Confusion Modelling for Automated Lip-Reading using Weighted Finite-State Transducers

Dominic Howell, Barry-John Theobald, Stephen Cox

School of Computing Sciences, University of East Anglia, Norwich, UK

{dominic.howell, b.theobald, s.j.cox}@uea.ac.uk

Abstract

Automated lip-reading involves recognising speech from only the visual signal. The accuracy of current state-of-the-art lip-reading systems is significantly lower than that obtained by acoustic speech recognisers. These poor results are most likely due to the lack of information about speech production that is available in the visual signal: for example, it is impossible to discriminate voiced and unvoiced sounds, or many places of articulation, from visual signals. Our approach to this problem is to regard the visual speech signal as having been produced by a speaker who has a reduced phonemic repertoire and to attempt to compensate for this. In this respect, visual speech is similar to dysarthric speech, which is produced by a speaker who has poor control over their articulators, leading them to speak with a reduced and distorted set of phonemes. In previous work, we found that the use of weighted finite-state transducers improved recognition performance on dysarthric speech considerably. In this paper, we report the results of applying this technique to lip-reading. The technique works, but our initial results are not as good as those obtained by using a conventional approach, and we discuss why this might be so and what the prospects for future investigation are.

Index Terms: automated lip-reading, weighted finite-state transducers, visual speech recognition, confusion modelling

1. Introduction

The goal of automated lip-reading is to use only visible information from a speaker to transcribe the words that he or she speaks. Recent studies have shown that automatic lip-reading performs significantly worse than audio speech recognition [1]. These poor results are most likely due to the lack of speech information available in a visual signal (for example, the position of some articulators cannot be seen, and there is no way to tell whether a sound is voiced or unvoiced). In addition, the purpose of speech is to produce distinctive sounds to convey a message, and the particular mouth-shapes used to produce these sounds are (usually) of no concern to the speaker: it is quite possible to produce a perfectly intelligible audio signal from mouth-shapes that are not distinct, something that is verified by human lip-readers who report that some people are much more "readable" than others. Furthermore, mouth shapes are severely affected by co-articulation [2]. Because of these limitations, human lip-readers make heavy use of pragmatics and contextual information to understand what is being spoken [3].

Visual speech has an interesting relationship to dysarthric speech. Dysarthric speakers have poor control over their articulators due to medical conditions that affect their motor functions (such as cystic fibrosis, stroke etc.). This leads to a phonemic repertoire that is both reduced and distorted, and hence to speech that has low intelligibility, and is difficult to recognise - an obvious parallel with visual speech, where certain sounds cannot be distinguished visually. In previous work on dysarthric speech recognition, we learnt patterns of phonemic confusions from a talker, and when these confusions were compensated at recognition time, recognition accuracy increased [4]. We take a similar approach to lip-reading: we model visual speech as a speech signal produced by a speaker who has a limited phonemic repertoire, and learn the patterns of confusion between the ground-truth phoneme sequences and the recognised sequences. At recognition time, we find the most likely interpretation of a reduced/distorted phoneme output sequence in the light of these patterns, as was successfully explored in [5].

Figure 1 illustrates the "standard" approach and our proposed approach. In this study, we use utterances of isolated words. Both approaches begin by converting the visual signal into a sequence of feature vectors (described in section 2). Hidden Markov models (HMMs) of each phoneme are then built, as described in section 5. In the standard approach, the input feature vector sequence is decoded by forming a network of phoneme models such that any path through the network represents the transcription of one of the words in the vocabulary. The most probable route through this network is found using the Viterbi algorithm, and the word associated with this route is the recognised word. In our proposed approach, the recogniser first decodes a set of n-best phoneme sequences under the influence of a phone bigram language

model and represents this set as a transducer, P. These sequences are passed to a second transducer, C, which is a model of confusions made in this decoding, built by passing a hold-out set through the recogniser. C then expands P into a much larger set of hypotheses, together with their associated probabilities. A third transducer D allows only hypotheses that represent vocabulary words to be decoded. The word associated with the most likely path through the transducer cascade is selected as the recognised word.

2. Data Capture and Features

The data used in these experiments were captured from a single female speaker who spoke six repetitions of a set of 211 isolated words. The videos were captured with the speaker at a full-frontal pose and recorded using a Sanyo Xacti camera at 1080p resolution with a progressive scan at 59.94 frames per second.

The words were chosen to provide a high coverage of bigram phoneme pairs. There were an average of 4.6 phones in each word, with the shortest words having only two phones and the longest having seven phones.

For visual speech features, an Active Appearance Model (AAM) was used. This choice was motivated by the work conducted in [6], which concluded that model-based features (such as AAMs) perform significantly better than other techniques such as discrete cosine transform features or eigenlips. More about building AAMs and extracting visual features can be found in [7].

Sufficient modes of variation are retained to capture 85% of the variation in both shape and appearance. Velocity (Δ) and acceleration ($\Delta\Delta$) features were also added. Finally, all dimensions of the feature-vector are z-score normalised across the utterance. Utterance files across the six recording sessions are randomly shuffled to remove any bias due to a particular session.

3. Weighted Finite-State Transducers

Weighted finite-state Transducers (WFSTs) have been extensively applied to problems in audio speech recognition (ASR) and natural language processing (NLP). Similar to a finite-state machine, a WFST is a network of states with directed transitions that has an additional ability to map from an *input* symbol to an *output* symbol. Transitions between states are weighted with probabilities so that any path through the transducer has an associated likelihood [8] [9].

Formally, a WFST can be defined as an eight-tuple $(\Sigma, \Omega, Q, E, i, F, \lambda, p)$ consisting of: Σ, a finite, non-empty set of input symbols, Ω, a finite, non-empty set of output symbols, Q, a finite, non-empty set of states, E, a finite set of transitions that define the relationship between states (Q), i: an initial state ($i \in Q$), F, a set of final states ($F \subseteq Q$), λ, an initial weighting and p, a function to define the final weighting.

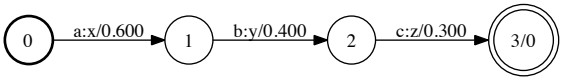


Figure 2: A simple weighted finite-state transducer

The example WFST shown in figure 2 takes an input string (in this case abc) and produces an output string (xyz) according to the shortest path for a given input sequence (in this case there is only one path). These weightings are typically defined by negative log probabilities when the topology is initialised.

Two WFSTs (A and B) can be *composed* together ($A \circ B$) to form a single WFST using the output of A as the input to B. Although this provides a method to implement NLP translations, when composing multiple WFSTs, the networks increase dramatically in size. *Determinisation* and *minimisation* both provide methods for pruning WFST networks, therefore reducing computational expense [10].

3.1. WFSTs in Speech Recognition

There have been many applications of WFSTs in speech recognition [4, 10, 8]. The speech recognition transduction cascade can be defined as a composition of the transducers P^*, C, D and M [10], defined as:

1. P^*: a transducer representing the recognised sequence of phones. When performing recognition using the standard approach, it is well-known that the "top" output sequence is not necessarily the most accurate, and so the n-best sequences are often used. For use in our own technique, this set of sequences must first be converted to a transducer. All n phone sequences are aligned to one another using dynamic programming (DP) to produce n aligned phone sequences of the same length. These sequences are then modelled as a WFST (an example is shown in figure 3a). We explored the use of up to 15-best phone sequences but found that our system performed best when we modelled the top 9-best phone sequences as a WFST.

2. C: a transducer modelling the possible confusions and reduced phonemic repertoire (represented with negative log probabilities) of *insertions*, *substitutions* and *deletions*.

3. D: the dictionary transducer mapping sequences of recognised phones sequences into complete words

4. M: defines the legal sequences of words with a word-level language model. For the purpose of the experiments conducted in this paper on an isolated word dataset, this transducer model is omitted.

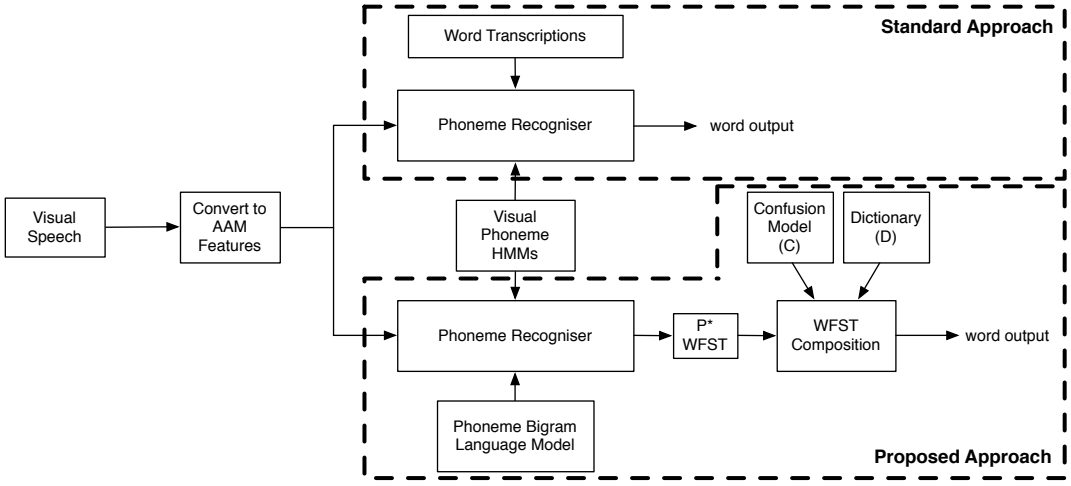


Figure 1: A comparison of the standard approach and proposed approaches to lip-reading

Figure 3 shows the four transducers representing each of the aspects of the speech recognition cascade. The most likely recognised sequence can be computed by finding the best path (λ^*) through the composed and pruned transducer network:

$$\lambda^* \Big(min(det(P^* \circ C \circ D \circ M)) \Big), \qquad (1)$$

where determinisation ($det()$) and minimisation ($min()$) are WFST pruning techniques and P^*, C, D, M are composed to build the cascade. Even with a relatively small vocabulary, building the composition of these four WFSTs introduces efficiency issues. This is especially the case when decoding (i.e. finding the lowest cost path through the network). The OpenFST library [11] was used to build the WFST networks described in this paper.

4. Experiments

4.1. Baseline1

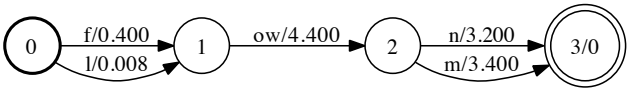
We performed standard lip-reading recognition as a baseline system using phone-level HMMs as performed in previous lip-reading work [12]. Each phone HMM was built with a left-to-right topology consisting of five emitting states and eleven mixture components, which was found to give the highest accuracy. A total of 43 HMMs were trained, one for each phone, with an additional silence HMM. The models were trained using the phone-level transcription of each word and a "flat-start" procedure [13] using ten re-estimations of embedded training. For all results, we use cross-fold validation to train and test the systems: five repetitions of the vocabulary are selected for training and the remaining repetition for testing. This is then repeated six times, and the reported result is the mean over the six tests.

4.2. Baseline2

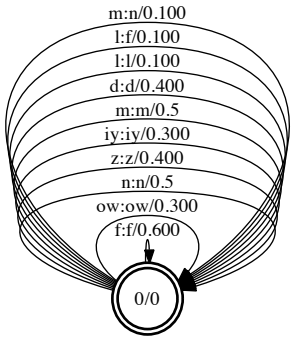
It has been known for many years that the method of decoding a phoneme string and then finding the sequence of words that best match this string leads to sub-optimal performance. In practice, it is always better to use the technique described here as the "standard" approach, in which the recogniser uses a network that permits it to decode only legal words. However, the confusion modelling approach adopted here requires that a phoneme string is decoded initially. We were curious to see how well a system that used the method of decoding a phoneme string and then finding the best matching sequence of words that match this string (without any confusion modelling) would work. This would give us a baseline against which to measure the gain introduced by adding confusion modelling. Hence we built a system that decoded the best phoneme string and then identified the best-matching word by using DP between this string and the phoneme transcriptions of all the words in the vocabulary (no confusion modelling was used here).

4.3. Identity Confusion Matrix (ICM)

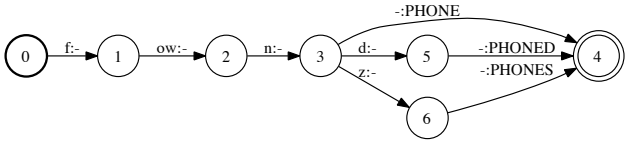
Training the weights for the confusion transducer model (introduced as C in section 3.1) is the most challenging task in a WFST cascade. Our first experiment used the identity matrix as a confusion model, i.e. we pretend that the recogniser is perfect and there are no confusions modelled. To avoid $-\infty$ log probabilities on off-diagonal elements, we add a small probability mass to every element in the identity matrix.



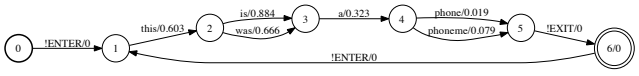
(a) P^* represents the n-best noisy transcription produced by the phoneme recogniser. Transcriptions are warped together to form a single transducer with alternative paths representing the n-best options. For this example, the 2-best transcriptions were chosen and represented as a WFST: */f/ /ow/ /n/* and */l/ /ow/ /m/*. These can be combined to form a total of four possible phone sequences.



(b) C represents elements in the confusion matrix where row i is the input and column j is the transitioned output. The element from the probability confusion matrix is converted into a negative log weighting for the WFST.



(c) D provides the phoneme-to-word mapping, restricting only valid sequences of phonemes to valid words. In this example, the vocabulary consists of three words: PHONE, PHONED and PHONES. In the transducer cascade, the output from the confusion matrix is restricted to provide only valid words using this dictionary model.



(d) Finally, M represents the sentence-level language model - mapping sequences of words produced by the dictionary to valid sentences. For the purposes of this work on isolated words, the M transducer is omitted from the cascade.

Figure 3: A visual representation of the WFST cascade used to recognise a simple example sentence. (a) represents the input noisy string, (b) models the confusion patterns, (c) provides a strict model to force phone sequences into words and (d) maps the output word sequences to valid sentences.

4.4. Confusion Matrix derived from the Standard Approach (SCM)

We next used a confusion-matrix produced by using the output from the standard approach. DP was used to align the phoneme string corresponding to each output recognised word with the phoneme string corresponding to the correct word, and the phoneme alignment pairs were counted. These counts were then converted to probabilities by normalising across the rows.

4.5. Use of timing information to estimate the Confusion Matrix (TCM)

When alignments produced by the above technique were analysed, it was found that there were many cases where the purely symbolic alignment used here was highly inaccurate because the time-registration of the aligned phoneme pairs was very different. We therefore attempted to introduce some timing information into the estimation of confusions so that only phoneme pairs that occurred at approximately the same point in time would be regarded as genuine confusions: other alignments would be disregarded. Each recognised phoneme sequence produced by the phoneme recogniser was aligned to the corresponding ground-truth transcription using DP. The "offset" of an aligned phoneme pair is defined as the absolute timing difference between the central point of the reference phoneme and the central point of the aligned output phoneme (both sequences contain information about the time-registration of each phoneme in the sequence). Hence we were able to produce a distribution of offsets for each label phone. Using the absolute mean offset values for a particular ground-truth phoneme and a specified window (typically between 0.5 and 3 standard deviations from the mean offset), each phoneme alignment pair is either accepted or rejected as a genuine confusion based on whether it falls inside this acceptance window. A further constraint was imposed on the acceptance criteria to remove any confusions between vowels and consonants. Figure 4 demonstrates the use of the offset window and the system for accepting phoneme confusions based on the timing information.

For this experiment, the data were split into three sets; training (four repetitions), confusion-matrix estimation (one repetition) and testing (one repetition).

4.6. Smoothing the Confusion Matrix

Element C_{ij} of a confusion-matrix is an estimate of the probability that phoneme p_i will be confused with phoneme p_j. We also include an extra row to account for insertions of phonemes and an extra column to account for deletions. In practice, the diagonal elements ($C_{i,i}$) usually dominate a row, and it is necessary to re-distribute some of this "probability mass" from the diagonal to off-diagonal elements of the row in order to increase the like-

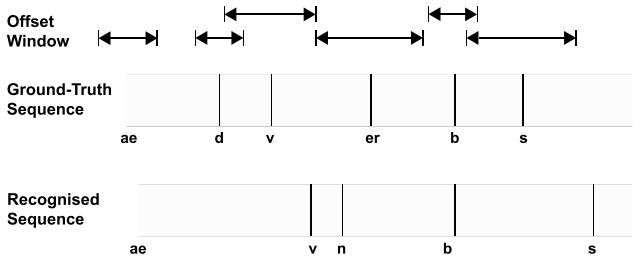


Figure 4: Alignment of the ground-truth transcription and noisy output from HMM recognition for the word *adverbs*. The offset window is characterised with a mean (μ) and standard deviation (σ) computed by observing offsets between ground-truth and recognised sequences at training. Each recognised phone is subject to a decision based upon whether it falls within the offset window (defined as $\pm x$ standard deviations away from the mean offset).

lihood of confusions appearing in new hypotheses. We refer to this as "smoothing" and we have used use three smoothing methods.

4.6.1. Base Smoothing (TCM-base)

In base smoothing, we simply re-assign a fixed percentage of the diagonal probability to all the other classes on the same row. In this work, re-distributing about 10% of the diagonal provided the best performance.

4.6.2. Exponential Smoothing (TCM-exp)

Exponential smoothing gives us more control over the redistribution by using the parameter α as shown in equation 2.

$$z_{i,j} = \frac{e^{-\alpha C_{ij}}}{\sum_k e^{-\alpha C_{kj}}} \qquad 0.5 < \alpha < 5, \qquad (2)$$

where C_{ij} is the value of the original confusion matrix element, $z_{i,j}$ is the resulting smoothed element, and α is a parameter that controls the degree of smoothing. As $\alpha \to 0$, the probability mass is equally distributed over a row, and as $\alpha \to \infty$, the mass is concentrated in the highest element.

4.6.3. Base Smoothing using visemic classes (TCM-base-vis)

This method uses base smoothing (described as 4.6.1) but smoothes based upon prior knowledge of confusions in visual speech. Firstly, we re-distribute an amount from the diagonal count across the row *except* the phones within its particular visemic class. For these phones, the new diagonal count is split evenly and re-distributed to these classes. For example, in the mapping defined in [14], the phonemes */b/, /p/* and */m/* are in the same visemic class. Therefore, we distribute 10% of the diagonal count for phoneme */b/* across the row, ignoring the phonemes in the same class (*/b/, /p/* and */m/*). We then distribute the left-over diagonal count evenly between these three classes, making them equally likely.

The viseme groupings that were used are taken from previous work on phoneme-to-viseme mappings [14].

5. Results

Table 1 compares the results of the experiments on isolated word recognition. The Baseline 1 result (59.95%), using the standard approach, is quite good for a vocabulary of 211 words, although it should be noted that this is a speaker-dependent system. However, if we use the sub-optimal approach of decoding a phoneme string and then finding the best-matching word to this string (Baseline 2), accuracy drops dramatically to 20.16%. Turning to the experiments on our proposed method, using an identity confusion-matrix (i.e. no confusion modelling) gives an accuracy of 35.36%. It is interesting that this is considerably higher than Baseline 2. This is because the confusion-matrix must have very small confusion probabilities off the diagonal in order for any legal word to be decoded by the D transducer, which enables a rich set of candidate words. However, DP uses a cost function that finds only the closest match.

If the confusion-matrix is estimated from the output of the baseline recogniser, accuracy falls to 21.42%. This result suggests that many of the alignments are not genuine confusions, but are in fact an artefact of the recognition process, something that was commented upon in section 4.5. Using timing information improves accuracy hugely with base smoothing giving a better result (49.70%) than exponential smoothing (42.68%). However, the best result from the WFSTs is still about 10% lower than the "standard" approach.

6. Discussion

We have described a new approach to automatic lipreading in which a model of the confusions in the visual signal is used to correct the errors from a visual phoneme recogniser. We have shown that this method is effective, in that when it is compared with a similar system that does not use confusion modelling, word accuracy increases from 20.16% to 49.7%. However, the accuracy produced by this method is still lower than that produced by a "standard" system that uses a constrained network to decode (59.9%). Despite this initial result, we remain convinced that this technique has promise. Our future work will be mostly focused on improving the confusion-modelling. Incorporating the timing information into confusion-modelling gave a very large gain in performance, but it is clear that the confusion matrix still contains noisy entries, and we are working on the use of

Technique	Word Accuracy (std. deviation)
Standard system, as shown in figure 1 (Baseline 1)	59.95% (4.19)
Phone decoding followed by string-matching (Baseline 2)	20.16% (1.43)
WFSTs with identity confusion matrix (ICM)	35.36% (2.27)
WFSTs with confusion matrix produced by standard approach (SCM)	21.42% (3.30)
WFSTs with timing confusion matrix and base smoothing (TCM-base)	49.70% (1.60)
WFSTs with timing confusion matrix and exponential smoothing (TCM-exp)	42.68% (2.30)
WFSTs with timing confusion matrix and base smoothing using visemic classes (TCM-base-vis)	46.58% (3.09)

Table 1: Word-level recognition results (word accuracy and standard deviation)

(a) iterative techniques to minimise error in the confusion model and (b) building more reliable estimates based on a level of confidence that "islands" of the decoded signal were recognised correctly. In the task described here, the amount of data available for confusion-matrix estimation was very small and the task was particularly simple for a standard recogniser. We have recently completed the recording of a much larger dataset of continuous speech which should alleviate both of these problems and enable us to develop and test our technique under more realistic conditions.

7. References

[1] Stephen Cox, Richard Harvey annd Yuxuan Lan, Jacob L Newman, and Barry-John Theobald, "The challenge of multispeaker lip-reading," *In Proceedings of the International Conference on Auditory-Visual Speech Processing*, pp. 179 – 184, 2008.

[2] P.L. Jackson, "The theoretical minimal unit for visual speech perception: Visemes and coarticulation.," *The Volta Review*, 1988.

[3] P. Hansen and M. Coleman, "Speechreading skill and visual movement sensitivity are related in deaf speechreaders," *Perception*, vol. 34, pp. 205–216, 2005.

[4] Santiago Omar Cabellero Morales, *Error Modelling Techniques to Improve Automatic Recognition of Dysarthric Speech*, Ph.D. thesis, School of Computing Sciences, University of East Anglia, May 2009.

[5] Omar Caballero Morales and Stephen Cox, "Application of weighted finite-state transducers to improve recognition accuracy for dysarthric speech," *Proceedings of 11th International Conference on Spoken Language Processing*, September 2008.

[6] Yuxuan Lan, Barry-John Theobald, Richard Harvey, Eng-Jon Ong, and Richard Bowden, "Comparing visual features for lipreading," *Proceedings of the International Conference on Auditory-Visual Speech Processing*, pp. 102–106, 2009.

[7] Tim Cootes, C J Edwards, and C J Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681–685, 2001.

[8] Mehryar Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 20, no. 1, 1994.

[9] Mehryar Mohri, "Weighted finite-state transducer algorithms: An overview," Tech. Rep., AT and T Research - Shannon Laboratory, 2004.

[10] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Proceedings of the ISCA Tutorial and Research Workshop, Automatic Speech Recognition: Challenges for the new Millenium*, September 2000.

[11] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: A general and efficient weighted finite-state transducer library," *Implementation and Application of Automata*, pp. 11–23, 2007.

[12] Iain Matthews, Tim Cootes, J Andrew Bangham, Stephen Cox, and Richard Harvey, "Extraction of visual features for lipreading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 198–213, 2002.

[13] Steve Young, G Evenmann, D Kershaw, G Moore, J Odell, D Ollason, V Valtchev, and P Woodland, *The HTK Book (version 3.2.1)*, 2002.

[14] C Fisher, "Confusions among visually perceived consonants," *Journal of Speech and Hearing Research*, vol. 11, pp. 796 – 804, 1968.