Developing tools for audio-visual speech recognition

Sonia M Marshall

MInf Project (Part 1) Report

Master of Informatics
School of Informatics
University of Edinburgh

2021

Abstract

Audio-visual speech recognition combines automatic speech recognition with lip reading to improve recognition performance. This project focused on the lip reading aspect, which is a challenging task due to many sources of variation found in real-world environments. The experiments investigated how well a pre-trained lip reading model can generalise to a new dataset. The model, based on the transformer architecture, was evaluated on the LRS2 and LRS3-TED datasets. Further training using LRS3-TED was performed to fine-tune the model, experimenting with different combinations of batch sizes and learning rates. The best fine-tuned model achieved a small improvement in Word Error Rate (WER) on the LRS3-TED test dataset, achieving 78.0% WER. Future work is suggested to further improve performance, including using a TED language model and training for more epochs.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Lip reading is a technique used to understand speech by interpreting movements of a speaker's lips. The McGurk effect [26] demonstrates that listeners use this visual information in addition to audio to understand what a speaker is saying. By training a suitable model it is possible for a machine to perform lip reading. While lip reading, or visual speech recognition, uses only the visual input, it can be combined with automatic speech recognition (ASR) in a multi-modal approach, referred to as audio-visual speech recognition (AVSR).

ASR is already used in many practical applications, such as video captioning and personal assistants on our mobile phones. In these real-world applications the input is not always clean, potentially containing background noise or audio from another speaker. Since lip reading uses only the visual aspect, it is unaffected by background noise in the audio. It is therefore advantageous to make use of both sources of information: AVSR models have been found to improve performance compared to models using only lip reading or only ASR [4].

Lip reading is a more challenging task than ASR - even state-of-the-art lip reading models and professional human lip readers achieve a much higher Word Error Rate (WER) than ASR systems on audio-visual datasets. In [31], the state-of-the-art V2P lip reading model achieved 40.9% WER on the Large-Scale Visual Speech Recognition (LSVSR) dataset and the older LipNet model achieved 72.7%, while professional lip readers achieved a WER of 86.4% (with context) and 92.9% (without context). In comparison, an ASR model trained on the LSVSR audio achieved a WER of 18.3%. These results demonstrate firstly the difficulty of lip reading, but also that lip reading models can provide significant improvements compared to human performance.

Why is lip reading so challenging? It faces difficulties common to many image processing tasks. The models must be able to cope with variation in lighting, background and the appearance of speakers, as well as the possibility of occlusion and variation in pose as the speaker moves their head. Additionally, it is difficult to distinguish between some words purely from lip reading. A phoneme is the contrasting sound that

changes the meaning of a word in a pair of words which are otherwise the same. We can perceive a difference in sound when we hear different phonemes, however, we cannot distinguish all phonemes visually because not all articulators are externally visible. For example the words 'file' and 'vile', or 'ship' and 'chip', sound different but look the same on a speaker's lips. Words that are visually difficult to distinguish tend to be more easily distinguishable in the audio, so by using audio-visual speech recognition this ambiguity can be more easily resolved.

Large training datasets are required for lip reading models to learn to cope with the variation found in real-world applications. It is not easy to acquire such large quantities of suitably pre-processed audio-visual data. The ideal scenario would be to have a model which can generalise well to new lip reading contexts having been trained on an existing dataset - which is the motivation for this project's investigation into lip reading models.

1.2 Project Goals and Achievements

Since this was the first year of a two-year project on audio-visual speech recognition, my approach was to focus first on the visual aspect, and to incorporate the auditory aspect next year. This year my goals were to become familiar with the current (and historical) field of visual speech recognition and acquire a working lip reading model, evaluate its generalisation to an unseen dataset, and improve its performance by modifying the model or training it further.

I have achieved these goals: I found and set up a pre-trained lip reading model and acquired suitable datasets for the experiments. I evaluated the lip reading model on LRS2 [4] (a dataset it was trained on) and LRS3-TED [3] (an unseen dataset) and compared the performance. I implemented code to train the lip reading model and trained it on a subset of the LRS3-TED training data, achieving a Word Error Rate (WER) of 78.0% on the LRS3-TED test set.

I had planned to also train a language model more suitable for the LRS3-TED dataset than the LRS2 language model, and evaluate the performance using that model. However since training the lip reading model took more time than originally expected this has been left for next year, as detailed in Section 6.2.

Chapter 2

Background

2.1 Lip reading

As deep learning techniques have become more advanced and large audio-visual datasets have become available (see Section 2.2) there has been a shift in visual speech recognition methods. Lip reading systems are moving away from using the more traditional approaches, towards using deep learning methods which can provide improved performance. [16] offers a detailed survey of automatic lip reading systems from 2007 to 2018.

Lip reading begins with a sequence of video frames which undergo pre-processing - face detection and facial landmark localisation, particularly lip localisation - then visual features are extracted from each frame, and finally these features are classified into a sequence of speech units (e.g. phonemes, characters, words) [18]. Early lip reading methods used traditional handcrafted visual features. These included geometric features [29], appearance-based approaches [8], image transforms [30] and combinations of these methods. The resulting visual features were usually classified using Hidden Markov Models (HMMs), since they are effective at modelling temporal sequences - HMMs were also commonly used in traditional ASR systems [17].

By using a deep learning approach we avoid having to use handcrafted features as described above, and instead allow a neural network to learn the most useful features from the data. A common combination is to use a network architecture suited to extracting visual features, such as a convolutional neural network (CNN), followed by a network architecture suitable for modelling sequences, such as a Long Short-Term Memory (LSTM) network or Gated Recurrent Unit (GRU) [6] [12] [10].

Lip Reading in the Wild [11] was one of the first end-to-end trainable lip reading models, using CNNs to predict at the word level. Word-level predictions are however not ideal for a real-world application, since word boundaries must be known beforehand and words not in the vocabulary cannot be recognised. Different levels of classification can be used - for example, systems can output sequences of characters or phonemes which are then decoded into words or sentences/phrases.

LipNet [6] was the first end-to-end sentence-level lip reading model. It used spatio-

temporal CNNs and GRUs, predicting at the character level and using connectionist temporal classification (CTC) loss to decode the sequence of words. This model achieved a much lower WER compared to human lip readers on the Grid corpus [13], and was able to generalise well to unseen speakers from the dataset. However, as detailed in Section 2.2 the Grid corpus has a very specific and limited syntax for sentences. When tested on the more challenging LSVSR dataset [31] this model had a much higher WER.

The Vision to Phoneme (V2P) neural network [31] had a similar architecture to LipNet, but used LSTMs instead of GRUs, amongst other changes, and achieved improved performance on the LSVSR dataset. V2P predicts a sequence of phoneme distributions, which are then fed into a phoneme-to-word decoder to produce a sequence of words. Compared to systems performing word-level predictions, this separation brings the advantage that the vocabulary can be extended without retraining the network. This model was found to generalise very well to the LRS3-TED dataset [3], which it had not been trained on.

[2] compared three different DNN architectures for lip reading, predicting at the character level. One used a bidirectional LSTM architecture, another was fully convolutional and the third was based on the transformer architecture [34]. The transformer model was the best performing model out of the three, achieving the lowest WER on the LRS2 dataset [4], and so was made publicly available - this was the lip reading model used in this project.

The speech recognition models mentioned so far have used only lip reading - no audio. Since lip reading is more challenging than ASR and results in higher WER, potentially its most interesting use is in combination with ASR. As mentioned in Section 1.1, audio-visual speech recognition enables improvements in performance in comparison to using only one source. The Watch, Listen, Attend and Spell network (WLAS) [10] introduced a dual attention mechanism that allowed the model to use either or both the audio and visual input. This dual attention mechanism was also used in the transformer based architectures of [4]. When using both input sources the WER decreased compared to using audio only, and this decrease in WER was larger for noisier audio. This demonstrates the power of using both sources - lip reading allows us to make the most of the visual information which is unaffected by the noise in the audio.

2.2 Audio-visual Datasets

The audio-visual datasets used to train and test lip reading models are central to the task of lip reading, and their development has been closely linked with the development of novel lip reading methods. The shift towards using deep learning techniques for lip reading has been matched by a shift from small and constrained audio-visual datasets towards much larger and more varied datasets. Audio-visual datasets typically contain video clips of speakers, closely cropped to their faces, and a transcription of the audio matching each video. The majority of datasets are for the English language, although there are a few in other languages [28] [33]. [16] gives a comprehensive comparison and review of audio-visual datasets up to 2018.

The older datasets tended to be small and quite constrained. They were often produced from scratch in a controlled lab environment, from a small number of speakers reading from a script. Some of the earliest datasets focused on the simplest task of recognising digits or letters of the alphabet [27] [25]. Others were more relevant for the general task of recognising natural speech - containing words, phrases or sentences. For example the Grid corpus [13], which has been widely used to evaluate and compare the performance of lip reading models, contains sentences constructed from a limited vocabulary and grammar.

The problem with many of these older datasets is that lip reading models trained on them will not generalise well to real-world scenarios. They will be unable to cope with environments with fewer or no constraints. For example, recordings in lab environments do not typically contain background noise that would normally be found in an uncontrolled environment. In the presence of background noise humans subconsciously modify their speech production in what is known as the Lombard effect [23]. As well as acoustic and phonetic modification, there are also articulatory adjustments [5]. This means that noise in the audio can affect the visual aspect as well, and so should be considered when creating audio-visual datasets. To help overcome this some researchers have added speech-shaped noise in the background during recording of the speakers [5], or recorded speech in different noise conditions such as a vehicle's windows being open or closed [22]. However a better solution has more recently become possible. Rather than recording datasets from scratch, they can instead be built using existing videos of real-world speech.

Lip Reading in the Wild (LRW) [11] was a key dataset that started the trend to build larger, less constrained datasets by extracting clips from existing video content (in particular, from BBC TV news broadcasts). Using real-world videos provides a larger and more diverse set of speakers, larger vocabulary, and a greater variety of lighting and backgrounds than could be produced in a controlled lab environment. The videos capture how people naturally speak, rather than test subjects reciting a script. The size and diversity of these datasets make them more suitable than the older datasets for training deep learning models capable of generalisation.

LRW was built for the purpose of recognising individual words. This is quite limited - a more interesting task is recognising sentences or phrases. The same researchers hence followed up LRW with the Lip Reading Sentences (LRS) dataset [10], built from a similar selection of BBC TV news broadcasts. There have since been a variety of sentence-level audio-visual datasets created from various sources including: BBC TV programmes (LRS2 [4]), TED and TEDx talks (LRS3-TED [3], AVSpeech [15]) and YouTube videos (LSVSR [31]). LSVSR is the most recent of these 'in the wild' datasets, currently the largest and containing the most diverse content.

While many datasets focus on frontal or near-frontal views of faces [13] [11] [10], it is realistic to assume that the angle of a speaker's face may vary due to natural head movements. One approach to provide robustness to this variation in viewpoint is to record both frontal and profile views of speakers [5]. However, this requires storing twice as much video data for each utterance and the videos show only two views and no angles in between. Recent improvements in face and facial landmark

detection have allowed better detection of profile faces in existing videos, allowing the 'in the wild' datasets to contain a wider range of viewpoints within each video clip [12] [4] [3]. It was found that lip reading in profile is more difficult than lip reading frontal views, but performance can be significantly improved by training on a variety of viewpoints, compared to training only on frontal views [12]. Therefore, datasets containing multiple viewpoints are more challenging but can generalise better to real-world scenarios.

In this project the LRS2 and LRS3-TED datasets were used. They were chosen as they can be readily acquired for use, unlike for example the larger and more varied LSVSR dataset. LRS3-TED has the benefit that it is built from publicly available TED data, so its use is not as restricted as LRS2 which uses data from the BBC. The 'in the wild' nature of the data makes them preferable to the older more constrained datasets such as the Grid corpus. They both contain a variety of viewpoints. Both are challenging datasets which were produced by the same group, using a similar pipeline, but from different sources, which makes them interesting to compare.

Chapter 3

Technical Background

This chapter describes a few deep learning architectures relevant to lip reading. Each architecture builds upon the previous, aiming to solve its issues and improve performance.

3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [14] are a neural network architecture that is useful for modelling sequences. Unlike feedforward neural networks, RNNs contain feedback loops that allow them to remember information between time steps, by passing a hidden state from one time step to the next. This hidden state acts as the memory of the network - it is a representation of all the previous inputs - and so allows the modelling of dependencies in the sequence.

RNNs however suffer from the vanishing gradient problem. The weights in the network are updated using backpropagation through time. As the gradients are propagated back through the layers (time steps) they get smaller and smaller, so that the early layers learn much more slowly than later layers. As a result the network 'forgets' the information from the early inputs - the later hidden states will contain little information from these early inputs. This short-term memory problem means that RNNs cannot model long range dependencies very well.

3.1.1 Long Short-Term Memory

The Long Short-Term Memory (LSTM) [20] architecture is a variation of the RNN architecture introduced to solve the vanishing gradient problem and allow modelling of longer sequences with long term dependencies.

In addition to the hidden state, the LSTM has a memory cell which uses a gradient of 1 between time steps, and so does not suffer from the vanishing gradient problem. The LSTM uses gates to decide what information from the current input and previous hidden state to store in the memory cell at each time step. This means that only relevant

information is kept while less relevant information can be forgotten. Each gate is a neural network layer.

The forget gate regulates what information to keep and what to remove from the previous memory cell. The input gate is used to update the current memory cell, and finally the output gate computes the next hidden state. The separation of the memory cell and hidden state allows information from early inputs to be retained for later in the memory cell, without necessarily being passed immediately to the hidden state, which is used for predictions.

3.2 Encoder-Decoder Architectures

Lip reading involves transforming sequences of video frames into sequences of speech units (e.g. phonemes, characters, words). Usually the number of input video frames will not be the same as the number of output speech units, and this difference in input and output length is also the case for other sequence processing tasks, such as machine translation. A single RNN or LSTM is therefore not suitable for these tasks, because it cannot be used to map an input sequence to an output sequence of a different length.

The solution to this is using a sequence-to-sequence model [32], which uses two neural networks (e.g. LSTMs). One acts as an encoder, mapping the entire input sequence to a fixed-size vector (the hidden state that the encoder outputs at the final time step). The second acts as a decoder, taking the encoder vector as input and predicting a sequence of outputs.

3.2.1 Attention

A problem with the basic encoder-decoder architecture is that the internal representation of the input produced by the encoder is a fixed size. This means that long sequences are represented by the same size of vector as shorter sequences. Some useful information about the long sequences may be lost due to this limited representation. To improve the performance, particularly for decoding long sequences, the attention mechanism [7] can be used.

Instead of mapping the entire input sequence to one vector representation, a context vector is created to represent each item in the input sequence. Different lengths of input sequences will therefore be represented by different numbers of vectors. The attention mechanism uses an alignment model to allow the decoder to focus on only the most relevant parts of the input sequence at each time step. To predict the next output the decoder uses the context vectors for the parts of the input that are most relevant to the previous output.

3.2.2 Transformers

[34] proposed the transformer architecture which removes the use of RNNs, instead using only the attention mechanism. Removing the sequential nature of the model makes it more efficient and scalable. This architecture is also better at modelling long range

dependencies, since it reduces the path length between input and output positions, and therefore less information is lost.

The transformer architecture uses two types of the attention mechanism described in the previous section. Self-attention [9] allows the model to transform an input sequence into a representation which relates different items in the input sequence to each other. Multi-head attention means that there are multiple parallel attention layers, which enables the model to attend to multiple different parts of the input at once.

Since the sequential nature of the model has been removed the encoder is given an embedding of the whole input sequence at once, alongside a positional encoding vector to provide information about the order of the input sequence. The encoder uses multi-head self-attention to encode the input sequence into an internal representation made up of key, value pairs. The decoder is fed the previous decoder output and corresponding positional encoding vector and performs multi-head self-attention to produce a query vector. A third multi-head attention block is used to compare the decoder query vector to the encoder key vectors to find the most relevant values to use. In this way the decoder can attend to any and all important positions in the input sequence and use this information to predict the next output.

Chapter 4

Experimental Setup

In this project I aimed to train a lip reading model suitable for the LRS3-TED dataset, by adapting a pre-trained LRS2 lip reading model from [2]. This chapter introduces the datasets and lip reading model that were used in more detail.

4.1 Datasets

The audio-visual datasets used in this project were LRS2 and LR3-TED (first introduced in Section 2.2). The datasets are very large, taking up 51GB and 135GB of space respectively. LRW was also acquired, but finally was not used since this project focused on recognising full sentences.

The datasets consist of square video clips of speakers' faces and corresponding text files containing the transcription. See Figure 4.1 for example video frames extracted from two different videos in the LRS3-TED test dataset. The top row shows a varying viewpoint within a single clip, while the bottom row shows a frontal view of the speaker's face throughout the clip. The transcription for the top row example is: `HOW DO YOU KNOW FOR SURE.`

Figure 4.1: Video frames extracted from two different videos in the LRS3-TED test set [3].

LRS2 contains sentences up to 100 characters in length, while LRS3-TED contains longer sentences. The longest sentence in the LRS3-TED data used in this project has 149 characters (see Table 4.2).

LRS2 is split up into pre-train, train, validation and test sets. LRS3-TED is split up into pre-train, trainval and test sets, and the data is structured into folders where each folder contains videos from the same TED talk (hence, the same speaker). The number

Table 4.1: Number of speakers, utterances, words and vocabulary in the different partitions of the LRS3-TED and LRS2 datasets. [3] [4]

Dataset	**# speakers**	**# utterances**	**# word instances**	**Vocab**
LRS3-TED Pre-train	5,090	118,516	3.9M	51k
LRS3-TED Trainval	4,004	31,982	358k	17k
LRS3-TED Test	412	1,321	10k	2k
LRS2 Pre-train	-	96,318	2,064,118	41,427
LRS2 Train	-	45,839	329,180	17,660
LRS2 Validation	-	1,082	7,866	1,984
LRS2 Test	-	1,243	6,663	1,698

Table 4.2: Number of utterances, speakers and longest videos and transcriptions of the datasets used in the experiments.

Dataset	**# utterances**	**# speakers**	**Longest video (frames)**	**Longest transcription (characters)**
LRS3-TED training	10000	1172	157	149
LRS3-TED validation	500	499	157	142
LRS3-TED test	1321	412	157	129
LRS2 test	1243	-	145	96

of utterances and speakers in each partition of the datasets, amongst other details, are given in Table 4.1.

Since in this project I was adapting a pre-trained lip reading model it was not necessary to train on the entire LRS3-TED trainval set. Using all the data would have led to large training times unfeasible for this project, so I used a subset of the LRS3-TED trainval data. The training and validation sets used in my experiments were created by selecting training data from the top and validation data from the bottom of an ordered list of the trainval files. This ensured that the two sets contained no overlapping speakers.

The training set was formed by selecting the first 10,000 samples of the ordered list. Their order was then shuffled so that samples from the same speakers were not next to each other in training. In that way, each training batch contained samples from a variety of speakers, rather than all of the samples from just a few speakers.

The validation set was formed by selecting the last 5,000 samples from the ordered list, shuffling them, then selecting the bottom 500 samples. This was done so that the set contained a large variety of speakers, to better represent unseen data. The size of the validation set was chosen to be 500 so that it was fast to evaluate the model on.

Since the lip reading model does not generalise well to longer sentences than it has seen in training [2], it was ensured that the training set contained some sentences with length greater than or equal to the longest sentences in the validation and test sets. The sizes of the datasets used in the experiments and the lengths of the longest samples are given in Table 4.2.

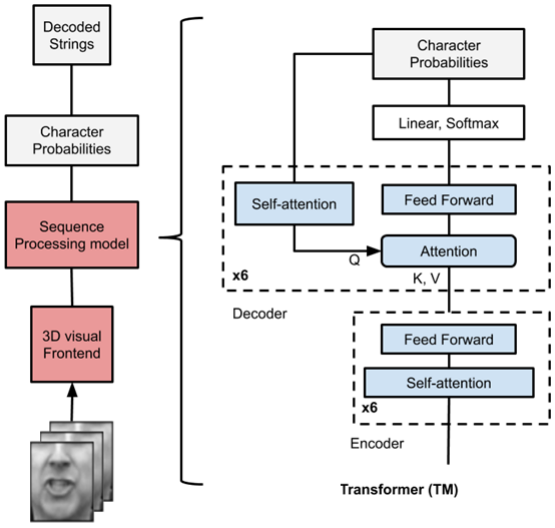


Figure 4.2: The architecture of the lip reading model. K, V and Q denote the Key, Value and Query tensors for the multi-head attention. Diagram adapted from [2].

4.2 Lip reading model

The lip reading model used in this project is the best performing model from [2], based on the transformer architecture [34]. The model architecture can be seen in Figure 4.2. It works by passing a sequence of input video frames cropped to the lip region through the visual frontend (a spatio-temporal residual network) which outputs feature vectors. The sequence of feature vectors is then input to the sequence-to-sequence transformer model which outputs character probabilities. Beam search is then used to decode the output character probabilities into the final predicted sentence. Optionally, an external language model can be used during decoding to improve performance.

[2] trained the lip reading model using the LRW, MV-LRS and LRS2 datasets. They used the Adam optimiser [21] with an initial learning rate of 10^{-3}, reduced to 10^{-4} upon plateau, and all other Adam parameters were set to the default. The transformer model took the longest time to train compared to their other two proposed architectures - for full details of training see [2]. The character-level language model provided by [2] uses LSTMs and was trained on the subtitles of the full source videos used to generate the LRS2 training set. The language model is optional since the transformer lip reading model learns a language model internally via the teacher forcing training method.

4.3 Implementation details

[2] implemented the model using TensorFlow and the original implementation is available in their GitHub repository [1]. Modifications made during the project can be found in the submitted code repository.

In order to run my experiments which involved training the lip reading model, I had to implement code to run the training. The code to evaluate the model was already

provided. The model can be run in the training mode or the evaluation mode by running `main.py` with command line arguments specified in `config.py`.

To implement the training I edited `lip_model/training_graph.py`, `main.py` and `config.py` - setting up a training graph with an optimiser, updating the model weights after each batch and saving model checkpoints. I also added a decaying learning rate, and implemented gradient accumulation to allow larger batch sizes.

4.3.1 Training code

I added code to the training graph to create an Adam optimiser. The optimiser was then used to update the weights to minimise the loss during training.

The original implementation contained code to restore a model from a checkpoint. I added code to save model checkpoints during training. The saved checkpoints could then be restored later to evaluate the trained model.

At first I passed the optimiser a learning rate which remained the same throughout training. I later decayed the learning rate by passing the initial learning rate into an `inverse_time_decay` function. There are several different learning rate decay functions available in TensorFlow, including exponential decay, polynomial decay and cosine decay. However I decided that for this project it was not important to investigate which of these would work best, and instead chose one and used it throughout.

4.3.2 Gradient Accumulation

Due to limitations of GPU memory it was only possible to run training with a maximum batch size of 8 samples. Small batch sizes have been found to produce a lower generalisation error compared to large batches, however when batch normalisation and a large dataset is used, a slightly larger batch size has been found to be useful [24]. Larger batch sizes also reduce the training time, since the weights are updated fewer times. It is therefore beneficial to be able to train the model using a batch size larger than 8.

I implemented gradient accumulation to allow larger batch sizes than the GPU can handle at once. This works by computing the gradients for each minibatch (maximum size 8) but only applying the gradients (updating the weights) after accumulating the gradients of a certain number of minibatches. Instead of using the Adam optimiser's `minimize()` function which computes and applies the gradients in one step, this can be split into `compute_gradients()` and `apply_gradients()`. A command line argument `n_minibatches` was added to enable configuration of the batch size. The total batch size is the `batch_size` (e.g. 8) multiplied by `n_minibatches`.

4.4 Problems

Several issues were encountered which meant that the lip reading model was first evaluated on the full LRS3-TED dataset several weeks after I decided to use it and downloaded the code.

A disadvantage of this implementation of the model is that it is written in Python 2, which is no longer supported, and uses an old version of TensorFlow (version 1.12). This caused issues when setting up the Conda environment for running the model - due to deprecation, there were some package dependencies that could not be met. This was resolved by not including the dependencies required for optional visualisation of the input videos, attention matrices and predictions. While it is unfortunate that these cannot be visualised, it is preferable to the model not running at all. It is however not sustainable to keep using this implementation in the long run, and the code would at some point need to be ported to Python 3 and TensorFlow 2.0. Due to the size and complexity of the code it was deemed that this would take more time out of the project than it would be worth. There were additional issues with exceeding my allocated disk quota when installing the large TensorFlow dependencies, but these were resolved by clearing out unused packages.

Incorrectly set arguments led to a couple of errors when first running the model. The LRS3-TED videos are larger than the LRS2 videos, with a width and height of 224 pixels as opposed to 160 pixels. I initially resized the LRS3-TED videos to 160x160 pixels so that the model would run, before realising that the `img_width` and `img_height` arguments could be set to the appropriate size (224) instead. The other error occurred because the longest video in LRS3-TED was longer than the longest video in LRS2 (see Table 4.2). The arguments `time_dim` and `maxlen` had to be set appropriately to the maximum input video length in video frames or the maximum output sequence length in characters (whichever was larger).

Chapter 5

Experiments and Results

The lip reading model can be evaluated with or without using beam search, and with or without using the LRS2 language model. Table 5.1 shows the three different model setups used in the following experiments. The parameter values shown were the best values determined in [2].

5.1 Evaluating the LRS2 lip reading model

The aim of this experiment was to investigate and compare the performance of the LRS2 lip reading model on the LRS2 test set and the LRS3-TED test set.

The experiments were run on a single GeForce GTX TITAN GPU with 6083MiB memory.

5.1.1 Results

The WER, Character Error Rate (CER) and the time taken to run the model on the LRS2 test set, LRS3-TED test set and LRS3-TED validation set are shown in Table 5.2. The results for the LRS2 and LRS3-TED test sets are also visualised in Figure 5.1 for easier comparison.

Below are a few examples of predictions from the LRS3-TED test set using the LM model setup, in the format: `(wer) <true-transcription> --> <model-prediction>`.

Table 5.1: Three different setups used to evaluate the lip reading model.

Model setup	**Language model (LM) (y/n)**	**beam search (y/n)**	**Beam width**	**Test Augmentation Times**	**Length penalty**	**LM penalty**
no beam, no LM	n	n	0	0	-	-
no LM	n	y	5	2	0.6	0
LM	y	y	15	2	0.7	0.1

Examples with 0% WER (49 in total):

```
(wer=0.0) YOU-WANT-TO-WORK-FOR-HIM --> YOU-WANT-TO-WORK-FOR-HIM

(wer=0.0) NOW-I'M-READY-FOR-MY-INTERVIEW --> NOW-I'M-READY-FOR-MY-INTERVIEW

(wer=0.0) SO-IT'S-REALLY-IMPORTANT-THAT-YOU-KNOW-THAT-RIGHT-NOW-WE-HAVE-OVER
--> SO-IT'S-REALLY-IMPORTANT-THAT-YOU-KNOW-THAT-RIGHT-NOW-WE-HAVE-OVER

(wer=0.0) WE-CAN-DO-THIS --> WE-CAN-DO-THIS
```

Examples of long sentences:

```
(wer=84.6) SO-PEOPLE-HEAR-ABOUT-THIS-STUDY-AND-THEY'RE-LIKE-GREAT-IF-I-
WANT-TO-GET-BETTER-AT-MY-JOB-I-JUST-NEED-TO-UPGRADE-MY-BROWSER
--> PEOPLE-HEAR-ABOUT-THIS

(wer=70.0) AND-SO-ONE-OF-THE-MAJOR-CHALLENGES-OF-OUR-PROJECT-REALLY-
IS-TO-FIND-PHOTOGRAPHS-THAT-WERE-TAKEN-BEFORE-SOMETHING
--> AND-ONE-OF-THE-MAJOR-CHALLENGES

(wer=60.0) WHEN-YOU-REALLY-LOOK-AT-IT-HOW-IS-IT-THAT-YOUNG-PEOPLE-
SPEND-MOST-OF-THEIR-TIME-USING-NEW-TECHNOLOGIES
--> WHEN-YOU-REALLY-COULD-LOOK-AT-HOW-HE'S-GOING-TO-COME-AND-SPEND-
MOST-OF-THINGS-BECAUSE-YOU-COULD

(wer=38.1) SO-YOU-WANT-TO-GO-TO-THAT-BOARD-MEETING-BUT-YOU-ONLY-WANT-
TO-PAY-ATTENTION-TO-THE-BITS-THAT-INTEREST
--> SO-YOU-WON'T-GO-TO-THAT-POINT-ME-BUT-YOU-ONLY-WANT-TO-PAY-
ATTENTION-TO-MISS
```

5.1.2 Discussion

It can clearly be seen from Figure 5.1 that the LRS2 lip reading model did not generalise well to the unseen dataset, LRS3-TED. The WER and CER on the LRS3-TED datasets were higher by at least an absolute value of 30% and 23% respectively, compared to the LRS2 dataset. Therefore, the experiment in Section 5.2 aimed to tune the LRS2 lip reading model to better fit the LRS3-TED data.

It is important to note that in [4], a similar sequence-to-sequence transformer model (pre-trained on MV-LRS, LRS2 and LRS3-TED) achieved a WER of 48.3% on LRS2 after being fine-tuned on LRS2 training data, and 58.9% on LRS3-TED after being fine-tuned on LRS3-TED. This suggests that LRS3-TED is a more challenging dataset, perhaps due to containing a larger vocabulary and longer sentences. Therefore, a higher WER (and CER) on LRS3-TED is to be expected, although perhaps not as much as 30% higher. Additionally, I would suggest one difference between the datasets that may make it more difficult to generalise from LRS2 to LRS3-TED: microphones. In

Table 5.2: CER and WER on the LRS2 and LRS3-TED test sets and LRS3-TED validation set using the LRS2 lip reading model.

Dataset	Model Setup	CER (%)	WER (%)	Runtime (mins)
LRS2 test	no beam, no LM	38.3	58.4	19
	no LM	33.8	51.2	97
	LM	33.0	48.8	202
LRS3-TED test	no beam, no LM	61.8	89.3	36
	no LM	58.2	82.1	130
	LM	58.1	80.7	240
LRS3-TED validation	no beam, no LM	62.1	90.5	12
	no LM	60.0	83.2	24
	LM	63.0	82.7	38

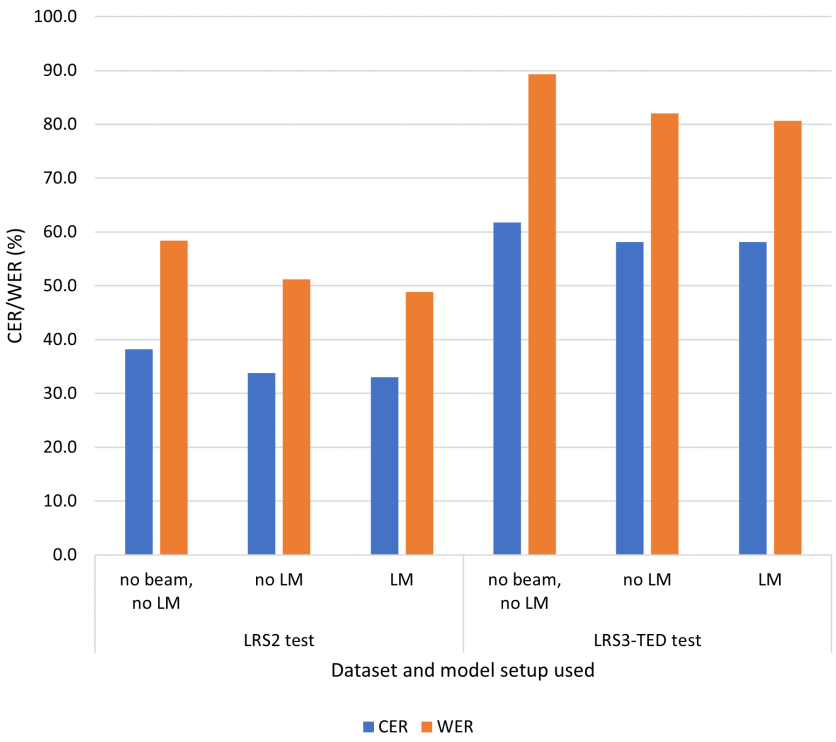


Figure 5.1: CER and WER on the LRS2 and LRS3-TED test datasets using the LRS2 lip reading model.

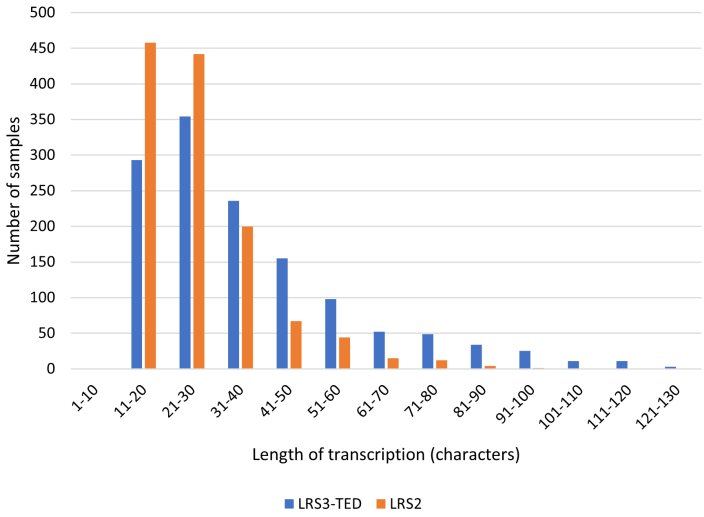


Figure 5.2: Distribution of LRS2 and LRS3-TED test set transcription lengths

many of the LRS3-TED videos the speakers have a microphone visible near their lips (see Figure 4.1) - this could potentially affect feature extraction, since the model is not used to recognising faces with microphones.

Table 5.2 shows the time taken to run each evaluation of the model, alongside the WER and CER, in different setups. The speed of running the most basic setup (no beam, no LM) makes it perfect for quickly evaluating performance, for example checking the WER of the validation set at various points during training (see Section 5.2), which takes only 12 minutes. Using beam search (and no LM) takes longer to run but improves the WER by around 7% for all the datasets, making it worth the time trade-off when evaluating on a test set. The wider the beam width, the lower the WER, but the longer it takes to run. Wider beam widths were briefly experimented with but it was concluded that the small decrease in WER that this achieved was not worth the much longer evaluation time. For example, increasing beam width from 15 to 35 (while using the LM) increased the runtime by around 3 hours, for a decrease in WER of just 0.05% on LRS2. Hence for all experiments the beam widths chosen in [2] and specified in Table 5.1 were used.

Figure 5.2 shows the distribution of the LRS2 and LRS3-TED test set transcription lengths (note that the LRS3-TED test set has 78 more sentences in total compared to the LRS2 test set). From this chart we can see that LRS2 has larger number of very short sentences (11-30 characters) compared to LRS3-TED, while LRS3-TED contains sentences longer than any found in LRS2 (above 100 characters) and also has a larger number of sentences in each length category above 31 characters. This suggests that the length penalty parameter used during decoding may not have the same optimal value for the LRS3-TED data as the value found in [2] for the LRS2 data. A value which favours slightly longer sentences may result in a reduction in WER on the LRS3-TED test set. It would be worth investigating the effect of changing the length penalty on the WER in a future experiment.

Although using an external language model is not necessary due to the sequence-to-

sequence transformer architecture (see Section 4.2), adding the LRS2 language model resulted in a small improvement in performance on all datasets compared to using beam search with no LM. There was an absolute decrease of 2.4% in WER for the LRS2 test set, 1.4% for the LRS3-TED test set and 0.5% for the LRS3-TED validation set. Using beam search with the LM took longer than running beam search without the LM. Since the language model is trained on transcriptions of the source BBC videos for the LRS2 training set it is well suited to predicting LRS2 data, which explains why its use results in a greater decrease in WER for the LRS2 data than for the LRS3-TED data. TED talks are a different genre to BBC TV programmes, and may contain more technical jargon. A language model trained on TED talk transcriptions would therefore be more suitable for predicting the LRS3-TED data and may give a greater improvement in WER.

In Section 5.1.1 there are a few examples listed of LRS3-TED transcriptions predicted by the model. The samples which the model predicted perfectly (0% WER) tended to be fairly short, although there were a few longer exceptions. It is known that the model does not generalise well to longer sentences than it has seen in training. This can be seen by a few long sentences (usually longer than any sentence in LRS2) whose prediction was only a few words long. These first few words were often correct or close to the actual transcription, and it seemed as if the sentence had been cut off short. However, this was not the case for all long sentences - there were also long samples which were predicted with a low WER.

5.2 Training an LRS3-TED lip reading model

The aim of this experiment was to fine-tune the LRS2 lip reading model to better predict LRS3-TED data, by training it further using the LRS3-TED training set. Different batch sizes and learning rates were investigated.

The experiments were run on a single GeForce GTX 1080 Ti GPU with 11178MiB memory. This larger memory compared to the GPU used in Section 5.1 allowed larger batch sizes and faster run time.

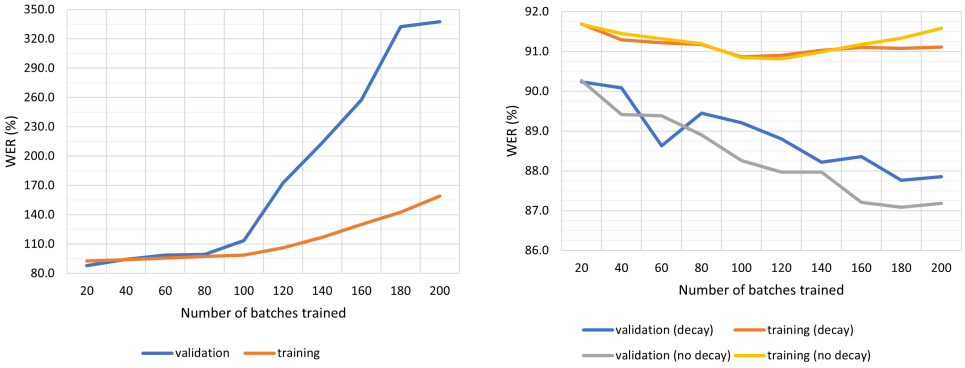
5.2.1 Results

Table 5.3 gives the CER and WER on the LRS3-TED test set for three models trained with different batch sizes and learning rates, as well as the time taken to train each model.

Figures 5.3a and 5.3b show the training and validation WER during training of models with the same batch size (50) and different learning rates. A model checkpoint was saved after every 20 batches of training, and the WER of the validation set was evaluated at each checkpoint. The values shown for the training WER are the cumulative WER on the training data used so far, measured every 20 batches.

Table 5.3: Training time for models trained with different batch sizes on the LRS3-TED, and their CER and WER on the LRS3-TED test dataset. The models all had initial learning rate 10^{-6}, the models trained with a decaying learning rate used inverse time decay with decay step 10 and decay rate 0.1.

Batch size	Decay (y/n)	Training time (hrs)	Model setup	CER (%)	WER (%)
8	y	15.7	no beam, no LM	61.5	88.7
			no LM	58.0	78.4
			LM	60.0	78.3
50	n	7.4	no beam, no LM	61.9	88.3
			no LM	59.2	78.7
			LM	60.6	78.6
50	y	7.6	no beam, no LM	60.8	87.8
			no LM	57.5	78.8
			LM	58.5	78.0



(a) Deteriorating model. Initial learning rate 10^{-5} and using inverse time decay with decay rate 0.1 and decay step 10.

(b) Successful models. Initial learning rate 10^{-6}, one using inverse time decay with decay rate 0.1 and decay step 10 and the other no decay.

Figure 5.3: Training and validation WER during one epoch of training for three different models, each using batch size 50 and different learning rates.

5.2.2 Discussion

The best performance achieved on the LRS3-TED test set in this project was a WER of 78.0%, which was achieved by the model trained using a batch size of 50, initial learning rate 10^{-6} and decaying the learning rate with inverse time decay (see Table 5.3). This was achieved when evaluating using beam search and the external language model. This is an absolute decrease of 2.7% compared to WER achieved in Section 5.1 by the original LRS2 lip reading model.

As mentioned in Section 4.2, the final learning rate used in [2] was 10^{-4}. I hence began my experiments using an initial learning rate of 10^{-4}, however, this caused the model's performance to deteriorate rapidly. After training a few batches the model started producing very odd results - a couple of examples are shown below. The predictions began to contain repeated words, often beginning with 'AN' - even if the true transcription was not even vaguely similar - including words that are not in the English language. Eventually the performance degraded even further, with the model producing long sequences of repeated characters. This correspondingly resulted in a large WER on the training data.

Deteriorating model predictions:

```
(wer=100.0) SO-THE-BEST-THING-THAT-HAPPENED-TO-US-SO-FAR-IN-THE-
MEDICAL-ARENA-IN-CANCER-RESEARCH-IS-THE-FACT-THAT-THE
--> AND-AN-AREA-AND-AN-AREA-ANITA-ANITA-ANNESO-ANTIONESEANITE

(wer=100.0) I-AM-CONVINCED-THAT-AFRICA'S-FURTHER-TRANSFORMATION-
AFRICA'S-ADVANCEMENT-RESTS
--> AN-ARTINITAN-ANNINA-ANTIOCOINEEEEEIANEEEEEEEEEEEEEEAIAIOIEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEIIAIEEEEEEEEEEEEEEEEEEE
```

Reducing the learning rate to 10^{-5} and to 10^{-6} still resulted in the model deteriorating, but at a later stage in training. Using an initial learning rate of 10^{-6} and decaying the learning rate throughout training fixed this, resulting in a successfully trained model. Figure 5.4 demonstrates how the learning rate changes over the course of 200 batches of training when using inverse time decay with the decay parameters used in these experiments. Although using this learning rate worked, it is a very small value so the weights do not change by very much with each update. It is possible that this is why the improvement in WER achieved by the model compared to the LRS2 model was quite small.

Figure 5.3a shows the training and validation WER for one model which deteriorated as explained above. One would expect to see the training WER decrease as training progresses, and would hope to see validation WER decreasing at the same time. Instead, the training and validation WER both increase as the training progresses - slowly at first, and then faster after about 100 batches. The validation WER increases to over 300%, much higher than the WERs achieved on the same dataset in Table 5.2, indicating that the training went badly wrong.

Figure 5.3b shows the training and validation WER for two successful models. The

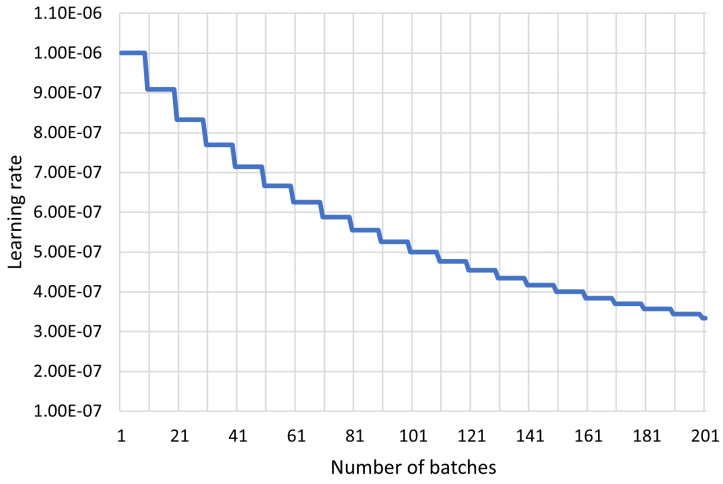


Figure 5.4: Learning rate value over 200 batches, using inverse time decay with decay rate 0.1, decaying every 10 batches.

validation WER shows a decreasing trend for both models, which indicates the training is progressing successfully. The training WER begins to increase slightly in the late stages of training for the model with no learning rate decay, which could be a sign that the model is on the verge of deteriorating like the model in Figure 5.3a. However, while the validation WER is calculated over a fixed set of samples at different stages of training, the training WER is measured on different sets of samples, since it is measured at different stages of one epoch of training. The training WER should therefore be used as more of a sanity check that the model is not deteriorating rapidly, rather than a trend that can be analysed in the same way as the validation WER. The model only sees each sample once during the training, and it is possible that the samples at the end of the training set are simply more challenging than the ones at the beginning, which would cause the cumulative WER to increase slightly near the end of training. If the model were to be trained for more than one epoch then the training WER at the end of each epoch could be measured and analysed more usefully.

For the successful models we can compare the WER and CER achieved and the time taken to train (see Table 5.3). Increasing the batch size from 8 to 50 led to a significant reduction in the time taken to train a model - the time was reduced by over half, from almost 16 hours to around 7 and a half hours - which is much more reasonable for running multiple training experiments. When the batch size is larger the weights are updated fewer times, which leads to the reduction in time. Increasing the batch size (while keeping the learning rate decay the same) resulted in a very slight decrease in CER and WER, with the exception of the WER for the model setup with beam search and no LM, which increased very slightly. Using a decaying learning rate decreased the CER and WER compared to using no decay. This can be seen for batch size 8, since when no decay was used, the model deteriorated as explained above (and hence is not shown in the table). For batch size 50, decaying the learning rate caused a slight decrease in CER and WER (except, again, for the model setup with beam search and no LM, which increased very slightly), which can be seen in the table.

Assuming that the training sets have a similar distribution of sentence lengths as the test sets (see Figure 5.2), then even though LRS3-TED does have sentences above 100 characters in length (longer than LRS2) the number of these sentences is very small compared to the number of shorter sentences. It is known that the lip reading model does not generalise well to longer examples than it has seen in training. I hoped that by training on LRS3-TED, the performance on longer sentences would have improved. However, the long sentences whose predictions were cut off (mentioned in Section 5.1.2), were still predicted similarly with the trained models. This may be because the number of very long sentences seen is so small compared to the number of short sentences, that predictions of long sentences are given a higher cost than shorter sentences.

The improvement in WER achieved by training the LRS2 lip reading model on LRS3-TED data was very small. This may be because it was only trained on a subset of the available LRS3-TED training data and only trained for one epoch. I chose to do this because the LRS2 lip reading model had already been trained for 12 epochs, and I thought that just a small amount of training would be enough to adapt the model to better predict the LRS3-TED data. Possible changes that could be made in the future to decrease the WER further would be to train on the full training set and to train for more than one epoch. It would also be worth experimenting further with changing the batch size, changing the initial learning rate and the learning rate decay parameters.

Chapter 6

Conclusions

6.1 Summary of results

In this project I evaluated how well a lip reading model generalised to an unseen dataset, and further fine-tuned the model to improve performance on that dataset.

As discussed in Section 5.1, the lip reading model pre-trained on the LRS2 dataset did not generalise well to the unseen LRS3-TED dataset. The lowest WER on the LRS2 test set was 48.8% while on LRS3-TED it was 80.7% - a considerable difference. Using the LRS2 language model during decoding was found to improve performance on both datasets, but had a greater impact on the LRS2 test data, reducing WER by an absolute value of 2.4% compared to 1.4% for the LRS3-TED data.

By training the lip reading model for one epoch on a 10,000 sample subset of the available LRS3-TED training data the WER on the LRS3-TED test set was reduced to 78.0% - an absolute decrease of 2.7%. As presented in Section 5.2, it was found that using an initial learning rate higher than 10^{-6} caused the model to deteriorate and produce strange predictions, meanwhile decaying the learning rate proved beneficial. Increasing the batch size from 8 to 50 had little impact on the WER, but roughly halved the training time and hence was useful for running experiments efficiently.

6.2 Future Work

Next year, in the second half of this MInf Project, I will continue to improve and work with the lip reading model from this year. I intend to perform the following further work:

- Train a language model on TED talk transcriptions (e.g. using TED-LIUM corpus [19])
- Test the performance of the lip reading model on the LRS3-TED dataset when using the TED language model, compared to using the LRS2 language model
- Link up to the video pre-processing pipeline created by another student to test

the whole process of:

- starting with full, uncropped videos of speakers
- producing a dataset of cropped videos
- training a lip reading model on the dataset
- evaluating the model on the dataset

In addition, I intend to bring the audio into this audio-visual speech recognition project, by combining the lip reading model with an ASR model.

Finally, I am intrigued by the possibility of working with datasets in other languages. The authors of LRS3-TED [3] have stated that they are creating an LRS3-Lang dataset containing 13 different languages. I will keep an eye out for progress on this dataset and consider whether it can be used in my project.

Bibliography

[1] T. Afouras. Deep lip reading. https://github.com/afourast/deep_lip_reading. Accessed 10-04-2021.

[2] T. Afouras, J. S. Chung, and A. Zisserman. Deep lip reading: a comparison of models and an online application. In *INTERSPEECH*, 2018.

[3] T. Afouras, J. S. Chung, and A. Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. In *arXiv preprint arXiv:1809.00496*, 2018.

[4] Triantafyllos Afouras, Joon Son Chung, A. Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[5] Najwa Alghamdi, Steve Maddock, Ricard Marxer, Jon Barker, and Guy J Brown. A corpus of audio-visual Lombard speech with frontal and profile views. *The Journal of the Acoustical Society of America*, 143(6):EL523–EL529, 2018.

[6] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. LipNet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*, 2016.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] C. Bregler and Y. Konig. "Eigenlips" for robust speech recognition. In *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume ii, pages II/669–II/672, 1994.

[9] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561. ACL, November 2016.

[10] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016.

[12] J. S. Chung and A. Zisserman. Lip reading in profile. In *British Machine Vision Conference*, 2017.

[13] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120 5 Pt 1:2421–4, 2006.

[14] J. Elman. Finding structure in time. *Cogn. Sci.*, 14:179–211, 1990.

[15] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Trans. Graph.*, 37(4), July 2018.

[16] A. Fernandez-Lopez and F. Sukno. Survey on automatic lip-reading in the era of deep learning. *Image Vis. Comput.*, 78:53–72, 2018.

[17] M. Gales and S. Young. The Application of Hidden Markov Models in Speech Recognition. *Found. Trends Signal Process.*, 1:195–304, 2007.

[18] Ahmad BA Hassanat. Visual speech recognition. *Speech and Language Technologies*, 1:279–303, 2011.

[19] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *International Conference on Speech and Computer*, pages 198–208. Springer, 2018.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Bowon Lee, Mark Hasegawa-Johnson, Camille Goudeseune, Suketu Kamdar, Sarah Borys, Ming Liu, and Thomas Huang. AVICAR: Audio-visual speech corpus in a car environment. In *INTERSPEECH-2004*, pages 2489–2492, 2004.

[23] E. Lombard. Le signe de l'élévation de la voix. *Annales des Maladies de L'Oreille et du Larynx*, 37:101–119, 1911.

[24] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[25] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[26] H. McGurk and J. Macdonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[27] Kieron Messer, Jiri Matas, Josef Kittler, Juergen Luettin, and Gilbert Maitre. XM2VTSDB: The extended M2VTS database. In *Second international conference on audio and video-based biometric person authentication*, volume 964, pages 965–966, 1999.

[28] A. Ortega, F. Sukno, E. Lleida, A. Frangi, A. Miguel, L. Buera, and E. Zacur. AV@CAR: A Spanish multichannel multimodal corpus for in-vehicle automatic audio-visual speech recognition. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.

[29] Eric David Petajan. *Automatic Lipreading to Enhance Speech Recognition (Speech Reading)*. PhD thesis, University of Illinois at Urbana-Champaign, USA, 1984.

[30] G. Potamianos, H. P. Graf, and E. Cosatto. An image transform approach for HMM based automatic lipreading. In *Proceedings 1998 International Conference on Image Processing.*, pages 173–177 vol.3, 1998.

[31] B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, et al. Large-scale visual speech recognition. *arXiv preprint arXiv:1807.05162*, 2018.

[32] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[33] Satoshi Tamura, Chiyomi Miyajima, Norihide Kitaoka, Takeshi Yamada, Satoru Tsuge, Tetsuya Takiguchi, Kazumasa Yamamoto, Takanobu Nishiura, Masato Nakayama, Yuki Denda, et al. CENSREC-1-AV: An audio-visual corpus for noisy bimodal speech recognition. In *Auditory-Visual Speech Processing 2010*, 2010.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.