performance is

the normalised edit distance between the ground truth and the predicted strings of graphemes, known as the error rate and defined as follows:

$$\textit{Accuracy}\ [\%] = \frac{\textit{Correct} - \textit{Ins} - \textit{Del} - \textit{Sub}}{\textit{Total}} \cdot 100 \tag{2.59}$$

$$\textit{Error Rate}\ [\%] = 100 - \textit{Accuracy} \tag{2.60}$$

Intuitively, this metric indicates the minimum number of changes (Insertions, Deletions, Substitutions) needed to transform an input string into a target string. The optimal number of changes is typically obtained using a dynamic programming algorithm computing the Levenshtein edit distance. The error rate is lower bounded at 0% when the two strings are identical, but it is not upper capped since the hypothesis string may have more insertion errors than the total length of the target string.

The well established state-space models described in Section 2.4 use a word-based lexicon to decode the sequence of phone states to a sequence of words. This made the Word Error Rate (WER) a widely adopted metric for assessing a speech recogniser. In contrast, the end-to-end sequence to sequence neural network is typically constructed with character-based units in speech recognition. This is owed to the decoder module requiring an embedding matrix of all the units in the vocabulary, prohibiting the use of a large number of words. As a consequence, a typical sequence to sequence neural network decodes speech as a string of characters, enabling a Character Error Rate (CER) metric. Since the modelled alphabet typically includes punctuation, the WER can be then trivially computed by merging character sub-strings delimited by the blank space token.

For systems modelling viseme or phoneme units in Chapter 3, we will report Viseme and Phoneme Accuracy scores respectively. For the rest of this thesis we will mainly report CER, with the following reasoning. First, CER is better aligned with the label-wise cross-entropy training objective than WER. Second, CER offers a higher granularity of the error metric than WER. For example, decoding a sequence of consistently misspelled words may lead to a WER close to 100%, dismissing altogether the correctly predicted characters in each work. As we discussed in Section 2.2, the visual modality of speech may help disambiguate only certain categories of sounds for which their place of articulation is visible. CER would allow us to identify an eventual improvement when using an AVSR system without necessarily having to develop strong audio baselines trained on large amounts of data. Third, one major goal of this thesis is to investigate the fusion strategy between auditory and visual speech representations. As a result,

we are interested in assessing the raw decoding performance of the acoustic and visual models, thus we do not rescore the models' predictions with an external language model that may obfuscate the differences between audio-only and AVSR systems. Finally, following the discussion in the previous section, achieving state of the art absolute WER/CER scores does not represent a primary goal of the thesis. Instead, we will emphasise the relative scores and re-implement the well established baselines in order to achieve a more fair comparison between alternative methods.

It is important to note that, despite their popularity in speech recognition, reporting average error rates has been criticised in the speech literature. Oviatt (2002) warns about the disadvantages of over-relying on a single metric, particularly on WER, when designing spoken language systems. Bourlard et al. (1996) take a stronger stance, supporting the idea that minimising the word error rate in automatic speech recognition can often suppress innovation. Their argument is that the well established techniques have been overly tuned to the test data of common benchmarks. This makes it difficult for new approaches to surpass the accuracy of the traditional ones. In turn, there is an incentive to make small incremental improvements to the leading approaches. As will become clearer in the following chapters, the major contributions of this thesis are not directly aimed at minimising average error rates. Instead, we will address several structural problems in AVSR and online decoding with sequence to sequence neural networks, as we will explore the asynchronous relationship between speech modalities, and discover a dynamic speech segmentation mechanism.

3 Lipreading

3.1 Introduction

Extracting good visual representations is one of the major challenges in audio-visual speech processing. In this chapter we will focus on lipreading, which is the task of decoding text from the visual modality of speech. My original contribution to knowledge in this chapter is an exploration of what defines a good visual representation and how could such properties be later transferred to audio-visual speech recognition.

From the literature review of lipreading in Section 2.6.1, it can be observed that most models fall short of reasonably accurate decoding of fluent visual speech into words. Petridis et al. (2017b) report an accuracy of 94.7% at classifying 10 short visual utterances. Next, Stafylakis and Tzimiropoulos (2017) tackle the more difficult task of word classification from a vocabulary of 500 words, obtaining an accuracy of 83%. Moving to the more challenging conditions of unconstrained fluent speech, Shillingford et al. (2019) go a greater length than anyone else to train a large scale lipreading model on 3,886 hours of audio-visual speech recordings which span a vocabulary of 127,055 words. They find that the average word error rate of their best model is 40.9%, increasing to 53.6% when no language model is used. Their CTC-based neural model made use of phoneme abstractions, and the authors show that decoding directly to characters further increases the word error rate to 76.8%. This striking contrast between restricted and unrestricted vocabularies, coupled with the low human level performance at lipreading reported by Shillingford et al. (2019), raises the following question: is there another way to formulate the lipreading task that allows a neural network to more easily learn how to process visual speech ?

Given the inherent ambiguity in visual speech owed to the obscured articulatory information, the spoken message may only be recovered from context and guessed based on prior statistical knowledge of speech, often to a partial extent. We see two problematic aspects in decoding visual speech. First, the lipreading

task in computers is almost always specified as to decode the genuine spoken message that was manually transcribed by annotators having available both the audio and video channels. It is then likely that the visual information plus the prior knowledge of speech are insufficient for the reliable prediction of the provided labels. Second, lipreading models not only have to cope with visual challenges posed by illumination conditions, dynamic viewing angles, speaker appearance diversity or speaking style, but also need to develop strong language modelling abilities. From the study of Shillingford et al. (2019) it is not entirely clear how much each of these factors contribute to the relatively low lipreading accuracy on unconstrained speech, and the entangled nature of the end-to-end task does not facilitate such analysis. The professional lipreaders they hired could only decode words with a mean accuracy 7.1%, increasing to 13.6% when they are additionally provided with the topic and the first 6 words in the video. Despite the human participants having already mastered vision, the lexical ambiguity in the visual modality appears to be a major bottleneck in lipreading.

In this chapter we investigate the fluent speech lipreading performance when the modelled unit is the viseme. This is the visual equivalent of a phoneme and is discussed in Section 3.2. The aim in this choice is to lessen the label ambiguity associated with character or phoneme units, allowing a neural network to spend more capacity on addressing the remaining challenges associated with the visual channel and speaker variance. The viseme units have been used for a long time in lipreading research (Goldschen et al., 1997; Rogozan, 1999) before end-to-end models became feasible in speech and computer vision. It is not fully clear how we should define the visemes (Cappelletta and Harte, 2012), and how useful viseme-based models can become in practice. We will first investigate two engineered visual features in lipreading, namely DCT coefficients and AAM parameters, and model their temporal dynamics using GMM-HMM systems. These visual features were commonly used in lipreading at the outset of this thesis, before being superseded by CNN representations learnt from data. Then, we will compare their performance with sequential recurrent neural networks that learn visual representations from raw pixels using backpropagation, and make fewer simplifying assumptions than HMMs. Such neural models will be extended with a multimodal processing component in Chapter 4.

We initially study the lipreading performance on the TCD-TIMIT dataset introduced in Section 2.8.1, which was designed to further control some of the variables of the visual channel, specifically the relatively constant head pose, clear and identical illumination conditions for all speakers, high image resolution, and a constant distance from the video camera. Next, we run the same experiments on

the larger and unconstrained LRS2 dataset introduced in Section 2.8.2 to analyse the impact of the visual channel on the decoding accuracy. Our work does not attempt to advocate the use of visemes over phonemes or characters, but instead is aimed at investigating whether or not reducing the label ambiguity increases the decoding accuracy, and what kind of decodable knowledge exists in the visual modality.

3.2 Visemes

The concept of visemes was introduced by Fisher (1968) to describe *"any individual and contrastive visually perceived unit"* of speech. This definition makes the visemes an attractive option for lipreading models, as it has the potential to lower the amount of ambiguity in the target signal. There have been many proposals to define an inventory of visual units based on expert knowledge (Bear and Harvey, 2017; Bozkurt et al., 2007; Cappelletta and Harte, 2012; Goldschen et al., 1996; Jeffers and Barley, 1980; Rogozan, 1999; Taylor et al., 2012), typically involving the clustering of visually similar phonemes. Still there is no consensus on what is the optimal grapheme to viseme mapping in lipreading. Furthermore, Taylor et al. (2012) argue that visual speech could be better described using dynamic units rather than static ones. In our work we choose the mapping of Jeffers and Barley (1980), as it was also used by Harte and Gillen (2015) and allows us a direct comparison on the TCD-TIMIT dataset. This mapping comprises a set of 12 visemes, listed in Table 3.5 of Section 3.6.4.

3.3 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) represents a common choice for visual feature extraction in many lipreading tasks, as seen in the literature surveys of Fernandez-Lopez and Sukno (2018); Potamianos et al. (2003); Zhou et al. (2014). The DCT is a core operation in image and video compression, as it enables a frequency decomposition of the energy of a signal, with the noise being naturally concentrated in the higher frequency range (Ahmed et al., 1974). Matthews et al. (2001) found that image-based transforms, including the DCT, outperform the representations obtained with deformable models of facial shape and appearance on the task of large vocabulary audio-visual speech recognition. Similarly, Seymour et al. (2007) found DCT features to achieve a lower error rate than wavelet and principal component based transformations on the task of lipreading continuously spoken digits, and also show the highest robustness to image corruption through blurring. Coupled with the widespread availability

0	1	5	6	14
2	4	7	13	15
3	8	12	16	21
9	11	17	20	22
10	18	19	23	24

Table 3.1: *Zig-zag pattern in a DCT matrix that prioritises low frequency coefficients.*

of fast DCT software implementations, and its properties to be invertible and to achieve an ordering of the signal components by frequency, we select DCT as a baseline for our investigation.

For an image X of size MxN, the 2D DCT is commonly defined as the following linear transformation:

$$Y_{u,v} = \sqrt{\frac{\alpha_u}{N}}\sqrt{\frac{\alpha_v}{M}} \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} X_{n,m}\, cos\left(\frac{\pi(2n+1)u}{2N}\right) cos\left(\frac{\pi(2m+1)v}{2M}\right) \tag{3.1}$$

where $\alpha_0 = 1$, $\alpha_i = 2$ for $i > 0$, and $0 \leq u \leq N-1$, $0 \leq v \leq M-1$.

To obtain a DCT-based feature in our framework, a region of interest (ROI) has to be first localised and isolated from the full-sized image. As the initial work of Harte and Gillen (2015) provided normalised mouth ROI images for TCD-TIMIT, we obtained their coordinates in the full image through cross-correlation-based template matching in order to apply different post-processing steps. The extracted ROI is converted to grey-scale, then down-sampled to 36x36 pixels using cubic interpolation, and finally a 2D DCT transform is applied. The 36x36 representation is vectorised by keeping a truncated set of low frequency coefficients, as this is a common DCT feature vectorisation method in lipreading (Harte and Gillen, 2015; Seymour et al., 2007; Thangthai and Harvey, 2018). More precisely, the coefficients are chosen from the top left corner of this matrix in a zig-zag pattern, which is illustrated in Table 3.1. The feature vector is made of the first 44 coefficients (without the first coefficient at position [0,0] which only encodes the average pixel intensity) and is concatenated with its first and the second derivatives across time. The derivatives are computed using a central finite differences scheme that is fourth order accurate in the Taylor series expansion, and the same order is preserved at the boundaries by using forward and backward schemes. Accordingly, we obtain a DCT feature vector of size 132 for each video frame.

Since we are keeping the feature size constant, there is a trade-off between the spatial frequency range captured by the selected DCT coefficients and the granularity of the representation. The choice for the window size was made experimen-

tally, after testing values of 24, 28, 32, 36 and 40 pixels per side. Koumparoulis et al. (2017) study the effect of ROI resolution in lipreading in more depth and achieve the best result for a resolution of 60x60 pixels, yet the absolute error differences between multiple resolutions do not appear to be substantial enough to suggest an abrupt loss of spatial information beyond a specific threshold.

3.4 Active Appearance Models

Another commonly used feature in lipreading is the parametrisation of an Active Appearance Model. (Lan et al., 2010) found AAM features to outperform DCT on lipreading continuous speech from a small vocabulary of 51 words.

An AAM is a deformable statistical model of shape and appearance that learns the variance of an annotated set of training images. The shape consists of a set of landmarks $\mathbf{s} = [x_1, y_1, ..., x_N, y_N]$ placed on the object to be modelled, which are a priori aligned using Generalised Procrustes Analysis to reduce the effect of translation, rotation and scaling. Applying Principal Component Analysis (PCA) on the set of aligned training shapes leads to a shape model expressed as:

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{n} p_i s_i = \bar{\mathbf{s}} + \mathbf{Sp} \tag{3.2}$$

where any shape s is a linear combination of the shape eigenvectors s_i with the weights p_i also known as shape parameters, plus the mean shape $\bar{s}$.

To construct the appearance model, the pixels within the training shapes are first warped to their corresponding locations in a common reference shape (typically the mean shape $\bar{s}$) using techniques such as piecewise affine warping or thin plate splines. PCA is applied again on the serialised warped image denoted here with x, such that any appearance $A(x)$ could be expressed as a mean appearance $\bar{A}(x)$ plus a linear combination of the appearance eigenvectors $A_i(x)$:

$$\mathbf{A}(x) = \bar{\mathbf{A}}(x) + \sum_{i=1}^{m} c_i A_i(x) = \bar{\mathbf{A}}(x) + \mathbf{Ac} \tag{3.3}$$

where the weights c_i denote the appearance parameters.

Since the number of shape and appearance parameters is as large as the number of landmarks and the number of pixels respectively, a trade-off can be made between the representation power of the models and the size of the parameter vectors by analysing the cumulative ratio of the corresponding eigenvalues.

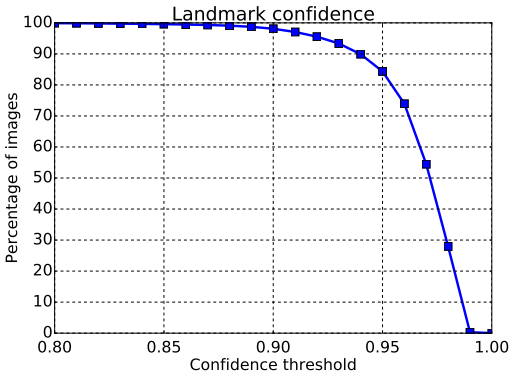


Figure 3.1: *OpenFace landmark confidence on TCD-TIMIT*

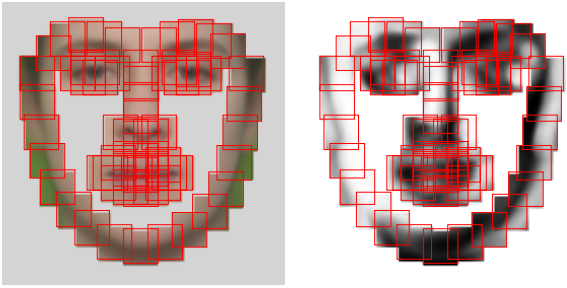
For unlabelled images, when a good initialisation of the shape can be provided (e.g. the mean shape aligned on a face localised using a face detector), several fitting algorithms can be applied to iteratively update the parameters that minimise an error between the given image and the model instance. Alabort-i Medina and Zafeiriou (2017) classify these algorithms with respect to the cost function, type of composition and optimisation method. The parameters obtained at the last iteration constitute the foundation of the AAM-based visual features.

3.4.1 AAM training

An annotated set of images is required to train AAMs. Previously, this has been a time-consuming step for most datasets. In Lan et al. (2009) and Bear et al. (2014), a few frames per speaker are manually annotated, then person-specific AAMs are trained and fitted on the remaining frames. Thanks to the recent advancements in face alignment implemented in the open-source tool OpenFace of Baltrusaitis et al. (2016) and the public availability of annotated facial data for training, we noticed that such bootstrapping annotation techniques are no longer necessary for training facial AAMs, as the generic landmark estimates provide a sufficiently good initialisation for the iterative optimisation algorithms. We used OpenFace to obtain 68 facial landmark estimates and their confidence scores for each video frame. We then analysed the cumulative distribution of these confidence scores on TCD-TIMIT, shown in Figure 3.1. This reveals an overall high confidence, which means that most frames have reliable landmark labels. From a visual inspection we observed that most landmarks above a confidence score of 0.9 were very accurate, with the exception of the lips region.

Training generative models such as AAMs with a massive amount of similar data, such as consecutive video frames, leads to poor performance in practice, so we apply a sampling strategy. Taking the faces that get detected successfully and

(a) *Holistic no-op* **(b)** *Holistic SIFT*



(c) *Patch no-op* **(d)** *Patch SIFT*

Figure 3.2: *Overview of AAM types by warp and feature used in this work. The* Patch *models are evaluating local neighbourhoods of the landmarks instead of the entire appearance. The SIFT descriptors are robust alternatives to raw pixel intensities where no image transformation is applied (no-op). For the illustrations of the SIFT-based models in Figures 3.2b and 3.2d, we only display the first channel of the SIFT features.*

that have a high confidence score, we sort them by the amount of lip opening (distance between the upper and lower lips). We then sample between 3 to 6% of the sorted frames at evenly spaced intervals. For TCD-TIMIT we decided to use a confidence threshold of 0.94 to train our models, which kept 90% of the frames. In addition, we randomly selected only 5 training sentences per speaker from the available 67, further reducing the training data size to a total of around 1100 frames. The reference shape of the AAM, to which all the other faces will be aligned to, was chosen as the mean shape from the first video in the dataset (*01M/si2077*). We will refer to these models as *global*, since they use training data from each volunteer. The models built from the training samples of a single person will be coined *person-specific*.

The previous attempts at lipreading with AAMs have used the original formulation where the entire appearance texture within the landmark area was modeled. Tzimiropoulos and Pantic (2014) show that learning only small patches around the landmarks leads to robust models that outperform the state of the art at fitting to unseen faces. We considered both approaches, coined *Holistic* and *Patch* AAMs in Alabort-i Medina et al. (2014) (and illustrated in Figure 3.2), in order to compare their fitting and classification performance. In addition to the traditional pixel

intensities for appearance features (denoted in this work and in Alabort-i Medina et al. (2014) as *no-op*), we also considered scale-invariant feature transform (SIFT) image representations of Lowe (2004), which were shown by Antonakos et al. (2015) to largely outperform popular alternatives at fitting to unconstrained images, requiring at the same time fewer appearance components.

Modelling only a part of the face can be beneficial for lipreading (Berry et al., 2011; Papandreou et al., 2009), since the PCA energy would better describe the subtle movements of the more informative articulators. However, modelling a smaller area is prone to higher fitting errors. We build two additional models, one for the lips area only, and another for the whole chin and mouth area (further denoted as *chin*), the latter being chosen as a trade-off between relevancy to lipreading and fitting performance. The face and the chin models use a pyramid of three resolution levels (25%, 50%, 100%) as in Papandreou et al. (2009), whereas the lip models only use the last two. Other important parameters for our models were the image re-scaling to a diagonal of ≈ 150 pixels at full scale, 40 and 150 shape and appearance components respectively, and patch sizes of 17x17 pixels around landmarks for the Patch models. Accordingly, our AAMs describe each video frame with a vector representation of size 190.

Table 3.2 shows how well our models were able to represent the appearance of the training data. High values of the kept variance imply that model is able to reconstruct accurately any training face, provided that the optimisation algorithm finds the right parameters. More variance was kept using pixel intensities than SIFT features, likely because the colour images have only three channels whereas the SIFT ones have eight, hence 2.66 times more raw data is being modelled. The variance kept by the shape eigenvectors was close to 100% using 40 components, suggesting that there are strong correlations between the landmark locations.

Table 3.2: *Percentage of kept variance for the appearance models using 150 appearance components*

Model $\rightarrow$	**Holistic**		**Patch**		**Scale**
$\downarrow$ **Part**	**no-op**	**SIFT**	**no-op**	**SIFT**	
face	96.6	78.7	83.1	63.0	**25%**
	96.8	79.2	87.6	71.1	**50%**
	93.2	76.9	82.8	74.7	**100%**
chin	97.9	75.9	82.8	56.4	**25%**
	97.1	73.4	87.4	65.0	**50%**
	93.6	70.1	83.9	69.6	**100%**
lips	95.4	72.2	89.2	61.6	**50%**
	91.6	68.5	90.9	65.9	**100%**

3.4.2 AAM parametrisation

The AAM fitting process consists in the optimisation of a cost function (typically the error between a given image and the AAM reconstruction) with respect to the shape and appearance parameters, provided that a good initialisation is available. We initialise the shape by running the *dlib* face detector used in *Menpo* (Alabort-i Medina et al., 2014) that estimates the face bounding box, within which the mean shape is aligned. The Wiberg Inverse Compositional (WIC) algorithm was chosen for the optimisation problem, as it was shown by Alabort-i Medina and Zafeiriou (2017) to be an efficient alternative to state of the art algorithms. We ran 10 iterations of WIC for the first two resolution scales and 5 more for the full resolution model, with the exception of the Holistic no-op model that needed 20 iterations at the lowest scale to converge rather than 10 as the other models. For the *chin* and *lips* models, the shapes were initialised from a subset of the final *face* shape, iterating 10 more times per resolution scale to make room for corrections.

We used the shape and the appearance parameters after the last iteration as feature vectors for the lipreading models. The features are built either from the shape only, appearance only, and the concatenation of shape and appearance parameters. We build two additional features from the first derivative of the appearance alone and the concatenation of shape and appearance parameters. Among these five features, the highest performance in our initial experiments was achieved by the latter, which was our default choice in the subsequent experiments. The first four shape parameters were discarded, as they represented the global similarity transform used for normalisation. The AAM optimisation is a slow process, taking almost one day to process the 67 training files of each TCD-TIMIT speaker using the four face models alone in *menpo*. We ran the fitting process on a HPC cluster made of 16 nodes and 40 CPU cores each, achieving a theoretical speedup factor of 160.

3.4.3 AAM fitting evaluation

The overall lipreading performance partly depends on the accuracy of landmark localisation on unseen faces. In this experiment we compare the performance of our face AAMs in terms of face-normalised point-to-point Euclidean error between the WIC fitter prediction and the ground-truth shapes. Although the ground-truth landmarks are still estimates from OpenFace, we noticed that the confidence scores reported in Figure 3.1 are highly correlated with our expectation as we visually inspected various speakers in the TCD-TIMIT dataset. We ob-

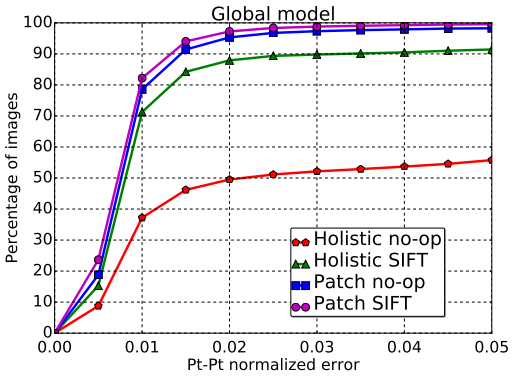
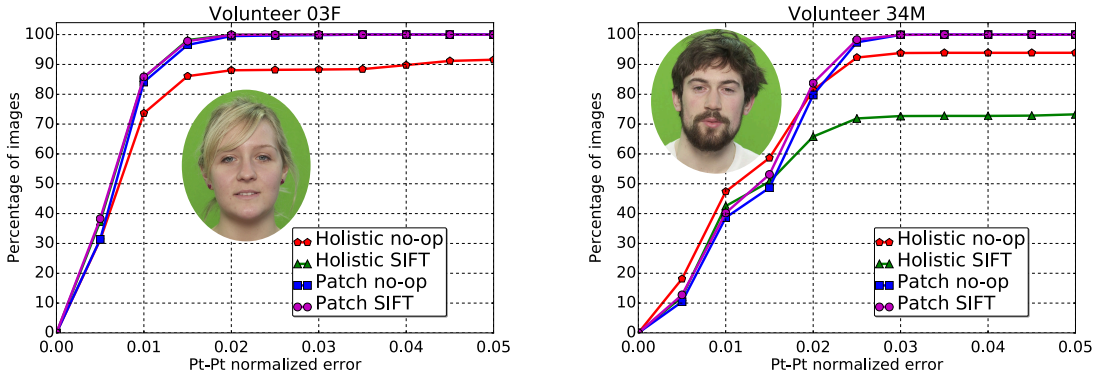


Figure 3.3: *AAM fitting convergence using global face models (trained on the full set of volunteers)*



(a) *volunteer 03F* **(b)** *volunteer 34M*

Figure 3.4: *AAM fitting convergence using person-specific models*

tained almost identical trends when considered the fitting performance only on the frames above 0.94 confidence.

Figure 3.3 shows the proportion of frames fitted with an error lower than a certain value displayed on the horizontal axis, using the global face models, while Figures 3.4a-3.4b show the same information using person-specific AAMs of two volunteers. The two speakers modelled individually were drawn from the top/bottom 10 performers in Harte and Gillen (2015), where volunteer *03F* was considered easier to lipread than *34M*, who had a full beard and moustache.

The Holistic models were outperformed by the Patch models in almost all cases, with the exception of volunteer *03F* where *Holistic SIFT* managed to match them, although for volunteer *34M* it did not cope well with the facial hair. Both Patch models achieved a convergence rate above 95% for an error of 0.02 and were almost indistinguishable in performance, demonstrating their robustness not only for fitting to unseen frames, but also when trained from less perfect landmarks that were estimated automatically.

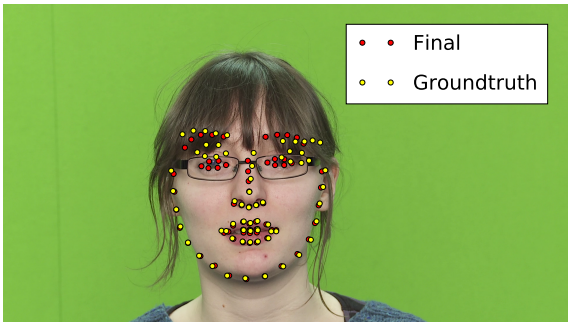


Figure 3.5: *Landmark correction for volunteer 05F wearing glasses and with the eyebrows occluded*

In most cases, AAMs were able to improve the pre-trained OpenFace estimates where the confidence score was low. One such example is shown in Figure 3.5, where the eyes and the eyebrows landmarks were corrected for volunteer *05F* wearing glasses with the eyebrows not visible. This leads to a better initialisation of the AAM fitting algorithm for faces that are otherwise more challenging to annotate. We conclude that the coarse landmark estimation from a generic model, followed by person-specific fine-tuning using AAM fitting, represents an efficient automatic pipeline for obtaining accurate facial landmarks.

3.5 Hidden Markov Model pipeline

Extracting sequences of visual representations for every video file in a dataset is only the first part of building an automatic lipreading system. The feature sequences are densely sampling the input video signal and do not expose the higher level speech message or its semantic segmentation. Speech recognition requires the learning of a mapping function translating the sequence of feature vectors into the sequence of output units, namely visemes in our case.

For training and evaluating HMMs in this chapter, we will leverage an existing implementation from the HTK toolkit of Young et al. (2015).

3.5.1 HMM training

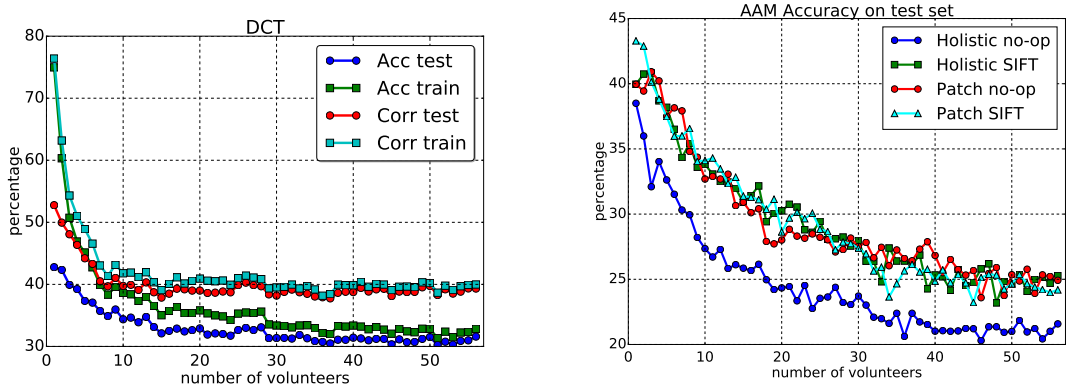
Our viseme recogniser was implemented in HTK 3.5, following the procedure described in Harte and Gillen (2015) as close as possible. For each of the 12 viseme classes we have built 3-state left-to-right HMMs with mixtures of 20 diagonal covariance Gaussian densities per state, initialised in flat start mode with *HCompV*. Additionally, for the silence viseme state we have added backward and skip transitions. Finally, we have applied 5 runs of embedded training using *HERest* for every increment of the mixture components. The reported correctness

and accuracy results are computed using *HResult* between the ground-truth transcriptions provided with TCD-TIMIT and the predicted ones. No language model was used. This allows a comparison of the raw lipreading decodable knowledge of the feature sets.

3.5.2 Viseme recognition performance with HMMs

We will first analyse the viseme recognition results obtained by training HMMs in a speaker-dependent partitioning, hence using 67 training sentences from each volunteer and testing on their remaining 31 unseen sentences. The predicted viseme sequence is computed using the HTK tool *HVite*.

To ablate the effect of the training data size on the system performance, we train multiple HMM systems on multiple subsets of the TCD-TIMIT train partition. We plot in Figure 3.6a the correctness and accuracy scores returned by *HResults* with the DCT features for an increasing number of volunteers added to the training set (ordered by their alphanumeric IDs). The accuracy on the entire set of volunteers (31.59%) is 3% below the one obtained in Harte and Gillen (2015). An increase of 1-2% was possible when we interpolated the features to double the frame rate and used 4-state HMMs, but we reverted to the original settings to have a fair comparison with the AAM features.



(a) *Correctness and Accuracy for DCT features* **(b)** *Accuracy scores for AAM features*

Figure 3.6: *HMM evaluation on the Speaker-Dependent partition of TCD-TIMIT*

In Figure 3.6b we show the accuracy obtained using AAM-based features using the same HMM framework. As anticipated, the *Holistic no-op* model has the lowest accuracy, since fitting converges on less than 60% frames on average. The other three models perform similarly, yet reaching an accuracy of $\approx$ 25% on the entire test set, considerably lower than DCT. It is important to note that our AAM features do not include temporal derivatives, as in the case of DCT repre-

sentations. This decision was made in order to limit the AAM vector size, which is already $\approx$ 44% higher than the DCT one appended with the first and second derivatives. As a consequence, the comparison between the two feature sets is bounded by the effectiveness of HMMs at modelling such short term dependencies in the input signal, and does not necessarily portray an intrinsic / stand-alone value of each feature set.

We repeated the experiment with features extracted using the two part models, *chin* and *lips*, on a subset of the first 33 volunteers, following the process described in Section 3.4.2. The results are displayed in Figure 3.7, showing the *chin* model to perform only marginally better, although the decreasing trend remains. This small increase comes with the cost of doubling the processing time, as it requires a cascade of two fittings.

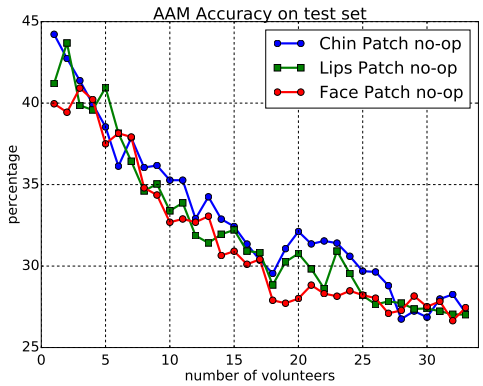


Figure 3.7: *HMM performance of the* chin*,* lips*, and full* face *AAM models*

Speaker-specific models

In order to see how much the quality of the AAM impacts the viseme recognition accuracy, we tested the case of person-specific AAMs for the two volunteers described in Section 3.4.3. If there was a problem with the global model, we should notice a significant increase in accuracy when switching to person-specific models. Table 3.3 shows the viseme recognition results obtained with both specific and global models for these two speakers, along with the DCT baseline. We could not find a significant advantage of the person-specific models, hence at this stage it would not be useful to attempt adapting a global AAM to particular faces in order to gain a performance boost.

3.5.3 Discussion

In this section we have explored the performance of hand-crafted visual features for lipreading large vocabulary fluent speech with a traditional HMM framework.

Table 3.3: *Recognition performance for person-specific models versus the global models. Since there was less data available for individual speakers, the highest values were obtained on average with 14 Gaussian mixture densities*

Speaker $\rightarrow$	**03F**		**34M**	
$\downarrow$ **Part**	**Corr**	**Acc**	**Corr**	**Acc**
Specific AAM				
face	51.84	42.62	51.63	43.24
chin	52.62	45.24	53.11	40.18
lips	50.87	43.98	52.62	43.14
Global AAM				
face	53.11	44.66	51.04	41.76
chin	53.88	43.50	51.53	41.95
lips	52.43	44.27	53.70	42.15
DCT	54.66	46.80	47.88	39.68

We first computed DCT-based features, reaching a similar result with Harte and Gillen (2015). Then we trained several AAMs using an automatic procedure and fitted them to each video frame to obtain the AAM-based features.

A first finding is that AAM features do not outperform the DCT ones in an identical recognition framework. This has been reported before on IBM ViaVoice (Neti et al., 2001; Potamianos et al., 2003). This dataset has 290 subjects and over 50 hours of speech. However their approach was to rescore audio-only lattices with visual unit HMMs. Their scenario therefore bypassed the issue of using visual features to find the viseme boundaries. On the other hand, the study of Lan et al. (2009) found AAM better than DCT on a lipreading task with a small vocabulary of 51 words, where word-level HMMs were used. Later work of Lan et al. (2010) reported results on a corpus of 12 speakers, each speaking 200 sentences from a vocabulary totalling 1000 words. Again AAM outperformed DCT, but the approach made use of Linear Discriminant Analysis (LDA) requiring frame-aligned viseme labels, while the facial landmarks were obtained semi-automatically from person-specific trackers. Potamianos et al. (2003) make use of further refinements to both AAM and DCT features, including LDA projections and *maximum-likelihood linear transform* (MLLT) rotations. Although they could have positively impact the results, such techniques have not been pursued in our work. As will become clearer in the following sections, our exploration of viseme units for lipreading will follow the direction of end-to-end neural networks, which shift the focus from feature engineering to learning algorithms with weaker prior assumptions. In another study, K. Paleček and Chaloupka (2013) used speaker-specific normalisation that makes the results less comparable. This is the most

comprehensive comparison between DCT and state-of-the-art AAM on an open vocabulary lipreading task that we are aware of.

The reported results are obtained using a visual speech model only, allowing raw performance comparison of the extracted features. Adding a simple bigram language model improves the viseme recognition accuracy by up to 10% for the AAM features, and 3% for DCT, narrowing the performance gap to less than 1%.

Modelling a subset of the face has only shown minor improvements of the recognition accuracy. The *chin* model seems to have a slightly better advantage versus the *lips* one, and this could be explained by two factors. The extra iterations of the part model ensured a more accurate fitting where there were more control points available. Also, the chin area contains additional visemic information, as speech articulators are not limited to the lips region.

A notable conclusion about AAMs is that the *Patch* models, especially when combined with SIFT image descriptors, are able to achieve a much lower fitting error and therefore a higher recognition accuracy than the traditional *Holistic* ones that have been used so far in lipreading.

There are multiple reasons for the relatively low lipreading accuracy with either feature set. One concerns the representation power of the features that allows the easy discrimination between different viseme classes. The second concerns the effectiveness of HMMs at modelling the sequence of visual speech representations. Investigating the latter cause is more challenging since it strictly depends on the performance of the upstream feature extraction task. Besides, as many other learning based systems, the explainability of HMMs quickly fades away in higher dimensions, therefore studying the Gaussian mixture densities does not scale up well with an increase of the feature size. For these reasons, we choose to focus on more promising approaches in the next section. First, we will study the performance of the DCT and AAM feature sets when the HMM framework is substituted with the more powerful sequence to sequence neural architecture. In ASR, a LSTM-based seq2seq model was shown in Chiu et al. (2018) to outperform the state of art HMM at recognising speech from voice search and dictation after being trained on 10,000 hours of speech recordings. Such a neural model makes fewer assumptions about the modelled signal than a HMM, and has consequently seen a growth in popularity as speech data became available for research in the recent years. Secondly, the neural seq2seq model facilitates end to end learning, which enables the learning of optimal representations directly from the raw video data. As it is not fully clear yet what are the ideal properties of

a visual representation for lipreading and what is the right filtering approach for the irrelevant data, it appears that learning with minimal assumptions from raw video frames is the most suitable approach for lipreading. In the next section we will compare our handcrafted feature sets against the end-to-end learnt representations within the same neural sequence to sequence architecture, showing that even for the relatively small TCD-TIMIT dataset it is advantageous to use the second approach.

3.6 Lipreading with sequence to sequence neural networks

Finding visual features and suitable models for lipreading tasks that are more complex than a well-constrained vocabulary has proven challenging. In this section we take a look at lipreading models using sequence to sequence neural networks in order to make a direct comparison with the HMM framework in Section 3.5. We will first reuse the DCT visual features to compare only the sequence modelling component of the lipreading system. We will then make use of Convolutional Neural Networks discussed in Chapter 2 to learn visual representations and evaluate the performance of an end to end trainable lipreading model with viseme units.

3.6.1 Sequence modelling

Motivated by the discussion in Section 2.5.1, we will use the neural sequence to sequence framework with attention for modelling the sequence of visual features. We used the attention mechanism variant proposed by Luong et al. (2015), as it outperformed the variant of Bahdanau et al. (2015) in our initial benchmark.

Training and decoding

In the training stage, the entire transcription is available to the decoder. A learned vector representation of the ground-truth symbol, called *embedding*, is fed to every decoding time step. Additionally, Bengio et al. (2015) proposed to randomly swap the ground truth token with the previously decoded one with a given probability in order to increase the robustness of the network to recover from mistakes at inference, process known as *scheduled sampling*. This training process implies that the predicted output transcription has an identical length with the ground-truth transcription, thus a cross-entropy (CE) loss function can be applied. In the evaluation stage, the decoder is likely to produce a transcription

of a different length than the ground truth, and a separate evaluation metric is needed. Hence, we evaluate the quality of the prediction by computing the Levenshtein edit distance with respect to the ground truth targets.

Combining the cross entropy loss with the Connectionist Temporal Classification (CTC) loss of Graves et al. (2006) could lead to several benefits, as pointed by Kim et al. (2017). First, the CTC loss helps the encoder to better focus on the input signal, as the implicitly learnt language model on the decoding side exhibits a strong early influence in training. In addition, the encoder is encouraged to learn representations that contain more decodable knowledge of the grapheme targets, as a CTC sub-network predicts a label for each input frame. This sub-network is a one-layer neural network having the size of our viseme alphabet plus one, uses a softmax activation function, and is trained jointly with the CE loss. Since Kim et al. (2017) obtained the best results for a mixing coefficient of 0.2 for the CTC loss, we only consider this case in our experiments.

3.6.2 Learning visual representations

For our lipreading task, we explored several use cases for CNN architectures. First, we considered the same 36x36 grey-scale mouth ROIs used in Section 3.3, and also a colour 36x36x3 RGB version, and a larger 64x64x1 grey one. These 2D CNNs have 4 layers with 16, 32, 64 and 128 feature detectors respectively, a small 3x3 convolution kernel and rectified linear activations. After the first layer, our convolutions use a stride of 2 to reduce the dimensionality. The activations of the last layer are flattened and fully connected to a new layer of 128 units, producing our frame-wise feature vectors. To quantify the impact of the implicitly learnt language model, we also present the results in the absence of a visual stream by replacing the features with zeros.

3.6.3 Experimental setup

We performed our experiments on the same speaker-dependent partition of TCD-TIMIT as in Section 3.5.

Network details

Both the encoder and the decoder of our sequence to sequence model use two LSTM layers with 128 units each. We also test a single layer bidirectional LSTM (BiLSTM) variant (Graves and Schmidhuber, 2005; Schuster and Paliwal, 1997), processing the sentence both in the forward and backward directions, while maintaining the same number of parameters as the two layer unidirectional LSTM

Table 3.4: *Lipreading viseme accuracy on TCD-TIMIT. The right column shows the number of iterations needed to reach convergence (or* nc *for* no convergence*).*

System	Accuracy	Iters
A. DCT + HMM baseline (Section 3.5)	31.59%	-
B. AAM + HMM baseline (Section 3.5)	25.28%	-
C. Eigenlips + DNN-HMM (Thangthai et al., 2017)	46.61%	-
D. zeros + LSTMs	45.87%	160
E. DCT + LSTMs	61.52%	250
F. DCT + BiLSTMs	60.72%	180
G. DCT + LSTMs w/o attention	48.29%	270
H. DCT + LSTMs + CTC loss	61.18%	180
I. CNN + LSTMs	-	*nc*
J. CNN + BiLSTMs	**66.27%**	400
K. CNN + LSTMs on RGB + CTC loss	66.20%	150
L. CNN + LSTMs on 64x64 + CTC loss	-	*nc*
M. CNN + LSTMs + CTC loss	64.61%	260
O. ResNet CNN + LSTMs	71.21%	120

model. Decoding was performed using a beam search strategy of width equal to 4.

We used several regularisation methods. We applied dropout to the recurrent cells (Zaremba et al., 2014), keeping the inputs, the states and the outputs with a probability of 0.9, and also to the activations of the CNN layers with a probability of 0.5. We also used weight decay on the recurrent and the convolutional weights, scaled by 0.0001 and 0.01 respectively before being added to the total loss. We enable gradient norm clipping with a threshold of 10.0 (Pascanu et al., 2013) and we also clip the LSTM cell states between -10.0 and 10.0 prior to cell output activation.

3.6.4 Speaker-dependent lipreading on TCD-TIMIT

The results of our study are shown in Table 3.4. We first observe a massive improvement over the HMM baseline. However, a large part is owed to the implicitly learnt LSTM-based language model which overfits on small datasets, as hypothesised in Bahdanau et al. (2016) and indicated by system **D**. In comparison, a bi-gram language model only increased the accuracy of the HMM system **A** to 35%. Looking at the predictions, we note that the model quickly learns to output only two visemes in an interleaved pattern, surrounded by the silence viseme delimiting the start and the end of each sentence. These correspond to the *Lips relaxed, narrow opening* and *Tongue up or down* classes (explained in Table 3.5), and together they account for 52.56% of the occurrences in TCD-TIMIT scripts.

Since the scripts are phonetically balanced, this viseme distribution only reflects a natural speech pattern, making it difficult to sample a flat target distribution. We identified this matter in all our experiments, typically taking at least 100 training iterations before the predictions start to look diverse. This suggests that the internal language model might slow down training convergence and reduce generalisation, while the use of viseme targets may further magnify this effect due to the relatively small inventory size (only 12 visemes in our case).

The use of DCT features with a Seq2seq model led to a substantial improvement over the best viseme lipreading system of Thangthai et al. (2017) on the TCD-TIMIT dataset. There is a noticeable boost in convergence speed from unidirectional to bidirectional LSTMs, yet it does not always translate into higher accuracy, as demonstrated by **E** and **F**. This could be explained by the fact that two single-layer networks are less powerful than a single two-layer variant. We tried another variant of two-layered bidirectional LSTM which did not improve the performance.

Also previously noted by Chung et al. (2017), the system **G** without an attention mechanism could not learn meaningful patterns from the input, predicting a similar transcription for most sentences. This could imply that either the temporal information vanishes during encoding, or the decoding process relies heavily on the language model. The attention-based system **E** alleviates these aspects, obtaining an absolute 13.23% improvement over this variant.

In Figure 3.8 we display a typical alignment learnt by the attention sub-network in the decoder, and we notice that each target viseme generally uses a narrow range of the encoder representations. Together with the monotonic trend of the alignment, this suggests that a local conditioning within the decoder-encoder attention mechanism may represent a good inductive bias that could be taken advantage of to improve the learning speed. On many alignments produced by the decoder we could observe that they tend to become fuzzy towards the end of the sentence. One possible explanation is that the thought vector may end up learning to summarise mostly the recent past, and the attention is only needed to assist the decoding of earlier events.

The use of CNN representations led to an additional $\approx$5% absolute improvement over the best performing DCT-based system, as is the case with system **J**. In this case, using BiLSTMs was crucial to prevent the system from getting stuck in a local minimum, as with **I**. However, our experiments with higher resolution images (64x64 pixels) did not reach convergence, showing the limits of a shallow CNN architecture.

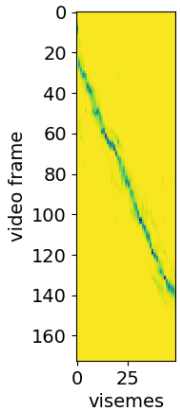


Figure 3.8: *A typical alignment learnt by System **J***

The use of the joint CTC-CE loss function significantly accelerates the training process. However, in our case, the test set accuracy was lower than for the cross-entropy loss function alone. The impact of the CTC loss may be twofold. It enforces a frame-wise classification on the encoder's outputs, which could lead to better gradients for the CNN layers. This is demonstrated by the performance achieved with systems **K** and **M**, which could not converge without the additional CTC loss. On the other hand, the two loss functions could have competing requirements for the state representation, and a proper weighting may be vital for optimal performance, as shown in Kim et al. (2017).

We have compared the viseme confusion matrices of systems **A**, the DCT + HMM baseline, and **J**, our top performing DNN-based lipreading system (at a later stage we developed system **O**, which represents an incremental improvement of the visual front-end, and will be discussed in the next section). Table 3.5 shows the relative performance increase across the viseme classes for these two systems. The table also shows the TIMIT phonemes mapped to each viseme class and their visibility, or ease of observation for a human. The improvement from **A** to **J** is ubiquitous with the exception of a single viseme corresponding to the *Lips rounded* shape. This viseme is most frequently confused with the *Lips relaxed narrow opening* viseme, suggesting that it is difficult even for the CNN to learn features that disambiguate them. Lower improvements are seen for *Lips forward* and *Tongue back*. The frontal view used as input does not capture any depth information, however the database includes a second view at an angle of 30° which could be useful for such visemes.

A general conclusion is that the Seq2seq model greatly outperforms HMM and hybrid DNN-HMM systems even without CNN-based feature extraction. The fully neural architectures achieved the highest accuracy in our experiments. This finding adds more empirical evidence to the discussion in Section 3.5.3 regarding

Table 3.5: *Viseme accuracy of the best DNN system **K** and relative change from HMM baseline (**A**). Visemes sorted by decreasing visibility.*

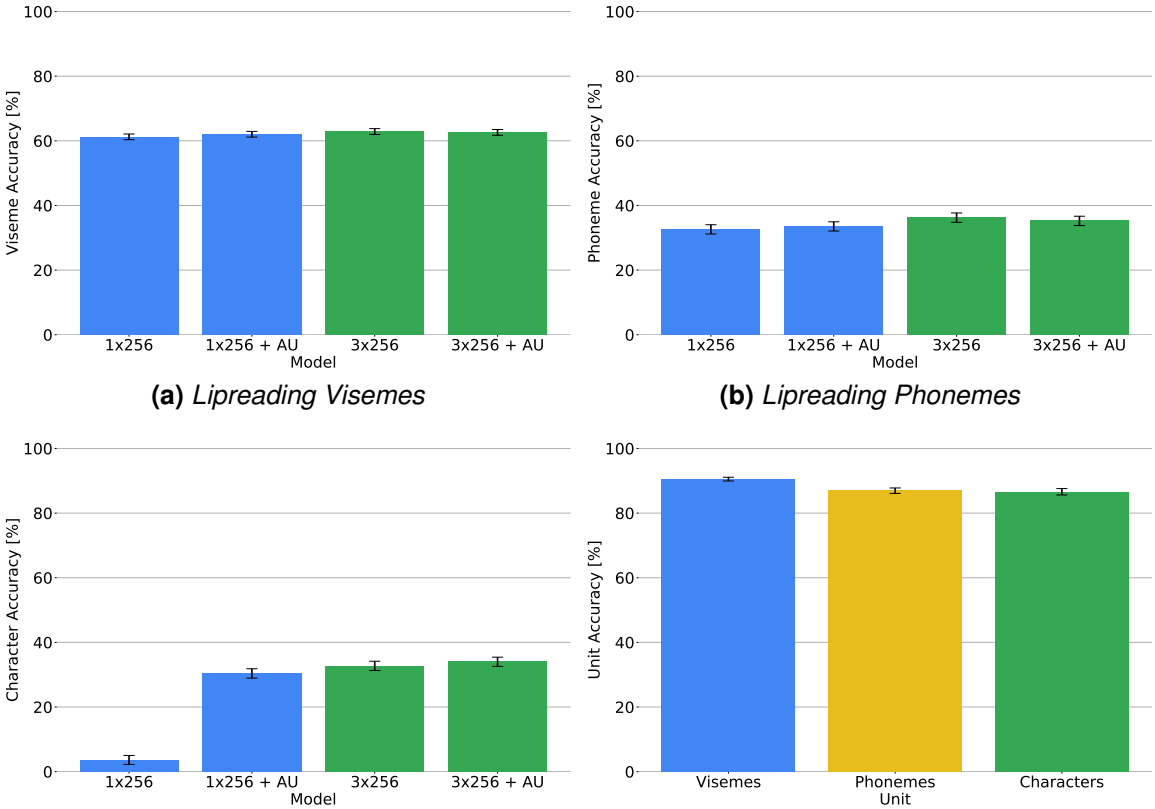
Viseme	TIMIT Phonemes	Accuracy K [%]	△ Accuracy K - A [%]
Lips to teeth	/f/ /v/	85.6	21.25
Lips puckered	/er/ /ow/ /r/ /q/ /w/ /uh/ /uw/ /axr/ /ux/	83.4	50.81
Lips together	/b/ /p/ /m/ /em/	94.8	30.40
Lips relaxed moderate opening to lips narrow-puckered	/aw/	45.7	25.90
Tongue between teeth	/dh/ /th/	58.4	27.79
Lips forward	/ch/ /jh/ /sh/ /zh/	65.4	18.26
Lips rounded	/oy/ /ao/	31.6	-8.41
Teeth Approximated	/s/ /z/	81.6	52.24
Lips relaxed narrow opening	/aa/ /ae/ /ah/ /ay/ /ey/ /ih/ /iy/ /y/ /eh/ /ax-h/ /ax/ /ix/	95.6	73.50
Tongue up or down	/d/ /l/ /n/ /t/ /el/ /nx/ /en/ /dx/	84.8	56.17
Tongue back	/g/ /k/ /ng/ /eng/	63.2	24.41
Silence	/sil/ /pcl/ /tcl/ /kcl/ /bcl/ /dcl/ /gcl/ /h#/ /#h/ /pau/ /epi/	93.6	0.21

the difficulty of modelling the lipreading task with HMMs.

3.6.5 Larger scale lipreading on LRS2

We have seen in the previous section that sequence to sequence neural models can easily overfit a dataset of the size of TCD-TIMIT by exploiting the highly repetitive patterns of the transcript. Currently the largest publicly available audio-visual speech dataset is LRS2, which, unlike the laboratory recorded TCD-TIMIT, contains uncontrolled illumination conditions and challenging viewing angles of the speakers. This joint increase in both visual task difficulty and transcript diversity on LRS2 allows a more accurate picture of the real world performance of such models.

Our original motivation for the viseme units in lipreading consisted in reducing the targets ambiguity and converting the task to a well posed problem. Given the increased amount of training data, in addition to the viseme targets used so far we will also explore phoneme and character targets. Character units have been initially explored by Assael et al. (2016) on a dataset with a restricted vocabulary of only 51 words, but with an alternative neural architecture that does not exploit the patterns in the target signal. Later work of Shillingford et al. (2019) showed that on larger datasets it is beneficial to resort to phoneme abstractions, coupled with a post-processing step based on finite state transducers to produce the



(a) *Lipreading Visemes* **(b)** *Lipreading Phonemes*

(c) *Lipreading Characters* **(d)** *Audio-only performance on clean speech*

Figure 3.9: *Lipreading performance on LRS2 using LSTM-based sequence to sequence models with multiple target units. For comparison, Figure 3.9d shows the equivalent audio-only performance with a Transformer model. The error bars denote the 95% confidence interval of the mean error, i.e.* $\pm 1.96\sigma/\sqrt{1222}$*, where* σ *is the standard deviation of the error on the test set of 1222 sentences.*

word-level output. A direct comparison between all three units for the lipreading task on LRS2 allows us to quantify the effect of target clustering on the decoding performance.

We re-train multiple seq2seq lip-reading models on LRS2, varying the size of the encoder stack. We increase the hidden state size to 256 units, and test both single-layer and three-layer encoders, further labelled as *1x256* and *3x256* respectively. The decoder uses a single LSTM layer of 256 units in all our experiments. For each of the two encoder sizes, we also investigate the effect of a visual regularisation loss based on regressing action units from the visual representations, which will be introduced in Chapter 4 in the context of audio-visual speech recognition and shown to have crucial role in learning good visual representations for that task. These regularised models are further labelled as *+ AU*. The visual convolutional front-end was updated to make use of residual connections as in He et al. (2016b). We plot in Figure 3.9 the performance of all four lipreading models for each of the viseme, phoneme, and character targets.

Increasing the size of the encoder only has a marginal effect on the decoding performance. Whereas the performance with phoneme and character units is very similar, there is an absolute difference of 28.87% between the best performing viseme and character models. The Action Unit regularisation loss does not have a significant impact on the overall lipreading performance, unlike on the AVSR task in Section 4.3.3. However, not reported here, we found the AU loss to have a major effect when we later swapped the LSTM layers with Transformer ones, although the final performance was similar to the one of the LSTM models.

The best viseme decoding accuracy on LRS2 we obtained, of 62.87%, is 3.4% below the lipreading result on the speaker-dependent partition of TCD-TIMIT. A later experiment on TCD-TIMIT with the updated residual learning visual front-end used here further improved the accuracy to 71.21% on this dataset (System **O** in Table 3.4). This difference quantifies the joint effect of hardened visual conditions and lower sentence predictability. Disentangling these two variables would involve the costly operation of recording a large scale audio-visual speech dataset in laboratory conditions similar to TCD-TIMIT.

With a related network based on the Transformer architecture instead of the LSTM one, which was used for our experiments in Section 4.3.9, we also analysed the performance at decoding these units from the audio modality. As seen in Figure 3.9d , the system can decode viseme sequences from audio representations with an accuracy of 90.52% on LRS2, whereas on phoneme and characters this figure is slightly lower, of 86.94% and 86.6% respectively. This result shows that the video model is unable to match the performance of the audio one at decoding viseme units that are specifically designed to denote visually distinguishable units. One cause could be the imperfect definition of visemes, which is supported by the variety of phoneme to viseme mappings proposed so far. However, a more likely cause is the relatively higher difficulty of identifying speech units in a video, with possible key factors being the low image resolution, the low sampling rate of only 25 frames per second, the partial views of the articulators, or simply the limited lip articulation of some speakers.

3.7 Conclusion

In this chapter we have studied the performance of traditional and more recent lipreading systems with viseme units. First, we compared two of the most used engineered features in lipreading, namely DCT coefficients and AAM parameters, within a GMM-HMM framework. We found the DCT coefficients to work better, although the viseme decoding accuracy was only 31.59%. Next, we re-

placed the GMM-HMM with an LSTM-based sequence to sequence modelling framework, and obtained an accuracy of 61.52% for decoding the same viseme units. This represents a relative performance increase of 94.74% obtained with a generic model based on neural networks. Further replacing the DCT features with learnt visual representations led to a decoding accuracy of 71.21%, which is only a 15.75% relative increase compared to the neural model using DCT features. This finding suggests that visual representations optimised for lipreading may be learnt from a relatively low amount of data. Additionally, we see that modelling the temporal dynamics of these visual features has a much stronger impact on the overall accuracy. This may be analogous to ASR, where Purwins et al. (2019) notes that the representations learnt from raw audio signals bring diminishing gains over spectral auditory features even with the latest neural speech models.

Observing that the neural model exploits the highly repetitive label patterns of the relatively small TCD-TIMIT, we analysed the performance of this model on the larger LRS2 corpus, seeing a lower viseme decoding accuracy of 61.52%. We have also seen that the same viseme targets can be decoded from the audio modality with an accuracy of 90.52%, with the large bias being attributed to the imperfect definition of visemes and to several noise sources specific to the visual modality. A drawback of choosing visemes as intermediate output units over phonemes or characters when building a large vocabulary visual speech recognition system is the requirement of a viseme-level lexicon. Auer and Bernstein (1997) found that approximately 54% of words in English remain distinct following their translation to 12 viseme classes. Our investigation was not concerned with measuring a word-level lipreading accuracy. It only focused on decoding a sequence of distinctive visual units of speech under either controlled or uncontrolled visual conditions.

One general conclusion is that the engineered features may not be so relevant to lipreading in the context of abundant speech data and expressive neural models. A limitation of AAMs is the large space of hyper-parameters to be manually tuned by experts, increasing the prospect of errors and their downstream propagation. Furthermore, AAMs are directly optimised to reconstruct the shape and appearance of an entire face, which in turn may put a lower emphasis on the subtle lip articulations that are relatively more important for lipreading. In the case of DCT coefficients, it is not entirely clear that only the low frequency ones are relevant to lipreading. As the coefficients are commonly selected with low-pass filtering, there is an inherent trade-off between the coarseness of the transform and the feature dimension. On the other hand, the trainable filters of the CNN architec-

ture allow the learning of an optimal transformation for the processing of visual speech, although a good regularisation of these models is still necessary. In this context of representation learning, we may then ask what structure of the visual signal can be manually filtered out from the raw pixel representation without affecting the task accuracy. As seen in Section 3.6.4, colour does not seem to play a substantial role in lipreading, and may be a potential candidate for filtering in order to improve generalisation especially on small datasets.

Obtaining a better performance at modelling visemes as opposed to phonemes and characters raises the following question: are visemes a necessary abstraction? One could build a viseme-level lexicon of canonical articulations for the English vocabulary, but with a limited range of applications. The recent work of Müller et al. (2019) suggests that neural networks have the ability to automatically cluster similar output classes. Such a property may already allow a neural network to learn relationships between the target characters or phonemes in the high dimensional embedding space without being manually enforced, as in the case of visemes.

Our goal with this chapter was to investigate what a good visual speech representation should look like, using the lipreading task as an experimental proxy. It is not fully clear if lipreading, i.e. learning to map sequences of visual speech inputs onto sequences of symbolic units, may be the optimal way to extract visual representations of speech when the task of interest is AVSR. This contrasts with one training strategy used in audio-visual speech where the entire audio modality is dropped out with a certain probability to discourage the over-reliance on the dominant audio modality (Chung et al., 2017; Ngiam et al., 2011). In the absence of the audio modality, such training strategy would essentially force a model to lipread visually ambiguous units. Due to the ill-conditioning of the task, a more natural use of the visual modality in speech is to play an assistive or complementary role for the audio modality, rather than a competing one. This observation represents the foundation of our audio-visual modelling strategy *AV Align* presented in the next chapter.

4 AV Align

4.1 Introduction

In the previous chapter, we investigated how well traditional and more recent lipreading models are able to decode symbolic units in fluent speech. Our experiments have shown that both speaker independent viseme lipreading on LRS2 and speaker-dependent viseme lipreading on TCD-TIMIT are difficult tasks even with the latest developments in neural networks, having obtained an absolute accuracy of 62.87% and 71.21% respectively with our best systems. Despite reducing the label ambiguity by clustering similar phonemes into viseme units, hence sacrificing the overall meaning, both the LSTM and Transformer models struggle to accurately decode the sequence of visemes on previously unseen speakers. However, the task of lipreading implicitly assumes that there is substantial symbolic knowledge to be decoded from the video modality under the form of graphemes from a low dimension alphabet. This assumption may be too restrictive since the visual modality is inherently limited in linguistic information as we discussed in Section 1.1. The findings of Shillingford et al. (2019) reinforce this hypothesis, reporting that even professional English speechreaders achieve high word error rates on their challenging dataset. A natural relaxation of this restriction is audio-visual speech integration, which allows the visual modality to play a descriptive role of what is being seen, without the guess work entailed in lipreading.

In this chapter, we explore the problem of multimodal fusion for the speech recognition task. Recently proposed DNN-based multimodal systems encode each modality separately, and the representations are fused when decoding (Afouras et al., 2018b; Chung et al., 2017; Petridis et al., 2018b). Instead, motivated by our conclusions from the lipreading experiments in Chapter 3, we propose a new method *AV Align* where the acoustic representations of speech are altered by the visual representations during a multimodal *encoding* process, before decoding starts. In other words, what the system sees influences what it hears. Another

distinct feature of *AV Align* is that the alignment is done at every acoustic frame, allowing the encoder representations to be partially reconstructed from the visual signal and limiting the propagation of uncertainties at future timesteps. This allows the learning of the natural asynchrony between sounds and lip movements. Being able to visualise the audio-visual alignments makes the architecture interpretable by design. *AV Align* is a flexible strategy that does not require the features from the two modalities to have identical sampling rates, as in (Afouras et al., 2018b; Petridis et al., 2018b). These properties make *AV Align* an attractive alternative to both traditional time-aligned feature fusion (Potamianos et al., 2003), and dual attention decoding (Chung et al., 2017), and we will explore in greater detail the strengths and weaknesses of these systems.

The *AV Align* architecture suffers from the same aforementioned convergence problem of the visual front-end discussed in Section 2.6.2, and confirmed by the experiments in Section 4.3.3. To address it, we regress Action Units (AUs) (Ekman and Rosenberg, 1997) from the visual representations and introduce an auxiliary loss function on the visual side which is jointly optimised with the character sequence loss of the decoder. Our strategy to regularise the network with a secondary AU loss addresses the convergence problem and enables a performance boost of the audio-visual system on the challenging LRS2 dataset (BBC and University of Oxford, 2017). We demonstrate that it is possible to efficiently train a DNN-based AVSR system with a mere 30 hours of audio-visual data.

In this chapter we show that the AVSR strategy *AV Align* implicitly discovers alignments with a monotonic trend between the acoustic and the visual speech representations. We find this to be a necessary condition to improve the speech recognition error rate of the multimodal system. In addition, we investigate the source of divergence on the challenging LRS2 dataset, and propose an architectural improvement to encourage the convergence of the visual front-end in training. Our improved system can now learn speaker and speech independent representations on uncontrolled recording conditions and vocabulary. We show how our architectural improvement also applies to the popular *Watch, Listen, Attend, and Spell* (WLAS) network of Chung et al. (2017), and also to simple feature fusion, effectively helping the system learn visual representations and substantially improve the speech recognition performance when compared to the architectures unaltered by the proposed improvement. AV Align is the first neural network based approach in the AVSR field which attempts to explicitly model the inherent alignment between the auditory and visual modalities in speech. As opposed to the dual attention mechanisms of WLAS in the decoder, AV Align as-

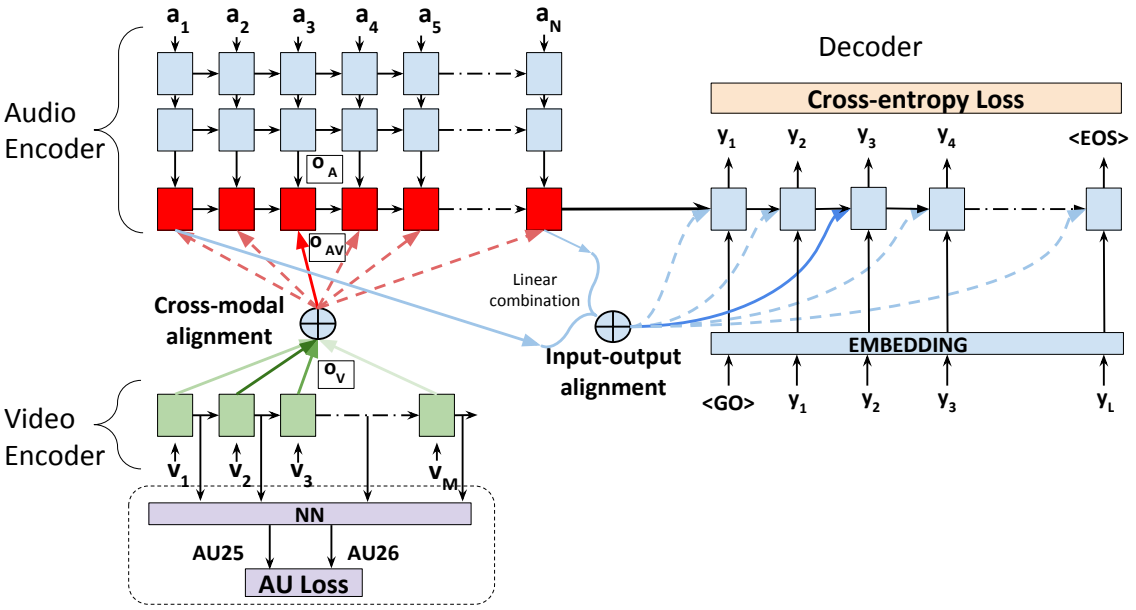


Figure 4.1: *The AV Align strategy. The top layer cells of the Audio Encoder take audio representations from a stack of LSTM layers (o_A) as inputs and attend to the top layer outputs of the Video Encoder (o_V, only one layer shown), producing the cross-modal alignment. The Decoder receives the fused Audio-Visual representations (o_{AV}), producing an input-output alignment through a second attention mechanism. We amend this architecture with an Action Unit prediction network (*NN + AU Loss*) motivated in Section 4.2.2. Dashed lines depict inactive states in a hard selection process, whereas shaded lines stand for a soft selection mechanism.*

sumes no decodable lexical knowledge in the visual modality and is designed to produce visual cues assisting the potentially noisy audio channel. This approach has further potential in fields outside AVSR that require an alignment between modalities that have time-varying contributions to the overall task.

4.2 Audio-Visual alignment and fusion

4.2.1 AV Align

Here we introduce the audio-visual speech alignment and fusion strategy *AV Align*, illustrated in Figure 4.1. Technically, it can be considered as the original sequence to sequence network with attention (Bahdanau et al., 2015) extended with an additional encoder and explicitly modelling the cross-modal correlations. Given a variable length acoustic sentence $A = [a_1, a_2, \dots, a_N]$ and its corresponding visual track $V = [v_1, v_2, \dots, v_M]$, we transform the raw input signals into higher level latent representations using stacks of LSTM layers (further denoted in Fig-

ure 4.1 by $o_A = [o_{A_1}, o_{A_2}, ..., o_{A_N}]$ and $o_V = [o_{V_1}, o_{V_2}, ..., o_{V_M}]$):

$$o_{A_i} = \text{LSTM}_A(a_i, o_{A_{i-1}}) \tag{4.1}$$

$$o_{V_j} = \text{LSTM}_V(v_j, o_{V_{j-1}}) \tag{4.2}$$

Next, one additional LSTM layer is stacked on top of the last acoustic LSTM layer, taking as input o_A. Its internal state h_i is correlated with all the entries in o_V at every audio timestep i to compute the visual context vector c_{V_i}:

$$h_i = \text{LSTM}_{AV}([o_{A_i}; o_{AV_{i-1}}], h_{i-1}) \tag{4.3}$$

$$\alpha_{ij} = \text{softmax}_i(h_i^T \cdot o_{V_j}) \tag{4.4}$$

$$\text{where } \text{softmax}_i(\mathbf{x}) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

$$c_{V_i} = \sum_{j=1}^{M} \alpha_{ij} \cdot o_{V_j} \tag{4.5}$$

Finally, the visual context vector c_{V_i} and the current LSTM hidden state h_i are concatenated and fed to a fully-connected neural network having n output units to produce at every timestep the fused representation o_{AV_i}:

$$o_{AV_i} = W_{AV}[h_i; c_{V_i}] + b_{AV} \tag{4.6}$$

$$\text{where } W_{AV} \in \mathbb{R}^{n \times 2n}, b_{AV} \in \mathbb{R}^n$$

Every input to the attention-enhanced LSTM layer is concatenated with the fused representation from the previous timestep $o_{AV_{i-1}}$, as seen in Equation (4.3). Both o_{AV_0} and h_0 are initialised with zeros.

The rest of the network is a character-level LSTM decoder ($LSTM_D$) that attends to the enhanced audio-visual representations (o_{AV}) instead of the audio-only ones (o_A), and outputs a variable length character sequence of posterior probabilities $p = [p_1, p_2, ..., p_L]$. Therefore, *AV Align* adds one cross-modal attention mechanism between the two stream encoders, but maintains the traditional attention mechanism between the decoder and encoder. We will denote the hidden state of the decoder LSTM with h_{D_k}, the audio-visual context vector with c_{AV_k}, and with β_{ki} the attention weight between the output timestep k and the input timestep

i, leading to the following expressions:

$$h_{D_k} = \text{LSTM}_D([y_{k-1}; o_{D_{k-1}}], h_{D_{k-1}}) \tag{4.7}$$

$$\beta_{ki} = \text{softmax}_k(h_{D_k}^T \cdot o_{AV_i}) \tag{4.8}$$

$$c_{AV_k} = \sum_{i=1}^{N} \beta_{ki} \cdot o_{AV_i} \tag{4.9}$$

$$o_{D_k} = W_D[h_{D_k}; c_{AV_k}] + b_D \tag{4.10}$$

$$\text{where } W_D \in \mathbb{R}^{n\,x\,2n}, b_D \in \mathbb{R}^n$$

$$p_k \equiv p(y_k|A_{1:N}, V_{1:M}, y_{k-1}) = \text{softmax}(W_\eta o_{D_k} + b_\eta) \tag{4.11}$$

$$\text{where } W_\eta \in \mathbb{R}^{\eta\,x\,n}, b_\eta \in \mathbb{R}^\eta$$

and η is the alphabet size.

Equations (4.3)-(4.6) and (4.7)-(4.10) represent the default behaviour of the *AttentionWrapper* class in TensorFlow (Abadi et al., 2016) using the Luong attention mechanism (Luong et al., 2015). The hidden state of the decoder's LSTM layer in Equation (4.7) is initialised as the final state of the audio-visual LSTM layer: $h_{D_0} = h_N$.

The system is trained using the cross-entropy loss function:

$$CE\ Loss = \frac{1}{L} \sum_k -y_k \log(p_k) \tag{4.12}$$

The motivation behind *AV Align* is to address a possible learning difficulty of the WLAS network, considered a state of the art method when we developed our approach. We speculated that the dual attention decoder of WLAS is overburdened with modelling tasks. On top of audio decoding and language modelling, the WLAS decoder is also required to learn cross-modal correlations. Instead, *AV Align* moves the cross-modal learning task to the encoder side, and the decoder attends to a fused audio-visual memory. Intuitively, AV Align can be seen as a way of reconstructing and enhancing the frame-level audio representations through the use of a dynamically computed visual context vector.

One notable design choice in AV Align is that only the top level representations from each encoder stack are used for cross-modal alignment. The top layers of stacked RNNs encode higher order features, which we assume to be easier to correlate than at the lower layers. Another design choice is the direction of cross-modal attention, which can be applied either from audio to video, or from video to audio. We choose that only the acoustic modality learns from the visual. This

is because in clean speech, the acoustic modality is dominant and sufficient for recognition, while the visual one presents intrinsic ambiguities: the same mouth shape can explain multiple sounds. The design assumes that acoustic encodings can be partially corrected or even reconstructed from visual encodings. One disadvantage is that an alignment score has to be computed for each timestep of the typically longer audio sequence, since usually $N > M$. Overall, these inductive biases have the goal of reducing the number of connections in the network in an educated way, as a more general variant is more prone to the curse of dimensionality.

4.2.2 Visual learning regularisation - the Action Unit loss

As discussed in Section 4.1, the original formulation of *AV Align* did not produce satisfactory results on the challenging LRS2 dataset, conflicting with the substantial improvements seen by Harte and Gillen (2015) in the controlled conditions of TCD-TIMIT. Initial experiments, reported below in Section 4.3.4, suggest a convergence problem of the visual front-end. Following the reasoning from Section 4.1, we want to avoid pre-training strategies and instead rely on the audio-visual data at hand, simplifying the network training methodology.

We suspect that there are two possible causes for the cross-modal attention convergence problem. One is the CNN not learning reliable visual features, as the error signal propagates over a long path susceptible to gradient vanishing. The second one relates to the cross-modal attention module not learning to correlate representations, extract reliable visual context vectors or enhance the acoustic representation. For the reason that the second factor might just be a consequence of the first one, we prefer to focus on improving the visual representations.

Our choice is to regress Action Units (AUs) (Ekman and Rosenberg, 1997) from the visual representations and apply an auxiliary loss function penalising the difference between the network's prediction and the targets externally estimated with the OpenFace toolkit (Baltrusaitis et al., 2018). We argue that learning to predict *certain* AUs is useful to the visual speech recognition task. The auxiliary loss provides a stronger error signal to the visual encoder than the cross-entropy loss on the decoder side, lessening the effect of gradient vanishing.

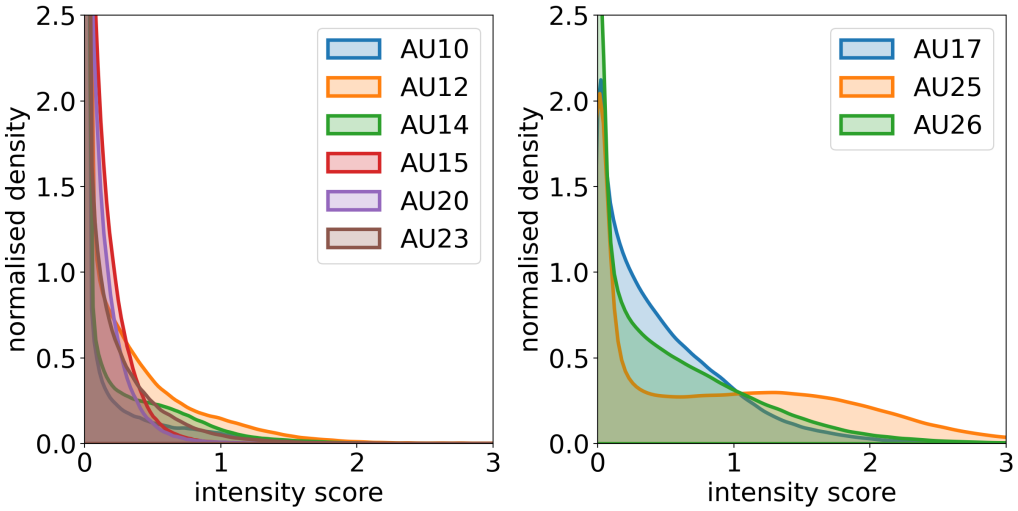


Figure 4.2: *Smoothed histograms of the lip-related Action Units on TCD-TIMIT. Similar histograms are obtained on LRS2.*

Of the 17 AUs estimated by OpenFace, only 9 refer to the lower face / lip area. A closer inspection of their histograms for TCD-TIMIT, displayed in Figure 4.2, reveals that only three of them (17, 25, 26) occur frequently in speech and could be used for our task. We obtained a similar trend on LRS2. AU17 (*Chin raiser*) appears to be estimated unreliably on our datasets. Consequently, we choose only two AUs: *Lips Part* (AU25) and *Jaw Drop* (AU26). These two AUs can be linked to lip opening movements defined by Jeffers and Barley (1980), which occur altogether for approximately one third of the time in speech. Although the visibility of the two AUs may be occluded when co-occurring with other action units in speech, estimating the annotations using the video-based OpenFace toolkit ensures that only the visible AUs are taken into account.

We amend the AV Align architecture from Section 4.2.1 with a fully connected neural network, as depicted inside the dashed box in Figure 4.1. This network, defined in Equation (4.13), takes as input the visual LSTM output o_{V_j}, and produces two outputs activated by the sigmoid function. Since AUs are dynamic attributes, we argue that they can be regressed more reliably from o_{V_j}, where the temporal context is taken into account, than from the frame level visual features v_j.

$$\widehat{AU}_{25,26}(j) = \text{sigmoid}(W_{AU}\, o_{V_j} + b_{AU}) \tag{4.13}$$
$$\text{where } W_{AU} \in \mathbb{R}^{2\,x\,n}, b_{AU} \in \mathbb{R}^2$$
$$\text{and sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$

To generate the target values, we normalise the intensities estimated with Open-

Face, which are real numbers from 0 to 5, by clipping to [0, 3] and dividing by 3 to match the output range of the sigmoid units. We define the AU Loss function (*AU Loss* in Figure 4.1) as the mean squared error between the predicted and target AUs, multiplied by a scale factor μ of 10.0 found empirically on our evaluation data:

$$AU\ Loss = \frac{\mu}{M}\sum_{j=1}^{M}(AU_{25,26}(j) - \widehat{AU}_{25,26}(j))^2 \tag{4.14}$$

The AU Loss is then added to the decoder's cross entropy loss from Equation (4.12).

4.2.3 Transformer variant

Since the AV Align strategy has a a generic formulation with respect to the input requirements, we can replace the LSTM encoders and decoders with the Transformer architecture of Vaswani et al. (2017).

The Transformer architecture is made of an Encoder and a Decoder stack. The Encoder stack contains repeated blocks of self-attention and feed-forward layers. The decoder stack contains repeated blocks of self-attention, decoder-encoder attention, and feed-forward layers. The inputs to these stacks are summed with positional encodings to embed information about the absolute position of timesteps within sequences. The Encoder and the Decoder stacks are schematically illustrated in Figure 4.3.

The Align stack

In order to adapt AV Align to the Transformer architecture, we define an additional Align stack as the repeated application of cross-modal attention and feed-forward layers. The Align stack is displayed in Figure 4.3 between the Encode and the Decode stacks. The cross-modal attention layer is a generic attention layer applied between the outputs of the two stream encoders. The Align block takes video outputs o_V and audio outputs o_A as keys/values and queries respectively, whereas the regular encoder-decoder attention layer receives audio representations and graphemes. Consequently, the Audio-Visual Transformer model implements a single generic attention operation, as originally defined in Vaswani et al. (2017), maintaining simplicity.

Formally, the audio-visual alignment and fusion steps of the attention layer in the

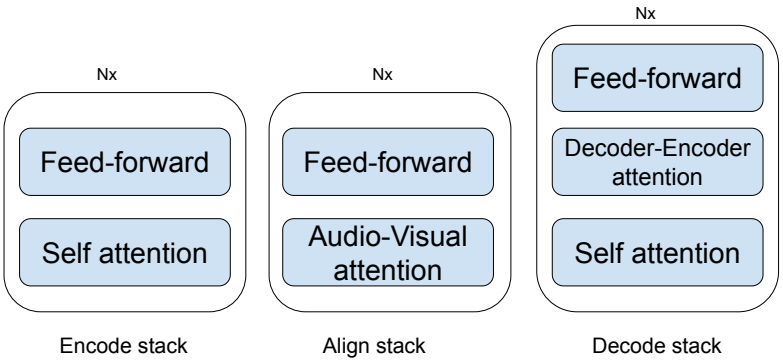


Figure 4.3: *The three main blocks of the Audio-Visual Transformer variant. The Encode and Decode stacks are the same as in the original model Vaswani et al. (2017). We introduce the Align stack which is based on generic attention and feed-forward layers. We keep the Nx notation from the original article to imply stacking together multiple blocks of the same structure.*

Align stack can be described as:

$$c_V = \mathit{Attention}(\text{queries} = o_A, \text{keys} = o_V) \tag{4.15}$$

$$o_{AV} = c_V + o_A \tag{4.16}$$

where c_V are the visual context vectors computed as linear combinations of the video source o_V. Both the LSTM and Transformer variants of AV Align use the concept of dot-product attention to align the higher level audio and video representation, as in Equation (4.15). However, whereas the LSTM model fuses the visual context vector with audio representation by concatenating them and projecting to a shared space using linear combination, as seen in Equation (4.6), the fusion operation in the Transformer is a residual connection, seen in Equation (4.16), which is a sub-case of the weighted summation with fixed weights (more precisely, $W_{AV} = [I_n; I_n], b_{AV} = 0$). Adding one layer's inputs to the attention output is the default fusion mechanism of the original Transformer model of Vaswani et al. (2017), also being used to fuse the decoder's input queries with the audio/audiovisual keys. We also explored a linear combination style fusion for the Transformer model, unreported here, without significant findings. Additionally, no statistically significant differences are also reported later in Section 4.3.6 when exploring multiple feature fusion strategies with the LSTM-based encoders.

Our implementation forks the Transformer model officially supported in The TensorFlow Model Garden (2020) and only adds the high level Align stack, together with the visual convolution front-end, reusing the existing implementations of attention and feed-forward layers. Compared to the multi-modal Transformer model proposed in Tsai et al. (2019), we do not make use of cross-modal attention at

every layer in the alignment stack. As we argued in Section 4.2.1, there may be limited correspondences between audio and video at the lower levels of representations, and aligning only the higher level concepts is likely to speed up the training convergence.

4.2.4 Comparison to related work

WLAS

A closely related audio-visual alignment strategy for speech recognition is *Watch, Listen, Attend, and Spell* (WLAS) by Chung et al. (2017). The main difference between them is that WLAS does not use an attention mechanism between the audio and video modalities as in AV Align, but instead defines two attention mechanisms between the decoder and each encoder, fusing the resulting context vectors. Correspondingly, WLAS uses the symbolic representations in the decoder as a proxy for aligning the audio and visual modalities, resulting in the fusion of two modality-specific context vectors associated with the same symbolic state. Our premise is that the dual attention mechanisms of WLAS overburden the decoder in Seq2seq architectures. In the uni-modal case, a typical decoder has to perform both language modelling and acoustic decoding. Adding another attention mechanism that attends to a second modality requires the decoder to also learn correlations among the input modalities. AV Align aims to make the modelling of the audio-visual correlation more explicit, while completely separating it from the decoder. Correspondingly, AV Align moves this task to the encoder side, and it explicitly models the alignment between each acoustic and all visual encodings. This elegantly addresses the problem of different frame rates, that traditionally required slower modalities to be interpolated in order to match the frame rate of the fastest modality (Potamianos et al., 2003).

Feature fusion

From the perspective of multimodal integration, AV Align shares the same concept with the commonly used feature fusion strategy, where the representations of the two input modalities are integrated directly through concatenation and/or weighting (Duchnowski et al., 1994; Makino et al., 2019; Petridis et al., 2018b; Potamianos et al., 2003; Wand et al., 2018). AV Align does the extra step of finding a time alignment between audio and visual representations, whereas the features are expected to be time-synchronous in feature fusion.

Referring to Equation (4.6), instead of fusing the audio state h_i with its corresponding context vector c_{V_i}, we can re-sample the modalities to the same frame-

rate, i.e. $A = [a_1, a_2, ..., a_M]$, $V = [v_1, v_2, ..., v_M]$, transform each modality independently by several layers of neural networks to obtain $o_A, o_V \in \mathbb{R}^{M x n}$, and finally concatenate the two into $o_{AV}^{concat} = [o_A; o_V] \in \mathbb{R}^{M x 2n}$.

We considered two variants of feature fusion for our experiments in Section 4.3.10.

Variant 1 (V1) linearly projects the concatenated representations to the dimension of the model hidden state:

$$o_{AV_i}^{V1} = o_{AV_i}^{concat} W_{AV} + b_{AV} \quad \text{where } W_{AV} \in \mathbb{R}^{2n x n}, b_{AV} \in \mathbb{R}^n \tag{4.17}$$

Variant 2 (V2) extends V1 with an additional LSTM or Transformer layer applied to the linearly projected fused representations o_{AV}^{V1}:

$$o_{AV}^{V2} = \text{TransformerEncoderStack}(o_{AV}^{V1}) \tag{4.18}$$

Although feature fusion through time-synchronous concatenation does not explicitly model the natural asynchronies of audio-visual speech, we suspect that a neural network modelling long range dependencies between input timesteps, such as the Transformer, may be able to internally re-organise the representations in attempting to leverage the visual modality. This is motivated by the internal structure of the LSTM cell which can gate and store past information in its cell state with high granularity (the forget gate has the same dimension as the state size), coupled with the findings that Transformers still learn the concept of recurrence similar to LSTMs, as we will show in Section 4.3.9.

4.3 Evaluating AV Align

We begin by presenting the data and the system training procedure, followed by a suite of experiments which offer more insights into the learning mechanisms of *AV Align*.

4.3.1 Input pre-processing

Our system takes auditory and visual input concurrently. The **audio** input is the raw waveform signal of an entire sentence. The **visual** stream consists of video frame sequences, centred on the speaker's face, which correspond to the audio track. We use the OpenFace toolkit of Baltrusaitis et al. (2018) to detect and align the faces, then we crop around the lip region. Complete details of the pre-processing of each stream now follow.

Audio input. The audio waveforms are re-sampled at 22,050 Hz in the case of TCD-TIMIT, whereas we maintain the original 16,000 Hz sampling rate for LRS2. The audio signals are additively mixed with several types of acoustic noise at different Signal to Noise Ratios (SNR) as explained in Section 4.3.2. We compute the log magnitude spectrogram of the input, choosing a frame length of 25ms with 10ms stride and 1024 frequency bins for the Short-time Fourier Transform (STFT), and a frequency range from 80Hz to 11,025Hz with 30 bins for the mel scale warp. We stack the features of 8 consecutive STFT frames into a larger window, leading to an audio feature vector a_i of size 240, and we shift this window right by 3 frames, thus attaining an overlap of 5 frames between consecutive audio windows.

Visual input. We down-sample the 3-channel RGB images of the lip regions to 36x36 pixels. A ResNet CNN (He et al., 2016b) processes the images to produce a feature vector v_j of **128 units** per frame. The details of the architecture are presented in Table 4.1.

Table 4.1: *CNN Architecture. All convolutions use 3x3 kernels, except the final one. The Residual Block (He et al., 2016b) is in its* full preactivation *variant.*

layer	operation	output shape
0	Rescale [-1 ... +1]	36x36x**3**
1	Conv	36x36x**8**
2-3	Res block	36x36x**8**
4-5	Res block	18x18x**16**
6-7	Res block	9x9x**32**
8-9	Res block	5x5x**64**
10	Conv 5x5	1x1x**128**

4.3.2 Training procedure

For our experiments, we train and evaluate audio-only and audio-visual speech recognition models based on the sequence to sequence architecture with attention (Bahdanau et al., 2015). The systems model speech at the character level, with an alphabet consisting of the 26 letters English alphabet *a–z*, plus blank space and apostrophe. Our choice for character units instead of sub-phonetic alternatives is aimed at minimising the amount of prior knowledge incorporated into our system, preserving its simplicity. To normalise the text, we convert it to lower case, all numbers are converted to words following the cardinal format, and punctuation is removed. We make our software implementation publicly available as two multimodal speech recognition toolkits based on TensorFlow (Abadi et al., 2016), which are listed in Section 1.5.2. The models using LSTM networks are

implemented in *avsr-tf1*, whereas for the Transformer networks we developed the TensorFlow 2.x based framework *Taris*, also used for the experiments in Chapter 5.

The visual LSTM encoder uses a single recurrent layer, as an ablation study, not reported here, showed a significant increase of training convergence rate for minimal loss in accuracy. The baseline system consists of a 11-layer ResNet (He et al., 2016b) to process the cropped lip images, one or three layers LSTM (Gers et al., 1999) encoders of 256 units for each modality, and a one-layer LSTM decoder of 256 units. For completeness and reproducibility, we provide the software implementation and all the hyper-parameters at `https://github.com/georgesterpu/avsr-tf1`.

The acoustic modality is corrupted with only *Cafeteria* noise, as this noise type was found the most challenging in our initial work (Sterpu et al., 2018a), and the noise source did not influence the conclusions. We train our systems in four stages, first on clean speech, then with a Signal to Noise Ratio (SNR) of 10db, 0db and finally -5db. Each time we increment the noise level we also copy the model parameters rather than train from scratch, speeding up the system's convergence.

4.3.3 Recognition accuracy

In this section we report the performance of the audio-only and multimodal systems on two datasets. We repeat each experiment 5 times and report the mean error of the best system, including the 95% confidence interval displayed as error bars. Additionally, we include the standard deviation of the mean error across the 5 repetitions, displayed with arrows at the bottom of the bar plots.

Speaker-Dependent TCD-TIMIT

We first train Audio-only and Audio-Visual systems on the speaker dependent (SD) partition of TCD-TIMIT, where 70% of data from 59 speakers is used in training, and the remaining 30% in evaluation.

Figure 4.4 shows the Character Error Rate (CER) of our systems for each noise condition. We notice performance improvements of *AV Align* over *Audio* starting from 7% on clean speech, going up to 30% at an SNR of -5db. When we apply the secondary AU loss, the *AV Align + AU* system achieves a similar performance to *AV Align*. This suggests that the AU loss is not detrimental to the performance of *AV Align* when such regularisation is not necessary, as we will show in Section 4.3.4.

A deeper dive into these results reveals that when comparing the audio-visual system with the audio-only one, performance gains extend not only to the already seen *sx* sentences, but also to the unique *si* ones. Note *sx* are the sentences repeated across many speakers, whereas the *si* sentences are unique to a speaker. Therefore we can deduce that DNNs can learn sentence independent speech representations. However, it would be much stronger to show that the learnt representations are also speaker independent.

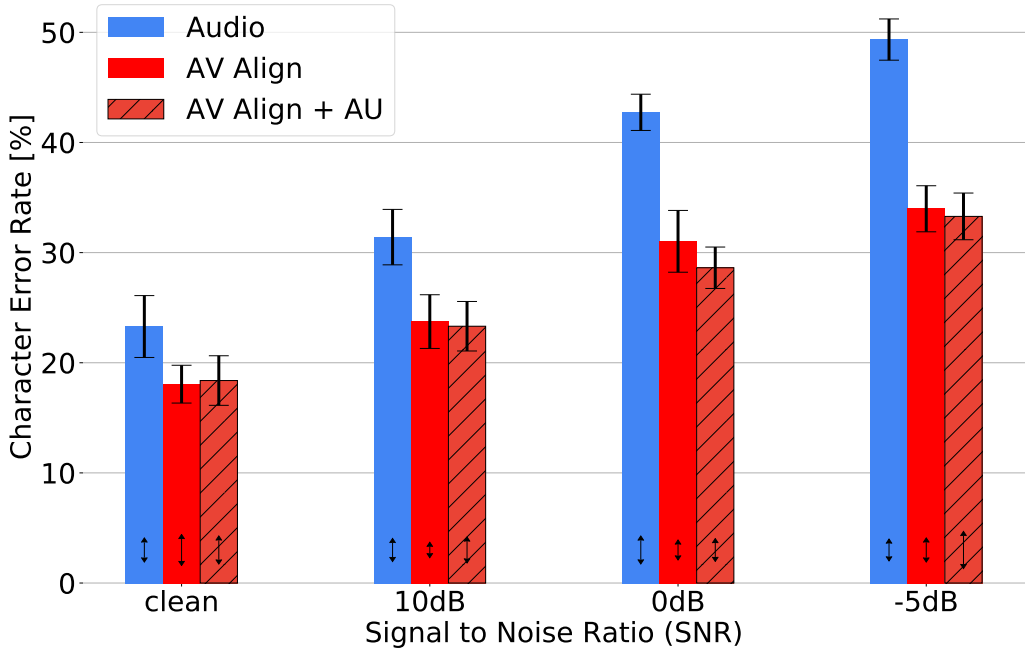


Figure 4.4: *Performance on the Speaker Dependent (SD) partition of TCD-TIMIT.*

Speaker-Independent TCD-TIMIT

We needed a new dataset partitioning scheme to achieve speaker independence. We thus assign each TCD-TIMIT volunteer either to the train or test partition, aiming at the same time to balance attributes such as gender and facial hair. Consequently, speakers *06M, 14M, 17F, 18M, 31F, 41M, 46F, 47M,* and *51F* are assigned to the test set, and the remaining 50 to the train set. Due to the large overlap with the volunteer sentences, the lipspeakers were not used here.

We retrained the audio and audio-visual systems from 5 different random initialisations on this new partition, and display the results in Figure 4.5. The confidence intervals and standard deviation are displayed as in Figure 4.4. Note overall a strong trend whereby performance for the repeated *sx* sentences is markedly better than for the unique *si* sentences. This is apparent in both the audio and audio-visual systems in this speaker-independent scenario. The global error rate is hence a misleading performance figure. This can likely be attributed

to an inherent language model in the decoder that becomes strongly tuned to the more frequently seen *sx* content due the imbalance between the two sentence types in TCD-TIMIT and the reduced sentence diversity, promoting memorisation (Arpit et al., 2017; Zhang et al., 2017). Both *AV Align* and *AV Align + AU* frequently converge to poor local optima where the error rate is similar to the one of the *Audio* system. As it will later be shown in Section 4.3.4, this corresponds to the case where the audio-visual alignments are not properly learnt. Overall it is difficult to offer definitive conclusions from these experiments. We can see that the variance in error across multiple random initialisations is reduced by introducing the AU loss, but ultimately it appears we do not have sufficient data for each speaker in TCD-TIMIT to train the speaker-independent system. An ideal dataset would have a much larger number of unique sentences from a larger cohort of speakers, but such a dataset does not exist in the research community. Experiments in the following sections will use LRS2 to allow a fuller exploration of how to optimally exploit the visual modality in speaker-independent AVSR using the *AV Align* strategy.

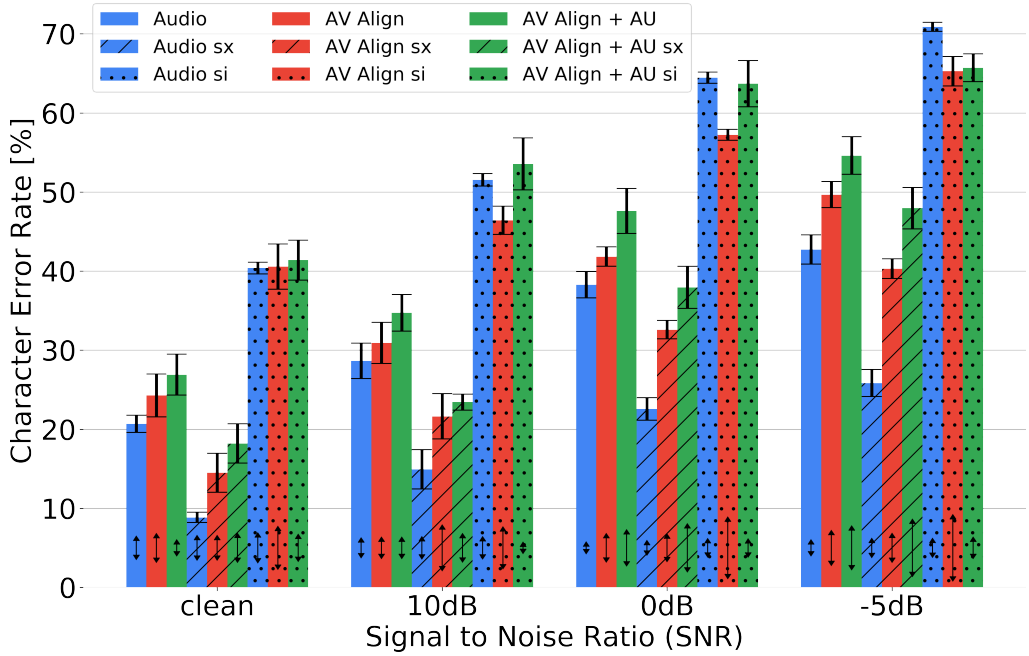


Figure 4.5: *Performance on the Speaker Independent partition of TCD-TIMIT. Note* sx *sentences are repeated across many speakers but* si *sentences are unique to a speaker.*

LRS2

Since the relatively small size of TCD-TIMIT restricts the learning power of a neural network, as seen in the previous experiment, we now evaluate *AV Align* on LRS2. Introduced in Section 2.8.2, LRS2 is currently the largest publicly available audio-visual dataset. We retrain the audio and audio-visual systems from scratch

with the same hyper-parameters, and discard approximately 2.77% of the LRS2 sentences due to the failures of the face detector. Our results are shown in Figure 4.7 where confidence intervals and standard deviation are displayed as before.

We notice a relative performance improvement of the *AV Align + AU* system over *Audio* starting at 6.4% on clean speech, going up to 31% in worsening audio conditions. These improvements rates had previously only been seen on the speaker dependent partition of TCD-TIMIT, whereas this time we are in the challenging setup of LRS2. The overlap of the 95% confidence intervals in clean speech suggests that the differences between our audio-only and audio-visual models are not statistically significant, despite the better mean error rate achieved by the best *AV Align + AU* system. Summerfield (1987) argued that bimodal speech perception plays a role in restricting the number of lexical hypotheses, regardless of the noise conditions. In this setting, the mean error rate may not be a sufficient metric to fully demonstrate the advantages of *AV Align + AU* over the *Audio* model.

This result brings evidence to support our rationale in Section 4.2.2 regarding the difficulties faced by *AV Align* in overcoming the fundamental problem of gradient vanishing. *AV Align* converges to a local minimum where only the audio representations are learnt effectively. We include in Figure 4.6 a visualisation of several gradient histograms in the convolutional kernels for both *AV Align* and *AV Align + AU* when trained on LRS2 data. Showing that the AU-regularised system consistently obtains larger gradients which do not vanish at later stages in training explains to some extent our intuition behind the choice for the AU Loss.

Figure 4.6: *Gradient Visualisation in the CNN layers of both AV Align (orange) and AV Align + AU (blue). Note the differing scales on the horizontal axis between systems.*

The multitasking design based on the proposed AU Loss was needed so the net-

work could start learning audio and visual representations from the beginning. Nevertheless, these solid improvements on LRS2, as opposed to the inconclusive results in Section 4.3.3, suggest that a strongly imbalanced dataset further contributes to the learning difficulties of the network, and special attention has to be paid to this aspect when collecting new datasets for AVSR.

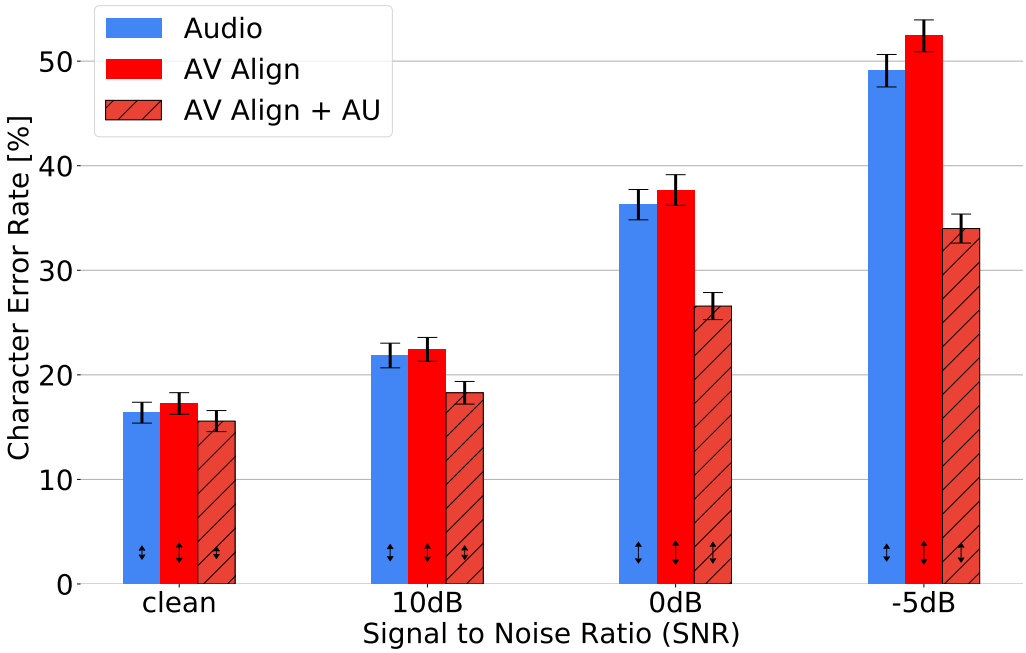


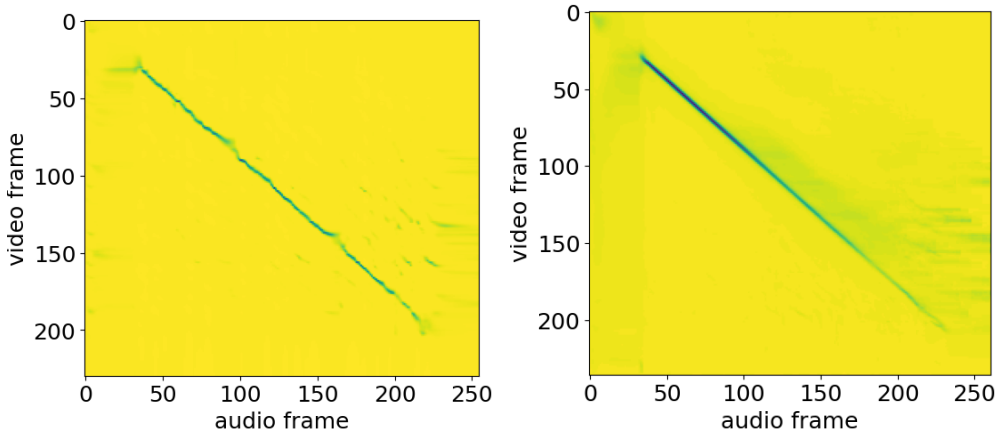
Figure 4.7: *System performance on LRS2*

4.3.4 Cross-modal alignment patterns

The *AV Align* architecture allows an explicit soft alignment between the audio and visual representations extracted by the two encoders. A question that arises is: does it *really* learn to align the two modalities of speech, or does it only exploit a spurious correlation in the dataset that would limit the generalisation power? So far we found that the method can decode up to 30% more accurately than an audio-only system on both TCD-TIMIT and LRS2. We have not yet identified the source of this improvement. In this section we will visualise the audio-visual alignments produced by AV Align.

For every sentence in the test set, we generate the alignment matrix between the two encoders, which is the α_{ij} variable in Equation (4.4) and has a size of [*MxN*] corresponding to the number of frames in each modality. Similarly, we also generate the alignment matrix between the decoder state and the fused audio-visual representations, represented by the β_{kj} variable in Equation (4.8) having a size of [*NxL*] where *L* is the number of decoded characters.

TCD-TIMIT



(a) *AV alignment on sentence 31F/si459* **(b)** *Mean AV alignment*

Figure 4.8: *Cross-modal alignment patterns of the system trained on TCD-TIMIT*

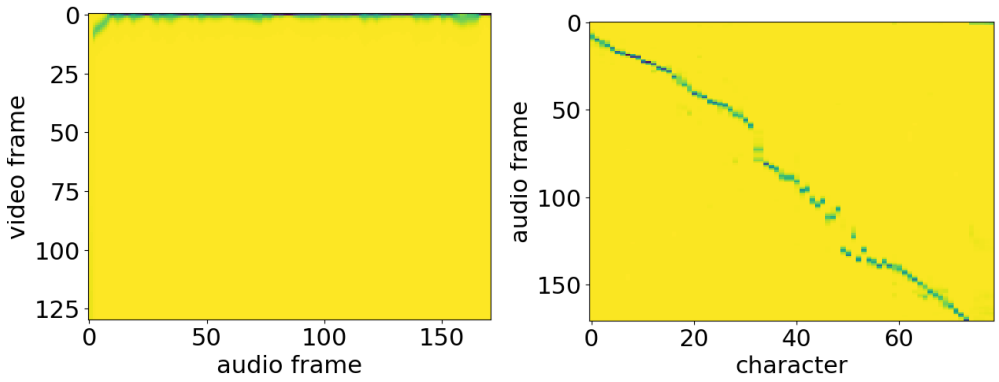
We display in Figure 4.8a the cross-modal alignment pattern of *AV Align* on a randomly chosen sentence from the speaker dependent test set of TCD-TIMIT. We observe that the alignment pattern looks almost monotonic in a weak sense, i.e. can be well approximated by a monotonic function. The lack of alignment at the start and end of the sentence is attributed to the recording conditions of TCD-TIMIT, where the speakers were instructed to leave a second of silence. We also aggregate all the alignments on the full test set in Figure 4.8b, noticing that the monotonicity property is preserved.

LRS2

In Figure 4.9a we display the cross-modal alignment patterns of *AV Align* on a randomly chosen example from LRS2, together with the decoder's text alignment in Figure 4.9b.

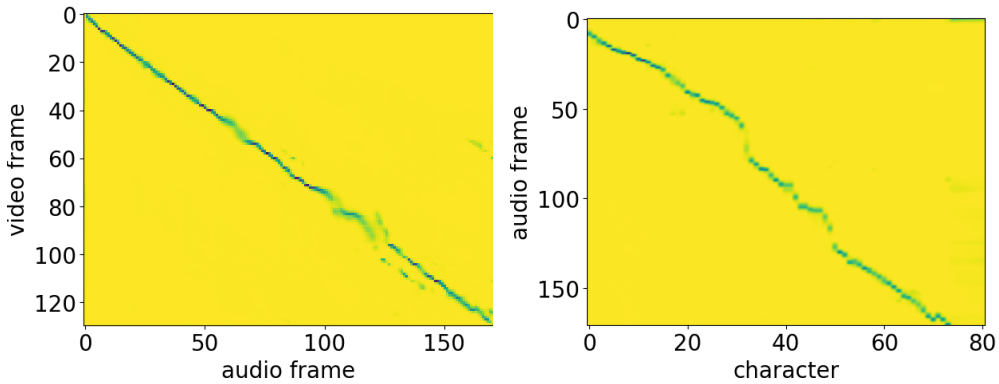
We observe that each audio frame is predominantly aligned to the first video frame, suggesting a failure of the cross-modal attention mechanism to converge. On the other hand, the second attention mechanism learns non-trivial and plausible alignments between text and inputs. Likely, the fused audio-visual representations are dominated by the audio modality. The performance similarity between *AV Align* and the audio system for all noise levels, illustrated in Figure 4.7, brings further evidence to support this claim. We find a similar pattern on the proposed speaker independent partition of TCD-TIMIT.

In Figure 4.10 we display the alignments of *AV Align + AU* on the same sentence as in Figure 4.9. This time we see that the system effectively learns proper cross-modal alignments, explaining the performance improvement shown in Figure 4.7.



(a) *Audio - Video* **(b)** *Decoder - Encoder*

Figure 4.9: *Alignment patterns on a single example from LRS2 (6349793037997935601/00008)*



(a) *Audio-Video* **(b)** *Decoder - Encoder*

Figure 4.10: *Alignment patterns on a single example from LRS2 (6349793037997935601/00008)* ***when training with AU Loss***

Overall, this suggests that monotonic audio-visual alignments are a necessary condition for *AV Align* to capitalise on the visual modality.

4.3.5 Additional control experiments - aligning without video

To validate that the monotonic alignments represent true correlations between audio and video, we propose three control experiments by corrupting the visual representations o_{V_j} attended to by the audio-visual LSTM layer. These experiments do not require re-training the systems and are only applied for inference.

We first replace the visual representations with random uniform noise. As shown in Figure 4.11a, the cross-modal alignment patterns are no longer monotonic as in Figure 4.10a. The error rate surges above 100%, indicating a limitation of the training strategy to cope with a mismatched data distribution. Next, we add segments of blank video frames between one and four seconds long, both in the

beginning and at the end of a sentence. We see in Figure 4.11b that the alignment patterns have shifted vertically for a proportional amount of timesteps. After reversing the time axis of the video representations, we observe in Figure 4.11c that the alignment patterns become horizontally flipped too. The error rate on the test remains identical in this case, whereas it only slightly increases by 0.31% when appending blank visual frames. These control experiments show that the audio and the visual representations are aligned only by their content, and not because the system implicitly learned to exploit the monotonicity of speech to guess where to look in a sentence.

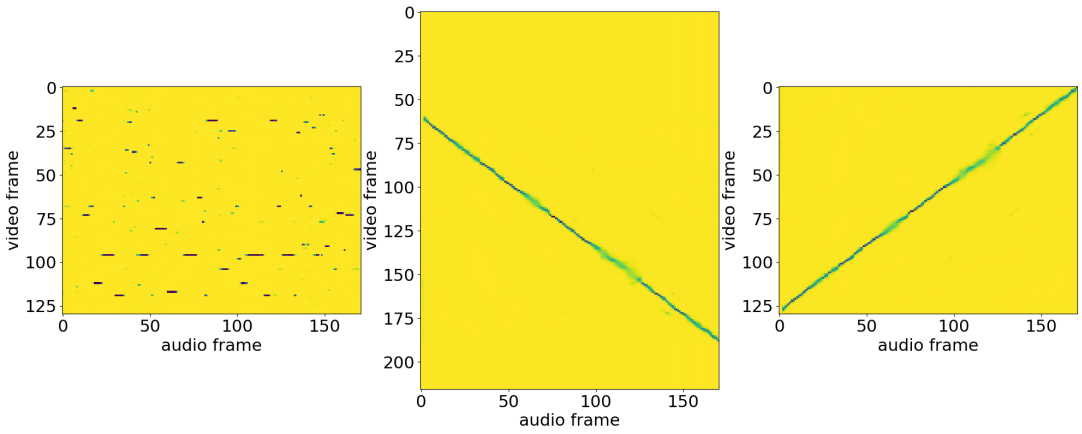


Figure 4.11: *The effect of corrupting the video memory with several transformations. Same sentence as in Figure 4.10.*

4.3.6 Enhancing the representation fusion layer

We have shown so far that *AV Align* is able to align audio and video representations, and consequently to fuse them into an informative visual context vector. In the standard formulation of the network, the fusion step is implemented as follows: the context vector c_{V_i} is concatenated with the current audio-visual encoder output h_i and processed by a *single* layer linear neural network. Using the shorthand notation $\text{nn}_i(x) = W_i x + b_i$, the fusion function defined in Equation (4.6) was $o_{AV_i} = \text{nn}_{AV}([h_i; c_{V_i}])$ (referred to as *baseline*). We want to explore deeper and

nonlinear fusion networks, and we propose the following fusion designs:

$$
\begin{aligned}
M1: \quad & o_{AV_i} = \tanh(\mathrm{nn}_1(\tanh(\mathrm{nn}_2([h_i; c_{V_i}])))) \\
M2: \quad & o_{AV_i} = h_i + \tanh(\mathrm{nn}_1(h_i)) + c_{V_i} + \tanh(\mathrm{nn}_2(c_{V_i})) + \\
& \quad + \tanh(\mathrm{nn}_3([h_i; c_{V_i}])) \\
M3: \quad & o_{AV_i} = h_i + \tanh(\mathrm{nn}_1(h_i)) + \tanh(\mathrm{nn}_2(c_{V_i})) + \\
& \quad + \tanh(\mathrm{nn}_3([h_i; c_{V_i}])) \\
M4: \quad & o_{AV_i} = h_i \cdot \mathrm{W_a} + c_{V_i} \cdot \mathrm{W_v}, \text{where} \\
& \quad \mathrm{W_a} = \mathrm{sigmoid}(\mathrm{nn}_1(h_i)), \\
& \quad \mathrm{W_v} = \mathrm{sigmoid}(\mathrm{nn}_2(c_{V_i})) \\
M5: \quad & o_{AV_i} = h_i \cdot \mathrm{W_a} + \tanh(\mathrm{nn}_1([h_i; c_{V_i}])) \cdot \mathrm{W_{av}}, \\
& \quad \text{where } \mathrm{W_{av}} = \mathrm{sigmoid}(\mathrm{nn}_2([h_i; c_{V_i}]))
\end{aligned}
$$

As can be seen in Figure 4.12, variants M1 and M5 are relatively up to 5.8%

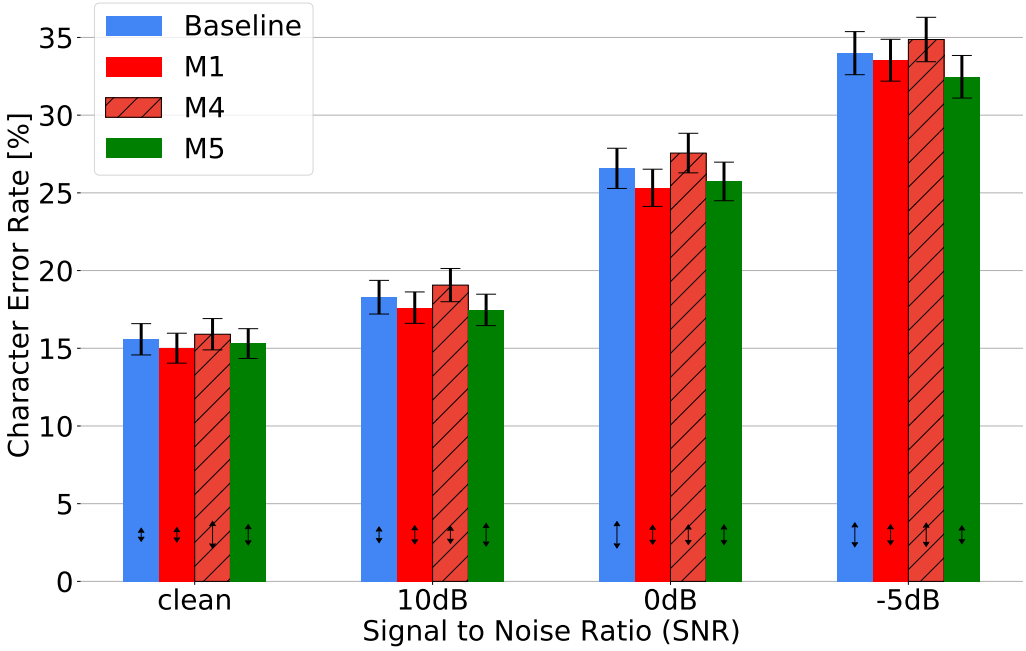


Figure 4.12: *System performance on LRS2 with several variations of the audio-visual fusion layer*

better than the baseline at certain noise levels. These statistically insignificant improvements suggest either that the original linear fusion is a sufficiently good approximation in AVSR, or that the nonlinearities are learnt by another component of the system, such as the input gate of the LSTM encoder. Interestingly, variant M4 is only 2% to 4% worse than the baseline, however it offers a greater interpretability potential since it assigns a confidence score between 0 and 1 to each modality at every timestep. For M1, we also experimented with one and

three layer variants using ReLU and sigmoid activation functions, all performing slightly below the presented variant. M2 and M3 were similar in performance to M5, and were both omitted from the plot for clarity. Despite not being supported by statistical significance, this experiment illustrates possible extensions of the linear fusion in *AV Align*, which may become useful on larger amounts of data and video corruption levels. Stewart et al. (2014) propose a stream weighting approach based on estimating the uncertainties in each modality, similar to our M4 but with a single weight per stream rather than per feature bin, showing that the benefit of stream weighting becomes more salient when the visual modality also is subject to noise corruption, which was not explored in this work.

4.3.7 Applicability of AU loss to WLAS

Having demonstrated the importance of the AU Loss in Section 4.3.4 for *AV Align*, we ask the question: does it also improve the WLAS network of Chung et al. (2017)? Our assumption is that the convergence problem is owed to the imbalance in the information carried by the two speech modalities, which is both data and model invariant.

We implement the WLAS network and follow an identical training and evaluation procedure on LRS2 as with *AV Align*, ensuring a fair comparison. Since we do not pre-train the audio and language models on an auxiliary corpus of 224,528 sentences as in Chung et al. (2017), and we do not make use of the curriculum learning or alternating training, essentially training it in the same way as *AV Align*, we denote the network by *AV Cat* instead of *WLAS*. The results are shown in Figure 4.13. The original design, *AV Cat*, performs just slightly worse than the audio system, as we saw with *AV Align*. The improved model using the AU Loss, *AV Cat + AU*, outperforms the audio model, however its performance is still inferior to *AV Align + AU*.

Additionally, we train *AV Cat* and *AV Cat + AU* on the SD partition of TCD-TIMIT, and we show the results in Figure 4.14. The same trend can be seen as in the case of LRS2.

We further investigate the two alignments produced by this architecture, consisting of the correlation of the decoder state with each encoded modality, which are displayed in Figure 4.15. The first column represents the *AV Cat* system, presenting a similar video convergence problem as with the cross-modal alignment of *AV Align* from Figure 4.9a. The next four columns illustrate the benefit of the AU loss for the video convergence of *AV Cat + AU* as the noise level increases. The text to video alignments are less pronounced in clean speech conditions, unlike

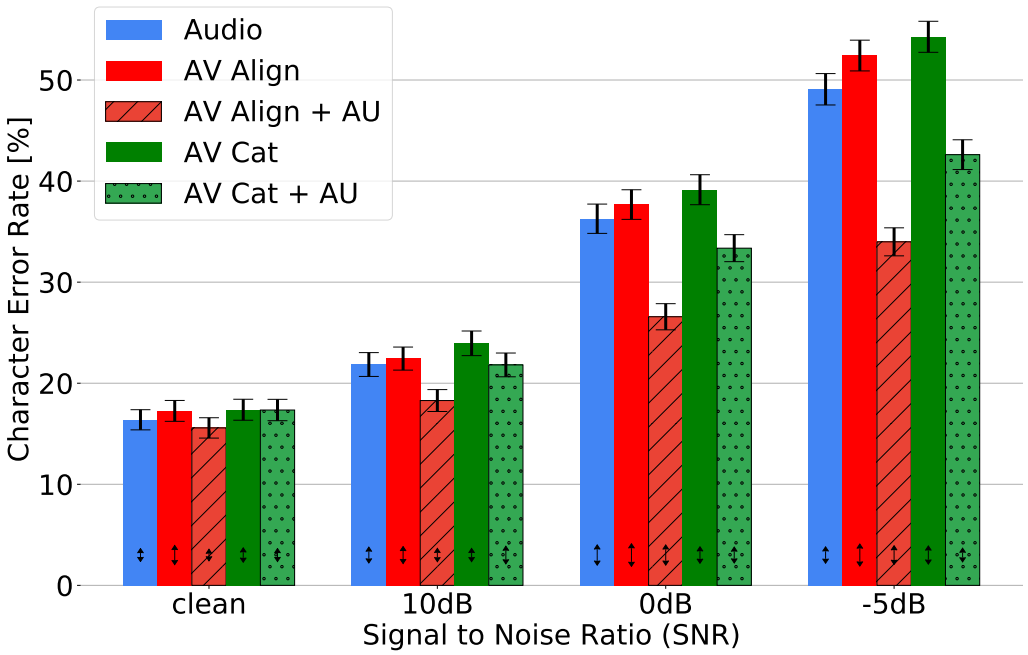


Figure 4.13: *Performance of all five systems on LRS2*

in Section 4.3.4, where audio-visual alignments emerge and remain crisp starting on clean speech. This suggests that aligning the voice with the lips may be a simpler task than correlating characters with lips. In fact, the latter may prove difficult even to human annotators, making *AV Align* more suitable for semi-supervised learning than *AV Cat* (e.g. when there is no text annotation available).

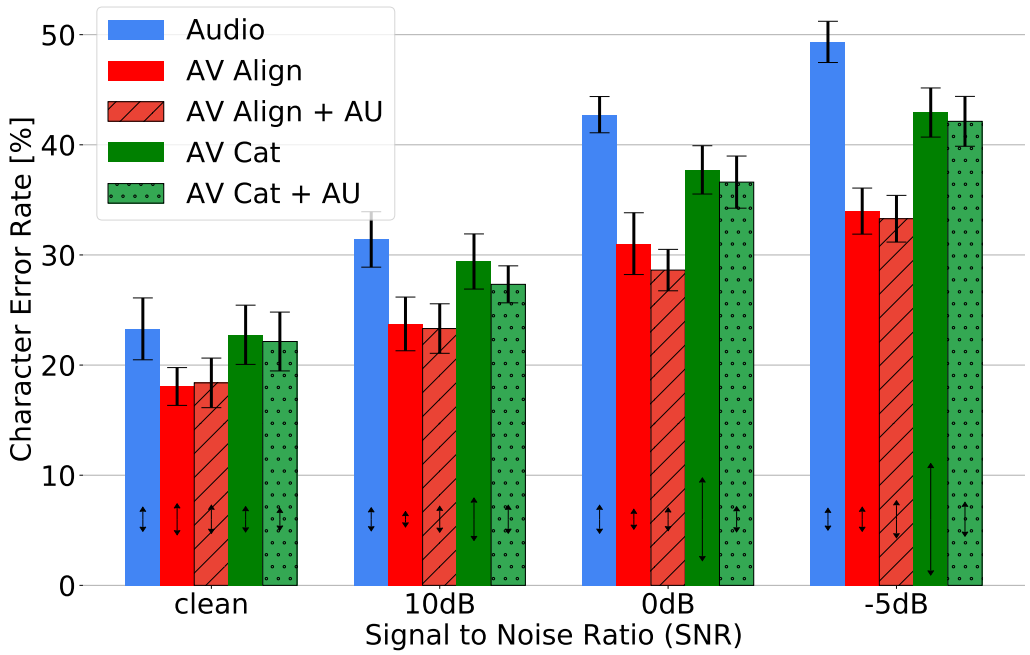


Figure 4.14: *Performance of all five systems on the SD partition of TCD-TIMIT*

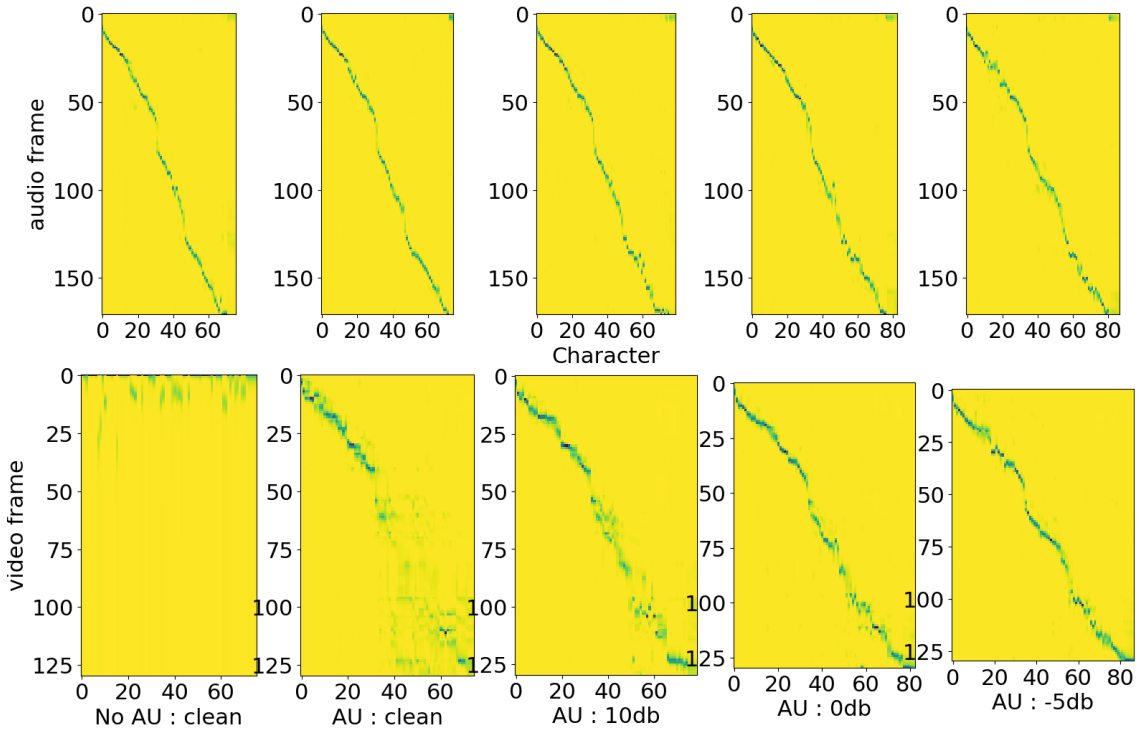


Figure 4.15: *Alignment patterns of the WLAS network Chung et al. (2017) trained on LRS2 and evaluated on the same sentence as in Figure 4.10. Top row displays the text to audio alignment, bottom row displays the text to video alignment.*

4.3.8 Error analysis

In Section 4.3.3 we showed that *AV Align + AU* performs 31% better than the audio-only system on the test set of LRS2. Since this is only an average value, it

would be interesting to know if this gain is general or restricted to a subset of the sentences. Therefore we have analysed the error rate on individual sentences. For each test sentence, we first compute the difference δ between the error rates of the *Audio* and *AV Align + AU* systems. Next, we estimate the predictability of each sentence by training a separate character-level language model on the train set, and evaluating the cross-entropy between the labels and the predictions. The language model is similar to the *AV Align + AU* decoder shown in Figure 4.1, but without any conditioning on the encoder. Since the analysis is fairly similar for all noise levels, we focus our attention on the most challenging -5db condition.

In Table 4.2 we list several examples of sentences ranked by their cross-entropy score. When the sentence is highly predictable (*thank you very much*, *something like that*), both systems provide the same prediction. In some cases, the Audio-visual system produces a worse explanation. For example, it replaced *choice* with *auctions* or *scores* with *that's all*, where the Audio system guessed the correct word. At a SNR of -5db, it may be possible that the Audio-visual system learns to trust more the video modality, erroneously dismissing informative auditory cues. On the other hand, in other cases the Audio-Visual system provides the right prediction where the Audio model does not succeed to guess even one word. This is the case of *"and our experts"*, which is only decoded perfectly from two modalities.

We plot the error difference δ in Figure 4.16. Although *AV Align + AU* performs better on average, there is still a number of sentences where the audio system scores better. A closer inspection on several examples where the error difference is -50% or lower shows an interesting pattern: while the audio system makes reasonable spelling mistakes at this noise level, the prediction of the audio-visual one looks highly uncorrelated with the input. For example, the sentence "was it your choice" is acoustically transcribed as "was in your choice", whereas the *AV Align + AU* prediction is "was in the auctions". This sentence belongs to a cluster of highly predictable sentences which are decoded almost perfectly by the audio system. We could not identify an obvious pattern in the visual domain on these sentences. We performed an analogous analysis between the audio and *AV Cat + AU* systems, and also between *AV Cat + AU* and *AV Align + AU*, all with similar findings. This result suggests a shortcoming of both audio-visual systems: they do not fall back to audio-only performance when not able to capitalise on the visual modality. Instead, the conditioning on the input seems to diminish, leading to a more prominent impact of the intrinsic language model.

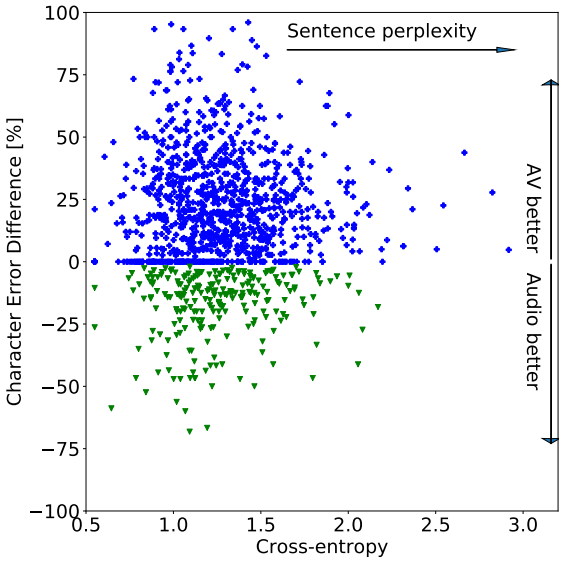


Figure 4.16: *Absolute error difference between the Audio and* AV Align + AU *systems on -5db speech, sorted by their predictability (easier sentences on left).*

In Figure 4.17 we illustrate the same difference δ as a cumulative distribution function, allowing us to compare rate of improvement under different noise conditions. In line with the results in Figure 4.7, we notice that more sentences see the benefit of the visual modality in worsening audio conditions.

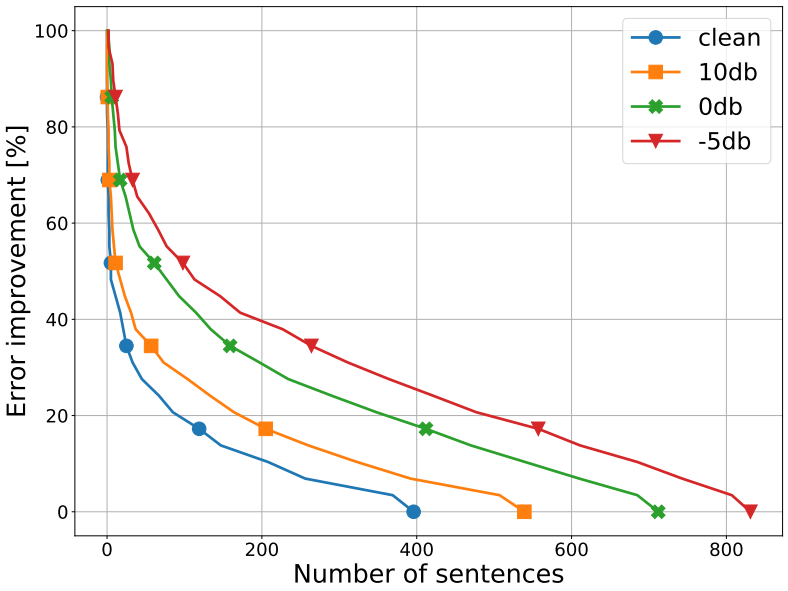


Figure 4.17: *Cumulative distribution function of the error improvement on LRS2, truncated to 0%, from* Audio *to* AV Align + AU*. The test set of LRS2 contained a number of* ***1,222*** *sentences.* X sentences see an improvement of at least Y%.

Table 4.2: *Examples of Sentences from LRS2 Ranked by their Cross-Entropy (CE) Score Reflecting Predictability*

Sentence	CE	[A] prediction	[AV] prediction
squirrel pox virus	3.11	spiral pops fires	squeer apots fires
puerto rican style	2.8	porture recan style	porture reconsile
great leonard cohen	2.43	rate leader cowin	rate lenent cowen
sausages in bacon	2.34	such a years in baken	such a year's impainent
the duke of gloucester	2.00	which you could prossed	the two coffloster
there aren't any biscuits in that barrel	1.78	there are antique biscuits in their barrow	there aren't any buscuity in their bear of
some decent scores	1.66	some piece of scores	some things that's all
was it your choice	1.36	was in your choice	was in the auctions
very close by the university	1.08	very close by the university	very close by the university
and our experts	1.05	i know where it's that	and our experts
i don't think so	0.89	i don't think so	but don't place so
something like that	0.63	something like that	something like that
thank you very much	0.56	thank you very much	thank you very much

4.3.9 AV Align Transformer

In this experiment we replace the LSTM cells of AV Align with the Transformer network. We want to know if a shorter gradient propagation path specific to the Transformer can render unnecessary the AU Loss. Such finding may imply that the previously observed visual convergence issue is specific to the LSTM network.

Neural network details

The Transformer model uses 6 layers in the Encoder and Decoder stacks, a model size d_{model} = 256, a filter size d_{ff} = 256, one attention head, and 0.1 dropout on all attention weights and feedforward activations. The Align stack is made of a single block of cross-modal attention and feed-forward layers, with one attention head. We performed an ablation study, noting that an increase in width and depth was not worth the additional computation time with respect to accuracy.

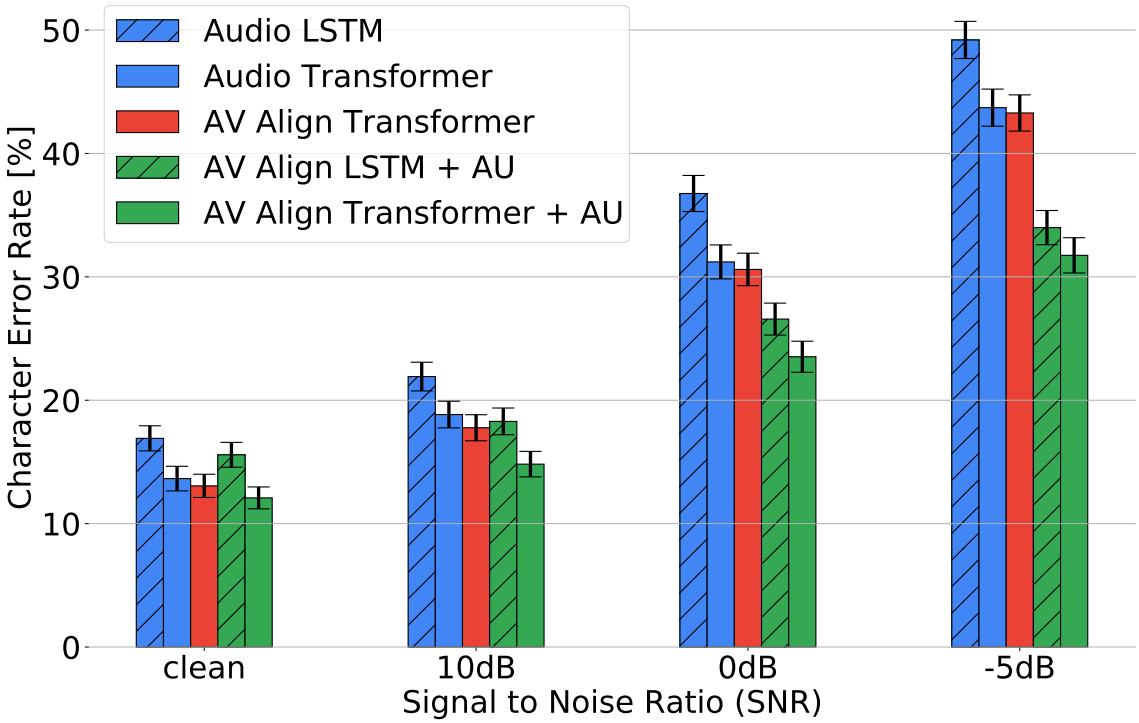
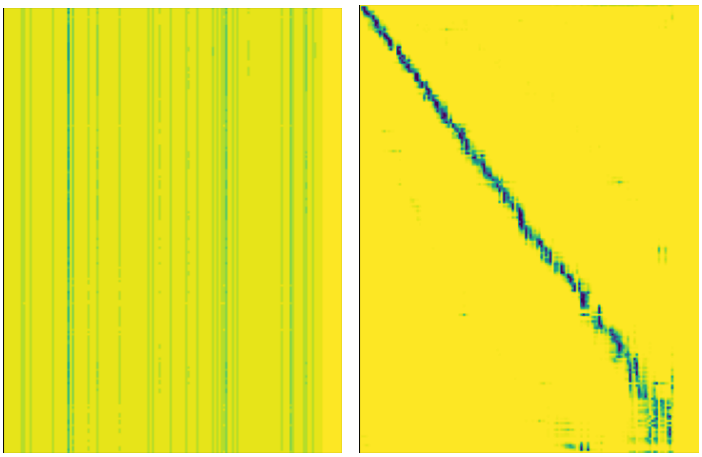


Figure 4.18: *Comparison between LSTM and Transformer models on LRS2. The former are shown with hatched bars. The audio-only models are shown in blue. The error bars denote the 95% confidence interval of the mean error.*

Audio-visual speech recognition accuracy

We train audio and audio-visual Transformer models on the same partition of LRS2 used to train our LSTM models. We follow an identical procedure with the one in Section 4.3.2, corrupting the audio modality in four stages with cafeteria noise. As in Section 4.3.3, we train an additional audio-visual model with the Action Unit loss enabled. The results are shown in Figure 4.18.

We notice that the AV Transformer achieves a similar performance to the Audio Transformer, suggesting that the video modality was not capitalised on. We start seeing performance improvements only when the AU loss is used, reproducing the finding in Section 4.3.3. The relative performance improvements of the AV Transformer + AU over the Audio Transformer start at 7.5% in clean speech, and go up to 26.6% in noised speech. Thus, the visual modality brings similar levels of relative improvements over the audio-only modality to both the Transformer and the LSTM trained in Section 4.3.3. The absolute error differences between the LSTM and the Transformer models are partly owed to the larger model size of the Transformer (25 MB Audio, 36 MB AV) over the LSTM one (9.3 MB Audio). In Section 4.4.1 we will discuss in greater detail the differences between the two architectures going beyond the error rates reported here.



(a) *Base model* **(b)** *Base model + AU Loss*

Figure 4.19: *The Audio-Visual alignments learnt by the Transformer models*

Audio-visual alignments

We inspect the alignment weights between the audio and visual representations, which are displayed in Figure 4.19. Without the AU Loss, the AV Transformer has the same difficulty as the Audio-Visual LSTM in Section 4.3.4 to learn cross-modal correspondences (Figure 4.19a), thought to be caused by the improper learning of visual representations. The alignments emerge as monotonic at the macro-block level with the AU loss (Figure 4.19b), reproducing the patterns obtained by the LSTM network.

4.3.10 Comparison to feature fusion

We have shown that AV Align represents a better inductive bias in audio-visual speech recognition than the dual attention decoder of WLAS. We still need to investigate if learning a soft cross-modal alignment between the two modality encoders is superior to a simpler direct feature fusion strategy that lacks the alignment component. In Section 4.2.4 we suspected that a Transformer network taking concatenated audio-visual inputs may still be capable to achieve an internal alignment in order to maximally take advantage of both modalities. We also notice there have not been any comparisons in the related work between WLAS and feature fusion either, strengthening the motivation for our investigation.

Since feature fusion requires both modalities to be sampled at identical rates, we choose to re-sample the audio modality. The LRS2 dataset has a visual frame-rate of 25 frames per second, which gives a frame time of 40 milliseconds (ms). Whereas we previously computed time-frequency audio representations every 10 ms, stacked 8 of them under a window of 80 ms and slid this window with

steps of 30 ms, we now slide the window with steps of 40 ms. In other words, we reduce the data overlap between two consecutive windows from 62.5% down to 50%. This allows us to obtain an audio feature for every 40 ms of speech, matching the visual frame time.

Each experiment was performed twice to increase our confidence that the random initialisation does not create higher error fluctuations that what we previously saw on the LRS2 dataset, and here we report the mean of the two experiments. The absolute error differences between the two experiments were predominantly below 1%. In Table 4.3 we report the character error rate of the early fusion systems on LRS2, together with the performance of the audio-only system using the new 40 ms window strides increased from 30 ms. The first five systems in this table are already plotted in Figure 4.18, and the 95% error confidence intervals of the last five systems are very similar to the ones obtained with the AV Align Transformer.

Table 4.3: *Comparison between AV Align and Feature Fusion on LRS2. The first five systems are also displayed with bar plots in Figure 4.18.*

System	Character Error rate [%]			
	clean	10db	0db	-5db
Audio LSTM	16.38	21.85	36.27	49.08
AV Align LSTM + AU	15.57	18.28	26.57	33.98
Audio Transformer	13.65	18.84	31.21	43.71
AV Align Transformer	13.06	17.77	30.60	43.28
AV Align Transformer + AU	12.08	14.82	23.52	31.74
Audio Transformer 40ms	14.14	19.34	35.21	47.46
Feature Fusion V1	15.00	19.13	34.15	46.10
Feature Fusion V2	14.71	19.67	33.23	44.32
Feature Fusion V1 + AU	13.83	17.31	27.78	37.81
Feature Fusion V2 + AU	12.96	16.25	25.53	34.68

We first notice there is a slight degradation in performance of the audio model trained on the 40 ms frame stride. This can be explained by the slightly lower overlap between consecutive audio frames, and also by the effect of input scaling on the overall accuracy of a neural network, as it was explored by Tan and Le (2019) on image classification. Next, we observe that the base Audio-visual models without the Action Unit regularisation loss achieve a comparable performance with the Audio model, confirming our previous findings with AV Align. When the AU loss is used, both V1 and V2 obtain improvements over the audio model alone, with the larger V2 model making use of an additional Transformer layer for fusion having a significant advantage over the smaller V1. Furthermore,

we see that V2 + AU is actually very close to the performance of AV Transformer + AU. Given that there are already differences between the 30ms and 40ms rate Audio models, this partly explain the difference between AV Align and Feature Fusion V2. Overall, this result suggests that feature fusion is still able to leverage the visual modality in speech recognition to a similar extent as AV Align, when it is coupled with a single-modality Transformer downstream.

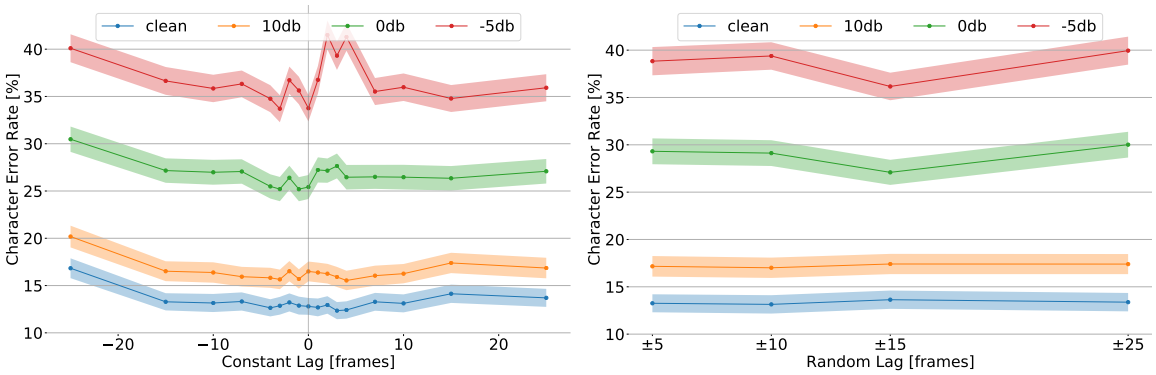
Artificial delay between modalities

Showing that direct audio-visual feature concatenation does not reduce the decoding accuracy considerably as opposed to learning a cross-modal alignment, brings more evidence to support our hypothesis in Section 4.2.4 that a Transformer can tolerate small delays between modalities owed to the natural asynchrony of multimodal speech. However, it would be much stronger to show that the Transformer can also tolerate larger delays created artificially.

To test this hypothesis, we design two new experiments where we time shift each one modality with respect with the other one, for either a constant or random amount of time. We achieve this by padding with zeros the two sequences of representations o_A and o_V. We will use the convention that a positive lag corresponds to the visual sequence shifted forward in time, or padded with zeros at its left extremity. Similarly, shifting the visual sequence backward in time is accomplished by padding zeros at its right extremity. Since both audio and visual sequences need to have an equal length for the direct concatenation operation, we pad the audio sequence anti-symmetrically with the same number of zero frames as the visual sequence was padded. For a shift D, we create a sequence Z_D made of $|D|$ zero vectors, where each vector has the same size as our feature size (i.e. 256). Formally, when the shift D is positive ($D > 0$), $o_V \leftarrow [Z_D; o_V]$ and $o_A \leftarrow [o_A; Z_D]$, whereas when the shift D is negative ($D < 0$), $o_V \leftarrow [o_V; Z_D]$ and $o_A \leftarrow [Z_D; o_A]$.

Since it obtained the best performance in the previous experiment, we choose the V2 + AU model and train it under two conditions. First, we consider D to be constant, and choose values in the range of $\pm1, \pm2, \pm3, \pm4, \pm7, \pm10, \pm15, \pm25$ frames, each corresponding to 40 ms of new data. Second, we experiment with a variable delay D sampled from a random uniform distribution with the minimum and maximum values between $\pm5, \pm10, \pm15, \pm25$. This variable delay is applied both in training and inference, and it is sampled independently for every batch of data.

We plot our results in Figure 4.20. As it can be seen, the decoding accuracy



(a) *Fixed* **(b)** *Random*

Figure 4.20: *Evaluation of Feature Fusion V2 + AU on LRS2 when a constant or random delay is artificially introduced between the two modalities. The duration of one frame is 40ms. Delays are simulated with zeros padding the start or end of sentences. Positive delays correspond to the video modality shifted forward in time (i.e. zeros are added to the start of the video modality and to the end of the audio modality respectively).*

remains relatively stable for a wide range of time shifts. Particularly in low noise conditions, the differences between systems appear to be insignificant in both experimental conditions. We only notice a performance degradation when the audio modality is delayed by 25 frames, or 1 full second, although this is not the case when the visual modality is delayed by the same amount. In high noise conditions we notice more fluctuations of the model accuracy. Given the increased difficulty of the task, we can partly attribute this effect to the learning algorithm choosing slightly worse optimum points, so this does not disprove our hypothesis regarding the toleration of modality delays in the Audio-Visual Transformer.

4.4 Discussion

AV Align is the first neural architecture for AVSR that explicitly and automatically learns the asynchronous alignments between the audio and visual modalities of speech. We have demonstrated our results on two large publicly available datasets in the AVSR domain, and the code is publicly shared. This is an important result because it allows the system to capitalise on the visual modality without requiring pre-training strategies, while creating the opportunity to carry out phonetic investigations thanks to its interpretability property. The system learns to discover audio-visual alignment patterns that provide informative visual cues to the audio modality, despite not being explicitly instructed to do so. This result is comparable with previous findings on traditional sequence to sequence neural networks learning the monotonicity of acoustic speech (Chan et al., 2016; Chorowski et al., 2015) or visual speech (Chung et al., 2017), also Chapter 3

in this work, as the decoded graphemes align with their corresponding modality representations. However, before this work it had never been demonstrated that this property holds for the cross-modal alignment between two encoders.

Many researchers have encountered difficulties in capitalising on the visual modality of speech given a dominant acoustic one under low noise conditions. Common solutions resorted to pre-training the visual front-end on a different vision task, or to an alternation between the two modalities in training, where one of them is randomly disconnected when learning the rest of the parameters in the system. The interpretability properties of *AV Align* have given us a greater insight into the nature of the optimisation problem, and motivated us to propose the regression of two lip-related Action Units from visual representations as a secondary objective. We expect the secondary objective to share a subset of visual representations with the primary decoder cross-entropy one. Consequently, we do not expect the AU loss to impact the model's effective representation capacity for the same parameter budget. Our approach greatly simplifies the training strategy, enabling our system to achieve competitive error rate reductions with a fraction of the training data required by other approaches.

Finally, we make a direct comparison with the more popular audio-visual fusion scheme proposed by Chung et al. (2017), although without making use of the full training procedure of WLAS. We show that such an approach can also benefit from the addition of the secondary AU loss, yet to a lesser extent than *AV Align*, confirming the difficulty of learning good visual representations in AVSR. A closer look at the alignment patterns suggests that learning cross-modal correlations as in *AV Align* may be a more suitable approach for AVSR than relating the state of the WLAS bimodal decoder to each modality separately.

A main take-home message from error analysis is that the performance improvements reported with multimodal systems are affected by a high deviation from the mean, leading to a considerable number of sentences where the audio system is ahead by a large error margin. This exposes a fundamental challenge in AVSR, that the visual modality needs to be integrated without impairing the auditory one, which in turn may require mechanisms for assessing the confidence in the visual content. This may warrant a re-evaluation of approaches originally designed for HMM frameworks such as those of Papandreou et al. (2009). Filtering out unreliable video sources may prove particularly important for challenging datasets such as LRS2, as our investigation suggests that neural networks have difficulties in learning this skill automatically. Future AVSR systems may need to be designed and tested with these observations in mind, so they could fall back to audio-only performance whenever the visual modality is not informative. We

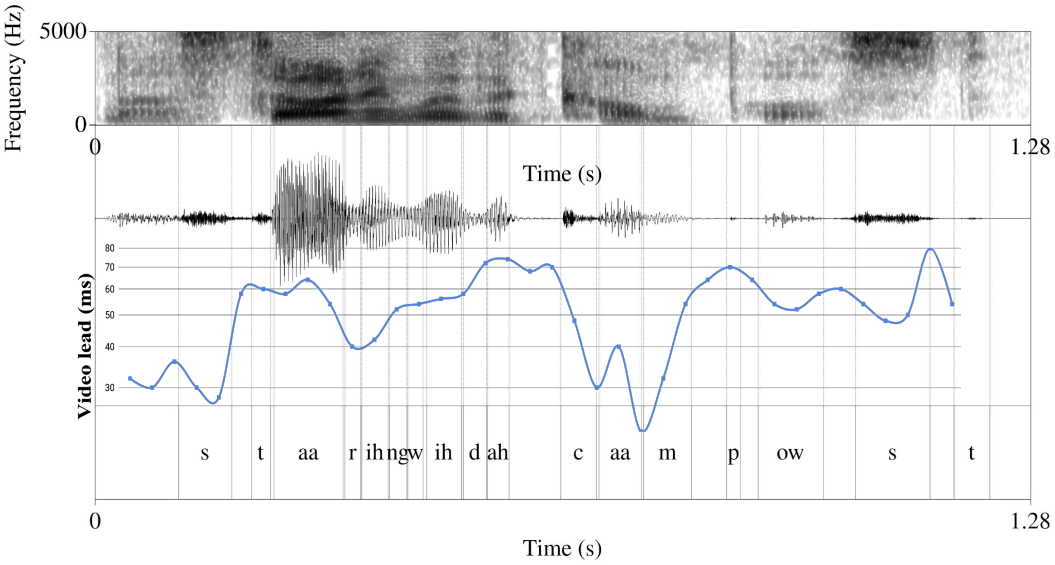


Figure 4.21: *Phonetic analysis of the modality lags predicted by AV Align for the sentence* "Starting with the compost"*, showing the speech spectrogram, waveform, modality lag, and transcription. The delay between modalities is estimated by fitting a normal distribution for each column (audio frame) of the cross-modal alignment matrix and selecting the mean.*

will discuss one such direction in Section 6.2.2.

Overall, this work brings more evidence to support the idea of cross-modal alignment in AVSR, which has been largely overlooked so far. Despite having the entire sentence available for alignment, *AV Align* learns to extract a visual context from a relatively narrow time window. The estimated timing of this context vector, shown in Figure 4.21, suggests that the learnt asynchronies between modalities vary between 20ms audio lead to 80ms video lead, with notable peaks associated with plosive sounds (*t, d, p, t*). This is in line with the precise phonetic measurements of Schwartz and Savariaux (2014), although a deeper analysis is needed to understand the learnt alignments. However, without setting more constraints on the internal states of the two encoder RNNs, it would be impossible to draw reliable conclusions from such analysis. Unlike in the work of Bengio (2002), our alignments are represented in a transformed domain, and we lack guarantees regarding the timing of the higher order representations.

4.4.1 Transformer or LSTM for AVSR?

The results show that the self-attention connections of the Transformer model can successfully substitute the recurrent ones of the LSTM-based AV Align. The cross-modal alignments emerge as locally monotonic based on the dot-product correlations between audio and video representations. Without the auxiliary Ac-

tion Unit loss, the AV Transformer presents the same learning difficulties as the LSTM variant of AV Align, and does not manage to learn monotonic alignments. We have previously speculated that the convergence problem of the visual module in AV Align was partly due to the longer propagation path of the error signal for the visual CNN and RNN in the sequence to sequence structure. Despite the great reduction of this path length in a Transformer network, our AV Transformer still required the AU loss. This demands a deeper investigation into the dominant modality problem in multi-modal machine learning, where patterns need to be discovered in the weaker visual signal. As we saw in Chapter 3, sequence to sequence models are known to be susceptible to encoder-decoder disconnect when the information distribution in the target signal can be exploited for localised optimisation of the decoder, and the audio-visual disconnect is another ramification of the same problem.

Our study does not reflect an analysis of the parameter efficiency of the Transformer network compared to the LSTM for this particular dataset. We opted for commonly used hyper-parameters for datasets of this size, noting that the Transformer model is larger, partly explaining the improvements in error rates. This is because the advantages and disadvantages of both strategies go beyond parameter efficiency, being reflected in hardware throughput and engineering effort, and are discussed in greater detail in Zeyer et al. (2019).

It has been suggested before that Transformers do learn the concept of recurrence from self attention connections. However, despite their highly parallel design conveying significant performance advantages over LSTMs, there is still a sense of wastefulness, particularly in speech, where distant inputs are unlikely to require connectivity. Additionally, the information from one speech frame to another does not change so much as to demand a full update of every representation in a layer.

Despite these inefficiencies, the Transformer architecture achieves faster computation speeds than LSTM on modern hardware for the majority of today's benchmarks. The LSTM blocks are facing more technical and engineering challenges in modern machine learning frameworks, which additionally leads to higher maintenance and development costs. Sutton (2019) argues that general purpose algorithms that best leverage computation scaling appear to be the most successful ones in the long run. The quintessential question becomes: is recurrence a concept that we want to embed into neural networks by hand, or is it preferable to opt for simpler architectures that allow the automatic learning of it?

4.4.2 Do we really need cross-modal attention in audio-visual speech recognition ?

The direct comparison in Section 4.3.10 between feature fusion and AV Align shows that the performance differences between the two methods are not substantial. Furthermore, we have seen that the learning of good visual representations is far more important than the design choices of the neural architecture, as none of the audio-visual systems obtained improvements without being assisted by the Action Unit loss. We may then reflect on whether the inductive bias represented by cross-modal alignment is still necessary in audio-visual speech.

We consider that cross-modal alignment poses unique benefits that make it an attractive option in speech. First, AV Align does not require any data resampling to match the lengths of the two inputs, which is required by the concatenation operation in feature fusion. This simplifies the input pipeline normally handling a wide range of audio and video sampling rates, and avoids the loss of information and redundancies introduced by down-sampling and interpolation respectively. Additionally, it allows the model to scale better with the time resolution of its inputs, connecting with the findings of Tan and Le (2019) that correctly balancing the input dimension may lead to performance improvements. Second, feature fusion does not necessarily bypass the need for modality alignment. Instead, such an operation is modelled internally by the self-attention block of the Transformer taking naturally asynchronous inputs. In other words, what is earned with a more general neural block over the cross-modal inductive bias is lost on the interpretability side. This is a rather bad trade-off for feature fusion, since it was exactly the interpretation of the explicit alignments in AV Align that led to the identification of the alignment problem and the proposal of the AU loss to address it. Our experiments in Section 4.2.4 on modality shifting shows a remarkable tolerance of the audio-visual Transformer architecture to large artificial delays between modalities. We only tested with maximum delays up to two seconds (in the range [-25, +25] frames). However, due to the full connectivity pattern of the Transformer, where any two distant representations in a layer have a direct connection within the self attention mechanism, there is no reason to believe that the lag tolerance property will not hold for even greater delays. This property may suggest that our Transformer model may waste resources for signals such as speech, and we argue that high lag tolerance may not be expected in practice from highly efficient models. Dodd (1977) shows that the human brain can still make use of auditory and visual cues that are phased by 400ms, although with significantly more errors than when the streams are synchronous. Therefore, for

limited time delays, lag tolerance may be an useful property to have for our multimodal Transformer. Limiting the cross-modal attention to well informed windows will be the subject of our investigation in the next chapter.

4.4.3 Limits of AV Align

A major limitation of AV Align is inherited from the formulation of the original sequence to sequence architecture. The decoding process conditions every output token on the entire input sequence, thus can only be used on pre-segmented sequences. Additionally, the computational complexity grows linearly with the input length. Because of this limitation, in speech datasets it is common to manually record short sentences, or force-align longer utterances and automatically split them when longer pauses occur. Consequently, the systems cannot be used in an online setting where the spoken content is expected to be decoded on-the-fly.

In the next chapter we address this limitation by developing a fully differentiable modification of the original sequence to sequence network. Our contribution removes the full sentence conditioning and enables online decoding.

5 Taris

5.1 Introduction

In the previous chapter we proposed a multimodal extension of the Sequence to sequence neural network architecture for the task of audio-visual speech recognition, called AV Align. Beyond the theoretical demonstration of its ability to capitalise on the visual modality of speech, it would be difficult to use AV Align in practice. As we noted at the end of that chapter, AV Align was formulated for the relatively short utterances found in TCD-TIMIT and LRS2, and uses an inefficient mechanism to search for contextualised representations over the full length of the utterance. It would be difficult to extend the method to considerably longer inputs given the quadratic time and memory complexity of global soft attention. In this chapter we are going to address this limitation affecting both AV Align and the family of sequence to sequence neural networks used in ASR.

The main research question addressed in this chapter concerns the revision of the sequence to sequence model for online decoding. My original contribution to knowledge is a strategy named Taris that aims to segment a spoken utterance by learning how to estimate the number of words in it. Taris uses fully differentiable objective functions and therefore greatly simplifies the training and inference stages for speech recognition systems based on neural networks.

Despite the remarkable progress in end-to-end automatic speech recognition technology based on sequence to sequence neural network architectures (Chiu et al., 2018), an unresolved issue is reducing the latency from full utterances down to a few words. The sequence to sequence model conditions every target unit on the full unsegmented audio sentence, being predicated on the principle that a decoder drives the soft segmentation of the input during training. Because the first output token can only be emitted once the entire input sequence has been encoded, this sentence-level, or offline conditioning, is a fundamental barrier in decoding speech in real-time, or online, with a sequence to sequence network. It has been shown that, once convergence is reached, there are pre-

dominantly local relationships between the output tokens and the audio representations in speech (Chan et al., 2016; Chorowski et al., 2015). We have also seen this in Chapter 4, where the monotonicity of the input-output alignments was found as a reliable indicator of the learning success on the speech task. Therefore, potentially incurring no loss in accuracy, an explicit local conditioning of the outputs on the inputs would break the offline limitation and reduce the algorithmic latency. The new challenge is to learn robust associations between input and output substrings which stand for the same linguistic concepts. We argue this is a necessary inductive bias in speech recognition, as the task specification sets no limit on maximum sequence lengths, and the truncation of long sentences is already performed during the collection of speech datasets.

Humans develop the ability to segment words in continuous speech from the earliest stages of life (Jusczyk and Aslin, 1995). There is evidence that we integrate a set of acoustic, phonetic, prosodic, and statistical cues in order to segment words in fluent speech (Johnson and Jusczyk, 2001). Cairns et al. (1994) describe the relationship between speech segmentation and recognition as a chicken-and-egg problem: segmenting units with meaning (e.g. words) from continuous speech posits the recognition of the unit, but the recognition of a unit presumes its a priori segmentation. This leads us to ask whether the ability to segment speech into *word* units with a neural network offers the potential to help crack the challenge of decoding online. This approach would take advantage of the monotonicity of speech, allow the network focus on local properties, and remove the offline conditioning.

To this end, we introduce *Taris*, a Transformer-based system for online speech recognition that learns to model the local relationships between text and audio in speech, relaxing the global conditioning constraint of the original model. We achieve this through self-supervision by introducing an auxiliary word counting task which facilitates the segmentation of speech. Taris allows efficient mini-batch training and introduces a negligible overhead compared to the original Transformer model, without trading off the recognition accuracy. The name *Taris* echoes the misuse of the strong-weak syllable stress rule when learning to segment words by infants exposed to the phrase *the guitar is* (Jusczyk et al., 1999).

5.2 Challenges of end-to-end online decoding

One major limitation of attention-based sequence to sequence neural networks, including the Transformers, is the quadratic time and memory complexity entailed by the sequence-level attention operation. Tay et al. (2020) review several

efforts into reducing this complexity in Transformers, finding that it has become an important topic with over a dozen of articles written in 2020. Related to our work are the approaches limiting the attention span to a local neighbourhood of the query input. For example, Parmar et al. (2018) develop 1D and 2D local self-attention models for multiple image processing tasks. In speech, Povey et al. (2018) plot the importance of the speech frames for the self-attention mechanism in the range of [-45; 45] audio frames relative to the query index, showing that the highest weights are predominantly allocated to frames around the origin. Moritz et al. (2020) also limit the attention span to a fixed size window for developing a latency controlled CTC-based speech recogniser. While encoding the audio signal in a windowed fashion appears straightforward and addresses the latency aspects of the HMM approach of Povey et al. (2018) or the CTC one of Moritz et al. (2020), an issue persists for the class of sequence to sequence models. More precisely, there is no obvious way of limiting the attention span from the decoder to the encoder. Each token in the label sequence naturally spans a variable number of speech frames, depending on the sound produced or the speaking rate among many other factors. As a consequence, online speech recognition with a sequence to sequence architecture poses an unique challenge. This leaves the question of mapping text labels to speech inputs in a time-restricted fashion an open problem.

It is important to note that this problem is specific to the attention-based sequence to sequence models, whereas alternative approaches may be more conveniently modified to decode online. The HMM enforces monotonicity in its left-to-right structure, and has available a set of exit states that inform about phoneme or word boundaries. Similarly, the CTC and RNN-T family of models making frame-synchronous predictions can also detect boundaries between units whenever a new prediction is non-blank and different from the previous one. This chapter aims to address the problem on the seq2seq side, as it will provide an alternative set of tools for tackling online decoding while side-stepping some of the limitations of the state-space models.

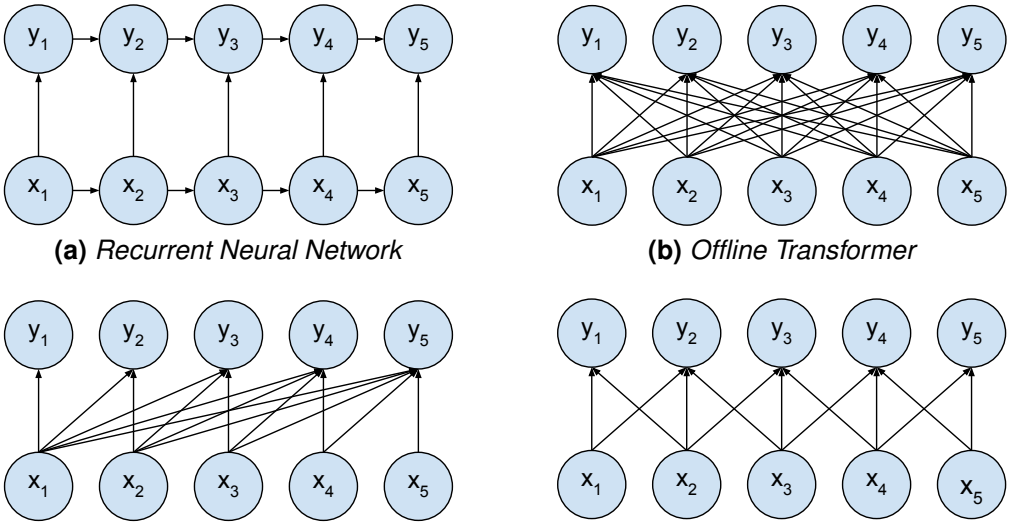
There have been several attempts to address the limitation of decoder-encoder global soft attention in seq2seq architectures. Chorowski et al. (2015) investigate a windowing technique that considers a fixed length sub-sequence of the encoder representations for each decoding timestep based on the median location of the alignment distribution at the previous timestep. Later work of Bahdanau et al. (2016) notes that median-centred windowing is highly dependent on the quality of the previous alignment. To overcome it, Bahdanau et al. (2016) heuristically define an even larger window based on the statistics of the silence segments at

the two ends of the input sequence, and of the character-to-audio-frame length ratio, both estimated on their training set. Although this approach may limit the attention span, it was only designed to provide a good initialisation for the decoder-encoder alignment distribution, without explicitly considering the character/word latency for online decoding. Another direction of research is represented by the explicit monotonicity constraints of Chiu and Raffel (2018); Raffel et al. (2017), using either hard (single frame) attention, or soft attention within a fixed length segment whose location it determined through a hard sampling process. As we discussed in Section 2.7, since sampling precludes the use of backpropagation, the method of Chiu and Raffel (2018) requires the computation of an expected value of the context vector at each decoding timestep. Zeyer et al. (2021) examine this family of strictly monotonic approaches, noting that an implicit assumption is the existence of a deterministic method for predicting the position of next target label. From their point of view, such assumption may be too strong in speech, and can potentially propagate wrong label predictions from which a beam search algorithm may not fully recover the correct transcript. Taris mitigates this through a couple of mechanisms, including segment-level soft attention corresponding to each character from each word, and a segment look-ahead design. This makes the prediction of each character in a word aware of the acoustic segments associated with a number of words in its left and right context.

The nature of the speech signal does not allow a robust identification of word boundaries right when they occur in fluent speech, as discussed by Auer Jr and Luce (2005). This means that the acoustic history alone does not provide sufficient information to predict word boundaries. It is then necessary to design a mechanism that delays the boundary decision until sufficient confidence has been gathered from boundary informative cues.

Recurrent Neural Networks define connections between input timesteps in a strictly causal way. This can be seen in Figure 5.1a, where any state y_i is a function of the current input state x_i and the previous state y_{i-1}. Consequently, this design may limit the ability to learn the speech segmentation task.

The Transformer proposed by Vaswani et al. (2017) is a better candidate for this task and we choose it as a foundation for our system Taris. Unlike the recurrent neural network that uses causal connections between timesteps (Figure 5.1a), the Transformer allows feature contextualisation at the sequence level through self-attention. Illustrated in Figure 5.1b, we see that a state y_i is also conditioned on inputs $x_{j>i}$. Although it maintains the sentence level conditioning, this offline modelling strategy provides a theoretical upper limit of the segmentation performance. The self-attention connections in the Transformer block can be adjusted



(a) *Recurrent Neural Network*

(b) *Offline Transformer*

(c) *Causal Transformer*

(d) *lookback and lookahead of k=1 frames*

Figure 5.1: *Illustration of the typical connectivity patterns at the sequence level for representation learning with RNNs and Transformers. Note that $x_{1:5}$ and $y_{1:5}$ denote sequences of abstract internal representations between any two intermediate layers in a deep stack, and not the actual input and output sequences.*

to allow causal modelling (Figure 5.1c) or non-causal modelling with a window (Figure 5.1d), allowing us to see how the word segmentation performance degrades as we limit the available context. The window length is directly linked to the algorithmic latency of Taris and its accuracy, and we investigate this trade-off in Section 5.5.4.

5.3 Taris

Taris takes as input a variable length sequence of audio vectors $A = [a_1, a_2, ..., a_N]$ and applies the Encoder stack of the Transformer model defined in Vaswani et al. (2017). Because of latency considerations, instead of the original full connectivity in Figure 5.1b, we use the type displayed in Figure 5.1d, with controlled look-back e_{LB} and look-ahead e_{LA} frames. We denote the outputs of the encoder as:

$$o_A = \text{EncoderStack}(A, e_{LB}, e_{LA}) \tag{5.1}$$

To obtain a soft, differentiable estimate of the word count from the encoder representations, we start by applying a sigmoidal gating unit on each encoder output

o_{A_i} to obtain a scalar score for each frame:

$$\alpha_i = \text{sigmoid}(o_{A_i} W_G + b_G) \tag{5.2}$$
$$\text{where sigmoid}(x) = \frac{1}{1 + \exp(-x)}, W_G \in \mathbb{R}^{d_{model} \times 1}, b_G \in \mathbb{R}^1$$

We assign to every single input frame i a segment index $\hat{w}_i$ by taking the *cumulative sum* of α and applying the *floor* function on the output:

$$\hat{w}_i = \left\lfloor \sum_{j=1}^{i} \alpha_j \right\rfloor \tag{5.3}$$

Namely, the first predicted segment is delimited by a cumulative sum of α between 0 and 1, the second segment by the same quantity between 1 and 2, and so on. This will be later illustrated in Figure 5.2, where the intersections between the horizontal lines and the plot of the cumulative sum of α designate the timestamps of the segment boundaries.

During training, the Decoder stack receives the labelled grapheme sequence $Y = [y_1, y_2, \ldots, y_L]$ made of English letters and the unique word delimiter SPACE. We assign to every grapheme k a word index w_k by leveraging the SPACE tokens in the labelled sequence:

$$w_k = \sum_{j=1}^{k} (y_j == \text{SPACE}) \tag{5.4}$$

Thus, whereas symbolic segmentation of speech uses a unique SPACE token to separate words, acoustic segmentation flags word boundaries by tracking the frame locations where the partial sum of the word counting signal α_i passes to the next integer value.

We modify the decoder-encoder connectivity of the Attention layer of Vaswani et al. (2017) to allow our decoder to perform soft-alignment over a *dynamic* window of segments estimated by the encoder. More precisely, we only allow those connections for which the following condition is met:

$$F = \widehat{W}_{ik} \leq (W_{ik} + d_{LA}) \textbf{ and } \widehat{W}_{ik} \geq (W_{ik} - d_{LB}) \tag{5.5}$$

In (5.5), d_{LA} and d_{LB} denote the number of segments the decoder is allowed to look-ahead and look-back respectively. The W and $\widehat{W}$ matrices are obtained from the w and $\hat{w}$ arrays by applying the tile operation, which repeats one sequence for a number of times equal to the length of the other one. For example, if we

assume a 4-word sentence with w = [000111222333] and $\hat{w}$ = [0123], tiling generates two matrices W and $\widehat{W}$ of the same shape 12x4 by repeating $\hat{w}$ 12 times row-wise, and w 4 times (and transposing). The association between the indices of these matrices is then extended to support segment look-back and look-ahead. More generally, F is a 2D matrix $\in \mathbb{R}^{N \times L}$ that defines the admissible connections between any decoder timestep and any encoder timestep, acting as a bias on the decoder-encoder attention. Setting F as a matrix of ones recovers the original Transformer model. The extension to 3D tensors that include the batch dimension is straightforward, offering Taris efficient minibatch training and inference.

The decoder implements a traditional character level auto-regressive language model that predicts the next grapheme in the sequence conditioned on all the previous characters and the dynamic audio context vector c_k:

$$c_k = \text{Attention}(\texttt{keys} = o_A, \texttt{query} = o_{D_{k-1}}, \texttt{mask} = F) \tag{5.6}$$

$$o_{D_k} = \text{DecoderStack}(Y, c_k) \tag{5.7}$$

$$p_k \equiv P(y_k | c_k, Y_{1:k-1}) = \text{softmax}(W_\eta o_{D_k} + b_\eta) \tag{5.8}$$

$$\text{where } W_\eta \in \mathbb{R}^{\eta \times d_{\text{model}}}, b_\eta \in \mathbb{R}^\eta$$

In (5.8), η is the alphabet size of 28 tokens representing the 26 English letters, space, and apostrophe. We measure the difference between the estimated word sum $\Sigma\hat{w} = \sum_i \alpha_i$ and the true word count $|w| = \sum_k (y_k == \text{SPACE})$ as:

$$\textit{Word Loss} = (|w| - \Sigma\hat{w})^2 \tag{5.9}$$

We define the training loss as:

$$\textit{CE Loss} = \frac{1}{L} \sum_k -y_k \log(p_k) \tag{5.10}$$

$$\text{Loss} = \textit{CE Loss} + \lambda \textit{ Word Loss} \tag{5.11}$$

In all our experiments we used a scale factor λ = 0.01 found empirically. The self attention connections of the Decoder are causal as in Figure 5.1c, since the model has to be auto-regressive.

Overall, Taris can be seen as a sequence to sequence neural network with a dynamic attention window between the decoder and the encoder, where the window location is driven by an indirect speech segmentation mechanism implemented in the encoder.

5.3.1 Latency analysis

Calculating the delay between the first audio frame timestamp and the first output unit is non trivial and depends on several factors.

First, the encoding look-ahead and look-back parameters e_{LA} and e_{LB} define the receptive field in audio frames of a learnt audio representation o_{A_i}. The absolute encoding delay is a function of e_{LA}, the number of layers in the Transformer network, and the audio frame duration. We provide in Section 5.5.2 a more precise measurement of the encoding latency.

Second, the decoding look-ahead parameter d_{LA} defines the number of audio segments required to start decoding the first grapheme in an output word unit. Although this number of segments is constant, the frame length of d_{LA} segments is dynamic and context dependent. In other terms, the first grapheme in the next word can be decoded once the cumulative sum of α_i becomes greater or equal to d_{LA}.

The dynamic context makes it difficult to provide any word decoding latency guarantees and measurements. The difficulty first lies in adequately defining what is meant by latency in our framework. Given that Taris tracks the cumulative sum of an ever increasing scalar variable α_i, a natural definition of latency in this case is the time difference between the event timestamp of this variable reaching an integer value and an oracle word boundary timestamp of the corresponding word. If we use a non-zero value for the encoder look-ahead e_{LA}, we would have to add the length of the right-sided receptive field as well. However, Taris does not offer the guarantee of a bijective correspondence between its estimated segments and the true word timestamps. This would only be possible in those cases where Taris decodes with a 100% accuracy the words in the evaluation sentence, without making any insertion or deletion errors. In principle, this resembles the setup of a forced aligner, where the model additionally receives the sequence of correct words as input, and only needs to align them with the audio signal. Yet, Taris is not designed to work in a forced alignment mode.

We conclude that devising a precise latency metric represents an interesting but substantial extension to our work. As a result, in this chapter we will illustrate the average word decoding latency of Taris using a simpler approach that relaxes the constraints of the definition above. In particular, we will first report the word counting loss from Equation (5.9) in order to verify if the counting task can be solved reasonably well using only the encoded audio representations. This metric alone would be insufficient for our online decoding task, since it does not provide an estimate of the segment boundary events. We will additionally plot

the histogram of the lengths of all segments obtained at inference. Any similarity with the distribution of the true word durations may be indicative of a non-trivial segmentation, and may at least offer a broad estimate of the segment decoding latency, despite not revealing if individual segments actually correspond to genuine words. As in this chapter we are mainly interested in the mechanics of limited context decoding with sequence to sequence neural networks, we will leave a deeper investigation as a future extension of this work.

Finally, we note that an alternative decoding approach in Taris is to gradually increase the segment look-ahead from 0 to d_{LA}, and consequently to provide up to $d_{LA} + 1$ updates for the same word. This can be more practical when an immediate, less accurate transcription is needed, accepting that it is subject to corrections depending on the future context.

5.3.2 Complexity analysis

Taris requires a negligible overhead in parameters and operations over the original Transformer. The only extra parameters are given by the W_G and b_G variables in Equation (5.2), which amount for $d_{model} + 1$ scalars in the total model size. Equations (5.2)-(5.5) describe the additional operators mainly consisting of a matrix vector multiplication followed by a sigmoid activation for every audio frame, and the update of a scalar cumulative sum. Since attention masking is already performed by the original Transformer to take into account the true input length in a minibatch, Equation (5.6) does not represent an overhead, and the only additional operation needed at each decoded timestep is the computation of the segment mask F in Equation (5.5). In training, this mask is computed only once per batch, since we have access to the full output sequence and know the positions of all the *SPACE* tokens in advance. The mask F is directly applicable on the tensor product performed by the Transformer architecture between the queries and keys by adding a large negative value outside the mask before applying the softmax operation. This grants Taris a highly efficient computation strategy that integrates with the Transformer.

Note that Taris applies a limit to the number of past audio frames when updating the encoder representations through e_{LB}, and to the number of past encoded segments for the context of each word in the targets through d_{LB}. While these two parameters do not affect the latency with respect to identifying segment boundaries, their purpose is to ensure that the complexity of soft attention for the unmasked locations does not grow with the sequence length. This property enables real-time inference on continuous streams of data, since the memory buffers can

be flushed once the past audio segments are no longer needed for making new predictions.

5.3.3 Considered alternatives

Attention-based sequence to sequence models learn an explicit alignment between the output tokens and the speech frames. We initially considered to leverage the alignments corresponding to each SPACE token of a pre-trained offline model, and use this as a supervision signal to train the gate α_i from Equation (5.2). However, visually inspecting these alignments revealed that there is no clear delimitation between the spoken words in English, with the softmax weights not being skewed towards a low number of frames. This observation is in line with our intuition that the SPACE token rarely corresponds to a short pause in English speech, and instead has a more analytic role which demands its inference from the acoustic differences between several words, or from the intrinsic language model in the decoder. In effect, such alignment information would only represent a crude approximation of the true boundaries between the spoken words, and it would be a very noisy supervision signal for our gate α_i to learn.

In Taris, we can choose to constrain the output of the gating unit α_i in Equation (5.2) to follow a specific distribution. For example, Hou et al. (2020) train their gating unit to follow a Bernoulli distribution, making values very close to 0, or very close to 1, more likely. During our initial experiments with a scaled sigmoid function (i.e. $1/(1 + exp(-kx)), k > 1$), we noticed that this unit does not typically have values close to the extremities of the range, and achieves a slightly higher word counting loss than standard unscaled sigmoid. We speculate that the gating unit learns to *accumulate* cues at the sub-word level in order to solve the word counting task, and an eventual binary output behaviour may only be feasible when coupled with a recurrent process to keep track of an internal state, which in turn would complicate the design. In our work, α_i is predicted directly from the hidden state o_{A_i} with a feed-forward neural network.

As an alternative to the sigmoid activation we also considered the hyperbolic tangent function, which has an output range between -1 and 1. The negative output values have the potential to enable a broader range of word counting strategies, such as assigning higher confidence scores and eventually correcting them later based on future evidence. With the sigmoid activation, the system does not have the opportunity to make corrections and has to adopt a more defensive approach. On the other hand, a sufficiently large receptive field may reduce the need for such corrections. In our initial experiments we did not see

a significantly improved word loss with the *tanh* activation, and, since Taris is a relatively new model, we decided to apply the law of parsimony and maintain the *sigmoid* until empirical evidence demands otherwise.

5.3.4 Comparison to related work

Dong et al. (2020) categorise end-to-end speech recognition models into label-synchronous and frame-synchronous models. The former refers to models that derive the contextual acoustic units from the soft alignment with the state of an auto-regressive decoder receiving grapheme labels as inputs. In contrast, the latter derive acoustic labels directly from the audio representations by removing the decoder, and thereby do not model the conditional dependence between the labels.

Taris is more closely related to the label-synchronous class, as it maintains a soft attention mechanism between the decoder and the encoder. However, Taris derives a segmentation signal directly from the audio representations, and the soft alignment is only allowed within a well defined dynamic window. This contrasts the model proposed by Dong and Xu (2020), which also predicts a normalised weight per frame, but uses these weights directly once they sum up to approximately 1.0 to linearly combine the corresponding audio representations into a single state from which the segment label is estimated. An approach similar to the one of Dong and Xu (2020) was previously introduced in Li et al. (2019), however they only test the method on Mandarin speech. Li et al. (2019) anticipate problems on languages such as English with less clear boundaries between linguistic units and complex orthographies.

Taris is more closely related to the approach of Hou et al. (2020) performing segment level attention. However, Hou et al. (2020) take a different approach to train the boundary detection unit by sampling from a Bernoulli distribution, which makes the model non-differentiable, and resort to policy gradients. Experimentally, they find that the cumulative sum of the boundary unit requires a dynamic threshold ranging from 0.2 to 0.55 for optimal decoding performance. This suggests that the approach we take with Taris not enforcing a specific distribution on the output of the gating unit, and only requiring the total sum to be close to the word count, is likely enabling the learning of a more flexible counting mechanism. The sigmoidal unit in Taris does not enforce the notion of a hard boundary, but instead we design the decoder to analyse a limited acoustic range covered by the cumulative sum of the gating unit.

5.3.5 Audio-Visual Taris

For the reasons discussed in Section 1.1, the visual modality does not contain sufficient linguistic information to allow the prediction of word boundaries. As a result, we cannot use the same counting strategy as with the audio modality in order to segment visual speech. Learning to segment the audio modality was necessary because the auditory and symbolic modalities of speech exist on different timescales, and we found the concept of words as the linking element between them. However, the audio and video modalities share the same time axis and can be integrated more easily, by only taking into account the different sampling rates. Having prior knowledge of the natural asynchrony between auditory and visual speech allows us to set an upper limit on the audiovisual integration window.

To this end, we describe an audiovisual extension of Taris. Given a sequence of visual representations $V = [v_1, v_2, ..., v_M]$ corresponding to the audio track $A = [a_1, a_2, ..., a_N]$ of the same spoken utterance, we define a symmetrical integration window of length $2B+1$ centred on a visual frame index j and apply a *constrained* cross-modal alignment between modalities:

$$c_{V_i} = \mathit{Attention}\,(o_{A_i}, o_{V\ j-B\ :\ j+B}) \tag{5.12}$$

$$j = \left\lfloor (i+1)\frac{N}{M} \right\rfloor - 1 \tag{5.13}$$

For any audio frame i, the index j is calculated as the nearest time-aligned video frame, e.g. audio frame 50 corresponds to the video frame 25 when the audio has twice the sampling rate of the video (i.e. $N = 2M$). Compared to c_{V_i} in Equation (4.5) on page 84 of the offline multimodal architecture in Chapter 4, alignment is performed within a window of $2B + 1$ visual frames, which is only a fraction of the full length M of the visual sequence. Consequently, an audio representation only depends on temporally local video representations, preserving the eager decoding property of Taris.

The audio and visual representations are integrated as we have presented before in Section 4.2.3:

$$o_{AV} = c_V + o_A \tag{5.14}$$

To complete the online audio-visual design, we only need to predict the gating signal α_i from the fused representations o_{AV} instead of the audio ones seen in

Equation (5.2):

$$\alpha_i = \text{sigmoid}(o_{AV_i} W_G + b_G) \tag{5.15}$$

This strategy allows us to investigate if Taris can learn to count words in fluent speech better from the fused audiovisual representations instead of the audio ones alone.

5.4 Why learn to count words?

Proper lexical segmentation of speech depends on context and semantics, as commonly illustrated by the example *how to wreck a nice beach* sounding similar to *how to recognise speech*. Thus, strategies incrementally scanning for hard boundaries (Dong and Xu, 2020; Hou et al., 2020; Li et al., 2019) are less suited to word units, prompting Dong and Xu (2020) to perform beam search on the entire sequence of sub-word tokens estimated from each segment. Instead, Taris has to develop intrinsic word counting mechanisms. One plausible strategy is to incrementally gather lexical evidence at the sub-word level, and learn to represent boundary-informative acoustic cues on a manifold where they can be accumulated.

We conjecture that learning the ability to count words facilitates the segmentation of speech into words, and we discuss below our intuition behind it.

In Figure 5.2 we illustrate the word counting sub-problem to be solved by the network. Starting in the bottom left corner, the network predicts scores for every audio frame in the sentence, and the cumulative sum is promoted to get as close as possible to the total word count, shown with a red circle. There is a very large number of paths that can be taken to reach the target count. However, when trained on large amounts of naturally distributed speech, we predict that Taris converges towards genuine word segmentation by having the cumulative sum cross all the intermediate word boundaries shown with yellow circles. In other words, the network may learn to self-normalise the accumulated probabilities for each word regardless of their length or cued structure.

We believe it suffices to train a system with the right amount of speech data, with the following intuition. As words appear in multiple contexts throughout a dataset, learning to count words may then have a normalisation effect on the fraction of $\Sigma\hat{w}$ allocated to each word in a sentence. Each word unit will approach a unitary mass allocation as its acoustic realisation is seen more often in multiple contexts. For the less frequent words, the correct allocation may happen by

marginalisation if the sentences they appear in contain relatively more frequent words. Loosely speaking, it is the task of solving a system of linear equations where the variables are the partial sums corresponding to the acoustic frames between two consecutive estimated boundaries.

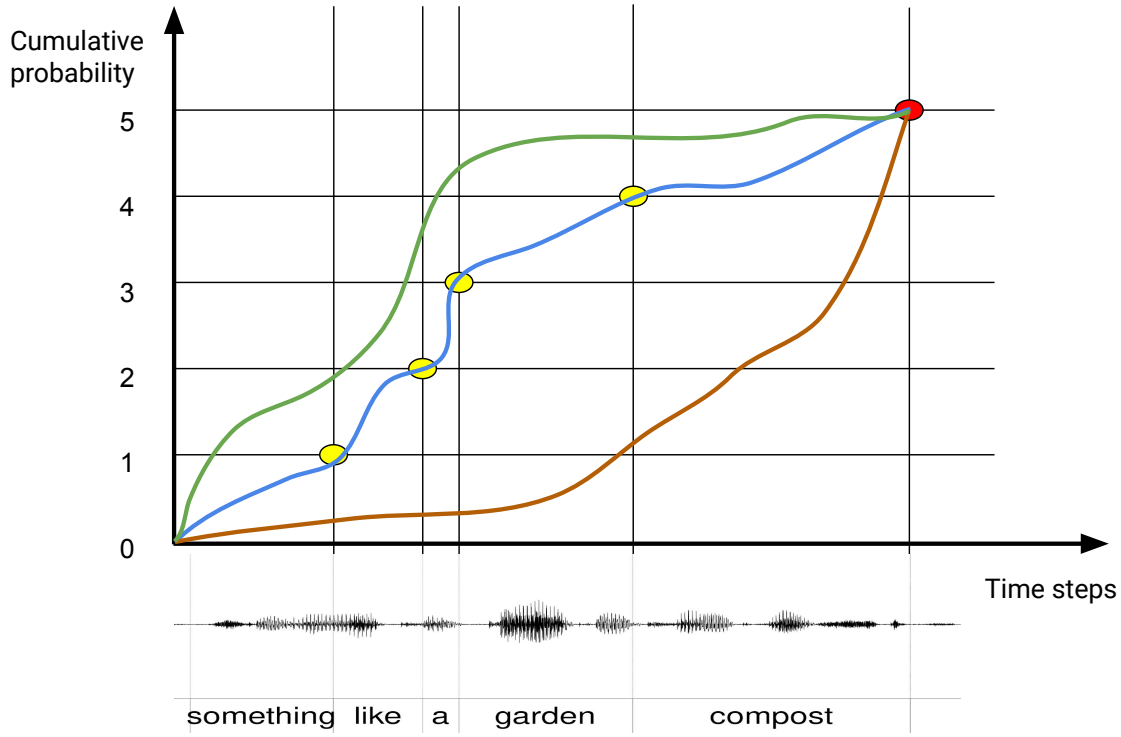


Figure 5.2: *Illustration of the word counting problem. Given a spoken phrase* something like a garden compost*, the network has to reach the correct word count 5 taking any path starting in the bottom left corner. The blue path allows an easy segmentation of speech into words. The green and dark orange paths are possible, but do not facilitate segmentation.*

Since we do not explicitly model the pauses between words, and the convergence towards the segmental behaviour is a mathematical conjecture without analytic proof for now, it is likely to observe deviations in practice on learnt solutions. However, Taris does not require a very strict approximation of word boundaries to function correctly. Instead, it is sufficient to just avoid frequent under- and over-segmentation, as it directly impacts the model's latency.

5.5 Experiments and results

We first conduct our experiments on the audio part of the unconstrained speech dataset LRS2 (BBC and University of Oxford, 2017) for rapid prototyping, and on the 100h partition of LibriSpeech (Panayotov et al., 2015) for empirical validation at a larger scale. Both datasets contain English speech. We later evaluate Taris

on Mandarin speech from the AISHELL-1 dataset (Bu et al., 2017). The audio features A in Equation (5.1) are extracted as in Chapter 4.

In the case of **LRS2**, we follow exactly the same setup as in Chapter 4, using the same dataset partitioning, the same audio features, and the same strategy for corruption with additive cafeteria noise at a SNR of 10db, 0db, and -5dB.

For **LibriSpeech** we choose the 100-hour training subset of clean speech containing 28,539 training samples, and we evaluate on the recommended clean test set of 2,620 samples. The data in LibriSpeech is derived from read audiobooks, which can be considered less challenging than the broadcast news content from LRS2. One benefit of LibriSpeech over LRS2 for our work is the considerably longer duration of sentences. This will allow us to evaluate more reliably the segment look-back and look-ahead mechanisms of Taris, as well as the accuracy of the word counting mechanism for a higher total word count.

For **AISHELL-1,** we use the recommended data partitioning from the original article of Bu et al. (2017). Our aim is to investigate how suitable is the word counting mechanism of Taris to a language different from English that is also not part of the Indo-European language family. AISHELL-1 contains 120,098 training samples and 7,176 test samples of Mandarin speech from different areas of China. The sentence length in this dataset varies from 1 second up to 14 seconds. More specific details of this dataset will follow in Section 5.5.7.

Our implementation of Taris forks the official Transformer model in TensorFlow 2 (The TensorFlow Model Garden, 2020). We train our LibriSpeech models for a total of 500 epochs at an initial learning rate of 0.001, decayed to 0.0001 after 400 epochs. The training time is approximately 200 seconds for a single epoch of LibriSpeech 100h on an Nvidia Titan XP GPU. The LRS2 models were trained with the same learning rates for 100 and 20 epochs respectively, on each noise level.

5.5.1 Neural network details

Our models use 6 layers in the Encoder and Decoder stacks, a hidden model size $d_{model} \equiv h = 256$, a filter size $d_{FF} = 256$, one attention head, and 0.1 dropout on all attention weights and feedforward activations. The models occupy 25 MB on disk, and are considerably smaller than the typical size of state-of-the-art models used in benchmarks. We chose this model size so we could train it on a single GPU of 12 GB of memory with a minibatch size of 64. We presume that a larger model may bring a similar level of improvement to both the online and the offline systems if we wanted to pursue a better absolute accuracy. This would come at

the cost of slower, more expensive training iterations.

5.5.2 Analysis of the receptive field

As we described our data pre-processing setup in Section 4.3.1, one audio frame is obtained by stacking 8 STFT frames taken over 25ms windows with 10ms strides. Each new audio frame includes the previous 5 STFT frames, so the additional non-overlapping information is represented by 3 STFT frames. In greater detail, the first audio frame achieves an effective range from 0ms to 95ms. The second audio frame starts at 30ms going up to 125ms, followed by the third frame from 60ms to 155ms, and so on.

The first layer in our Transformer encoder network has a receptive field in frames controlled by the e_{LA} and e_{LA} parameters. We preserve the same mask throughout the entire Transformer stack. This means that the superior layers can access a broader receptive field with respect with the audio input. A representation at position k in the Transformer layer l is then indirectly conditioned on the audio input up to the position $k + l \cdot e_{LA}$. We leave the fine tuning of this connectivity design for latency optimisation as future work.

5.5.3 The End-of-sentence token

During our initial experiments, we noticed that traditional evaluation and training strategies for sequence to sequence neural networks in speech recognition are commonly misusing the End-of-sentence (EOS) token, making it difficult to evaluate online systems. The commonly used ASR/AVSR datasets are a collection of variable-length utterances, and the system's accuracy is computed for each utterance using an edit distance based algorithm. These utterances are often fragments from full spoken sentences. For example, Afouras et al. (2018b) describe a multi-stage pipeline for constructing the LRS2 dataset from long recordings, which only retains those segments where the voice and the lips are in coherent. In addition, the segments are clipped to a maximum length of 10 seconds, or 100 characters, due to GPU memory constraints. The phrase illustrated in Figure 5.2, *"something like a garden compost"*, is one such instance of an excerpt from a sentence. Sometimes, the fragmentation includes the ending and the start of two consecutive sentences, with the punctuation removed from the ground truth transcription. This is the case with the LibriSpeech dataset, whose training partitions can have segments up to 35 seconds long. In other words, the ASR system does not receive full sentence units, and cannot develop the linguistic notion of an end of sentence. In our initial experiments it became obvious

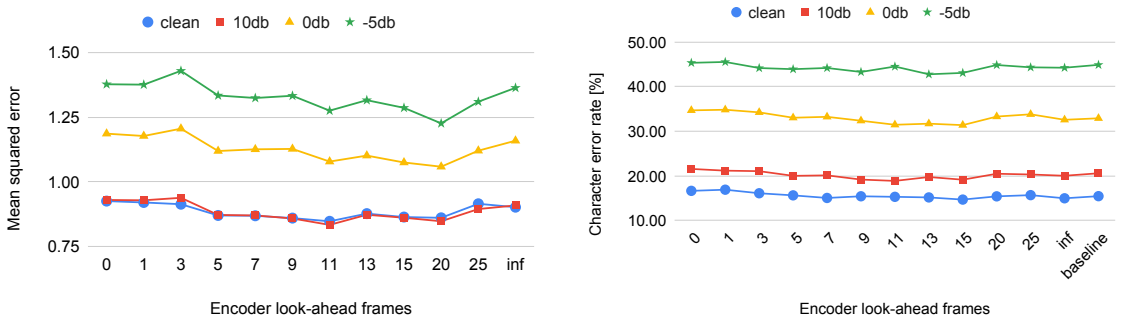
that one way the sequence to sequence model differentiates between an EOS token and a word delimiter (SPACE) likely comes from the a priori knowledge of the sentence length, and that EOS becomes more likely as the decoder-encoder alignment distribution advances towards the last remaining audio frames in the sentence.

The aspect above becomes problematic in an online setting for our approach, as the decoder is fed with a limited acoustic context. Given the nature of the dataset utterances, Taris does not have sufficient information to know when to stop the decoding process, as EOS cannot be estimated even spuriously anymore. Decoding with Taris would often stop after just a few words in an utterance, owed to the premature prediction of EOS, disparagingly biasing the accuracy on longer sentences. A potentially related issue of partial transcripts was reported by Chorowski and Jaitly (2017). In their case the problem stemmed from a higher cost associated with the continuation of the decoding process on long inputs, and their solution was to introduce an input coverage cost proportional to the number of unused speech frames during decoding. However, this solution assumes that the entire input sequence is available to the decoder to compute the coverage penalty, and is not applicable to an online setting.

To circumvent this problem, we made two important changes to the traditional seq2seq model. First, we replaced the EOS token in the labels, which cannot be predicted reliably, with the SPACE token. Second, we modified the stopping condition of the beam search inference decoder as follows: instead of stopping when all beams reach the EOS token, it now stops when the decoder predicts as many words as there were estimated by the audio encoder (the rounded value of $\Sigma\hat{w}$). This new strategy is mostly beneficial to the evaluation procedure, but should also be useful in practice as it allows the decoder to emit a controllable number of words. With this change, we are able to evaluate the error rate of Taris on full test sentences for which we lack any label alignments.

5.5.4 Learning to count words in auditory speech

We first investigate to what extent a sequence to sequence Transformer model can learn to count the number of words from audio data on LRS2. We train offline Transformer models with different values of the encoder look-ahead frames hyper-parameter e_{LA}. This will measure the change in the *Word Loss* cost from Equation (5.9) as more future context becomes available. All models use $e_{LB} = \infty$ frames and $d_{LA} = d_{LB} = \infty$ segments since it does not affect the decision latency of sigmoidal gating unit α. The results are shown in Figure 5.3a.



(a) *Word Loss on the test set of LRS2* **(b)** *Character Error Rate on the test set of LRS2*

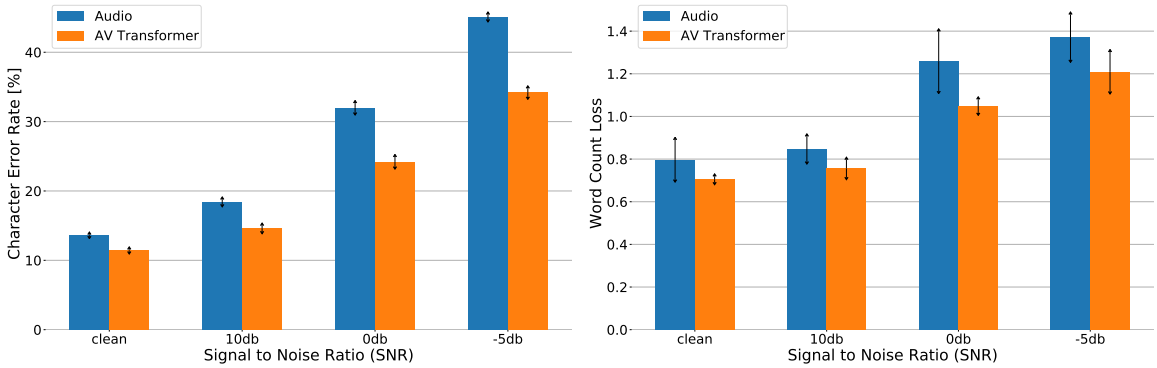
Figure 5.3: *Offline system evaluation for an increasing length of feature contextualisation in the encoder e_{LA}. The look-back parameter of the encoder e_{LB} is set to ∞. Both d_{LA} and d_{LB} are set to ∞. The "inf" label denotes that the encoder can access the entire future context of a sentence, therefore going up to N audio frames. The "baseline" label on the right plot denotes a standard Transformer encoder that does not use a Word Loss penalty.*

We see that the mean squared word count error is sub-unitary in clean speech and 10db noise, i.e., the estimated count is less than one word away from ground truth. This suggests that words can be counted relatively well from auditory speech. In addition, using a future context length of 11 frames offers the lowest counting error under all noise conditions.

In Figure 5.3b, we plot the mean Character Error Rate achieved by all our systems, including the offline Transformer baseline that does not use the auxiliary Word Loss. We observe no significant difference in the decoding accuracy. Not shown here to increase the plot intelligibility, the 95% confidence intervals of the mean errors are between 1% and 1.4%. Therefore, the auxiliary word count objective is not detrimental to the original accuracy obtained on LRS2 using only the cross-entropy loss. This suggests that our secondary task shares similar representations that are already learnt for decoding. Compared to Figure 5.3a, the curves plotted in Figure 5.3b appear more flat as e_{LA} varies. It is difficult to appreciate whether or not we should have seen steeper slopes in Figure 5.3a, since the generic nature of the sigmoidal gating network does not facilitate an investigation into the types of cues learnt.

5.5.5 Learning to count words in audio-visual speech

Next, we are interested in studying if the word count in fluent speech can be estimated with a higher accuracy from audio-visual cues than from the audio modality alone. In Section 5.5.4 we saw that the encoding look-ahead length does not have a major influence on either the word counting error or the character



(a) *Character Error Rate* **(b)** *Word Count Loss*

Figure 5.4: *Evaluation of the offline Audio and the Audio-Visual Transformer on LRS2 with the word counting loss enabled*

error rate. Therefore, in this experiment we limit our analysis to counting words from audio-visual representations with the offline models having infinite context available. We train Audio and Audio-Visual Transformer models on LRS2 and repeat the experiment for five different random initialisations. We plot the average Character Error Rate and the Word count loss of the two systems in Figure 5.4. The arrows indicate the standard deviation across the five trials. Following our findings in Chapter 4, the Audio-Visual Transformer uses the auxiliary Action Unit loss.

From Figure 5.4b it can be seen that the average word count loss of the Audio-Visual Transformer is slightly lower than the one of the Audio model, while the recognition accuracy shown in Figure 5.4a stays approximately the same as in Section 4.3.9, where we did not use the Word Loss. This aspect suggests that the visual cues may be informative of word boundaries in fluent speech, although it is difficult to draw conclusions regarding the statistical significance of this result from only 5 trials.

5.5.6 Online decoding accuracy

The decoder in our previous experiments had access to the entire encoder memory. For our online models in this section we opt for an encoder look-ahead e_{LA} of 11 frames and infinite look-back $e_{LB} = \infty$, as we showed in Section 5.5.4 that there are diminishing gains beyond this threshold.

LRS2

In this experiment we evaluate the Character Error Rate of Taris on LRS2 for an increasing number of decoder look-ahead segments d_{LA}, while setting the look-

back value $d_{LB} = \infty$. For a practical online model it may be a good trade-off to limit the decoder look-back context to a single sentence when transcribing continuously. We plot the Character Error Rate in Figure 5.5 for an increasing number of acoustic segments that the decoder is allowed to attend to.

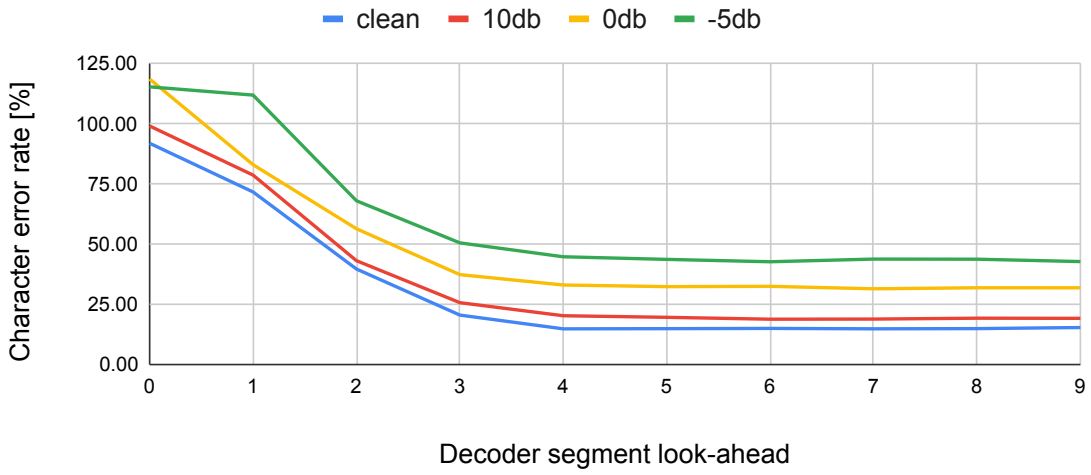


Figure 5.5: *Audio-only online decoding error rate on LRS2. We fix $e_{LB} = \infty$, $e_{LA} = 11$ frames, $d_{LB} = \infty$ and we only allow d_{LA} to vary.*

We notice that there are diminishing returns after a context look-ahead d_{LA} of 4 words. The overall accuracy beyond this threshold is comparable to the offline systems in Figure 5.3b. We can roughly estimate an average decoding latency of around 1 second for which we no longer expect Taris to update the partial hypotheses for the previously decoded words. This estimate is based on the histogram of the word length distribution later shown in Figure 5.6. This result shows that it is feasible to limit the attention span of a seq2seq decoder and still expect a comparable error rate with an unconstrained original model.

LibriSpeech

In the previous experiments we have used the LRS2 dataset for rapid prototyping. However, since this dataset contains many short sentences, the eventually higher decoding error rate of Taris on the longer sentences might have little effect on the reported average error rate. We re-train and evaluate our models on the 100 hour clean partition of the LibriSpeech dataset. This dataset contains considerably longer sentences than LRS2, but a simpler material of read speech derived from audio books. We display the mean error and 95% confidence interval (CI) around the mean in Table 5.1.

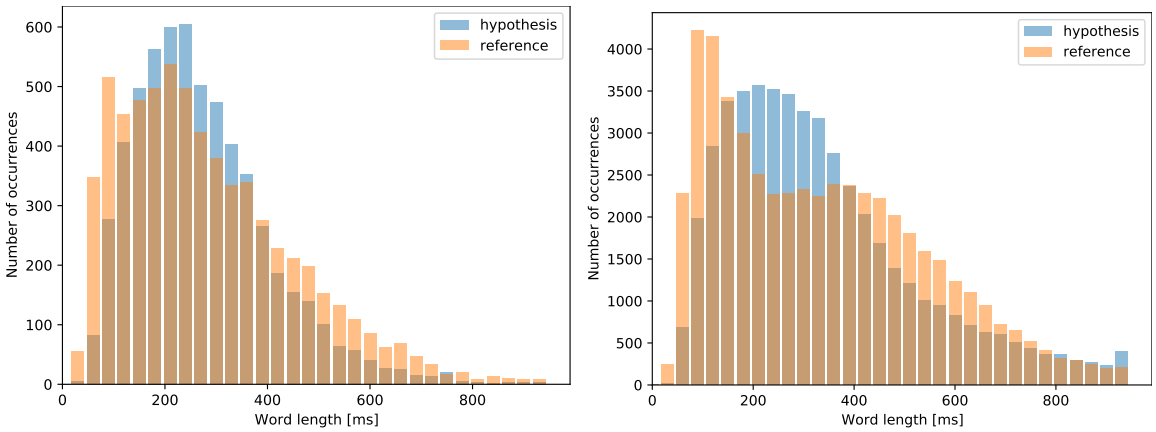
First, we notice that the systems achieve an error rate similar to the one obtained on LRS2, despite the increased amount of data, suggesting that further gains are

possible for larger model sizes. The error rate of Taris improves from 15.70% to 13.83% when limiting the look-back parameters of the encoder and the decoder. Technically, the variant with limited look-back has a more sparse connectivity pattern. If the gating mechanism of Taris succeeds in adequately segmenting the sentence, we suspect that decoding becomes an easier problem to optimise than when the search space of the optimal solution is unrestricted at the utterance level. We also notice that the word loss can be slightly detrimental to the overall decoding accuracy for the same network capacity, particularly for the models with unbounded attention span. This prompts a deeper investigation into the interplay between the cross entropy and word counting losses, as our constant scale factor λ is likely a less optimal solution to this multitask problem.

Table 5.1: *System evaluation on LibriSpeech 100h clean partition. Note the e_{LB} and e_{LA} parameters are expressed as a number of audio frames, whereas d_{LB} and d_{LA} denote the number of audio segments which contain a varying number of frames.*

Model	parameters				**CER**		**Word Loss**
	e_{LB}	e_{LA}	d_{LB}	d_{LA}	mean [%]	95% CI [%]	
Transformer	∞	∞	∞	∞	13.37	0.444	N/A
Transformer + Word Loss	∞	∞	∞	∞	14.64	0.451	0.92
Taris : infinite look-back	∞	11	∞	5	15.70	0.451	1.12
Taris : finite look-back	11	11	5	5	13.83	0.451	0.76

To investigate the typical duration of the estimated segments, we calculate their lengths by counting the number of audio frames between two successive passings of the word counting signal α_i to the nearest integer, and convert the frame count to milliseconds to compute the *hypothesis* length histogram. Additionally, we use the pre-trained Montreal forced aligner of McAuliffe et al. (2017) to compute the *reference* word length histogram on the same test set. We overlay both histograms in Figure 5.6 for a direct comparison. Not only are the histograms highly overlapped, but the one produced by Taris is in line with the average speaking rate of read speech. The small differences between the reference and hypothesis are likely owed to the short silences between words. The silences were excluded from the reference, whereas Taris does not explicitly model silences and includes them into segments. Latency has not received sufficient consideration in prior work to facilitate a direct comparison, as systems were trained with offline encoders (Moritz et al., 2019) or large receptive fields (Hou et al., 2020), relied on beam search over the output distribution (Dong and Xu, 2020), or used phoneme units (Jaitly et al., 2016). Sainath et al. (2020) presents a hybrid system combining a weaker online model with an offline rescorer that allows to revise online hypotheses with a final hypothesis. They introduce the



(a) *LRS2* **(b)** *LibriSpeech*

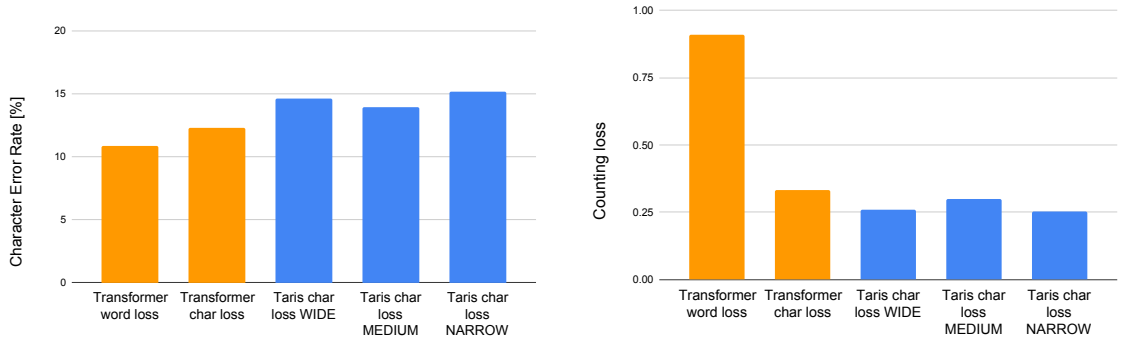
Figure 5.6: *Segmentation length distribution (in milliseconds) of Taris compared to the reference provided by the Montreal forced aligner*

notion of end-pointing latency. The offline rescorer is triggered after the utterance has been determined to be finished, that is when a threshold period has elapsed after a suspected end of utterance without further speech activity. Since our model is fully online and does not have to wait to rescore, this metric is not applicable to our system.

5.5.7 Evaluation on Mandarin speech

Since the word segmentation strategy in Taris is tailored for English, we are interested in extending the principle to Mandarin speech. Unlike English, Mandarin is characterised by a low number of morphemes per word and has almost no inflectional affixes, being considered a highly analytic language. In addition, the commonly used writing system belongs to the *scriptio continua* style, with no delimitation between words. On these grounds, we make a structural change in Taris. Instead of learning to count words (spaces between them), we let the system count the number of characters, similar to the quantity loss used in Dong and Xu (2020). This would drive the system to segment the acoustics associated with each character, which is almost always equivalent to a syllable.

For our experiment we use the Aishell-1 dataset Bu et al. (2017), which contains 165 hours of fluent speech recordings from 400 speakers coming mostly from the Northern area of China, and covers a broad range of topics. The transcription file comprises an inventory of 4333 characters, which will determine the final output size of the decoder. Since the labels also include candidate blank spaces between words, we also evaluate Taris at the word level as we did on English. Despite the larger dataset size, we maintain the same Transformer size as before for faster prototyping. We label the different parametrisations of Taris as follows:



(a) *Offline and online decoding error rate*

(b) *Unit counting loss*

Figure 5.7: *System Evaluation on Aishell-1*

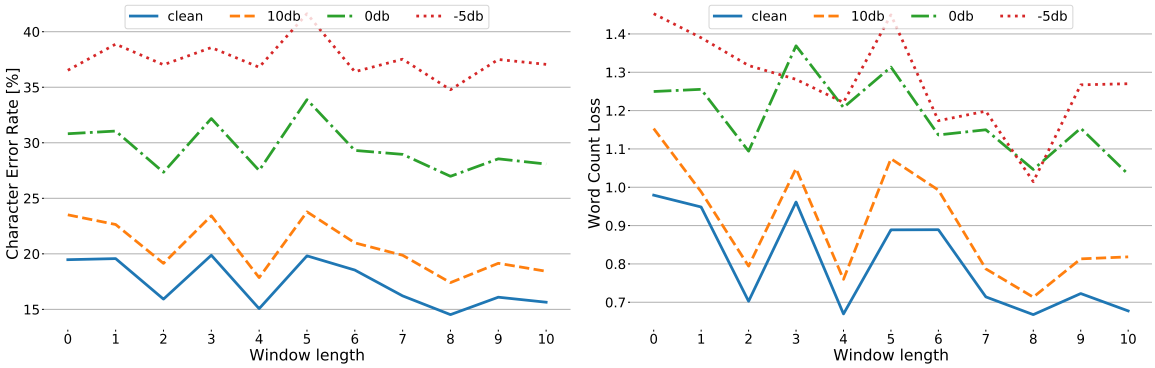
WIDE: $e_{LB} = e_{LA} = 11, d_{LB} = d_{LA} = 5$, MEDIUM: $e_{LB} = e_{LA} = 3, d_{LB} = d_{LA} = 5$, NARROW: $e_{LB} = e_{LA} = 2, d_{LB} = d_{LA} = 2$.

Figure 5.7 shows the error rates of the offline and online systems on the Aishell-1 corpus. Despite the small model size, the absolute decoding accuracy is comparable with the baseline results in Bu et al. (2017) obtained with the Kaldi toolkit. We notice that Taris does a much better job at learning to count the number of characters than the number of words, with our NARROW model obtaining a counting error of 0.2538. Since a Chinese character almost always corresponds to a single syllable, this result suggests that syllables may be easier to segment in fluent speech than words. Furthermore, the syllable level segmentation allows both the encoder and the decoder of Taris to use relatively low context lengths and further reduce the overall latency. Not shown on the figure, the error rate of a *word* counting Taris model is approximately 40%, implying that a good unit segmentation is essential for online decoding.

5.5.8 Online audio-visual decoding

In the previous experiment we have demonstrated that Taris can leverage the gating signal α to limit the dynamic range of decoder-encoder attention and still match the error rate of the offline Transformer. We now investigate how the audio-visual extension of Taris compares to the offline Audio-Visual Transformer.

In Section 5.5.5 we have seen that an Audio-Visual Transformer with infinite look-back and look-ahead encoding context achieves a slightly lower word counting cost than an Audio-only counterpart. Therefore, learning to count from the fused audio-visual representation does not degrade the word counting accuracy. The system may additionally take advantage of the visual modality to further improve its counting estimate.



(a) *Character Error Rate* **(b)** *Word Counting Loss*

Figure 5.8: *Evaluation of AV Taris on LRS2 when varying the size of the symmetrical window used for the soft-selection of the visual representation aligned with each audio representation*

We evaluate the AV Taris model for an increasing length of the cross-modal attention window, controlled by the length parameter B. The window length is defined as $len(w) = 2B + 1$, as it extends symmetrically in both directions. When $B = 0$ the system is similar to a down-sampled version of feature fusion that bypasses the requirement to have identical sampling rates for both modalities. For $B > 0$, the window is extended with B frames to the left and to the right respectively. The results are displayed in Figure 5.8.

We observe an absolute difference of approximately 3% at all the noise levels between AV Taris and the offline Transformer-based AV Align model. AV Taris is still superior to the audio-only models both in their offline and online variants at higher levels of noise. However, in low noise conditions up to 10db, the degradation of AV Taris over the AV Align Transformer entirely offsets the benefit of cross-modal alignment over audio-only modelling.

5.6 Discussion

We have proposed a simple, efficient, and fully differentiable solution for online speech recognition that does not require additional labels. Taris is inspired from early language acquisition in infants, and aims to segment a speech stream by learning to count the number of words therein. We show that our method matches the accuracy of an offline system once it listens to 5 dynamic segments. Lowering this latency remains a topic for exploration, e.g. by gradually reducing the look-ahead parameter d_{LA} later in training, explicitly modelling silences, or investigating the role of context, grammar, and semantics in lexical recognition.

Generalising to sentences of different lengths from the ones seen in training has

been recently identified as a major problem for neural online speech recognition systems (Chiu et al., 2019; Narayanan et al., 2019). By modelling only the local relationships in speech through finite look-back and look-ahead, we preserve the same property of the Neural Transducer (Jaitly et al., 2016) to effectively decouple the sentence length from the learnt representations, while allowing adaptive segments and simpler training.

Similar to the Neural Transducer (NT) model of Jaitly et al. (2016), Taris applies repeatedly a sequence to sequence model over consecutive audio windows. The NT model processes fixed length blocks, and does not need to *learn* a segmentation. Their mechanism increases the complexity of the optimisation algorithm. More precisely, the model introduces an additional end-of-block token in the output domain that needs to be emitted once per every audio window. This generates the problem of having to search for an optimal alignment in training between the longer sequence of predicted labels containing the additional token and the shorter ground truth sequence. Taris avoids this problem by not making use of end-of-block tokens. Instead, Taris analyses dynamic windows of speech centred on a word of interest. On the other hand, Taris does not guarantee the reliable segmentation of the spoken utterance into words. It only facilitates the compensation of eventual segmentation errors with a controllable number of look-back and look-ahead segments that the decoder is allowed to attend to. Studying the internal segmentation achieved by Taris remains a topic for future exploration.

An interesting behaviour of Taris concerns the handling of word contractions, such as *you're, that's, don't, it's, let's*, and others. In our work, we considered that written words are exclusively separated by spaces, as seen in Equation (5.4). Unless there is a systematic error in the transcriptions, Taris has the potential to learn the acoustic differences between "you're" and "you are". The system maintains its own segment counter (the cumulative sum of α_i), and has sufficient freedom to decide which form to transcribe. When "you are" is more likely, then the fraction of α_i added to $\Sigma\hat{w}$ may simply be one unit greater than when "you're" is preferred. Taris can also recover from potential errors since it uses a context window larger than a single segment. Depending on the intermediate scores α_i, the decoding of "are" in "you are" may then be conditioned on the acoustic representations corresponding to "you" and other adjacent segments. On the other hand, the general formulation of the word counting task in Taris may be problematic in the case of modelling silences. Since silences are generally not annotated in the human transcriptions, Taris implicitly includes all the audio frames not substantially modifying α_i to the adjacent segment. Consequently, the decoder performs

a soft alignment even over those uninformative silence frames. Increasing the efficiency of this process represents a possible direction of improvement.

It can be argued that Taris exploits human knowledge of the speech signal structure and embeds the concept of words and the local acoustic relationships, instead of being a more generic, self-organising neural network. Yet, the local processing of speech is merely the one dimensional equivalent of local convolutions applied to images, where the objects are replaced by words. Moreover, one-dimensional convolutions are commonly used in speech recognition (Abdel-Hamid et al., 2014; Kriman et al., 2020; Li et al., 2019a; Pratap et al., 2019). Given their major impact in research despite their lack of invariance to orientation, scaling, or even small perturbations, there is still much to be learned from engineered models in the pursuit of artificial general intelligence.

It is unlikely that humans learn to segment speech by counting words in full sentences. We are not offered the word count in a numeric format as a supervision signal. Why would it be appropriate to design a speech recognition system based on this aspect? We believe there are several reasons. First, this task would not be impossible for humans if it was formulated as a puzzle for finding patterns in a foreign language. Language acquisition in humans involves a long term process of teaching simpler, isolated words before gradually increasing the difficulty. These learning strategies have not fully matured in our machine learning technology. On the other hand, it is very common, and cheap, to produce a speech dataset annotated at the sentence level, without intermediate phone-level or word-level alignments. Therefore we are already asking computers to solve the speech recognition challenge differently from the way we learn a spoken language. We argue that learning to count words is a good compromise with respect to our existing technology and datasets when aiming to segment a spoken utterance.

6 Conclusion

6.1 Summary

In this thesis, we studied the problem of audio-visual integration in automatic speech recognition. Our aim has been to develop efficient algorithms that learn to take advantage of the visual modality in speech, by complementing the more informative auditory modality. We achieved this aim by leveraging our knowledge of the structure of the speech signal.

First, we used the assumption that there exists an underlying higher order alignment between the audio and visual modalities. We made the learning of this alignment explicit with the AV Align architecture, and used the alignment to troubleshoot the learning difficulties of neural networks on the task of multimodal speech recognition. Empirically, we have shown that AV Align discovers a monotonic trend in the cross-modal alignment pattern in order to achieve improvements in decoding accuracy over an audio-only system. This is in line with our expectation about the relationship between audio and visual representations of speech, although we do not impose any locality constraints on the alignments. AV Align does not always arrive at this alignment pattern. We have shown that the discovery of plausible alignments is correlated with the learning of good visual representations of speech. In addition, there is an incentive for our multimodal learning strategy to dismiss the visual modality and rely entirely on the auditory one. To correct this optimisation problem, we proposed a multitask learning strategy where we aim to regress the intensities of two lip-related Action Units from the visual representations. With the AU loss complementing the decoding cross-entropy loss, our systems are able to capitalise on the visual modality in the challenging settings of the LRS2 dataset. Compared to an audio-only system, we achieved average improvements in transcription accuracy ranging from 6.4% under clean speech conditions to around 31% at the highest level of audio noise. We then show that alternative AVSR architectures can also benefit from the regularisation of the visual representations through our AU loss. Over-

all, our findings underline that learning good visual representations for the task of audio-visual speech recognition remains a difficult problem for machine learning algorithms.

The multimodal system proposed in Chapter 4 was only applicable to short utterances that were segmented in advance. This limitation was in part inherited from the underlying sequence to sequence architecture with decoder-encoder attention, which could start decoding the first word only after encoding the entire input utterance. Furthermore, the cross-modal attention mechanism of AV Align introduced a limitation of the same nature as in the decoder-encoder attention mechanism. We wanted to devise a strategy that would break down a spoken utterance into smaller units which could be decoded more promptly. We started from the knowledge that speech is the vocalisation of a sequence of words, and that humans are able to identify the boundaries between words in continuous speech. We used this knowledge and developed Taris, an end-to-end differentiable neural network architecture that learns the word-level segmentation of a spoken utterance. Taris achieves this by predicting the number of words in a spoken sentence. The true word count can be easily inferred from the transcription. We show that, by learning to count words, Taris estimates acoustic segments of a duration that approximates the distribution of word lengths in English. Taris performs the segmentation task jointly with speech modelling and decoding. This automatic segmentation enables the decoding of speech into text with a controlled latency, as opposed to relying on an external tool that independently detects reasonable boundaries within an utterance in continuous speech. We evaluated Taris on English speech from the LRS2 and Librispeech datasets, obtaining a comparable accuracy with the equivalent Transformer model that decodes offline. Correspondingly, we revised the cross-modal attention mechanism in AV Align by limiting the attention span to a fixed window of video representations centred on each audio frame. As a result, we have achieved an audio-visual speech recognition system that can decode online. Experimentally, we found that the accuracy of the audio-visual extension of Taris lags behind the offline Transformer-based AV Align system by approximately 3%. The offline model could exploit the entire utterance for both cross-modal alignment and decoding, explaining one possible source for the difference.

We believe that the modelling assumptions in both AV Align and Taris are sufficiently general to transfer to other multimodal speech processing tasks. We will discuss several developments in Section 6.1.5.

6.1.1 Significance to AVSR

The results in Chapter 4 show that multimodal fusion through the cross-modal alignment of higher order audio-visual representations serves as a good inductive bias in multimodal speech recognition. Compared to a full connectivity between every data point and intermediate sequence representations as in the Transformer architecture, AV Align achieves a lower error rate with a much sparser network structure. Such sparsity is highly desirable in the current machine learning frameworks, allowing optimisation algorithms to converge faster and generalise better on specific, well-studied problems. This is particularly important for the task of audio-visual speech recognition. As the auditory modality has a higher contribution to the comprehension of the message under low noise conditions than the visual one, there is a strong early incentive to ignore the more difficult visual information.

Our motivation for AV Align sprung from the *Watch, Listen, Attend, and Spell* (WLAS) network of Chung et al. (2017), which is currently the most popular neural architecture in AVSR with 337 citations to date. To us, WLAS appeared to lightly deviate from perceptual studies endorsing an early integration of auditory and visual cues taking place before any lexical identification. WLAS extracts audio-specific and video-specific context vectors correlated with the state of a decoder that takes characters as inputs. It is exactly this lexical connection that made us rethink the integration strategy and propose AV Align. In favour of WLAS is the study of Barutchu et al. (2008), who show that lexical knowledge does play a role in the modulation of early audio-visual cue integration. Our work does not provide a singular and correct solution to the audio-visual speech integration task. Going forward, future architectures may want to retain two key components of AV Align. The first one is the earlier stage of feature integration compared to WLAS. The second one is the audio-to-video attention mechanism for its property to be invariant to the different sampling rates of the speech modalities. Before undertaking new studies for better audio-visual architectures, we recommend an investigation into the nature of the optimisation problem, a reassessment of the basic challenge, and a reconsideration of the system evaluation before undertaking. We will discuss these aspects in Section 6.2.

The work in Chapter 3 contributed to the design of AV Align. There, we noticed that visual speech recognition models struggle to decode characters from image sequences. This made us speculate that the modality dropout strategy of Chung et al. (2017) used to train their WLAS model may be sub-optimal. For this reason, AV Align always learns from both modalities simultaneously. A drawback of AV

Align over WLAS is the impossibility for our model to handle a missing audio modality. When this hybrid functionality is not required, we believe that AV Align can make better use of the model capacity than WLAS for capitalising on the visual modality of speech.

6.1.2 Context

The original contributions in this work were enabled by several key advancements in multiple disciplines. First, we had available larger audio-visual speech datasets, particularly LRS2. This allowed an investigation into the contribution of the visual modality to speech recognition for large vocabularies and continuous, unconstrained speech. The sequence to sequence architecture has become well established for text processing tasks and drove the initial developments. We based our experiments with neural networks on the TensorFlow framework. This open-source machine learning toolkit contains professional level software, and connects a large number of users and contributors for its maintenance, documentation, and ease of use. The data pre-processing pipeline of our systems rely on face detection tools such as OpenFace. Progress made in image and facial recognition methods enables us with reliable tools to assist visual speech processing. Their errors are sufficiently small to warrant the separate optimisation of face detection and speech recognition over a more expensive end-to-end solution. Concurrently, many research laboratories, including ours, have directed their investments towards continually improving dedicated hardware accelerators. On average, a full experiment loop of AV Align on each noise condition on LRS2 took approximately one day using a single GPU. This relatively short waiting time allowed us to test our hypotheses with low context switching.

6.1.3 Limitations

This thesis does not fully uncover the specific auditory and visual speech representations prior to integration. The features produced by AV Align are not interpretable in a human language, and, excepting the alignment pattern produced, most of the audio-visual network remains a black box. As we learnt from the literature review in Chapter 2, audio-visual integration is a process that happens at the subconscious level, prior to lexical categorisation. Therefore we do not have the option to generate the explanations with our neural network through supervised learning. As a consequence, we do not have an indicator of the frame-level contribution of the visual modality to speech recognition, but only an average error rate improvement on a speech corpus. A remaining challenge is to devise strategies for post-hoc explanations tailored to audio-visual speech or

interpretable schemes such as in Section 4.3.6 to demystify the multimodal integration process.

6.1.4 What went wrong

TensorFlow, the software toolkit used for the implementation of AV-Align, adopted a new design philosophy with the release of version 2.0. In order to encourage a coding style closer to the main principles of Python, TensorFlow 2.0 switched to an eager execution of the operators in a neural network by default, which traded the troubleshooting of the runtime execution for the speed of the graph compilation. In the background, a complex graph compiler had the potential to recover the performance gap by transforming the new eager code to a graph code in a convenient way. As we ported our AV Align implementation from TensorFlow 1.x to TensorFlow 2.0, we noticed that the default sequence to sequence LSTM model was no longer reaching the same level of accuracy in speech recognition. Furthermore, the cross-modal alignment mechanism of AV Align was not compatible with the graph compiler of TensorFlow 2.0, despite reporting multiple issues upstream and contributing to their solution. We ended up in a situation where our previous work could no longer be reproduced with the latest officially supported version of the largest machine learning toolkit available. Only later we identified differences in the implementations of the two versions. For example, the dropout strategy for the LSTM cells changed in TensorFlow 2.0, with no backward compatibility support for the old behaviour. Another example was the introduction of an undocumented multiplier that affected the scale of weight regularisation. We were only able to obtain similar results between versions 1.x and 2.0 by disabling weight regularisation, but the overall performance degraded much faster in noisy conditions than with a properly regularised network in 1.x. This has been an opportunity to understand the underlying engineering problems in the software library, and realise the hidden cost of maintaining an implementation leveraging inductive biases such as the LSTM cell. In contrast, the Transformer model takes considerably fewer lines of code and relies on more atomic operations shared with other components commonly used in machine learning frameworks. Combined with the complexity of the graph compiler, which was seen as a black box from the point of view of our work, it became clear that working with LSTMs was going to be very expensive. This realisation generated the discussion in Chapter 4 on the potential of replacing LSTM cells with the Transformer architecture, not only because of the advantages of parallelisation, but also because of the software issues and the lack of backward compatibility to enable reproducibility.

6.1.5 Broader impact

Thanks to the segmentation property of Taris discussed in Chapter 5, it is now possible to decode speech online with a sequence to sequence model with attention without resorting to advanced search strategies in training. Compared to alternative online models such as the RNN Transducer, Taris reduces the computational cost of training and the engineering cost of maintaining the hardware-specific software implementation of the RNN-T objective function. Additionally, it springs from the sequence to sequence model architecture that is currently outperforming alternative approaches. We believe that both the audio and the audio-visual variants of Taris represent a step forward for increasing the accessibility of audio-visual speech recognition technology, although they still require validation at a much larger scale than this thesis could afford. Connecting with the discussion in Section 2.1 about the current limits of ASR technology, we believe that the original contributions of this thesis will increase the adoption of AVSR solutions.

6.2 Future work

6.2.1 Cascaded optimisation

Between the experiments in cognitive psychology and the ones in machine learning there appears to be a contrast regarding the interpretation of the dominant modality in speech. For example, Dodd (1977) finds that, when audio and visual stimuli are conflicting, the subjects in the perceptual experiment pay more attention to the visual modality. Instead, in Chung et al. (2017); Ngiam et al. (2011) and in this thesis, we observe that machine learning systems learn to rely more on the auditory modality. Consequently, we needed different strategies in machine learning to ensure that the system does not become over-reliant on the audio channel, dismissing the visual signal. This apparent contrast suggests a possible mismatch between the multimodal integration strategies adopted by the human brain and by an artificial neural network. The convergence issues reported in machine learning based AVSR could be a consequence of the inadequacy of the systems offering higher incentives for short term gains.

Our system would be more compelling if we could sidestep the necessity for the Action Unit auxiliary objective. Currently, this secondary task requires an external tool trained on a separate dataset to estimate the intensity of the facial action units. Therefore, we introduce a new data stream of noisy information and increase the complexity of the training pipeline. Furthermore, the AU loss is

not transferable to other multimodal tasks outside audio-visual speech. Ideally, we want a better optimisation strategy that escapes the critical points associated with the audio modality alone where vision is neglected.

We can take advantage of the observations above and embed explicit priorities into the learning algorithm for the different parameters of the network. In particular, we can assign the highest priority to the learning of visual representations, followed by the learning of audio representations, followed by the learning of symbolic representations of speech. We outline below such a training algorithm that has the potential to learn good visual representations without resorting to the AU multitask. This algorithm alternates between the update of the visual-only parameters θ_V, audio and visual parameters θ_A and θ_V, and the full set of network parameters that includes the text-based parameters of the decoder θ_T. We define Δ as the absolute difference between the current batch loss J at step i and the running average of the batch loss over the previous τ steps:

$$\Delta = |J_i - \overline{J_{i-\tau\,:\,i}}| \tag{6.1}$$

We show the pseudocode for this learning strategy in Algorithm 1 below, and we also illustrate it in Figure 6.1.

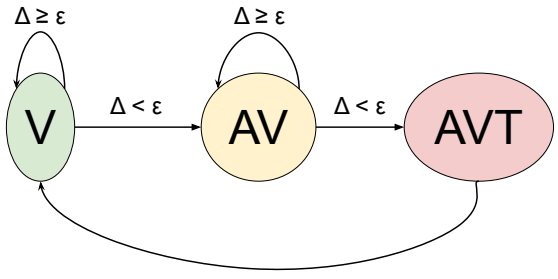


Figure 6.1: *Schematic illustration of the proposed cascaded optimisation strategy. The parameters associated with each node in the graph (Video, Audio, Text) are updated starting with the more impoverished visual speech modality until we can no longer measure significant improvements of the objective. In contrast, we always step only once when updating the parameters of the decoder before re-evaluating the improvement from updating the video parameters.*

Algorithm 1: Cascaded optimisation proposed for audio-visual speech recognition.

Input: ϵ, iterations
initialise $i = 0$, $J(0) = +\infty$;
evaluate Δ;
while $i \leq$ *iterations* **do**
    while *True* **do**
        update θ_V;
        i += 1;
        evaluate Δ;
        if $\Delta \geq \epsilon$ **then**
            break;
        end
    end
    while *True* **do**
        update θ_A, θ_V;
        i += 1;
        evaluate Δ;
        if $\Delta \geq \epsilon$ **then**
            break;
        end
    end
    update θ_A, θ_V, θ_T;
    i += 1;
    evaluate Δ;
end

6.2.2 Redefine the basic challenge AV > A

The evaluation of AV Align on the test partition of LRS2 in Section 4.3.8 has shown that our multimodal system transcribes with a lower accuracy than the audio-only system for a considerable number of sentences, in spite of showing significant improvements on average. In other words, the improvements are reflected at the corpus level, not on each separate sentence. As a consequence, it may be possible that the multimodal system learns to extract good visual representations from the easiest or more frequent content at the expense of the harder and less frequent.

This finding is not that surprising given that we explicitly optimise the system with an objective function defined as the average error on a minibatch. Depending on

our goals, we need to become more specific regarding the basic challenge to be solved by the audio-visual system. This has been previously defined as $AV > A$. Such formulation lacks specificity concerning the sentence-level or corpus-level improvement requirements. Assuming we want to encourage the audio-visual system to perform at least as well as the audio-only one on every sentence, we can adopt two possible strategies. We categorise these strategies into *full reference* and *no reference* setups.

In the **full reference** setup, we expect to have available a fully trained audio model ahead of training the audio-visual one. We can then decode the same sentence in parallel with the two systems. This allows us to introduce an additional constraint into the objective function of the multimodal system. Specifically, we can ask to maximise the difference between the audio-only and audio-visual cross-entropy losses. As a consequence, the audio-visual system has a secondary goal to outperform the audio system on each example in the batch.

$$\text{Disparity Loss} = \frac{1}{L}\sum_{k} -y_k \log(p_k^{AV}) - \frac{1}{L}\sum_{k} -y_k \log(p_k^{A}) \tag{6.2}$$

$$= \frac{-1}{L}\sum_{k} y_k \ \left[\log(p_k^{AV}) - \log(p_k^{A})\right] \tag{6.3}$$

$$= \frac{-1}{L}\sum_{k} y_k log\left(\frac{p_k^{AV}}{p_k^{A}}\right) \tag{6.4}$$

$$\text{Loss} = \textit{CE Loss} + \textit{Disparity Loss} \tag{6.5}$$

When the ratio p_k^{AV}/p_k^{A} in Equation (6.4) is lower than 1 for any output label, a small penalty is added to the main loss. However, when the same ratio is higher than 1, corresponding to an improved audio-visual prediction, it will subtract a value from Equation (6.4). To only penalise the system for worse audio-visual predictions, we can cap the ratio as follows:

$$\text{Disparity Loss} = \frac{-1}{L}\sum_{k} y_k min\left(0, log\frac{p_k^{AV}}{p_k^{A}}\right) \tag{6.6}$$

In the **no reference** setup, we do not know how an equivalent audio-only system would score on a particular sentence. Analysing Figure 4.16, we see a relatively large spread of the error. We believe it would be beneficial to explicitly minimise the standard deviation of the error on a minibatch, in addition to the average error. Some sentences will naturally contain more difficult content to lipread owed to the visual challenges and to the spoken material. Minimising the variance of the error in a minibatch of sentences may indeed lower the learning speed for the

simpler content. On the other hand, in the absence of this additional penalty, the stochastic gradient descent algorithm is very likely to choose the simpler over the more difficult trajectory. The error variance penalty could be defined as:

$$Var\ Loss = \frac{1}{B}\sum_{i=1}^{B} |CE_i - CE|^2 \tag{6.7}$$

where B is the batch size, CE_i is the cross entropy on sentence i in the minibatch, and CE is the average batch cross-entropy loss.

6.2.3 Joint lip tracking and AVSR optimisation

The audio-visual systems in this thesis receive a pre-segmented lip region from an external tool. As the amount of audio-visual speech data increases, it would be beneficial to perform the segmentation of the area of interest jointly with the modelling of the visual speech signal. The visual focus of attention could be directed based on the relevance to the AVSR task. Moreover, having more information available from the visual channel enables the learning of additional tasks sharing low-level visual representations. A current limitation is owed to the soft alignment principle of attention that still involves a dense evaluation regardless of the *effective* range. Therefore a remaining challenge is the design of efficient strategies to dynamically allocate more computational resources to the region of interest.

6.2.4 Sampling

In the experimental setup of this thesis, all training examples are treated equally. Every sentence has an equal contribution to the total training loss regardless of the spoken content or visual conditions. But some sentences are considerably easier to lipread and more informative than others (Macleod and Summerfield, 1987). Thus it would beneficial to let the system choose to learn *more* from those sentences that provide meaningful information about the audio-visual relationships in speech. We believe this has the potential to lessen the learning difficulties of AV Align by lowering the importance of those sentences that cannot be lipread reliably and in turn favour the dominance of the audio modality. Since it would be infeasible to manually annotate an audio-visual speech corpus at this level, this property could be learnt directly from the data. One possible approach would be to assign a lower weight to those training samples in a minibatch where the gradients of the objective function with respect to the visual parameters have a considerably smaller norm than the gradients associated with the audio pa-

rameters. The same technique could be used to identify those highly predictable sentences that favour a strong language model overfitting to the more frequently seen material.

6.2.5 New evaluation strategy

In this work we used the standard speech transcription error rate to compare our audio-only and audio-visual systems that were trained under identical noise conditions. This allowed us to measure an audio-visual benefit at a known SNR. Macleod and Summerfield (1987) pointed out a limitation of this approach. Since the error rate is lower bounded at 0%, the audio-visual improvement will be capped on those samples that can already be decoded accurately from audio alone. We have seen this in the error analysis in Section 4.3.8, where the audio system already achieves a 0% error rate on some sentences particularly under clean speech conditions. If we are interested in measuring the contribution of the visual modality to speech perception, the error rate metric would mask the potential improvements. Macleod and Summerfield (1987) proposed to evaluate the difference between audio and audio-visual perception at an SNR determined experimentally for each sentence, termed the *speech reception threshold* (SRT). They defined the SRT as the lowest SNR at which human listeners could barely understand the sentences selected for the experiment. Measuring the difference in dB between the audio SRT and the audio-visual SRT allowed them to assess the visual contribution while normalising for both the individual differences at speech perception and the lipreading difficulty of the speech material.

The same methodology could be used to evaluate our machine learning models, granting the following advantages. First, different researchers achieve different audio-only baselines, sometimes on the same data and with very similar models. The SRT of Macleod and Summerfield (1987) would partly compensate for the variations of the baselines, and would enable researchers to compare which method can capitalise more on the visual modality despite not having a bleeding-edge audio-only system. Second, the uncapped nature of the SRT would no longer produce identical audio and audio-visual results that were owed to the audio modality being self-sufficient on some sentences at higher SNRs. The main limitation of the strategy proposed by Macleod and Summerfield (1987) is the considerably higher effort needed to determine the SRT on each sentence. We see the use of new metrics such as the SRT as an imperative step towards a more reliable comparison between different audio-visual speech recognition models.

6.3 Final remark

Let us return to the opening paragraph of this thesis. According to Berwick and Chomsky (2015), there is a clear separation between language that is innate to the mind, and speech that is externalised through the sensorimotor interface. While speech has a linear or sequential property owing to the limitations of this interface, language has a hierarchical structure where the linear distance between words is less relevant. This may suggest that the current paradigm in neural networks where speech observations alter the internal state may not be entirely optimal. The traditional approach based on Hidden Markov Models saw observations as outputs, which roughly corresponds to the externalisation theory of Berwick and Chomsky (2015). To achieve a higher efficiency, the underlying process we may want to model could be the language itself, instead of its externalisation that was modulated by a complex sensorimotor interface. We hope this will inspire a more fundamental rethink of the existing approaches in speech modelling using neural networks to those willing to engage in more audacious research.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., and Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545.

Abdelaziz, A. H. (2018). Comparing fusion models for DNN-based audiovisual continuous speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):475–484.

Adjoudani, A. and Benoît, C. (1995). Audio-visual speech recognition compared across two architectures. In *Proceedings of the Fourth European Conference on Speech Communication and Technology, EUROSPEECH 1995, Madrid, Spain, September 18-21, 1995*. ISCA.

Afouras, T., Chung, J., and Zisserman, A. (2018a). LRS3-TED: a large-scale dataset for visual speech recognition. *CoRR*, abs/1809.00496.

Afouras, T., Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2018b). Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–11.

Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93.

Alabort-i Medina, J., Antonakos, E., Booth, J., Snape, P., and Zafeiriou, S. (2014). Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pages 679–682, New York, NY, USA. ACM.

Alabort-i Medina, J. and Zafeiriou, S. (2017). A unified framework for compositional fitting of active appearance models. *International Journal of Computer Vision*, 121(1):26–64.

Alpaydin, E. (2018). Classifying multimodal data. In Oviatt, S., Schuller, B., Cohen, P. R., Sonntag, D., Potamianos, G., and Krüger, A., editors, *The Handbook of Multimodal-Multisensor Interfaces: Signal Processing, Architectures, and Detection of Emotion and Cognition - Volume 2*, page 49–69. Association for Computing Machinery and Morgan & Claypool.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 173–182. JMLR.org.

Antonakos, E., i Medina, J. A., Tzimiropoulos, G., and Zafeiriou, S. P. (2015). Feature-based Lucas-Kanade and active appearance models. *IEEE Transactions on Image Processing*, 24(9):2617–2632.

Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. (2017). A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 233–242. JMLR.org.

Assael, Y. M., Shillingford, B., Whiteson, S., and de Freitas, N. (2016). LipNet: Sentence-level lipreading. *CoRR*, abs/1611.01599.

Auer, E. T. and Bernstein, L. E. (1997). Speechreading and the structure of the lexicon: Computationally modeling the effects of reduced phonetic distinctiveness on lexical uniqueness. *The Journal of the Acoustical Society of America*, 102(6):3704–3710.

Auer Jr, E. T. and Luce, P. A. (2005). Probabilistic phonotactics in spoken word recognition. In Pisoni, D. and Remez, R., editors, *The Handbook of Speech Perception*, chapter 25, pages 610–630. John Wiley & Sons, Ltd.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization. *arXiv preprint*, arXiv:1607.06450.

Bagby, T., Rao, K., and Sim, K. C. (2018). Efficient implementation of recurrent neural network transducer in Tensorflow. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 506–512.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Baker, J. (1975). The DRAGON system–an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29.

Baltrusaitis, T., Robinson, P., and Morency, L. P. (2016). OpenFace: An open source facial behavior analysis toolkit. In *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10.

Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L. (2018). Openface 2.0: Facial behavior analysis toolkit. In *Proceedings of the 13th IEEE International Conference on Automatic Face Gesture Recognition*, pages 59–66.

Baltrušaitis, T., Ahuja, C., and Morency, L. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

Barker, J., Watanabe, S., Vincent, E., and Trmal, J. (2018). The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. In *Proc. Interspeech 2018*, pages 1561–1565.

Barutchu, A., Crewther, S. G., Kiely, P., Murphy, M. J., and Crewther, D. P. (2008). When /b/ill with /g/ill becomes /d/ill: Evidence for a lexical effect in audiovisual speech perception. *European Journal of Cognitive Psychology*, 20(1):1–11.

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

BBC and University of Oxford (2017). The Oxford-BBC Lip Reading Sentences 2 (LRS2) Dataset. `http://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrs2.html`. Online, Accessed: 4 May 2020.

Bear, H. L. and Harvey, R. (2017). Phoneme-to-viseme mappings: the good, the bad, and the ugly. *Speech Communication*, 95:40 – 67.

Bear, H. L., Harvey, R., Theobald, B. J., and Lan, Y. (2014). Resolution limits on visual speech recognition. In *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, pages 1371–75.

Beck, E., Hannemann, M., Doetsch, P., Schlüter, R., and Ney, H. (2018a). Segmental encoder-decoder models for large vocabulary automatic speech recognition. In *Proc. Interspeech*, pages 766–770.

Beck, E., Zeyer, A., Doetsch, P., Merboldt, A., Schlüter, R., and Ney, H. (2018b). Sequence modeling and alignment for LVCSR-systems. In *Speech Communication; 13th ITG-Symposium*, pages 1–5.

Bengio, S. (2002). An asynchronous Hidden Markov Model for audio-visual speech recognition. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NeurIPS'02, page 1237–1244, Cambridge, MA, USA. MIT Press.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Berry, C., Kokaram, A., and Harte, N. (2011). An extended multiresolution approach to mouth specific AAM fitting for speech recognition. In *2011 19th European Signal Processing Conference*, pages 1959–1963.

Berwick, R. C. and Chomsky, N. (2015). *Why Only Us: Language and Evolution*. The MIT Press.

Binnie, C. A., Montgomery, A. A., and Jackson, P. L. (1974). Auditory and visual contributions to the perception of consonants. *Journal of Speech and Hearing Research*, 17(4):619–630.

Bourlard, H., Hermansky, H., and Morgan, N. (1996). Towards increasing speech recognition error rates. *Speech Communication*, 18(3):205–231.

Bourlard, H. A. and Morgan, N. (1993). *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, USA.

Bozkurt, E., Erdem, C. E., Erzin, E., Erdem, T., and Ozkan, M. (2007). Comparison of phoneme and viseme based acoustic units for speech driven realistic lip animation. In *2007 3DTV Conference*, pages 1–4.

Bu, H., Du, J., Na, X., Wu, B., and Zheng, H. (2017). AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5.

Cairns, P., Shillcock, R., Chater, N., and Levy, J. (1994). Lexical segmentation: The role of sequential statistics in supervised and un-supervised models. In *Proceedings of the 16th annual conference of the Cognitive Science Society*, pages 136–141.

Campbell, R. and Dodd, B. (1980). Hearing by eye. *Quarterly Journal of Experimental Psychology*, 32(1):85–99. PMID: 7367580.

Cangea, C., Veličković, P., and Liò, P. (2020). XFlow: Cross-modal deep neural networks for audiovisual classification. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3711–3720.

Cappelletta, L. and Harte, N. (2012). Phoneme-to-viseme mapping for visual speech recognition. In Carmona, P. L., Sánchez, J. S., and Fred, A. L. N., editors, *ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods, Volume 2, Vilamoura, Algarve, Portugal, 6-8 February, 2012*, pages 322–329. SciTePress.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

Chiu, C., Han, W., Zhang, Y., Pang, R., Kishchenko, S., Nguyen, P., Narayanan, A., Liao, H., Zhang, S., Kannan, A., Prabhavalkar, R., Chen, Z., Sainath, T., and Wu, Y. (2019). A comparison of end-to-end models for long-form speech recognition. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 889–896.

Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, E., Jaitly, N., Li, B., Chorowski, J., and Bacchiani, M. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778.

Chiu, C.-C. and Raffel, C. (2018). Monotonic chunkwise attention. In *Proceedings of the International Conference on Learning Representations*.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 577–585. MIT Press.

Chorowski, J. and Jaitly, N. (2017). Towards better decoding and language model integration in sequence to sequence models. In *Proc. Interspeech 2017*, pages 523–527.

Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017). Lip reading sentences in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chung, J. S. and Zisserman, A. (2017). Lip reading in the wild. In Lai, S.-H., Lepetit, V., Nishino, K., and Sato, Y., editors, *Computer Vision – ACCV 2016*, pages 87–103, Cham. Springer International Publishing.

Cooke, M., Barker, J., , Cunningham, S., , and Shao, X. a. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424.

Dodd, B. (1977). The role of vision in the perception of speech. *Perception*, 6(1):31–40. PMID: 840618.

Dong, L. and Xu, B. (2020). CIF: Continuous integrate-and-fire for end-to-end speech recognition. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083.

Dong, L., Yi, C., Wang, J., Zhou, S., Xu, S., Jia, X., and Xu, B. (2020). A comparison of label-synchronous and frame-synchronous end-to-end models for speech recognition.

Drexler, J. and Glass, J. (2020). Learning a subword inventory jointly with end-to-end automatic speech recognition. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6439–6443.

Duchnowski, P., Meier, U., and Waibel, A. (1994). See me, hear me: Integrating automatic speech recognition and lip-reading. In *Proceedings of Third International Conference on Spoken Language Processing (ICSLP 94)*, pages 547 – 550.

Ekman, P. and Rosenberg, E. L. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

Erber, N. P. (1969). Interaction of audition and vision in the recognition of oral speech stimuli. *Journal of Speech and Hearing Research*, 12(2):423–425.

Erber, N. P. (1975). Auditory-visual perception of speech. *Journal of Speech and Hearing Disorders*, 40(4):481–492.

Fan, R., Zhou, P., Chen, W., Jia, J., and Liu, G. (2019). An online attention-based model for speech recognition. In *Proc. Interspeech*, pages 4390–4394.

Fernandez-Lopez, A. and Sukno, F. M. (2018). Survey on automatic lip-reading in the era of deep learning. *Image and Vision Computing*, 78:53 – 72.

Fiscus, J. G., Fisher, W. M., Martin, A. F., Przybocki, M. A., and Pallett, D. S. (2000). 2000 NIST evaluation of conversational speech recognition over the telephone: English and Mandarin performance results. In *National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899*.

Fisher, C. G. (1968). Confusions among visually perceived consonants. *Journal of Speech and Hearing Research*, 11(4):796–804.

Forcada, M. L. and Ñeco, R. P. (1997). Recursive hetero-associative memories for translation. In Mira, J., Moreno-Díaz, R., and Cabestany, J., editors, *Biological and Artificial Computation: From Neuroscience to Technology*, pages 453–462, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Gales, M. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language*, 12(2):75–98.

Gales, M. and Young, S. (2008). The application of hidden Markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304.

Garofalo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L. (1993). The DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. *Linguistic Data Consortium*.

Gers, F., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with LSTM. *IET Conference Proceedings*, pages 850–855.

Goldschen, A. J., Garcia, O. N., and Petajan, E. D. (1996). Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. In Stork, D. G. and Hennecke, M. E., editors, *Speechreading by Humans and Machines: Models, Systems, and Applications*, pages 505–515. Springer Berlin Heidelberg, Berlin, Heidelberg.

Goldschen, A. J., Garcia, O. N., and Petajan, E. D. (1997). Continuous automatic speech recognition by lipreading. In Shah, M. and Jain, R., editors, *Motion-Based Recognition*, pages 321–343. Springer Netherlands, Dordrecht.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *ICML Representation Learning Worksop*.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

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

Harte, N. and Gillen, E. (2015). TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615.

He, K., Girshick, R., and Dollar, P. (2019). Rethinking Imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *ECCV 2016*, pages 630–645. Springer International.

Hennecke, M. E., Stork, D. G., and Prasad, K. V. (1996). Visionary speech: Looking ahead to practical speechreading systems. In Stork, D. G. and Hennecke, M. E., editors, *Speechreading by Humans and Machines: Models, Systems, and Applications*, pages 331–349. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Hou, J., Guo, W., Song, Y., and Dai, L.-R. (2020). Segment boundary detection directed attention for online end-to-end speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, 2020(1):3.

Huang, X., Baker, J., and Reddy, R. (2014). A historical perspective of speech recognition. *Commun. ACM*, 57(1):94–103.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Jaitly, N., Le, Q. V., Vinyals, O., Sutskever, I., Sussillo, D., and Bengio, S. (2016). An online sequence-to-sequence model using partial conditioning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 5067–5075. Curran Associates, Inc.

Jeffers, J. and Barley, M. (1980). *Speechreading (lipreading)*. Charles C. Thomas Publisher.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556.

Johnson, E. K. and Jusczyk, P. W. (2001). Word segmentation by 8-month-olds: When speech cues count more than statistics. *Journal of Memory and Language*, 44(4):548 – 567.

Jordan, T. R. and Sergeant, P. (2000). Effects of distance on visual and audiovisual speech recognition. *Language and Speech*, 43(1):107–124.

Jordan, T. R. and Sergeant, P. C. (1998). Effects of facial image size on visual and audio–visual speech recognition. In *Hearing by eye II: Advances in the psychology of speechreading and auditory-visual speech.*, pages 155–176. Psychology Press/Erlbaum (UK) Taylor & Francis, Hove, England.

Jusczyk, P. and Aslin, R. (1995). Infants' detection of the sound patterns of words in fluent speech. *Cognitive Psychology*, 29(1):1 – 23.

Jusczyk, P. W., Houston, D. M., and Newsome, M. (1999). The beginnings of word segmentation in English-learning infants. *Cognitive Psychology*, 39(3):159 – 207.

K. Paleček and Chaloupka, J. (2013). Audio-visual speech recognition in noisy audio environments. In *Proceedings of the 2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, pages 484–487.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Karpov, A., Ronzhin, A., Kipyatkova, I. S., and Zelezny, M. (2011). Influence of phone-viseme temporal correlations on audiovisual STT and TTS performance. In *Proceedings of the 17th International Congress of Phonetic Sciences, ICPhS 2011, Hong Kong, China, August 17-21*, pages 1030–1033.

Katsaggelos, A. K., Bahaadini, S., and Molina, R. (2015). Audiovisual fusion: Challenges and new approaches. *Proceedings of the IEEE*, 103(9):1635–1653.

Kim, S., Hori, T., and Watanabe, S. (2017). Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kolossa, D., Zeiler, S., Vorwerk, A., and Orglmeister, R. (2009). Audiovisual speech recognition with missing or unreliable data. In Theobald, B. and Harvey, R. W., editors, *Auditory-Visual Speech Processing, AVSP 2009, Norwich, UK, September 10-13, 2009*, pages 117–122. ISCA.

Koumparoulis, A., Potamianos, G., Mroueh, Y., and Rennie, S. J. (2017). Exploring ROI size in deep learning based lipreading. In *Proceedings of the 14th International Conference on Auditory-Visual Speech Processing*, pages 64–69.

Koumparoulis, A., Potamianos, G., Thomas, S., and da Silva Morais, E. (2020). Resource-Adaptive Deep Learning for Visual Speech Recognition. In *Proc. Interspeech 2020*, pages 3510–3514.

Kreutzer, J. and Sokolov, A. (2018). Learning to segment inputs for NMT favors character-level processing. In *Proceedings of the International Workshop on Spoken Language Translation*.

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

Lan, Y., Harvey, R., Theobald, B., Ong, E., and Bowden, R. (2009). Comparing visual features for lipreading. In *Proceedings of AVSP 2009*, pages 102–106.

Lan, Y., Theobald, B.-J., Harvey, R. W., Ong, E.-J., and Bowden, R. (2010). Improving visual features for lip-reading. In *Proceedings of AVSP*, pages 1–6, paper S7–3.

Le Cun, Y. (1986). Learning process in an asymmetric threshold network. In Bienenstock, E., Soulié, F. F., and Weisbuch, G., editors, *Disordered Systems and Biological Organization*, pages 233–240, Berlin, Heidelberg. Springer Berlin Heidelberg.

LeCun, Y. (1989). Generalization and network design strategies. In Pfeifer, R., Schreter, Z., Fogelman, F., and Steels, L., editors, *Connectionism in perspective*. Elsevier.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

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

Li, J., Lavrukhin, V., Ginsburg, B., Leary, R., Kuchaiev, O., Cohen, J. M., Nguyen, H., and Gadde, R. T. (2019a). Jasper: An end-to-end convolutional neural acoustic model. In *Proc. Interspeech 2019*, pages 71–75.

Li, J., Zhao, R., Hu, H., and Gong, Y. (2019b). Improving RNN transducer modeling for end-to-end speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 114–121.

Li, M., Liu, M., and Masanori, H. (2019). End-to-end speech recognition with adaptive computation steps. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6246–6250.

Lippmann, R. P. (1989). Review of neural networks for speech recognition. *Neural Computation*, 1(1):1–38.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lu, L., Kong, L., Dyer, C., Smith, N. A., and Renals, S. (2016). Segmental recurrent neural networks for end-to-end speech recognition. In *Proceedings of Interspeech 2016*, pages 385–389.

Luo, Y., Chiu, C., Jaitly, N., and Sutskever, I. (2017). Learning online alignments with continuous rewards policy gradient. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2801–2805.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Macleod, A. and Summerfield, Q. (1987). Quantifying the contribution of vision to speech perception in noise. *British Journal of Audiology*, 21(2):131–141. PMID: 3594015.

Makino, T., Liao, H., Assael, Y., Shillingford, B., Garcia, B., Braga, O., and Siohan, O. (2019). Recurrent neural network transducer for audio-visual speech recognition. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 905–912.

Massaro, D. W. and Stork, D. G. (1998). Speech recognition and sensory integration: A 240-year-old theorem helps explain how people and machines can integrate auditory and visual information to understand speech. *American Scientist*, 86(3):236–244.

Matthews, I., Potamianos, G., Neti, C., and Luettin, J. (2001). A comparison of model and transform-based visual features for audio-visual LVCSR. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2001)*, pages 825–828.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal forced aligner: Trainable text-speech alignment using Kaldi. In *Proc. Interspeech 2017*, pages 498–502.

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264(5588):746–748.

Moritz, N., Hori, T., and Le, J. (2020). Streaming automatic speech recognition with the transformer model. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078.

Moritz, N., Hori, T., and Roux, J. L. (2019). Triggered attention for end-to-end speech recognition. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5666–5670.

Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? In *Advances in Neural Information Processing Systems 32*, pages 4694–4703. Curran Associates, Inc.

Narayanan, A., Prabhavalkar, R., Chiu, C., Rybach, D., Sainath, T. N., and Strohman, T. (2019). Recognizing long-form speech using streaming end-to-end models. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 920–927.

Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., and Vergyri, D. (2001). Large-vocabulary audio-visual speech recognition: a summary of the Johns Hopkins Summer 2000 Workshop. In *Proceedings of the 2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564)*, pages 619–624.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696. ACM.

Oliver, B. M., Pierce, J. R., and Shannon, C. E. (1948). The philosophy of PCM. *Proceedings of the IRE*, 36(11):1324–1331.

Oviatt, S. (2002). Breaking the robustness barrier: Recent progress on the design of robust multimodal systems. In Zelkowitz, M. V., editor, *Advances in Computers*, volume 56, pages 305–341. Elsevier.

O'Neill, J. J. (1954). Contributions of the visual components of oral symbols to speech comprehension. *Journal of Speech and Hearing Disorders*, 19(4):429–439.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. In *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE.

Papandreou, G., Katsamanis, A., Pitsikalis, V., and Maragos, P. (2009). Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

Parker, D. B. (1985). Learning-logic: Casting the cortex of the human brain in silicon. Technical report, Center for Computational Research in Economics and Management Science, MIT.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org.

Petridis, S., Li, Z., and Pantic, M. (2017a). End-to-end visual speech recognition with LSTMs. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2592–2596.

Petridis, S. and Pantic, M. (2016). Deep complementary bottleneck features for visual speech recognition. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2304–2308.

Petridis, S., Stafylakis, T., Ma, P., Cai, F., Tzimiropoulos, G., and Pantic, M. (2018a). End-to-end audiovisual speech recognition. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6548–6552.

Petridis, S., Stafylakis, T., Ma, P., Tzimiropoulos, G., and Pantic, M. (2018b). Audio-visual speech recognition with a hybrid CTC/attention architecture. In *Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 513–520.

Petridis, S., Wang, Y., Li, Z., and Pantic, M. (2017b). End-to-end multi-view lipreading. In Tae-Kyun Kim, Stefanos Zafeiriou, G. B. and Mikolajczyk, K., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 161.1–161.12. BMVA Press.

Potamianos, G., Marcheret, E., Mroueh, Y., Goel, V., Koumbaroulis, A., Vartholomaios, A., and Thermos, S. (2017). Audio and visual modality combination in speech processing applications. In Oviatt, S., Schuller, B., Cohen, P. R., Sonntag, D., Potamianos, G., and Krüger, A., editors, *The Handbook of Multimodal-Multisensor Interfaces*, pages 489–543. ACM and Morgan & Claypool, New York, NY, USA.

Potamianos, G., Neti, C., Gravier, G., Garg, A., and Senior, A. W. (2003). Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326.

Povey, D., Hadian, H., Ghahremani, P., Li, K., and Khudanpur, S. (2018). A time-restricted self-attention layer for ASR. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878.

Prabhavalkar, R., Rao, K., Sainath, T. N., Li, B., Johnson, L., and Jaitly, N. (2017). A comparison of sequence-to-sequence models for speech recognition. In *Proc. Interspeech 2017*, pages 939–943.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., and Collobert, R. (2019). Wav2letter++: A fast open-source speech recognition system. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6460–6464.

Pundak, G. and Sainath, T. N. (2016). Lower frame rate neural network acoustic models. In *Interspeech 2016*, pages 22–26.

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S., and Sainath, T. N. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–1.

Rabiner, L. and Schafer, R. (2010). *Theory and Applications of Digital Speech Processing*. Prentice Hall Press, USA, 1st edition.

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

Raffel, C., Luong, M.-T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2837–2846. JMLR.org.

Rao, K., Sak, H., and Prabhavalkar, R. (2017). Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In *Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 193–199.

Reddy, D. R. (1976). Speech recognition by machine: A review. *Proceedings of the IEEE*, 64(4):501–531.

Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. (1994). Connectionist probability estimators in HMM speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2(1):161–174.

Renals, S. and Swietojanski, P. (2017). Distant speech recognition experiments using the AMI corpus. In Watanabe, S., Delcroix, M., Metze, F., and Hershey, J. R., editors, *New Era for Robust Speech Recognition: Exploiting Deep Learning*, pages 355–368. Springer International Publishing, Cham.

Robert-Ribes, J., Piquemal, M., Schwartz, J.-L., and Escudier, P. (1996). Exploiting sensor fusion architectures and stimuli complementarity in AV speech recognition. In Stork, D. G. and Hennecke, M. E., editors, *Speechreading by Humans and Machines: Models, Systems, and Applications*, pages 193–210. Springer Berlin Heidelberg, Berlin, Heidelberg.

Robinson, A. J. (1994). An application of recurrent nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305.

Robinson, T., Hochberg, M., and Renals, S. (1996). The use of recurrent neural networks in continuous speech recognition. In Lee, C.-H., Soong, F. K., and Paliwal, K. K., editors, *Automatic Speech and Speaker Recognition: Advanced Topics*, pages 233–258. Springer US, Boston, MA.

Rogozan, A. (1999). Discriminative learning of visual data for audiovisual speech recognition. *International Journal on Artificial Intelligence Tools*, 8(1):43–52.

Rosenblum, L. D. (2008). Primacy of multimodal speech perception. In Pisoni, D. B. and Remez, R. E., editors, *The Handbook of Speech Perception*, chapter 3, pages 51–78. John Wiley & Sons, Ltd.

Rosenblum, L. D., Miller, R. M., and Sanchez, K. (2007). Lip-read me now, hear me better later. *Psychological Science*, 18(5):392–396. PMID: 17576277.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, page 318–362. MIT Press, Cambridge, MA, USA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Saffran, J. R., Newport, E. L., and Aslin, R. N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35(4):606 – 621.

Sainath, T. N., Chiu, C., Prabhavalkar, R., Kannan, A., Wu, Y., Nguyen, P., and Chen, Z. (2018). Improving the performance of online neural transducer models. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5864–5868.

Sainath, T. N., He, Y., Li, B., Narayanan, A., Pang, R., Bruguier, A., Chang, S., Li, W., Alvarez, R., Chen, Z., Chiu, C., Garcia, D., Gruenstein, A., Hu, K., Kannan, A., Liang, Q., McGraw, I., Peyser, C., Prabhavalkar, R., Pundak, G., Rybach, D., Shangguan, Y., Sheth, Y., Strohman, T., Visontai, M., Wu, Y., Zhang, Y., and Zhao, D. (2020). A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6059–6063.

Saon, G., Kurata, G., Sercu, T., Audhkhasi, K., Thomas, S., Dimitriadis, D., Cui, X., Ramabhadran, B., Picheny, M., Lim, L.-L., Roomi, B., and Hall, P. (2017). English conversational telephone speech recognition by humans and machines. In *Proc. Interspeech 2017*, pages 132–136.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Schwartz, J.-L., Robert-Ribes, J., and Escudier, P. (1998). Ten years after Summerfield: A taxonomy of models for audio–visual fusion in speech perception. In Campbell, R., Dodd, B., and Burnham, D., editors, *Hearing by eye II: Advances in the psychology of speechreading and auditory-visual speech.*, pages 85–108. Psychology Press/Erlbaum (UK) Taylor & Francis, Hove, England.

Schwartz, J.-L. and Savariaux, C. (2014). No, there is no 150 ms lead of visual speech on auditory speech, but a range of audiovisual asynchronies varying from small audio lead to large audio lag. *PLOS Computational Biology*, 10(7):1–10.

Seymour, R., Stewart, D., and Ming, J. (2007). Comparison of image transform-based features for visual speech recognition in clean and corrupted videos. *EURASIP Journal on Image and Video Processing*, 2008(1):810362.

Shillingford, B., Assael, Y., Hoffman, M. W., Paine, T., Hughes, C., Prabhu, U., Liao, H., Sak, H., Rao, K., Bennett, L., Mulville, M., Denil, M., Coppin, B., Laurie, B., Senior, A., and de Freitas, N. (2019). Large-scale visual speech recognition. In *Proc. Interspeech 2019*, pages 4135–4139.

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

Stafylakis, T. and Tzimiropoulos, G. (2017). Combining residual networks with LSTMs for lipreading. In *Proc. Interspeech 2017*, pages 3652–3656.

Sterpu, G. and Harte, N. (2017). Towards lipreading sentences using active appearance models. In *AVSP*, Stockholm, Sweden.

Sterpu, G., Saam, C., and Harte, N. (2018a). Attention-based Audio-Visual Fusion for Robust Automatic Speech Recognition. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pages 111–115, New York, NY, USA. ACM.

Sterpu, G., Saam, C., and Harte, N. (2018b). Can DNNs Learn to Lipread Full Sentences? In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 16–20.

Sterpu, G., Saam, C., and Harte, N. (2020a). How to Teach DNNs to Pay Attention to the Visual Modality in Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1052–1064.

Sterpu, G., Saam, C., and Harte, N. (2020b). Should we hard-code the recurrence concept or learn it instead ? exploring the transformer architecture for audio-visual speech recognition. In *Proc. Interspeech 2020*, pages 3506–3509.

Sterpu, G., Saam, C., and Harte, N. (2021). Learning to count words in fluent speech enables online speech recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 38–45.

Stewart, D., Seymour, R., Pass, A., and Ming, J. (2014). Robust audio-visual speech recognition under noisy audio-video conditions. *IEEE Transactions on Cybernetics*, 44(2):175–184.

Stork, D. G. and Hennecke, M. E. (1996). Speechreading: an overview of image processing, feature extraction, sensory integration and pattern recognition techniques. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages XVI–XXVI.

Sumby, W. H. and Pollack, I. (1954). Visual contribution to speech intelligibility in noise. *The Journal of the Acoustical Society of America*, 26(2):212–215.

Summerfield, Q. (1987). Some preliminaries to a comprehensive account of audio-visual speech perception. In Dodd, B. and Campbell, R., editors, *Hearing by eye: The psychology of lip-reading.*, pages 3–51. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog), 13 March*.

Tan, M. and Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA. PMLR.

Tang, H., Lu, L., Kong, L., Gimpel, K., Livescu, K., Dyer, C., Smith, N. A., and Renals, S. (2017). End-to-end neural segmental models for speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1254–1264.

Tao, F. and Busso, C. (2018a). Aligning audiovisual features for audiovisual speech recognition. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Tao, F. and Busso, C. (2018b). Gating neural network for large vocabulary audiovisual speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(7):1290–1302.

Tao, F. and Busso, C. (2021). End-to-end audiovisual speech recognition system with multitask learning. *IEEE Transactions on Multimedia*, 23:1–11.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2020). Efficient transformers: A survey.

Taylor, S. L., Mahler, M., Theobald, B.-J., and Matthews, I. (2012). Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation*, EUROSCA'12, page 275–284, Goslar, DEU. Eurographics Association.

Thangthai, K., Bear, H. L., and Harvey, R. (2017). Comparing phonemes and visemes with DNN-based lipreading. In *Proceedings of the Workshop on Lip-Reading using deep learning methods*, BMVC 2017.

Thangthai, K. and Harvey, R. (2018). Building large-vocabulary speaker-independent lipreading systems. In *Proc. Interspeech 2018*, pages 2648–2652.

The TensorFlow Model Garden (2020). Transformer Translation Model. `https://github.com/tensorflow/models/tree/r2.1.0/official/nlp/transformer`. Online, Accessed: 25 May 2020.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Scripta series in mathematics. Winston, Washington, DC. Trans. from Russian.

Tsai, Y.-H. H., Bai, S., Liang, P. P., Kolter, J. Z., Morency, L.-P., and Salakhutdinov, R. (2019). Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569, Florence, Italy. Association for Computational Linguistics.

Tzimiropoulos, G. and Pantic, M. (2014). Gauss-Newton deformable part models for face alignment in-the-wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

Wand, M., Koutník, J., and Schmidhuber, J. (2016). Lipreading with long short-term memory. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6115–6119.

Wand, M., Schmidhuber, J., and Vu, N. T. (2018). Investigations on end- to-end audiovisual fusion. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3041–3045.

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

Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., Raj, D., Snyder, D., Subramanian, A. S., Trmal, J., Yair, B. B., Boeddeker, C., Ni, Z., Fujita, Y., Horiguchi, S., Kanda,

N., Yoshioka, T., and Ryant, N. (2020). CHiME-6 Challenge: Tackling multi-speaker speech recognition for unsegmented recordings. In *Proceedings of the 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, Barcelona, Spain.

Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In Drenick, R. F. and Kozin, F., editors, *System Modeling and Optimization*, pages 762–770, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423.

Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C. (2015). *The HTK Book, version 3.5*. Cambridge University Engineering Department, Cambridge, UK.

Yu, W., Zeiler, S., and Kolossa, D. (2021). Fusing information streams in end-to-end audio-visual speech recognition. *Pre-print accepted at ICASSP 2021*.

Yuhas, B. P., Goldstein, M. H., and Sejnowski, T. J. (1989). Integration of acoustic and visual speech signals using neural networks. *IEEE Communications Magazine*, 27(11):65–71.

Yuhas, B. P., Goldstein, M. H., Sejnowski, T. J., and Jenkins, R. E. (1990). Neural network models of sensory integration for improved vowel recognition. *Proceedings of the IEEE*, 78(10):1658–1668.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *CoRR*, abs/1409.2329.

Zeyer, A., Bahar, P., Irie, K., Schlüter, R., and Ney, H. (2019). A comparison of transformer and LSTM encoder decoder models for ASR. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15.

Zeyer, A., Beck, E., Schlüter, R., and Ney, H. (2017). CTC in the context of generalized full-sum HMM training. In *Proc. Interspeech 2017*, pages 944–948.

Zeyer, A., Irie, K., Schlüter, R., and Ney, H. (2018). Improved training of end-to-end attention models for speech recognition. In *Proc. Interspeech 2018*, pages 7–11.

Zeyer, A., Merboldt, A., Schlüter, R., and Ney, H. (2020). A new training pipeline for an improved neural transducer. In *Proc. Interspeech 2020*.

Zeyer, A., Schlüter, R., and Ney, H. (2021). A study of latent monotonic attention variants.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhou, P., Yang, W., Chen, W., Wang, Y., and Jia, J. (2019). Modality attention for end-to-end audio-visual speech recognition. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6565–6569.

Zhou, Z., Zhao, G., Hong, X., and Pietikäinen, M. (2014). A review of recent advances in visual speech decoding. *Image and Vision Computing*, 32(9):590 – 605.