Visual Units and Confusion Modelling for Automatic Lip-reading

Dominic Howell, Stephen Cox and Barry Theobald

School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK.

Abstract

Automatic lip-reading (ALR) is a challenging task because the visual speech signal is known to be missing some important information, such as voicing. We propose an approach to ALR that acknowledges that this information is missing but assumes that it is substituted or deleted in a systematic way that can be modelled. We describe a system that learns such a model and then incorporates it into decoding, which is realised as a cascade of weighted finite-state transducers. Our results show a small but statistically significant improvement in recognition accuracy. We also investigate the issue of suitable visual units for ALR, and show that visemes are sub-optimal, not because they introduce lexical ambiguity, but because the reduction in modelling units entailed by their use reduces accuracy.

Keywords: Lip-reading, Speech recognition, Visemes, Weighted finite state transducers, Confusion matrices, Confusion modelling

1. Introduction

In the past thirty years, the development of automatic speech recognition (ASR) has received enormous attention to the point where ASR is now a useful and reliable technology. By contrast, automatic lip-reading (ALR) has received very little attention. This is not surprising, since lip-reading is used by only a very small proportion of the population who have hearing difficulties, and although some of these users can apparently lip-read with high accuracy, it is an imperfect form of communication. Audio-visual speech recognition (AVSR) is now gaining in importance as attention turns towards making ASR more robust to interfering noise. A number of different techniques have been proposed for AVSR, but all of them would benefit from higher accuracy when decoding speech purely from a visual signal. Although this is the most significant motivation for researching ALR, it also has a number of possible applications in its own right in areas such as provision of automatic training systems for teaching lip-reading, as an aid for people who are able to make speech gestures but whose voice function has been removed, and in fighting crime, as well as being an interesting topic in speech communication.

Speech is primarily an audio form of communication, and a considerable amount of information about speech sounds is missing from the visual speech signal (47). The approach taken in this paper is to acknowledge that errors will occur in ALR because of this missing information, and to model and compensate for them, an approach which was inspired by previous work on dysarthric speech (43). Dysarthric speakers have poor control over their articulators because of medical conditions (such as cerebral palsy, stroke, brain tumour etc.) that affect their motor functions. This leads to a reduced phonemic repertoire and poor quality articulation, and hence to speech that has low intelligibility and is difficult for ASR systems to recognise. Similarly, in visual speech, certain speech sounds cannot be distinguished because they differ in a feature that is not present in the visual signal (e.g. voicing, place of articulation when it is in the rear of the vocal tract). In previous work on dysarthric speech recognition, patterns of phonemic confusions made by a talker were learnt by the system, and when these confusions were compensated at recognition time, recognition accuracy increased (43). In this work, we take a similar approach to lip-reading: we model visual speech as if it were a speech signal produced by a speaker who has a limited phonemic repertoire, and learn the resulting patterns of phoneme confusion by comparing the ground-truth phoneme sequences with the recognised sequences. At recognition time, we find the most likely interpretation (word-sequence) of the distorted phoneme output sequence in the light of these patterns. The approach is conveniently realised as a cascade of weighted finite-state transducers (WFSTs), one of which implements the confusion modelling, whilst the others implement familiar speech recognition tasks such as a pronunciation dictionary and language modelling. We compare this approach with the standard speech recognition approach in which no knowledge of confusions is used.

Until recently, the ALR community has concentrated (with a few exceptions) on small and restricted lip-reading tasks, usually isolated letters and/or digits, as this kind

Email address: `Dominic.Howell@uea.ac.uk`, `s.j.cox@uea.ac.uk`, `B.Theobald@uea.ac.uk` (Dominic Howell, Stephen Cox and Barry Theobald)

of task is appropriate in the initial stages of developing a technology. Here, we report ALR results on continuous speech utterances that have a medium-size (∼1000 words) vocabulary. We use a specially-recorded dataset consisting of videos of 3000 sentences spoken by a single speaker.

This unusually large corpus enables us to investigate a fundamental question in ALR, which is whether the use of phoneme-to-viseme mappings is effective. Visemes (discussed more thoroughly in section 4) are claimed to be the visual equivalent of phonemes i.e. they are units of visual speech. It is common practice to employ a phoneme-to-viseme mapping (several are available) in ALR on the grounds that there are many phonemes that cannot be distinguished visually, and indistinguishable phonemes should logically be grouped together as a single unit for purposes of recognition. Although there has been some work on testing these mappings (8; 1), it is not conclusive, and we investigate this in the first part of this paper.

The paper is organised as follows: in Section two, we set the scene for our work by reviewing the state-of-the-art in ALR. Section three describes the two databases that we recorded for these experiments, and section four describes our work in exploring the mapping between phonemes to visemes. Section five gives a brief background to WFSTs and describes our new approach in detail. Results based on the two databases used are described in sections six and seven respectively. We conclude with a discussion in section eight.

2. Previous work

The first attempts to automatically recognise speech from a visual signal date back to the 1980s and the work of Petajan (50; 51). Even from that date, the focus was on using the visual signal to enhance audio ASR, and most work since then has concentrated on such integration rather than lip-reading *per se*. However, this work was important in laying the foundations for techniques of deriving features suitable for speech recognition from visual images. These early systems tended to use very small vocabularies, such as a subset of the alphabet or the ten digits, uttered by a single speaker (57; 59), and used classification techniques such as hidden Markov models (5), neural networks (13) or hybrid models (4; 3). Work on continuous speech began about 2000 with continuously spoken digits (14). A summer workshop at Johns Hopkins in 2000 (45) enabled major advances in AVSR by recording a very large database of 290 speakers speaking material with a vocabulary of 10500 words (unfortunately it is unavailable). It pioneered the use of active appearance models (AAMs, (10)) as visual features and produced some of the first sets of speaker-independent ALR and AVSR results. Since then, there have been many different approaches to AVSR (52) including coupled HMMs (44), dynamic Bayesian networks (20), use of articulatory-based features (32), segment-based approaches (23; 22) and more recently, deep neural networks (48; 26). A recent review of AVSR research that considers especially the selection of visual features for visual speech is (63).

Work in ALR itself has grown significantly in the last ten years, although many authors use the term “lip reading” to describe work in AVSR rather than ALR. The work has covered essentially three areas: development of new visual features (30; 56; 31; 34), research into suitable units for lip reading (62; 25; 7) and exploration of new classification techniques (54; 56; 49). Much of this work still uses small datasets of isolated words from a single speaker but a recent paper (58) presents speaker-independent results on a 1000 word connected speech task.

3. Data and Visual Features

We recorded two datasets for the experiments in this work. A single speaker was recorded in each to eliminate the variation in visual features between speakers. We consider that this is a good strategy when exploring an innovative technique such as the one proposed here. In other recent work using multiple speakers from the large LiLiR dataset (29), we have shown how to compensate (to some extent) for speaker variation by using techniques such as speaker adaptive training and deep neural networks, and these techniques can be added later to the work described here.

The first dataset, called ISO-211, was an audio-visual database of 211 isolated words. It was designed for rapid experimentation in developing WFSTs for lip-reading. ISO-211 has a vocabulary of 211 phonetically rich words which were chosen to give maximum bigram coverage. The data were captured in a specialised recording environment using a Sanyo Xacti camera in portrait orientation at 1080 x 1920 pixel resolution using progressive scan at a sampling frequency of 59.94 frames per second. Audio was captured using a clip microphone at a sampling frequency of 48 kHz. A single native English speaking female speaker spoke six repetitions of each word.

The second dataset, called RM-3000, consists of audio-visual recordings of 3000 sentences spoken by a single native English-speaking male speaker. The sentences were randomly selected from the 8000 sentences in the Resource Management (RM) Corpus (53). The motivation for recording RM-3000 was to obtain a large database of continuous visual speech that had a medium size vocabulary and that was spoken by a single speaker. Sentences from the RM Corpus were chosen because its format (sentences of varying length whose grammar can be well-modelled with a language model) and its vocabulary size (1000 words) are ideal for research into lip-reading in its current state of development. The recording setup was the same as for the ISO-211 dataset.

Phoneme transcriptions of the sentences were derived from the BEEP Dictionary (6). Some statistics about the two databases are shown in Table 1.

	ISO-211	RM-3000
Total number of sentences	-	3000
Total Number of unique words	211	979
Total number of unique phonemes	45	45
Total number of word tokens	1255	26114
Total Number of phoneme tokens	7040	105561
Average number of words per sentence	-	8.70
Average number of phonemes per sentence	-	35.19
Average number of phonemes per word	5.61	4.04

Table 1: Statistics of the ISO-211 and RM-3000 corpora.

3.1. Features for lip-reading

In (46), three video resolutions (640 x 360, 1080 x 720 and 1920 x 1080) were compared in a visual-phone lip-reading recognition task, and it was found that there was no significant difference in the accuracy obtained. Therefore, to improve the efficiency of the feature extraction and modelling processes, all videos were down-sampled to a third of their original resolution to 360 × 640 pixels. Between 20 and 30 frames from each recording session were selected for hand-labelling: we labelled frames that described the extremities of mouth movements to capture as much variance of shape and appearance possibilities as possible. In each selected frame, 111 points were labelled over the whole face to ensure stability when tracking, which was done using the inverse compositional project-out AAM algorithm (37). An example frame is shown in Figure 1 with landmark points on the face: eight points on each eyebrow, 12 points on each eye, 2 points per nostril, 19 points around the chin and up the edge of the head to eye-level, 28 points on the outer lip contour, and 20 on the inner lip contour. After tracking the complete datasets, only the inner and outer lip contour points were retained prior to the AAM feature extraction process.

It seemed possible that the RM-3000 database (recorded by a male speaker) might be "noiser" than the ISO-211 database (recorded by a female speaker) because of the presence of facial hair and the lack of makeup (particularly lipstick) on the former recording. In practice, these differences did not seem to affect tracking or accuracy of segmentation in the feature extraction process.

AAMs encode the shape and appearance information of the lips. The *shape*, $\mathbf{s}$, of an AAM is described by the x and y-coordinates of a set of n vertices that delineate the lips: $\mathbf{s} = (x_1, y_1, \ldots, x_n, y_n)^T$,. These points are obtained using the tracking method described above. A compact model that allows a linear variation in the shape is given by:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{m} p_i \mathbf{s}_i, \tag{1}$$

where $\mathbf{s}_0$ is the mean shape and $\mathbf{s}_i$ are the eigenvectors corresponding to the m largest eigenvectors of the covariance matrix—these vectors accounted for 95% of variation in the shape mode and 90% variation in the appearance mode. The coefficients p_i are the shape parameters that define the contribution of each eigenvector in the representation of $\mathbf{s}$. Such a model can be computed using Principal Component Analysis (PCA).

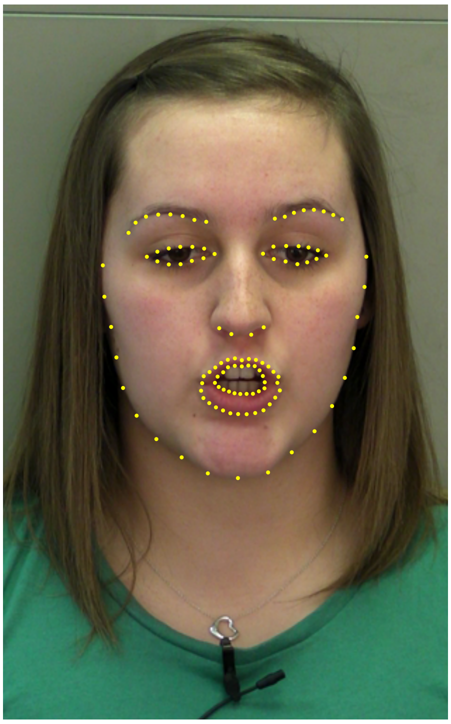

Figure 1: An example frame from the isolated-word dataset. Landmarks are hand-labelled on 20 to 30 images of the face to aid tracking. Points on other parts of the face are discarded for feature extraction.

The *appearance*, A, of an AAM is defined by the pixels that lie inside the base shape $\mathbf{s}_0$. AAMs allow linear appearance variation, so A can be expressed as a base appearance A_0 plus a linear combination of l appearance images A_i:

$$A = A_0 + \sum_{i=1}^{l} \lambda_i A_i, \tag{2}$$

where λ_i are the appearance parameters. The mean appearance A_0 and basis appearance images A_i can be computed by applying PCA to the images after warping to the mean shape, $\mathbf{s}_0$ (10). Although separate shape and the appearance components of an AAM can be used as features for lipreading, combined AAM features (10) are more discriminative (28), and we used these. Velocity (Δ) and acceleration ($\Delta\Delta$) features are added, and we apply a per-speaker z-score normalisation to the features to remove the mean and normalize the standard deviation.

4. Recognition Units for Lip-Reading

A phoneme can be defined as 'The smallest contrastive linguistic unit which may bring about a change of meaning' (12). A speaker must be capable of producing sounds that are recognisable as distinct phonemes for their speech to be understood. However, there is no requirement for a speaker's *visual* signals (e.g. mouth shapes) to form contrastive patterns, and hence there is no precise visual equivalent of the phoneme. The term *viseme* is loosely defined (16) to mean a visually indistinguishable unit of speech, and a set of visemes is usually defined by grouping together a number of phonemes that have a (supposedly) indistinguishable visual appearance. Several many-to-one mappings from phonemes to visemes have been proposed and investigated (16; 18; 9; 1).

For visual speech recognition, it seems intuitive that the units of recognition to be modelled should be visemes rather than phonemes, since the phonemes that are mapped to a single viseme are (supposedly) not visually distinguishable. However, because of the many-to-one mapping of phonemes to visemes, two words that have distinct phonemic transcriptions may have identical visemic transcriptions. These words are termed *homophenous* words—they sound different but look identical (e.g. 'bat', 'pat' and 'mat'). So it seems that for visual speech recognition, we are faced with a choice: model visemes, and deal with ambiguous word transcriptions; or model phonemes, and thus attempt to model events that are apparently indistinguishable. Here, we investigate the results from these two approaches.

Some studies have calculated that as many as 40%–60% of English spoken words could be homophenous, something that poses a significant problem for visual speech recognition (2). Here, we define a set of words to be homophenous if they all have the same viseme transcription in whatever phoneme/viseme mapping we are using. Of the 979 different words spoken in our database, 106 (10.83%) are homophenous when the Fisher phoneme-to-viseme mapping ((16), Table 2) is used. However, because of the uneven distribution of words in the 3000 sentences, these homophenous words account for 8988 (34.42%) of all word tokens out of a total of 26114 tokens. Therefore, even with perfect viseme recognition, the recogniser's performance could be as low as 65.58% if it were always to make the wrong choice between a group of homophenous words.[1]

Viseme Class	Mapped Phonemes
V1	/b/ /p/ /m/
V2	/f/ /v/
V3	/t/ /d/ /s/ /z/ /th/ /dh/
V4	/w/ /r/
V5	/k/ /g/ /n/ /l/ /ng/ /hh/ /y/
V6	/ch/ /jh/ /sh/ /zh/
V7	/eh /ey/ /ae/ /aw/ /er/ /ea/
V8	/uh/ /uw/
V9	/iy/ /ih/ /ia/
V10	/ah/ /ax/ /ay/
V11	/ao/ /oy/ /ow/ /ua/
V12	/aa/
V13	/oh/
V14	/sil/

Table 2: Description of the Fisher phoneme-to-viseme mappings to collapse 45 phoneme classes into 14 viseme classes. A viseme is reserved for the silence model (/sil/)

4.1. Experiments

For our recognition experiments, we used a conventional HMM/GMM system, an approach that has been successful for automated lip-reading (11; 35; 24). We trained monophone models of recognition units using 20 iterations of the embedded Baum-Welch re-estimation algorithm. An exhaustive search was performed to find the optimum number of states (three) and mixture components (19 per state). A short-pause model (*sp*) was tied to the centre state of the HMM that modelled silence to allow a short-duration silence between words. Ten-fold cross-validation was used, so that 2700 sentences of the RM-3000 dataset were used for training and the other 300 for testing. We built word, viseme and phoneme bigram language models (as required for a particular experiment) from the transcriptions of the 5000 RM sentences not used to make the RM-3000 dataset. The grammar-scale factor was optimised to give the best results.

Figure 2 shows the results obtained for audio and visual recognition as a function of the number of sentences used as training data. Note that the accuracy here is the accuracy of the recognition *unit* used, not *word* accuracy. We used phoneme and viseme units for both audio and visual data. As the terms used to describe the units used might be confusing, Table 3 clarifies their meaning. Figure 2 shows that, as expected, we can achieve very good phoneme recognition accuracy on single-speaker audio data. It is interesting to note that viseme recognition accuracy is actually a little lower (about 2%) than phoneme accuracy when using audio data, despite the number of viseme classes being less than one third of the number of phoneme classes. We can attribute this to the fact that the phoneme-to-viseme mapping of Table 2 groups to-

[1] *Homophones* share the same *phonetic* transcription e.g. 'for' and 'four'. Although these are a nuisance in speech recognition, they make up a tiny proportion of all words, unlike homophenous words.

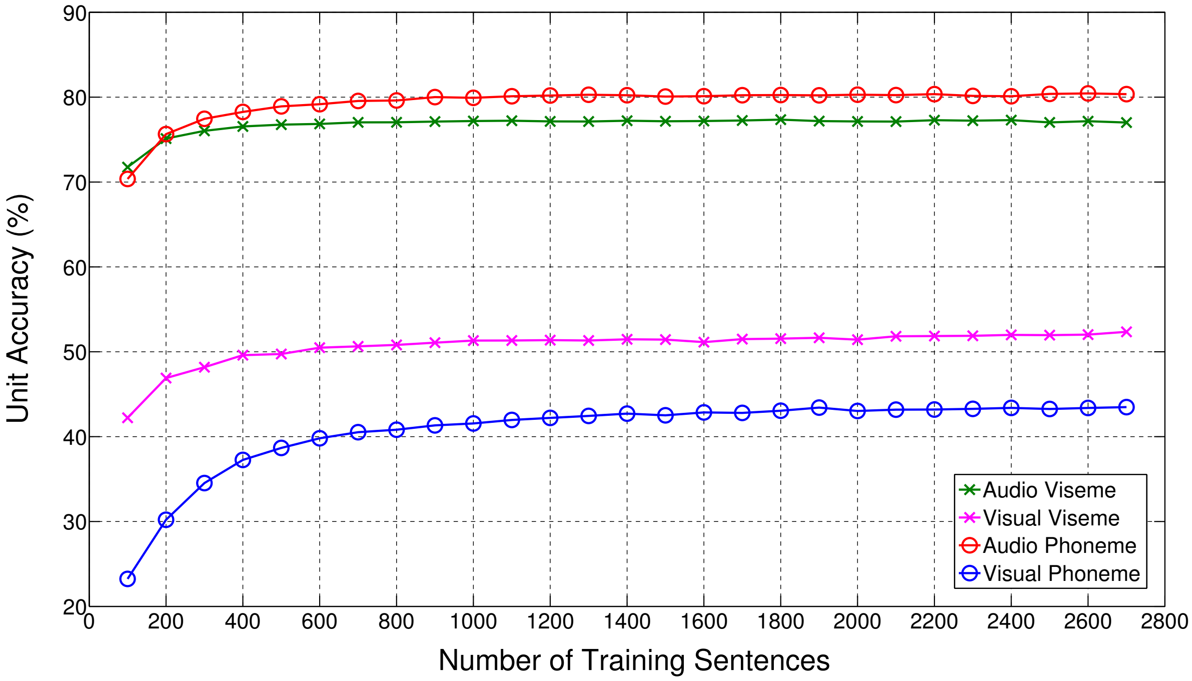

Figure 2: Unit recognition accuracy on 3000 speaker-dependent sentences from the Resource Management Corpus (RM). See Table 3 for an explanation of the units used. Error bars (a result of testing on different folds) have been omitted because they are too small to discern.

Term	**Description**
Audio phoneme	Results obtained on audio data using 45 monophone units.
Audio Viseme	Results obtained on audio data using 14 viseme units (mapping as per Table 2.)
Visual Phoneme	Results obtained on visual data using 45 monophone units.
Visual Viseme	Results obtained on visual data using 14 viseme units (mapping as per Table 2.)

Table 3: Clarification of the terms used in the experimental results.

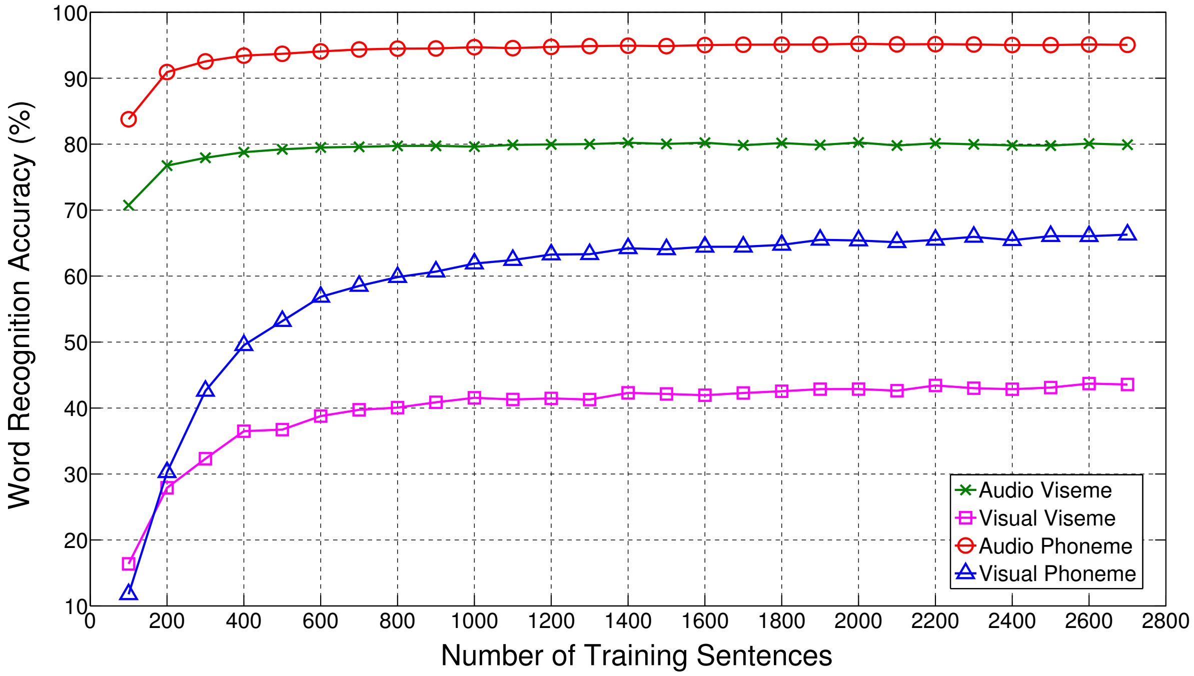

Figure 3: Word recognition performance on 3000 speaker-dependent sentences from the Resource Management Corpus (RM). See Table 3 for an explanation of the units used. Error bars on points omitted because they are too small to discern.

gether phonemes that have very different acoustic features, and so the variation in the features within the classes is high, and therefore difficult to model. Using visual data, the situation is reversed: we obtain better accuracy (near 10% better) using visemes rather than phonemes, which is what we would expect from using the phoneme-to-viseme mapping, which is designed to combine visually similar phonemes into a lower number of relatively homogeneous classes. However, the accuracy is significantly lower than that obtained with audio data.

But *unit* recognition accuracy is not of great interest (8)—we could reduce the number of units to two and get probably near 100% unit accuracy, but word accuracy would be very low. Figure 3 shows what happens when we use either phoneme or viseme units to recognise words. For audio data, the best performance (about 96% accuracy) is obtained when the units used are phonemes, and when viseme units are used with audio data, performance suffers considerably (about 15% lower), because one is combining sounds that may be quite different into a single unit. This effect is even more pronounced when using visual data: word recognition accuracy is about 23% worse when using viseme rather than phoneme units. Given that the viseme recognition rate is higher than the phoneme recognition rate, it is tempting to attribute this result to the presence of homophenous words. In other words, the decoded viseme strings may actually be more accurate than the decoded phoneme strings, but because there are often two or more words that share the same viseme transcription, performance is low because of the difficulty of selecting the correct word. In the next section, we demonstrate that this explanation is wrong, and give an alternative explanation for the drop in performance.

For phoneme, viseme or word recognition, Figure 2 and Figure 3 show that with audio data, optimum recognition performance is obtained with about 600 training sentences, whereas for visual data, performance is still increasing when the full set of 2700 sentences has been used for training. This implies that lip-reading requires considerably more data to reach optimum performance than audio ASR. It is also interesting to note that word recognition performance is about the same using both viseme and phoneme units when only 200 sentences are used for training, but performance using phonemes outstrips performance using visemes as more training sentences are added. This may be explained by the fact that phonemes require more training to achieve maximum performance because there are three times as many phoneme classes as viseme classes.

4.1.1. Analysis of the effect of homophonous words on recognition performance

We expected to get increased word accuracy for visual speech by combining 'indistinguishable' visemes into the same class, but performance was actually considerably lower using visemes than using standard phoneme units. Was this due to the formation of homophenous words, which now constituted 34% of the spoken vocabulary? We devised an experiment to see how well a word bigram language model was able to disambiguate the correct word from a set of homophenous words within a given context in a sentence. During decoding, the relative influence of the acoustic and language models on word selection is controlled by the grammar scale factor (GSF). The higher the GSF, the more weight is placed on word sequences that are *a priori* likely (i.e. trained by the language model) rather then ones suggested by the evidence from the viseme models.

We synthesised a set of 'perfect' features for a number of sentences in our corpus in the following way. Firstly, each sentence was transcribed as a sequence of visemes. The resulting viseme sequence was replaced by the corresponding sequence of concatenated HMMs, $\mathcal{S}$, and the viseme feature vectors corresponding to the sentence were force Viterbi-aligned to $\mathcal{S}$. Suppose N_i feature vectors had been Viterbi-aligned to state s_i of $\mathcal{S}$. Then the mean vector of the most-frequently used mixture component of state s_i was duplicated N_i times, and the resulting vector sequence added on to the end of a store. This resulted in a sequence of synthetic feature vectors of the same length as the original utterance that matched perfectly to the sequence of viseme HMMs corresponding to the sequence of words in the sentence. However, when decoding this sequence to a word sequence, two ambiguities must be resolved:

1. different possible segmentations of the viseme string into words;
2. homophenous words.

These ambiguities are resolved by the language model. Figure 4 shows the effect on the word accuracy of increasing the GSF when the 'perfect' features were decoded by the recogniser. When the GSF is 0 the language model has no effect, and the word accuracy is rather low (92%) because of the above ambiguities. If the GSF is increased to 1, the language model now chooses more correctly from the possible segmentations and from the sets of homophenous words, and accuracy increases to about 98%. However, if the GSF is further increased, accuracy falls, because the recogniser now places too much weight on high-probability word sequences that it has learnt from the training data at the expense of likelihood information from the viseme HMMs. An analysis of the remaining 2% errors showed that they were indeed caused by the ambiguity of homophenous words. A pair of confused words usually had the same viseme transcription and could plausibly appear in the same position in the decoded sentence i.e. the associated bigrams with the surrounding words presumably had similar probabilities. Examples are the pairs 'hepburn/campbell', 'westpac/rathburn', 'mind/miles', 'sensors/texas', 'barge/march', 'six/since' etc. Another common error was a confusion of plural/singular versions of a word that ends with a phoneme in the same viseme group as the phoneme /s/ e.g. 'threat/threats', 'speed/speeds', 'length/lengths'. The final /s/ of of these words is the same viseme (V3) as the preceding phoneme. So our conclusion is that, providing that a suitable language model is used to resolve ambiguity, the presence of homophenous words adds only a small error to performance.

This result implies that the deterioration in performance when visemes rather than phonemes are used is due to deficiencies in modelling of visual features rather than language issues. This is not surprising when one considers that audio ASR accuracy is increased if contextual modelling is performed by the use of triphones and quinphones. In practice, many phonemes have one or more allophones, different sounds that are perceived as the same phoneme, and coarticulation, which depends on context, alters the realisation of phonemes. So we should not be surprised if the same is true of visemes, especially as co-articulation is even more pronounced in visual speech. Because different phonemes occur in different contexts, by modelling phonemes in visual speech, one is, in effect, modelling different contexts of a viseme.

The issue of units for audio-visual speech recognition has been investigated by others e.g. (19; 33), most notably by Hazen (22). Although he did not consider the effect of homophenous words, he also came to the conclusion that a viseme representation was not beneficial for recognition (in fact he used tri-visemes). The work described here was performed on data from a single speaker and so the conclusion that visemes are sub-optimal units should be treated with caution. However, recent work by Hassanat (21) showed that visemes were sub-optimal recognition units for each of 27 male and female speakers and Yu (61) also made a similar finding using different data from two different speakers.

5. A Weighted Finite State Transducers Model

A finite-state automaton (FSA) is a mathematical model of a sequence of events. An FSA is defined by a finite set of *states* which are connected using *transitions*. Weighted finite-state transducers (WFSTs) are similar to FSAs, except that every transition also has an associated *transduction* between an input and an output symbol. Additionally, the transitions have *weights* associated with them that can be used to favour certain paths though the automaton over others. Figure 5 shows a very simple 3-state WFST. States are depicted by circles and transitions by arrowed lines. Starting states are defined by a bold outline surrounding the state (state 0 in Figure 5) and final states are defined by double-line borders around the state (state 3 in Figure 5). This transducer has the sole function of converting the input string *abc* to the output string *xyz*, and simultaneously producing an associated weighting of $1.2 + 3.2 + 3.3 = 7.7$. The *composition* operation provides the ability to combine multiple transducers using the binary relationship between the input and output symbol domains. If the transduction $x \rightarrow y$ is performed by transducer T_1 and the transduction $y \rightarrow z$ is performed by transducer T_2, then $T_1 \circ T_2$ (i.e. the transducer built from the composition of T_1 and T_2) models the transitive transduction $x \rightarrow z$. If several transducers are composed

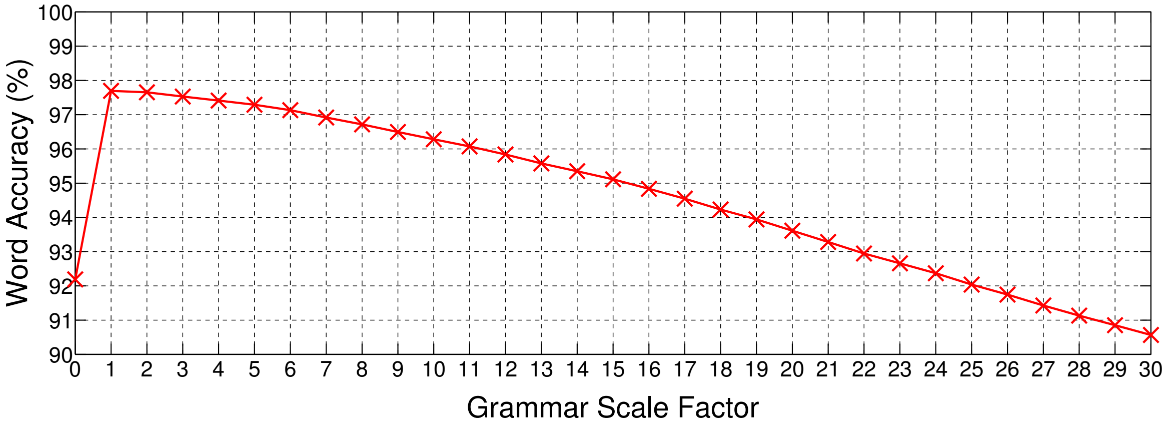

Figure 4: The effect of the language model on word accuracy when the recogniser is given 'perfect' features (i.e. ground-truth features generated by the trained HMMs). With a grammar scale factor of zero, the bigram word-pairings are preserved but each has equal probability. Thereafter, the bigram language model has an increasing influence on the decoding. Error bars on points omitted because they are too small to discern

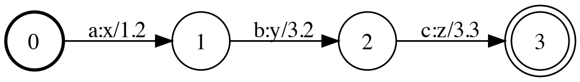

Figure 5: A example of a weighted finite-state transducer that translates the string '*abc*' to '*xyz*'.

one after the other in this way, the resulting system is known as a transducer *cascade*, and this has been found to be very useful in both speech and language processing (55; 27; 42; 38). A comprehensive introduction to WFSTs would go on to describe the operations of Union, Epsilon Removal, Closure, Determinization and Minimization as applied to WFSTs. Space does not allow us to do this here, so we refer the interested reader to articles by Mohri and Riley which give detailed descriptions of the underlying theory of WFSTs and their application to speech recognition problems (39; 38; 40).

Figure 6 gives an overview of the architecture of our WFST-based system. On the left-hand side, an N-best list of phoneme sequences is output from a visual phoneme recogniser controlled by a phoneme bigram language model. One or more of these sequences are fed to a cascade of four WFSTs, marked 'P*', 'C', 'L' and 'G' in the diagram, whose function we describe below. The construction of the 'C' transducer is also shown on this diagram: note that it is built from a dataset that is independent of the sets used to train or test the phoneme recogniser. The output text is produced by an algorithm that finds the 'best' path through the transducer cascade, where 'best' means the path that produces the minimum summed transducer weights.

5.1. The Input Transducer (P^*)

The input transducer has the function of converting output from the phoneme recogniser into the form of a transducer so that it can subsequently be composed with the rest of the transducer cascade. This transducer can represent the 1-best decoding of the phoneme recogniser, the N-best-decodings, or a phoneme lattice. In the case of the 1-best decoding, the transducer is a finite-state automaton with no transduction i.e. the output sequence is identical to the input sequence. For N-best-decodings, we build a WFST for each of the N decodings and then form the *union* of these WFSTs. The resulting transducer is determinized and minimized to enhance performance. However, this approach (which we term *N-Best-1* for future reference) restricts us to processing a single one of the N-best decodings at any time, and it seems plausible that a closer approximation to the correct phoneme sequence could be found by taking a route through several of the N-best decodings. Suppose the longest decoding D_L consists of N_L words. We use dynamic programming to align each decoding $D_1, D_2, \ldots, D_N$, to D_L so that our decodings can now be represented by an $N \times N_L$ matrix. In columns (time slots) of this matrix where all the decodings agree with each other, the same decoded phoneme label occurs in every row. Where decodings do not agree, there are multiple phoneme labels present in a column. It is straightforward to construct a WFST that is capable of traversing all possible paths through this matrix. A typical example of the resulting transducer is shown in Figure 7. This technique produces more compact WFSTs than the first technique which enables faster computation, although the increase in *out-degree* (the number of arcs exiting from a state) has the opposite effect. This way of expressing hypotheses has been termed a confusion-network (or 'sausage') (60) and was first proposed in a different context in (36). We term it *N-Best-2*.

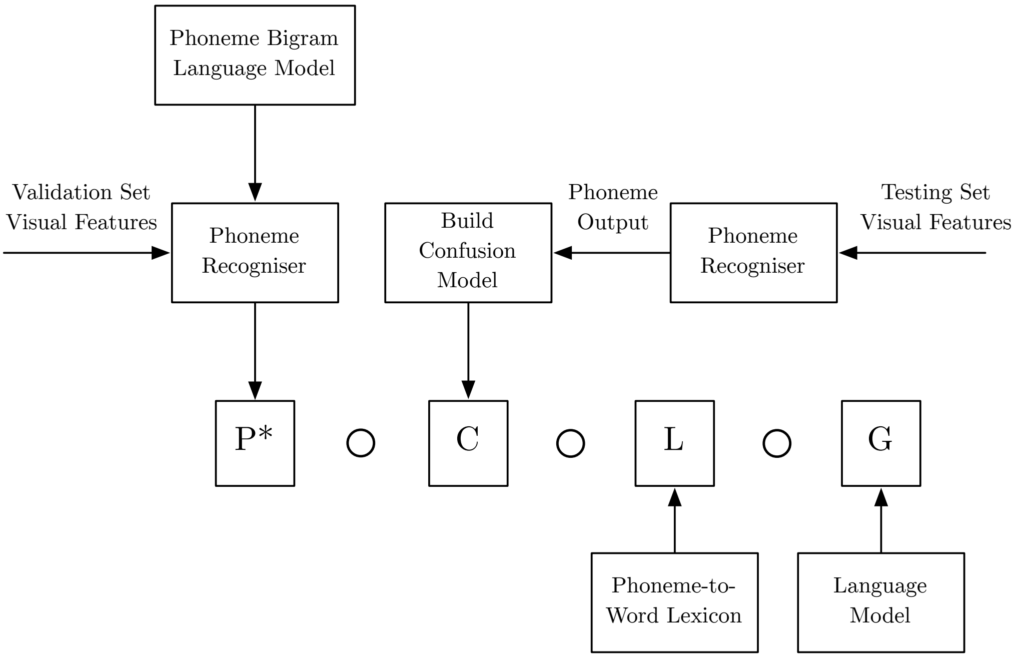

Figure 6: Our proposed WFST lip-reading system

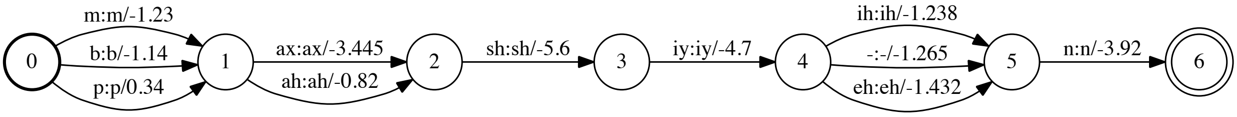

Figure 7: An example P* transducer. This transducer models the N-best output from the phoneme recogniser in response to the visual features for the word ‘Machine’.

5.2. The Confusion Transducer (C)

This transducer models the observed pattern of errors (substitutions, deletions and insertions) made by the phone recogniser. Its function is to input a set of errorful phoneme strings from the phone recogniser, and, using the observed error patterns, process these into a rich set of output strings that can then be processed into word sequences by the lexicon and language model transducers. The transducer is built by forming a confusion-matrix from the phone recogniser output and then converting this matrix into a transducer. The confusion-matrix is in turn built by aligning (using dynamic programming) the output of the phoneme recogniser to the ground-truth phoneme string and processing each pair of aligned symbols in turn. An example aligned phoneme string and the resulting confusion-matrix are illustrated in Figure 8. Note that the values shown in this confusion-matrix are counts, and these can easily be converted to probabilities $\Pr(p_j|p_i)$ (the probability that phoneme p_j is recognised when the ground truth is phoneme p_i) by normalising across a row.

	Sub.			Ins.		Del.		
Ground-Truth:	d	ih	f		r	ax	n	t
Recognised:	t	ih	f	v	r		n	t

Response

		d	f	n	t	r	v	ax	ih	DEL
Input	d	0	0	0	1	0	0	0	0	0
	f	0	1	0	0	0	0	0	0	0
	n	0	0	1	0	0	0	0	0	0
	t	0	0	0	1	0	0	0	0	0
	r	0	0	0	0	1	0	0	0	0
	v	0	0	0	0	0	0	0	0	0
	ax	0	0	0	0	0	0	0	0	1
	ih	0	0	0	0	0	0	0	1	0
	INS	0	0	0	0	0	1	0	0	

Figure 8: An example alignment between the ground-truth and recognised sequences using dynamic programming for the phonetic transcription of the word *different* (top) and the resulting confusion-matrix (bottom).

The WFST shown in Figure 9 illustrates a key concept in our system, namely how a WFST can correct a phoneme string that contains errors. The transducer shown is a very specific one that corrects the string ‘t ih f v r n t’ to ‘d ih f r ax n t’ (‘different’). The weights shown in this case are illustrative only. Where no errors were made, the transducer’s input and output symbols are the same i.e. ‘t/t’, ‘n/n’, ‘r/r’, ‘f/f’, ‘ih/ih’. The deleted phoneme ‘ax’ is re-inserted by means of the transduction ‘–/ax’ and the inserted phoneme ‘v’ is deleted by the transduction ‘v/–’.

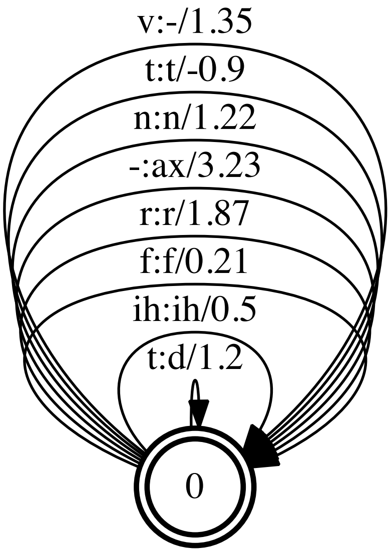

Figure 9: A cyclic confusion weighted finite-state transducer to correct the hypothesised sequence produced by the recogniser in Figure 8. The deletion of the phoneme *ax* in the hypothesised sequence is modelled in the confusion transducer using the epsilon symbol (-) to reverse the error and insert a phoneme into the hypothesised sequence. This epsilon symbol is reserved to allow ‘free’ transitions between states and is used to model both insertions and deletions.

The phoneme ‘t’ appears twice in the input. On the first occasion, it is correct, but on the second, it is an error and should be corrected to ‘d’. Hence there are two entries with ‘t’ as the first phoneme, one that maps it to ‘t’ and another that maps it to ‘d’. The entries have different weights, and these weights are actually the negative log probability values in the confusion-matrix: hence the higher the probability value, the lower the weight. In this case, the lower weight associated with the transduction ‘t/t’ makes it more likely that this transduction will be preferred in a situation where ‘t’ and ‘d’ are both possible responses to an input ‘t’. In practice, we know neither what the input sequence will be nor what the ground-truth should be, so the confusion WFST has an arc for every single non-zero entry in the confusion-matrix and hence produces a very large number of possible strings in response to an input sequence.

5.2.1. Estimation of the Probabilities in the Confusion Transducer

Experimentally, we have found that for the transducers to function well at correcting the strings, we need to distribute some of the probability mass from the diagonal of the confusion-matrix to off-diagonal elements. We term the simplest method of doing this *base smoothing* (41). Here, each off-diagonal element in a row receives the same proportion of the diagonal element from that row:

$$S(i,j) = \begin{cases} C(i,j) + \eta C(i,i) & \text{if } i \neq j \\ C(i,j)(1-(N-1)\eta) & \text{if } i = j. \end{cases} \tag{3}$$

In equation 3, $C(i,j)$ is the original confusion matrix i.e. the estimated probability that phoneme p_i was misrecognised as phoneme p_j, S is a smoothed version of C, N is the number of phonemes, and η (> 0) is a constant that controls what proportion of the diagonal of C is redistributed along the row. η must clearly be sufficiently small that $0 \leq S(i,j) \leq 1$.

A variant on base smoothing is *Exponential Smoothing* (41) in which

$$S(i,j) = \frac{e^{\alpha C(i,j)}}{\sum_k e^{\alpha C(i,k)}}, \tag{4}$$

where α is a constant that controls the degree of smoothing applied. When $\alpha = 0$, $S(i,j) = 1/N \; \forall \; i,j$ i.e. the probability mass of row i is equally distributed over the columns of the row. As $\alpha \to \infty$, the probability mass concentrates in the largest element of the row (which is generally the element on the diagonal).

One problem encountered when building these matrices is the very large number of deletions in the output of the phoneme recogniser when visual features are input to it. These deletions can lead to spurious alignments between the ground-truth phoneme sequences and the decoded sequence, which in turn lead to poorly-estimated confusion-matrices. Consider the example shown in Figure 10, which compares the purely symbolic alignment of the ground-truth phoneme sequence (top) with a timing diagram that shows where the phonemes start and end in the speech (bottom). The alignment of /ea/ and /sil/ is evidently correct, but the deletion of the phonemes /b/ and /th/ has led to the alignment of /ih/ with /aa/. These two events are a long way apart in time, and hence /ih/ and /aa/ are unlikely to be a genuine 'confusion-pair'—it is more likely that they are an artefact of the alignment process. To alleviate this problem, we only accepted confusion-pairs if both members of the pair occurred within a certain time of each other in the speech stream. We noted that some phonemes were more prone to mis-alignment than others, and so made the threshold for acceptance of a confusion-pair different for each phoneme. This threshold was estimated by performing a symbolic alignment using the training-data whilst simultaneously recording the difference between the start-times of the ground-truth and recognised phonemes. For each ground-truth phoneme class p_i, the mean difference, μ_i, and the standard deviation, σ_i, of this difference was then estimated. A confusion-pair was only accepted for inclusion in the confusion-matrix if the difference between the start-times of each phoneme lay with the range $\mu_i \pm \beta\sigma_i$, where β was a positive constant. The effect of β on the number of accepted pairs is shown in Figure 11. Figure 11 shows that using a threshold window whose width is $\pm 3\sigma$ reduces the number of observed confusion pairs from 85000 to 76200 (11.8%) compared with using no window.

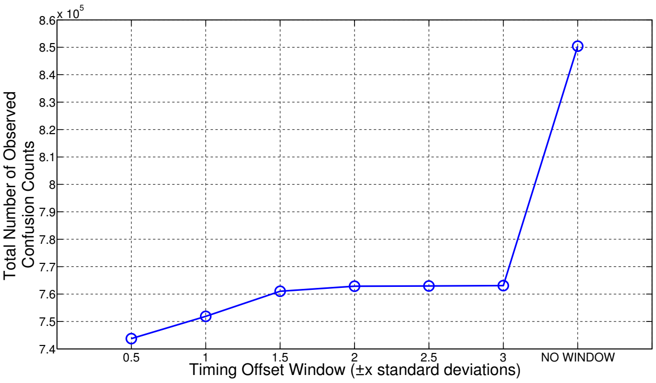

Figure 11: Analysis of the number of confusion patterns that are accepted as a function of the timing window. Error bars are not shown here because they are too small.

5.2.2. *Bigram Confusion Transducer*

It is beneficial in language modelling to employ higher-order N-gram modelling if sufficient data are available to train such models, and we expected that the extra contextual information introduced by modelling confusions between *pairs* of symbols would aid recognition performance. Conventional N-gram language models employ a back-off procedure to revert to the unigram model when a previously unseen word bigram is encountered at test time. In the same way, we maintain the unigram confusion matrix described in Section 5.2.1 as a back-off model. The bigram confusion matrix is populated using the same alignment procedure as described in Section 5.2.1 but with a window covering two phonemes instead of one. For unseen bigrams, the confusion model allows for back-off to the unigram confusion probability. An example of a bigram confusion model is shown in Figure 12. This model has been constructed using a vocabulary of three symbols: a, b, and c. Arc weights are defined using the negative logarithm of the entries in the bigram and unigram probability matrices. Owing to the strong influence of the unigram confusion matrix, a back-off weight β is applied to each unigram probability (where $0 < \beta < 1$), and the bigram probabilities are weighted by $(1 - \beta)$. Experiments were conducted using a β value with a 0.1 increment from 0.1 to 0.9 with best accuracy achieved when $\beta = 0.7$. The extension of the confusion model came at little computational cost. Only 6785 unique bigram confusions were observed during training and they increase the size of the transducer by only about 35%.

Finally, the lexicon transducer (L) is an inverse pronunciation dictionary i.e. it maps sequences of phonemes to whole words. The language model transducer (G) implements a word bigram language model with backoff to unigram. These transducers are standard and are described in detail in e.g. (39).

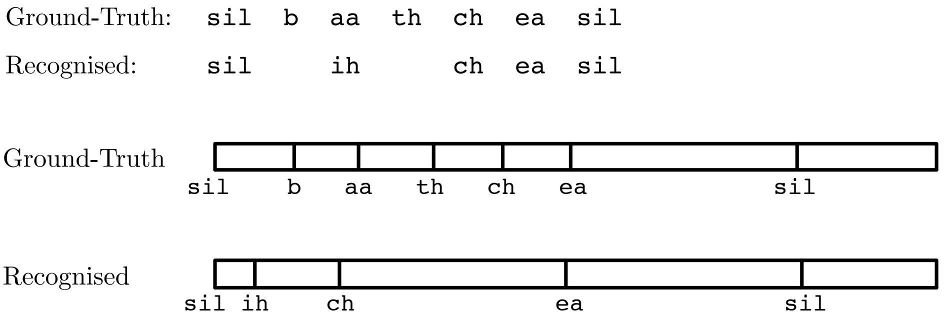
Ground-Truth: sil b aa th ch ea sil
Recognised: sil ih ch ea sil
Ground-Truth
sil b aa th ch ea sil
Recognised
sil ih ch ea sil

Figure 10: Top: the purely symbolic alignment between the ground-truth phoneme sequence and the output of the phoneme recogniser. Bottom: the relative timing of the two phoneme strings. The timing diagram shows that the alignment of /ea/ and /sil/ is correct. However, the deletion of the phonemes /b/ and /th/ has led to the alignment of /ih/ with /aa/, but these events are a long way apart in time and this is unlikely to be a genuine confusion-pair.

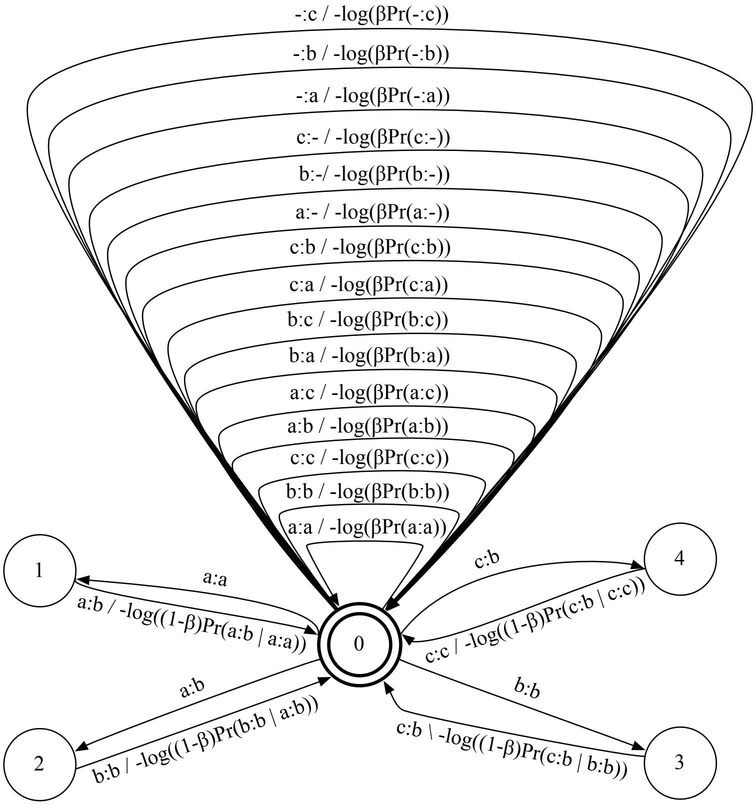

Figure 12: An illustration of a bigram confusion model with backoff weights. The vocabulary consists of three symbols: *a*, *b*, and *c*. The unigram backoff arcs (above the state marked '0') are derived from the unigram confusion matrix, which contains fifteen entries. Four possible bigram arcs have been added. A backoff weight, β, is applied to the unigram probabilities and a weight $(1 - \beta)$ is applied to the bigram probabilities.

6. Results on the ISO-211 Dataset

For the baseline 'standard' system, the 1256 words were split into six folds. Words were randomised between folds such that no word appeared in the same fold more than once. The six folds were then split into a training set consisting of five folds and a testing set consisting of the remaining fold. Cross-fold validation was performed with each fold used in turn for testing. For the WFST approach, the additional confusion model also requries training using a further held-out segment of the data. Therefore, the dataset was divided into three segments: a model training set (four folds), a fold used to train the confusion model and a validation fold to produce results. Table 4 summarises the results of our approach on the isolated word database. Results for two baseline systems are shown here. These are:

1. Baseline 1: A 'standard' HMM system. This used five-state monophone HMMs of each of the 44 phonemes (plus Silence), with eleven component Gaussian mixture models (GMMs) associated with each state and with a bigram phoneme language model. The parameters five states and eleven components were determined after an exhaustive search over the parameter space.
2. Baseline 2: A system that took the 1-best phoneme output from the phoneme recogniser and found the lowest alignment cost (using dynamic programming) to the phoneme transcriptions of each of the vocabulary words.

Table 4 gives a comparison of the results of the experiments on isolated word recognition. Baseline 1 (result A), a standard HMM approach, achieved nearly 60% word accuracy. The system of Baseline 2 is essentially unconstrained phoneme recognition, and its low performance shows that the visual phoneme recognition rate is low. Compare this result with system C, which uses a confusion WFST followed by a lexicon WFST. The confusion WFST was not estimated from data, but built by taking an identity matrix and redistributing a small amount of the diagonal element of a row equally to all other elements on the row. This creates a confusion-matrix that gives a high weight to mapping an input phoneme symbol to itself, but allows mapping to *any* other symbol, albeit with a low weight. The result, 35.36%, is considerably better than Baseline 2, and interestingly, considerably better than system D, in which the confusion-matrix was formed by purely symbolic alignment of the phoneme recogniser output and the ground-truth phoneme strings. In fact system D is almost no better than Baseline 2, which shows how poor the symbolic alignment process is. But when the confusion-matrix is formed from data with a timing constraint (system E), performance increases to over 46%. Using N-best decodings rather than just the top decoding is not beneficial if they are combined using *N-best-1* (system F, i.e. we are effectively allowed to process all N decodings in parallel but not combine them). However, using *N-best-2* (system G), in which decodings can be combined, leads to a further increase in performance to 49.7%. We found that base smoothing was always better than exponential smoothing: e.g. for system G, the difference is 49.70% versus 42.68%. Finally, using a bigram confusion-matrix adds another 3% to accuracy. However, the best result using WFSTs is still 7% worse than the standard approach. It seemed that the sparsity of data available to estimate the confusion-matrix entries was a problem when using the ISO-211 dataset: it was useful in developing the WFST techniques initially, but was too small to enable the full potential of the technique to be realised. In the next section, we report results on the RM-3000 dataset.

7. Results on the RM-3000 Dataset

In these experiments, the data were divided into ten folds, six of which were used for training the models, two for training the confusion-model, and two for testing. The folds were rotated to give cross-validation results.

The problem of deletions of phonemes in visual speech (mentioned in Section 5.2.1) becomes more acute when continuous speech rather than isolated words is recognised. If the phoneme recogniser is run to maximise phoneme accuracy (defined as $(N - D - S - I)/N$, where N is the total number of symbols in the ground-truth strings, D the number of deletions, S the number of substitutions and I the number of insertions), deletions account for over one quarter of the errors, and sequences of up to six deleted phonemes are sometimes seen: it seems very unlikely that any system could correct such a large gap in the output. By altering the 'insertion penalty' of the decoder, deletions can be traded to some extent for insertions, and these are easier for our system to correct. However, the overall accuracy figure goes down when the insertion penalty is altered to a non-optimal setting. We ran our WFST system on the RM-3000 data with the phoneme recogniser optimised to reduce deletions at the expense of extra insertions. The results were disappointing: a word accuracy of just 12.8% compared with an accuracy of 66.3% for a conventional HMM system that used monophone models with GMMs.

7.1. Enhancing the performance of a conventional word decoder

Our essential approach in this work consists of phoneme recognition, followed by generation of a set of string hypotheses using the confusion transducer, followed by decoding using a network of legal words whose sequences are constrained by a language model. It seemed likely at this point that this approach might not be as successful as the conventional approach of allowing only legal words to be decoded from the outset by using a network of words, because the raw visual phoneme recognition accuracy was too low.

However, our conventional word decoder provided fairly accurate sets of word hypotheses from visual speech, and it seemed to us that these could be enriched by the phonetic confusion transducer by first converting them into phoneme hypotheses. The enriched phoneme hypotheses can then be converted back to word hypotheses using the lexicon and language model transducers. The advantage of this approach is that hypotheses that may not have been considered or have been rejected early on by the conventional word decoder can be re-instated by the phonetic confusion transducer on the basis of possible phonetic confusions.

Figure 13 shows the architecture of our proposed system. The confusion transducer here is built by aligning phoneme strings that are transcriptions of the recognised word strings to phoneme strings formed by writing the ground-truth word strings as phoneme strings. In fact, we

	System	**% Word Accuracy (Std. Deviation)**
A	'Standard' HMM System (Baseline 1)	59.9 (4.19)
B	Phone decoding followed by string-matching (Baseline 2)	20.1 (1.43)
C	WFSTs with a near-identity confusion matrix to avoid $-\infty$ log probabilities on off-diagonal elements: a small probability mass is added to every element. Uses top decoding only.	35.4 (2.27)
D	WFSTs with confusion-matrix formed from purely symbolic alignment using top decoding only and base smoothing	21.4 (3.30)
E	WFSTs with confusion-matrix formed using timing information, using top decoding only and base smoothing	46.1 (1.03)
F	WFSTs with confusion-matrix formed using timing information combined using algorithm *N-best-1* and base smoothing.	36.0 (0.88)
G	As above, but combined using algorithm *N-best-2*.	49.7 (1.60)
H	WFSTs using a *bigram* confusion matrix with timing, with the backoff weight (β) set to 0.7	52.9 (3.31)

Table 4: Isolated word recognition accuracy results obtained on the ISO-211 dataset.

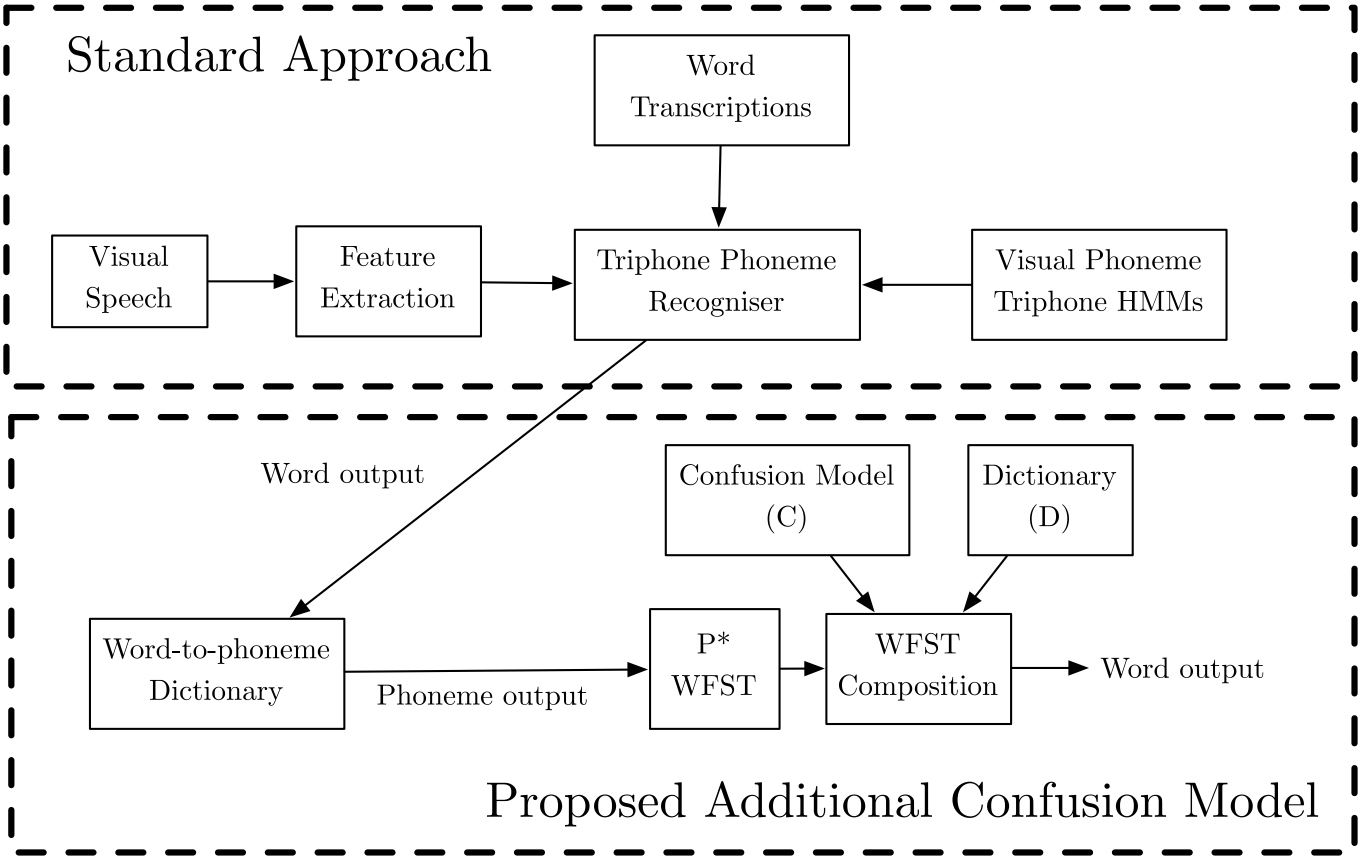

Figure 13: The architecture of a system that enriches phoneme hypotheses. The word hypotheses obtained from a conventional word decoder are converted to a set of phoneme strings which are input to our transdcuer cascade to be converted back to word hypotheses.

force-align the ground truth word strings to the appropriate model sequences in order to get timing information for both the aligned phoneme strings so that we can use timing restrictions to select confusion pairs as described in Section 5.2.1. The RM-3000 dataset was split into ten folds, each containing 300 sentences. From these folds, three sets were formed: a training set (consisting of eight folds) which was used to train the triphone word recog-

	Conventional HMM Triphone Word Decoder	**Addition of WFSTs**
No. of Correct Words	20,500	20,477
No. of Deleted Words	2,308	2,363
No. of Substituted Words	3,306	3,274
No. of Inserted Words	763	596
Total No. of Words	26,114	26,710
Word Accuracy (%)	75.58	76.14

Table 5: Comparison of the word recognition statistics between the standard approach triphone system and the WFST confusion modelling system using the triphone decodings.

niser, a testing set (one of the two remaining folds) which was used to train the confusion model, and a validation set (the final fold) which was used as test data. Cross-fold validation was performed with each validation set used as unseen test data. A comparison of the performance of this system with the conventional word decoder is given in Table 5. Table 5 shows that the proposed system achieves a small gain in accuracy (0.58% absolute) over the conventional system. McNemars test (17) shows that the difference between the two systems is statistically significant with $p < 0.001$. Interestingly, most of this gain seems to have come from reducing the number of inserted words.

8. Discussion

This paper has (a) discussed the issue of the choice of units for automatic lip-reading (ALR) and (b) proposed novel systems based around the use of a phonetic confusion model to enhance the recognition accuracy of ALR.

Our experiments with units showed firstly that the introduction of homophenous words into the lexicon (caused by mapping from phonemes to a smaller set of visemes) led to a decrease in accuracy of ALR. However, this decrease was small compared with the loss in accuracy incurred by using visemes rather than phonemes, and so we conclude that the use of visemes is not beneficial for ALR. We say ‘unlikely’ because we investigated only one viseme mapping, but it seems clear that provided enough data is available, modelling the context of visual speech is beneficial. This confirms the result that Hazen found for audio-visual speech recognition units (22).

We then proposed a new architecture for ALR that was based on the idea that visual speech has similarities with dysarthric speech, in that its phonemic repertoire is limited because some acoustic features are invisible. The technique learns the probabilities of phoneme confusions and incorporates them into its estimation of word hypotheses. This is all done within the framework of a cascade of weighted finite-state transducers (WFSTs), which makes it fast and efficient. We demonstrated that this architecture operated successfully on a small dataset of isolated words. However, its performance was slightly lower than a conventional system, which we attributed to the lack of data available to estimate the confusions reliably. We therefore recorded a large database of continuously spoken audio-visual speech consisting of 3000 sentences from the Resource Management (RM) dataset, spoken by a single male speaker. Continuous speech exposed the poor quality of visual phonetic recognition, and we found that our system worked best by enhancing the output from a conventional word decoder, where it achieved a modest improvement in word accuracy. We achieved a single-speaker word accuracy of over 76% on this 1000-word task.

Although the improvement in accuracy obtained thus far is small, these are our first results using this architecture for lip-reading and we believe that it holds promise. Firstly, an obvious way of increasing accuracy is to combine results from the word decoder and the confusion system in a ROVER-like (15) confidence-measure based system —an analysis of our results showed that accuracy would rise by nearly 10% if the correct decision was chosen when the two systems disagreed. Secondly, there are still aspects of our post word-decoder system that need to be explored, such as higher-order confusion models, the relative weightings of the phonetic confusion and the language model probabilities and the use of techniques such as conditional random fields (which are good at utilising context) for prediction of substituted/inserted/deleted phones. Finally, we need to confirm that the system is effective in a speaker-independent environment and this will depend on whether confusion-matrices are similar across different speakers.

References

[1] Helen L Bear, Richard W Harvey, Barry-John Theobald, and Yuxuan Lan. Which phoneme-to-viseme maps best improve visual-only computer lip-reading? In *Advances in Visual Computing*, pages 230–239. Springer, 2014.

[2] KW Berger. Visemes and homophenous words. *Teacher of the Deaf*, 70(415):396–399, 1972.

[3] Christoph Bregler, Herman Hild, Stephan Manke, and Alex Waibel. Improving connected letter recognition by lipreading. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 1, pages 557–560. IEEE, 1993.

[4] Christoph Bregler and Yochai Konig. eigenlips for robust speech recognition. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 2, pages II–669. IEEE, 1994.

[5] NM Brooke, MJ Tomlinson, and RK Moore. Automatic speech recognition that includes visual speech cues. *Proceedings of the Institute of Acoustics*, 16:15–22, 1994.

[6] Cambridge-University. BEEP Dictionary. `http://mi.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html`, 2012.

[7] Luca Cappelletta and Naomi Harte. Viseme definitions comparison for visual-only speech recognition. In *Signal Processing Conference, 2011 19th European*, pages 2109–2113. IEEE, 2011.

[8] Luca Cappelletta and Naomi Harte. Phoneme-to-viseme mapping for visual speech recognition. In *ICPRAM (2)*, pages 322–329, 2012.

[9] Luca Cappelletta and Naomi Harte. Phoneme-to-viseme mapping for visual speech recognition. In *International Conference on Pattern Recognition Applications and Methods*, pages 322–329, 2012.

[10] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *European Conference on Computer Vision 1998*, pages 484–498, 1998.

[11] S. Cox, R. Harvey, Y. Lan, J. Newman, and B. Theobald. The challenge of multispeaker lip-reading. In *International Conference on Auditory-Visual Speech Processing*, pages 179–184, 2008.

[12] Alan Cruttenden. *Gimson's Pronunciation of English.* Routledge, 7th edition, 2008.

[13] Paul Duchnowski, Uwe Meier, and Alex Waibel. See me, hear me: integrating automatic speech recognition and lip-reading. In *ICSLP*, volume 94, pages 547–550. Citeseer, 1994.

[14] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *Multimedia, IEEE Transactions on*, 2(3):141–151, 2000.

[15] Jonathan G Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354. IEEE, 1997.

[16] C Fisher. Confusions among visually perceived consonants. *Journal of Speech and Hearing Research*, 11(4):796 – 804, 1968.

[17] Laurence Gillick and Stephen J Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 532–535. IEEE, 1989.

[18] Alan J Goldschen, Oscar N Garcia, and Eric Petajan. Continuous optical automatic speech recognition by lipreading. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 1, pages 572–577. IEEE, 1994.

[19] Alan J Goldschen, Oscar N Garcia, and Eric D Petajan. Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. In *Speechreading by Humans and Machines*, pages 505–515. Springer, 1996.

[20] John N Gowdy, Amarnag Subramanya, Chris Bartels, and Jeff Bilmes. Dbn based multi-stream models for audio-visual speech recognition. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–993. IEEE, 2004.

[21] Ahmad Basheer Hassanat. Visual words for automatic lipreading. *arXiv preprint arXiv:1409.6689*, 2014.

[22] Timothy J Hazen. Visual model structures and synchrony constraints for audio-visual speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(3):1082–1089, 2006.

[23] Timothy J Hazen, Kate Saenko, Chia-Hao La, and James R Glass. A segment-based audio-visual speech recognizer: Data collection, development, and initial experiments. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 235–242. ACM, 2004.

[24] S. Hilder, R. Harvey, and B.J. Theobald. Comparison of human and machine-based lip-reading. In *In the Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP)*, pages 86–89, 2009.

[25] Sarah Hilder, Barry-John Theobald, and Richard Harvey. In pursuit of visemes. In *AVSP*, pages 8–2, 2010.

[26] Jing Huang and Brian Kingsbury. Audio-visual deep learning for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7596–7599. IEEE, 2013.

[27] L. Karttunen. Applications of finite-state transducers in natural language processing. *Implementation and application of automata*, pages 34–46, 2001.

[28] Y. Lan, R. Harvey, B.J. Theobald, E.J. Ong, and R. Bowden. Comparing visual features for lipreading. In *Procs. of Int. Conf. Auditory-visual Speech Processing. Norwich, UK*, 2009.

[29] Y. Lan, B.J. Theobald, R. Harvey, E.J. Ong, and R. Bowden. Improving visual features for lip-reading. In *Auditory-Visual Speech Processing 2010*, 2010.

[30] Yuxuan Lan, Richard Harvey, B Theobald, Eng-Jon Ong, and Richard Bowden. Comparing visual features for lipreading. In *International Conference on Auditory-Visual Speech Processing 2009*, pages 102–106, 2009.

[31] Yuxuan Lan, Barry-John Theobald, Richard Harvey, Eng-Jon Ong, and Richard Bowden. Improving visual features for lipreading. In *AVSP*, pages 7–3, 2010.

[32] Karen Livescu, Ozgür Cetin, Mark Hasegawa-Johnson, Simon King, Chris Bartels, Nash Borges, Arthur Kantor, Partha Lal, Lisa Yung, Ari Bezman, et al. Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 jhu summer workshop. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–621. IEEE, 2007.

[33] Patrick Lucey, Terrence Martin, and Sridha Sridharan. Confusability of phonemes grouped according to their viseme classes in noisy environments. In *Proc. of Australian Int. Conf. on Speech Science & Tech*, pages 265–270, 2004.

[34] Patrick Lucey and Gerasimos Potamianos. Lipreading using profile versus frontal views. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 24–28. IEEE, 2006.

[35] J. Luettin and N.A. Thacker. Speechreading using probabilistic models. *Computer Vision and Image Understanding*, 65(2):163–178, 1997.

[36] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373 – 400, 2000.

[37] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[38] M. Mohri. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311, 1997.

[39] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[40] Mehryar Mohri and Michael Riley. Weighted determinization and minimization for large vocabulary speech recognition. In *Eurospeech*, 1997.

[41] S. O. C. Morales. *Error Modelling Techniques to Improve Automatic Recognition of Dysarthric Speech.* PhD thesis, School of Computing Sciences, University of East Anglia, May 2009.

[42] S. O. C. Morales and S. Cox. Application of weighted finite-state transducers to improve recognition accuracy for dysarthric speech. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[43] S. O. C. Morales and Stephen J Cox. Modelling errors in automatic speech recognition for dysarthric speakers. *EURASIP Journal on Advances in Signal Processing*, 2009:2, 2009.

[44] Ara V Nefian, Luhong Liang, Xiaobo Pi, Liu Xiaoxiang, Crusoe Mao, and Kevin Murphy. A coupled hmm for audio-visual speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–2013. IEEE, 2002.

[45] Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin, Iain Matthews, Hervé Glotin, and Dimitra Vergyri. Large-vocabulary audio-visual speech recognition: A summary of the

Johns Hopkins Summer 2000 Workshop. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 619–624. IEEE, 2001.

[46] Jacob Newman. *Language Identification using Visual Features.* PhD thesis, School of Computing Sciences, University of East Anglia, May 2011.

[47] J.L. Newman, B.J. Theobald, and S.J. Cox. Limitations of visual speech recognition. In *Int. Conference on Auditory-Visual Speech Processing*, 2010.

[48] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011.

[49] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha. Unsupervised random forest manifold alignment for lipreading. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 129–136. IEEE, 2013.

[50] E Petajan. *Automatic lipreading to enhance speech recognition.* PhD thesis, University of Illinois, 1984.

[51] E. Petajan, B. Bischoff, D. Bodoff, and N.M. Brooke. An improved automatic lipreading system to enhance speech recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 19–25. ACM, 1988.

[52] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[53] Patti Price, William M Fisher, Jared Bernstein, and David S Pallett. The darpa 1000-word resource management database for continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 651–654. IEEE, 1988.

[54] N Puviarasan and S Palanivel. Lip reading of hearing impaired persons using hmm. *Expert Systems with Applications*, 38(4):4477–4481, 2011.

[55] Emmanuel Roche and Yves Schabes. *Finite-state language processing.* MIT press, 1997.

[56] Alaa Sagheer, Naoyuki Tsuruta, Rin-Ichiro Taniguchi, and Sakashi Maeda. Visual speech features representation for automatic lip-reading. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–781. IEEE, 2005.

[57] D.G. Stork, G. Wolff, and E. Levine. Neural network lipreading system for improved speech recognition. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 2, pages 289–295. IEEE, 1992.

[58] Kwanchiva Thangthai, Richard Harvey, Stephen Cox, and Barry-John Theobald. Improving lip-reading performance for robust audiovisual speech recognition using dnns. In *Proc. FAAVSP, 1st Joint Conference on Facial Analysis, Animation and Audio-Visual Speech Processing*, September 2015.

[59] MJ Tomlinson, MJ Russell, and NM Brooke. Integrating audio and visual information to provide highly robust speech recognition. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 821–824. IEEE, 1996.

[60] Gökhan Tür, Jerry H Wright, Allen L Gorin, Giuseppe Riccardi, and Dilek Z Hakkani-Tür. Improving spoken language understanding using word confusion networks. In *Proc. Third Conference of the International Speech Communication Association, Interspeech*, 2002.

[61] Dahai Yu. *The application of manifold based visual speech units for visual speech recognition.* PhD thesis, Dublin City University, September 2008.

[62] Ziheng Zhou, Xiaopeng Hong, Guoying Zhao, and Matti Pietikainen. A compact representation of visual speech data using latent variables. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(1):1–1, 2014.

[63] Ziheng Zhou, Guoying Zhao, Xiapeng Hoong, and Matti Pietikainen. A review of recent advances in visual speech decoding. *Image and Vision Computing*, 32:590–605, 2014.