Lip Reading Sentences in the Wild

Joon Son Chung[1] joon@robots.ox.ac.uk
Andrew Senior[2] andrewsenior@google.com
Oriol Vinyals[2] vinyals@google.com
Andrew Zisserman[1,2] az@robots.ox.ac.uk

[1] Department of Engineering Science, University of Oxford [2] DeepMind

Abstract

The goal of this work is to recognise phrases and sentences being spoken by a talking face, with or without the audio. Unlike previous works that have focussed on recognising a limited number of words or phrases, we tackle lip reading as an open-world *problem – unconstrained natural language sentences, and in the wild videos.*

Our key contributions are: (1) a 'Watch, Listen, Attend and Spell' (WLAS) network that learns to transcribe videos of mouth motion to characters; (2) a curriculum learning strategy to accelerate training and to reduce overfitting; (3) a 'Lip Reading Sentences' (LRS) dataset for visual speech recognition, consisting of over 100,000 natural sentences from British television.

The WLAS model trained on the LRS dataset surpasses the performance of all previous work on standard lip reading benchmark datasets, often by a significant margin. This lip reading performance beats a professional lip reader on videos from BBC television, and we also demonstrate that if audio is available, then visual information helps to improve speech recognition performance.

1. Introduction

Lip reading, the ability to recognize what is being said from visual information alone, is an impressive skill, and very challenging for a novice. It is inherently ambiguous at the word level due to homophemes – different characters that produce exactly the same lip sequence (*e.g.* 'p' and 'b'). However, such ambiguities can be resolved to an extent using the context of neighboring words in a sentence, and/or a language model.

A machine that can lip read opens up a host of applications: 'dictating' instructions or messages to a phone in a noisy environment; transcribing and re-dubbing archival silent films; resolving multi-talker simultaneous speech; and, improving the performance of automated speech recogition in general.

That such automation is now possible is due to two developments that are well known across computer vision tasks: the use of deep neural network models [22, 33, 35]; and, the availability of a large scale dataset for training [31]. In this case the model is based on the recent sequence-to-sequence (encoder-decoder with attention) translater architectures that have been developed for speech recognition and machine translation [3, 5, 15, 16, 34]. The dataset developed in this paper is based on thousands of hours of BBC television broadcasts that have talking faces together with subtitles of what is being said.

We also investigate how lip reading can contribute to *audio* based speech recognition. There is a large literature on this contribution, particularly in noisy environments, as well as the converse where some derived measure of audio can contribute to lip reading for the deaf or hard of hearing. To investigate this aspect we train a model to recognize characters from both audio and visual input, and then systematically disturb the audio channel or remove the visual channel.

Our model (Section 2) outputs at the character level, is able to learn a language model, and has a novel dual attention mechanism that can operate over visual input only, audio input only, or both. We show (Section 3) that training can be accelerated by a form of curriculum learning. We also describe (Section 4) the generation and statistics of a new large scale Lip Reading Sentences (LRS) dataset, based on BBC broadcasts containing talking faces together with subtitles of what is said. The broadcasts contain faces 'in the wild' with a significant variety of pose, expressions, lighting, backgrounds, and ethnic origin.

The performance of the model is assessed on a test set of the LRS dataset, as well as on public benchmarks datasets for lip reading including LRW [9] and GRID [11]. We demonstrate *open world* (unconstrained sentences) lip reading on the LRS dataset, and in all cases on public benchmarks the performance exceeds that of prior work.

1.1. Related works

Lip reading. There is a large body of work on lip reading using pre-deep learning methods. These methods are thoroughly reviewed in [40], and we will not repeat this here. A number of papers have used Convolutional Neural Net-

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.367

works (CNNs) to predict phonemes [27] or visemes [21] from still images, as opposed recognising to full words or sentences. A *phoneme* is the smallest distinguishable unit of sound that collectively make up a spoken word; a *viseme* is its visual equivalent.

For recognising full words, Petridis *et al.* [30] trains an LSTM classifier on a discrete cosine transform (DCT) and deep bottleneck features (DBF). Similarly, Wand *et al.* [38] uses an LSTM with HOG input features to recognise short phrases. The shortage of training data in lip reading presumably contributes to the continued use of shallow features. Existing datasets consist of videos with only a small number of subjects, and also a very limited vocabulary (<60 words), which is also an obstacle to progress. The recent paper of Chung and Zisserman [9] tackles the small-lexicon problem by using faces in television broadcasts to assemble a dataset for 500 words. However, as with any word-level classification task, the setting is still distant from the real-world, given that the word boundaries must be known beforehand. A very recent work [2] uses a CNN and LSTM-based network and Connectionist Temporal Classification (CTC) [15] to compute the labelling. This reports strong speaker-independent performance on the constrained grammar and 51 word vocabulary of the GRID dataset [11]. However, the method, suitably modified, should be applicable to longer, more general sentences.

Audio-visual speech recognition. The problems of audio-visual speech recognition (AVSR) and lip reading are closely linked. Mroueh *et al.* [26] employs feed-forward Deep Neural Networks (DNNs) to perform phoneme classification using a large non-public audio-visual dataset. The use of HMMs together with hand-crafted or pre-trained visual features have proved popular – [36] encodes input images using DBF; [14] used DCT; and [28] uses a CNN pre-trained to classify phonemes; all three combine these features with HMMs to classify spoken digits or isolated words. As with lip reading, there has been little attempt to develop AVSR systems that generalise to real-world settings.

Speech recognition. There is a wealth of literature on speech recognition systems that utilise separate components for acoustic and language-modelling functions (*e.g.* hybrid DNN-HMM systems), that we will not review here. We restrict this review to methods that can be trained end-to-end.

For the most part, prior work can be divided into two types. The first type uses CTC [15], where the model typically predicts framewise labels and then looks for the optimal alignment between the framewise predictions and the output sequence. The weakness is that the output labels are not conditioned on each other.

The second type is sequence-to-sequence models [34] that first read all of the input sequence before starting to predict the output sentence. A number of papers have adopted this approach for speech recognition [7, 8], and the most related work to ours is that of Chan *et al.* [5] which proposes an elegant sequence-to-sequence method to transcribe audio signal to characters. They utilise a number of the latest sequence learning tricks such as scheduled sampling [4] and attention [8]; we take many inspirations from this work.

2. Architecture

In this section, we describe the *Watch, Listen, Attend and Spell* network that learns to predict characters in sentences being spoken from a video of a talking face, with or without audio.

We model each character y_i in the output character sequence $\mathbf{y} = (y_1, y_2, ..., y_l)$ as a conditional distribution of the previous characters $y_{<i}$, the input image sequence $\mathbf{x}^v = (x_1^v, x_2^v, ..., x_n^v)$ for lip reading, and the input audio sequence $\mathbf{x}^a = (x_1^a, x_2^a, ..., x_m^a)$. Hence, we model the output probability distribution as:

$$P(\mathbf{y}|\mathbf{x}^v, \mathbf{x}^a) = \prod_i P(y_i|\mathbf{x}^v, \mathbf{x}^a, y_{<i}) \tag{1}$$

Our model, which is summarised in Figure 1, consists of three key components: the image encoder `Watch` (Section 2.1), the audio encoder `Listen` (Section 2.2), and the character decoder `Spell` (Section 2.3). Each encoder transforms the respective input sequence into a fixed-dimensional state vector s, and sequences of encoder outputs $\mathbf{o} = (o_1, ..., o_p)$, $p \in (n, m)$; the decoder ingests the state and the attention vectors from both encoders and produces a probability distribution over the output character sequence.

$$s^v, \mathbf{o}^v = \texttt{Watch}(\mathbf{x}^v) \tag{2}$$
$$s^a, \mathbf{o}^a = \texttt{Listen}(\mathbf{x}^a) \tag{3}$$
$$P(\mathbf{y}|\mathbf{x}^v, \mathbf{x}^a) = \texttt{Spell}(s^v, s^a, \mathbf{o}^v, \mathbf{o}^a) \tag{4}$$

The three modules in the model are trained jointly. We describe the modules next, with implementation details given in Section 3.5.

2.1. Watch: Image encoder

The image encoder consists of the convolutional module that generates image features f_i^v for every input timestep x_i^v, and the recurrent module that produces the fixed-dimensional state vector s^v and a set of output vectors $\mathbf{o}^v$.

$$f_i^v = \texttt{CNN}(x_i^v) \tag{5}$$
$$h_i^v, o_i^v = \texttt{LSTM}(f_i^v, h_{i+1}^v) \tag{6}$$
$$s^v = h_1^v \tag{7}$$

The convolutional network is based on the VGG-M model [6], as it is memory-efficient, fast to train and has a decent classification performance on ImageNet [31]. The

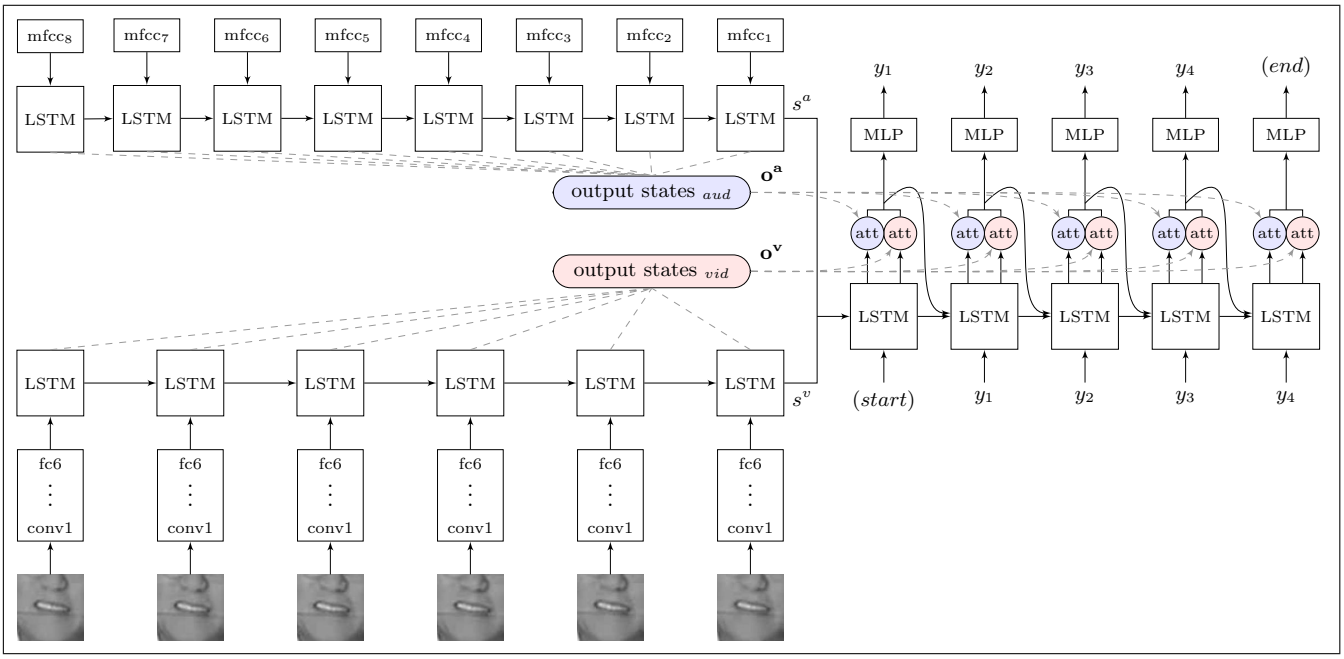

Figure 1. *Watch, Listen, Attend and Spell* architecture. At each time step, the decoder outputs a character y_i, as well as two attention vectors. The attention vectors are used to select the appropriate period of the input visual and audio sequences.

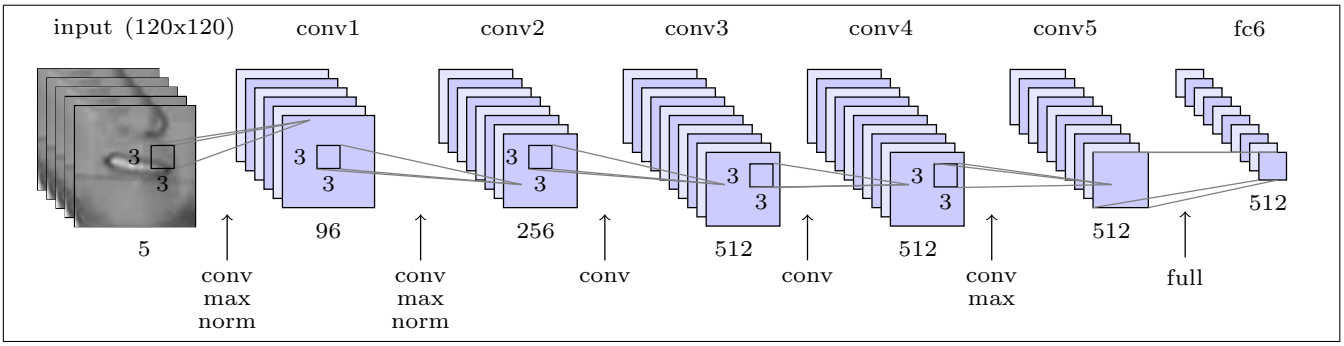

Figure 2. The ConvNet architecture. The input is five gray level frames centered on the mouth region. The 512-dimensional fc6 vector forms the input to the LSTM.

ConvNet layer configuration is shown in Figure 2, and is abbreviated as *conv1* $\cdots$ *fc6* in the main network diagram.

The encoder LSTM network consumes the output features f_i^v produced by the ConvNet at every input timestep, and generates a fixed-dimensional state vector s^v. In addition, it produces an output vector o_i^v at every timestep i. Note that the network ingests the inputs in reverse time order (as in Equation 6), which has shown to improve results in [34].

2.2. Listen: Audio encoder

The `Listen` module is an LSTM encoder similar to the `Watch` module, without the convolutional part. The LSTM directly ingests 13-dimensional MFCC features in reverse time order, and produces the state vector s^a and the output vectors $\mathbf{o}^a$.

$$h_j^a, o_j^a = \texttt{LSTM}(x_j^a, h_{j+1}^a) \tag{8}$$

$$s^a = h_1^a \tag{9}$$

2.3. Spell: Character decoder

The `Spell` module is based on a LSTM transducer [3, 5, 8], here we add a dual attention mechanism. At every output step k, the decoder LSTM produces the decoder states h_k^d and output vectors o_k^d from the previous step context vectors c_{k-1}^v and c_{k-1}^a, output y_{k-1} and decoder state h_{k-1}^d. The attention vectors are generated from the attention mechanisms $\texttt{Attention}^{\texttt{v}}$ and $\texttt{Attention}^{\texttt{a}}$. The inner working of the attention mechanisms is described in [3], and repeated in the supplementary material. We use two independent attention mechanisms for the lip and the audio input streams

to refer to the asynchronous inputs with different sampling rates. The attention vectors are fused with the output states (Equations 11 and 12) to produce the context vectors c_k^v and c_k^a that encapsulate the information required to produce the next step output. The probability distribution of the output character is generated by an MLP with softmax over the output.

$$h_k^d, o_k^d = \texttt{LSTM}(h_{k-1}^d, y_{k-1}, c_{k-1}^v, c_{k-1}^a) \quad (10)$$

$$c_k^v = \mathbf{o}^v \cdot \texttt{Attention}^{\texttt{v}}(h_k^d, \mathbf{o}^v) \quad (11)$$

$$c_k^a = \mathbf{o}^a \cdot \texttt{Attention}^{\texttt{a}}(h_k^d, \mathbf{o}^a) \quad (12)$$

$$P(y_i|\mathbf{x}^v, \mathbf{x}^a, y_{<i}) = \texttt{softmax}(\texttt{MLP}(o_k^d, c_k^v, c_k^a)) \quad (13)$$

At $k = 1$, the final encoder states s_l and s_a are used as the input instead of the previous decoder state – *i.e.* $h_0^d = \texttt{concat}(s^a, s^v)$ – to help produce the context vectors c_1^v and c_1^a in the absence of the previous state or context.

Discussion. In our experiments, we have observed that the attention mechanism is absolutely critical for the audio-visual speech recognition system to work. Without attention, the model appears to 'forget' the input signal, and produces an output sequence that correlates very little to the input, beyond the first word or two (which the model gets correct, as these are the last words to be seen by the encoder). The attention-less model yields Word Error Rates over 100%, so we do not report these results.

The dual-attention mechanism allows the model to extract information from both audio and video inputs, even when one stream is absent, or the two streams are not time-aligned. The benefits are clear in the experiments with noisy or no audio (Section 5).

Bidirectional LSTMs have been used in many sequence learning tasks [5, 8, 17] for their ability to produce outputs conditioned on future context as well as past context. We have tried replacing the unidirectional encoders in the `Watch` and `Listen` modules with bidirectional encoders, however these networks took significantly longer to train, whilst providing no obvious performance improvement. This is presumably because the `Decoder` module is anyway conditioned on the full input sequence, so bidirectional encoders are not necessary for providing context, and the attention mechanism suffices to provide the additional local focus.

3. Training strategy

In this section, we describe the strategy used to effectively train the *Watch, Listen, Attend and Spell* network, making best use of the limited amount of data available.

3.1. Curriculum learning

Our baseline strategy is to train the model from scratch, using the full sentences from the 'Lip Reading Sentences' dataset – previous works in speech recognition have taken this approach. However, as [5] reports, the LSTM network converges very slowly when the number of timesteps is large, because the decoder initially has a hard time extracting the relevant information from all the input steps.

We introduce a new strategy where we start training only on single word examples, and then let the sequence length grow as the network trains. These short sequences are parts of the longer sentences in the dataset. We observe that the rate of convergence on the training set is several times faster, and it also significantly reduces overfitting, presumably because it works as a natural way of augmenting the data. The test performance improves by a large margin, reported in Section 5.

3.2. Scheduled sampling

When training a recurrent neural network, one typically uses the previous time step ground truth as the next time step input, which helps the model learn a kind of language model over target tokens. However during inference, the previous step ground-truth is unavailable, resulting in poorer performance because the model was not trained to be tolerant to feeding in bad predictions at some time steps. We use the scheduled sampling method of Bengio *et al.* [4] to bridge this discrepancy between how the model is used at training and inference. At train time, we randomly sample from the previous output, instead of always using the ground-truth. When training on shorter sub-sequences, ground-truth previous characters are used. When training on full sentences, the sampling probability from the previous output was increased in steps from 0 to 0.25 over time. We were not able to achieve stable learning at sampling probabilities of greater than 0.25.

3.3. Multi-modal training

Networks with multi-modal inputs can often be dominated by one of the modes [13]. In our case we observe that the audio signal dominates, because speech recognition is a significantly easier problem than lip reading. To help prevent this from happening, one of the following input types is uniformly selected at train time for each example: (1) audio only; (2) lips only; (3) audio and lips.

If mode (1) is selected, the audio-only data described in Section 4.1 is used. Otherwise, the standard audio-visual data is used.

We have over 300,000 sentences in the recorded data, but only around 100,000 have corresponding facetracks. In machine translation, it has been shown that monolingual dummy data can be used to help improve the performance of a translation model [32]. By similar rationale, we use the sentences without facetracks as supplementary training data to boost audio recognition performance and to build a richer language model to help improve generalisation.

3.4. Training with noisy audio

The WLAS model is initially trained with clean input audio for faster convergence. To improve the model's tolerance to audio noise, we apply additive white Gaussian noise with SNR of 10dB (10:1 ratio of the signal power to the noise power) and 0dB (1:1 ratio) later in training.

3.5. Implementation details

The input images are 120×120 in dimension, and are sampled at 25Hz. The image only covers the lip region of the face, as shown in Figure 3. The ConvNet ingests 5-frame sliding windows using the Early Fusion method of [9], moving 1-frame at a time. The MFCC features are calculated over 25ms windows and at 100Hz, with a time-stride of 1. For `Watch` and `Listen` modules, we use a three layer LSTM with cell size of 256. For the `Spell` module, we use a three layer LSTM with cell size of 512. The output size of the network is 45, for every character in the alphabet, numbers, common punctuations, and tokens for `[sos]`, `[eos]`, `[pad]`. The full list is given in the supplementary material.

Our implementation is based on the TensorFlow library [1] and trained on a GeForce Titan X GPU with 12GB memory. The network is trained using stochastic gradient descent with a batch size of 64 and with dropout and label smoothing. The layer weights of the convolutional layers are initialised from the visual stream of [10]. All other weights are randomly initialised.

An initial learning rate of 0.1 was used, and decreased by 10% every time the training error did not improve for 2,000 iterations. Training on the full sentence data was stopped when the validation error did not improve for 5,000 iterations. The model was trained for around 500,000 iterations, which took approximately 10 days.

4. Dataset

In this section, we describe the multi-stage pipeline for automatically generating a large-scale dataset for audio-visual speech recognition. Using this pipeline, we have been able to collect thousands of hours of spoken sentences and phrases along with the corresponding facetrack. We use a variety of BBC programs recorded between 2010 and 2016, listed in Table 1, and shown in Figure 3.

The selection of programs are deliberately similar to those used by [9] for two reasons: (1) a wide range of speakers appear in the news and the debate programs, unlike dramas with a fixed cast; (2) shot changes are less frequent, therefore there are more full sentences with continuous facetracks.

The processing pipeline is summarised in Figure 4. Most of the steps are based on the methods described in [9] and [10], but we give a brief sketch of the method here.

Video preparation. First, shot boundaries are detected by comparing colour histograms across consecutive frames [24]. The HOG-based face detection [20] is then performed on every frame of the video. The face detections of the same person are grouped across frames using a KLT tracker [37]. Facial landmarks are extracted from a sparse subset of pixel intensities using an ensemble of regression trees [19].

Channel	Series name	# hours	# sent.
BBC 1 HD	News†	1,584	50,493
BBC 1 HD	Breakfast	1,997	29,862
BBC 1 HD	Newsnight	590	17,004
BBC 2 HD	World News	194	3,504
BBC 2 HD	Question Time	323	11,695
BBC 4 HD	World Today	272	5,558
All		4,960	118,116

Table 1. Video statistics. The number of hours of the original BBC video; the number of sentences with full facetrack. †BBC News at 1, 6 and 10.

Audio and text preparation. The subtitles in BBC videos are not broadcast in sync with the audio. The Penn Phonetics Lab Forced Aligner [18, 39] is used to force-align the subtitle to the audio signal. Errors exist in the alignment as the transcript is not verbatim – therefore the aligned labels are filtered by checking against the commercial IBM Watson Speech to Text service.

AV sync and speaker detection. In BBC videos, the audio and the video streams can be out of sync by up to around one second, which can cause problems when the facetrack corresponding to a sentence is being extracted. The two-stream network described in [10] is used to synchronise the two streams. The same network is also used to determine who is speaking in the video, and reject the clip if it is a voice-over.

Sentence extraction. The videos are divided into invididual sentences/ phrases using the punctuations in the transcript. The sentences are separated by full stops, commas and question marks; and are clipped to 100 characters or 10 seconds, due to GPU memory constraints. We do not impose any restrictions on the vocabulary size.

The training, validation and test sets are divided according to broadcast date, and the dates of videos corresponding to each set are shown in Table 2. The dataset contains thousands of different speakers which enables the model to be speaker agnostic.

Table 3 compares the *'Lip Reading Sentences'* (LRS) dataset to the largest existing public datasets.

4.1. Audio-only data

In addition to the audio-visual dataset, we prepare an auxiliary audio-only *training* dataset. These are the sentences in the BBC programs for which facetracks are not available. The use of this data is described in Section 3.3. It is only used for training, not for testing.

Figure 3. **Top:** Original still images from the BBC lip reading dataset – News, Question Time, Breakfast, Newsnight (from left to right). **Bottom:** The mouth motions for 'afternoon' from two different speakers. The network sees the areas inside the red squares.

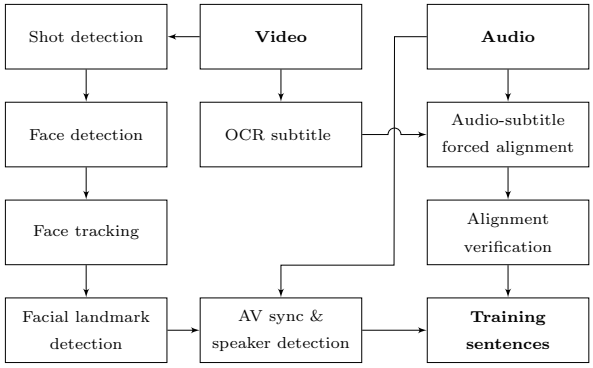

Figure 4. Pipeline to generate the dataset.

Set	Dates	# Utter.	Vocab
Train	01/2010 - 12/2015	101,195	16,501
Val	01/2016 - 02/2016	5,138	4,572
Test	03/2016 - 09/2016	11,783	6,882
All		118,116	17,428

Table 2. **The Lip Reading Sentences (LRS) audio-visual dataset.** Division of training, validation and test data; and the number of utterances and vocabulary size of each partition. Of the 6,882 words in the test set, 6,253 are in the training or the validation sets; 6,641 are in the audio-only training data. **Utter:** Utterances

Name	Type	Vocab	# Utter.	# Words
GRID [11]	Sent.	51	33,000	165,000
MODALITY [12]	Sent.	182	5,880	8,085
LRW [9]	Words	500	450,000	450,000
LRS	Sent.	17,428	118,116	807,375

Table 3. Comparison to existing large-scale lip reading datasets.

Set	Dates	# Utter.	Vocab
Train	01/2010 - 12/2015	342,644	25,684

Table 4. Statistics of the Audio-only training set.

5. Experiments

In this section we evaluate and compare the proposed architecture and training strategies. We also compare our method to the state of the art on public benchmark datasets.

To clarify which of the modalities are being used, we call the models in lips-only and audio-only experiments *Watch, Attend and Spell* (WAS), *Listen, Attend and Spell* (LAS) respectively. These are the same *Watch, Listen, Attend and Spell* model with either of the inputs disconnected and replaced with all-zeros.

5.1. Evaluation.

The models are trained on the LRS dataset (the train/val partition) and the Audio-only training dataset (Section 4). The inference and evaluation procedures are described below.

Beam search. Decoding is performed with beam search of width 4, in a similar manner to [5, 34]. At each timestep, the hypotheses in the beam are expanded with every possible character, and only the 4 most probable hypotheses are stored. Figure 5 shows the effect of increasing the beam width – there is no observed benefit for increasing the width beyond 4.

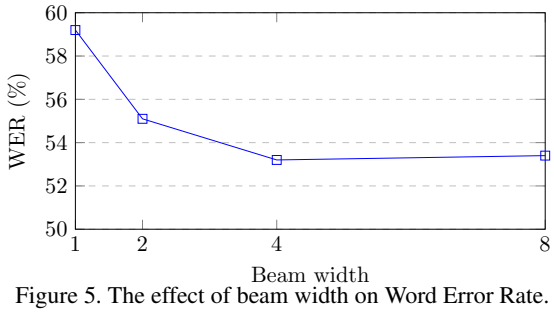

Figure 5. The effect of beam width on Word Error Rate.

Evaluation protocol. The models are evaluated on an independent test set (Section 4). For all experiments, we report the Character Error Rate (CER), the Word Error Rate (WER) and the BLEU metric. CER and WER are defined as $\texttt{ErrorRate} = (S+D+I)/N$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions to get from the reference to the hypothesis, and N is the number of words in the reference. BLEU [29] is a modified form of n-gram precision to compare a candidate sentence to one or more reference sentences. Here, we use

the unigram BLEU.

Method	SNR	CER	WER	BLEU†
Lips only				
Professional‡	-	58.7%	73.8%	23.8
WAS	-	59.9%	76.5%	35.6
WAS+CL	-	47.1%	61.1%	46.9
WAS+CL+SS	-	42.4%	58.1%	50.0
WAS+CL+SS+BS	-	39.5%	50.2%	54.9
Audio only				
Google Speech API	clean	17.6%	22.6%	78.4
Kaldi SGMM+MMI⋆	clean	9.7%	16.8%	83.6
LAS+CL+SS+BS	clean	10.4%	17.7%	84.0
LAS+CL+SS+BS	10dB	26.2%	37.6%	66.4
LAS+CL+SS+BS	0dB	50.3%	62.9%	44.6
Audio and lips				
WLAS+CL+SS+BS	clean	7.9%	13.9%	87.4
WLAS+CL+SS+BS	10dB	17.6%	27.6%	75.3
WLAS+CL+SS+BS	0dB	29.8%	42.0%	63.1

Table 5. Performance on the LRS test set. **WAS**: *Watch, Attend and Spell*; **LAS**: *Listen, Attend and Spell*; **WLAS**: *Watch, Listen, Attend and Spell*; **CL**: Curriculum Learning; **SS**: Scheduled Sampling; **BS**: Beam Search. †Unigram BLEU with brevity penalty. ‡Excluding samples that the lip reader declined to annotate. Including these, the CER rises to 78.9% and the WER to 87.6%. ⋆ The Kaldi SGMM+MMI model used here achieves a WER of 3.6% on the WSJ (eval92) test set, which is within 0.2% of the current state-of-the-art. The acoustic and language models have been re-trained on our dataset.

Results. All of the training methods discussed in Section 3 contribute to improving the performance. A breakdown of this is given in Table 5 for the lips-only experiment. For all other experiments, we only report results obtained using the best strategy.

Lips-only examples. The model learns to correctly predict extremely complex unseen sentences from a wide range of content – examples are shown in Table 6.

MANY MORE PEOPLE WHO WERE INVOLVED IN THE ATTACKS
CLOSE TO THE EUROPEAN COMMISSION'S MAIN BUILDING
WEST WALES AND THE SOUTH WEST AS WELL AS WESTERN SCOTLAND
WE KNOW THERE WILL BE HUNDREDS OF JOURNALISTS HERE AS WELL
ACCORDING TO PROVISIONAL FIGURES FROM THE ELECTORAL COMMISSION
THAT'S THE LOWEST FIGURE FOR EIGHT YEARS
MANCHESTER FOOTBALL CORRESPONDENT FOR THE DAILY MIRROR
LAYING THE GROUNDS FOR A POSSIBLE SECOND REFERENDUM
ACCORDING TO THE LATEST FIGURES FROM THE OFFICE FOR NATIONAL STATISTICS
IT COMES AFTER A DAMNING REPORT BY THE HEALTH WATCHDOG

Table 6. Examples of unseen sentences that WAS correctly predicts (lips only).

Audio-visual examples. As we hypothesised, the results in Table 5 demonstrate that the mouth movements provide important cues in speech recognition when the audio signal is noisy; and also give an improvement in performance even when the audio signal is clean – the character error rate is reduced from 16.2% for audio only to 13.3% for audio together lip reading. Table 7 shows some of the many examples where the WLAS model fails to predict the correct sentence from the lips or the audio alone, but successfully deciphers the words when both streams are present.

GT	IT WILL BE THE CONSUMERS
A	IN WILL BE THE CONSUMERS
L	IT WILL BE IN THE CONSUMERS
AV	IT WILL BE THE CONSUMERS
GT	CHILDREN IN EDINBURGH
A	CHILDREN AND EDINBURGH
L	CHILDREN AND HANDED BROKE
AV	CHILDREN IN EDINBURGH
GT	JUSTICE AND EVERYTHING ELSE
A	JUST GETTING EVERYTHING ELSE
L	CHINESES AND EVERYTHING ELSE
AV	JUSTICE AND EVERYTHING ELSE

Table 7. Examples of AVSR results. **GT:** Ground Truth; **A:** Audio only (10dB SNR); **L:** Lips only; **AV:** Audio-visual.

Attention visualisation. The attention mechanism generates explicit alignment between the input video frames (or the audio signal) and the hypothesised character output. Figure 6