interspeech 2008 incorporating SST 08

Application of Weighted Finite-State Transducers to Improve Recognition Accuracy for Dysarthric Speech

Omar Caballero Morales and Stephen Cox

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

S.Caballero-morales@uea.ac.uk, sjc@cmp.uea.ac.uk

Abstract

Standard speaker adaptation algorithms perform poorly on dysarthric speech because of the limited phonemic repertoire of dysarthric speakers. In a previous paper, we proposed the use of "metamodels" to correct dysarthric speech. Here, we report on an improved technique that makes use of a cascade of Weighted Finite-State Transducers (WFSTs) at the confusion-matrix, word and language levels. This approach outperforms both standard MLLR and metamodels.

Index Terms: weighted finite-state transducers, speech recognition accuracy, dysarthric speech, speaker adaptation

10.21437/Interspeech.2008-485

1. Introduction

Speech from dysarthric speakers often has low or very low intelligibility: the disorder means that the speaker has poor control over his or her articulators, and the result is that their speech is often less than 100% intelligible, depending on the degree of dysarthria. The condition is usually associated with a restricted phonemic repertoire, and hence, when using automatic speech recognisers, a high number of substitutions, deletions and insertions of phonemes is observed [8]. In previous work [3], we described a technique for incorporating a model of a dysarthric speaker's confusion matrix into the ASR process in such a way as to increase recognition accuracy. In this work, we extend the technique to a more powerful model that uses weighted finite-state transducers, and we demonstrate increased performance.

Most speaker adaptation algorithms are based on the principle that it is possible to apply a set of transformations to the parameters of a set of acoustic models of an "average" voice to move them closer to the voice of an individual. Whilst this has been shown to be successful for normal speakers, it may be less successful in cases where the phoneme uttered is not the one that was intended but is substituted by a different phoneme or phonemes, as often happens in dysarthric speech. In this situation, we argue that a more effective approach is to combine a model of the substitutions likely to have been made by the speaker with a language model to infer what was said. We suppose that the speaker wished to utter a word sequence W_{in} which can be transcribed using a dictionary into the phoneme sequence S_{in}. The sequence of phones decoded by the speech recogniser is S_{out}, and we describe here a cascade of Weighted Finite-State Transducers (WFSTs) that estimate W_{in} from S_{out}. The transducers model the speaker's phonetic confusions, the mapping from phonemes to words, and the mapping from words to a word sequence described by a grammar.

In [3], we proposed using transducers that we termed "metamodels". A metamodel is a discrete hidden Markov model (HMM) of a particular phone that models in a stochastic manner the pattern of substitutions, deletions and insertions made by a particular dysarthric speaker when the intended phone is presented to the recogniser. By forming a network of metamodels that represent legal sequences of words, the intended word sequence W_{in} can be estimated from the noisy string supplied by a phone recogniser S_{out}. The metamodels are trained on pairs of transcriptions of correct and recognised phone strings. Although we reported some success using metamodels for this task, they suffered from two disadvantages:

1. The models had a particular problem dealing with deletions. If the metamodel network defining a legal sequence of words is defined in such a way that it is possible to traverse it by "skipping" every metamodel, the decoding algorithm fails because it is possible to traverse the complete network of HMMs without absorbing a single input symbol. We attempted to remedy this problem by adding an extra "deletion" symbol, but as this symbol could potentially substitute every single phoneme in the network, it led to an explosion in the size of the dictionary, which was unsatisfactory.
2. The metamodels were unable to model specific phone sequences that were output in response to individual phone inputs. They were capable of outputting sequences, but the Markov property ensured that these sequences were conditionally independent, and so specific sequences could not be modelled.

WFSTs [7] are an attractive alternative to metamodels for this task. A WFST can be regarded as a network of automata, each of which accepts an input symbol and outputs one of a finite set of outputs, each of which has an associated probability. The outputs are drawn (in this case) from the same alphabet as the input symbols and can be single symbols, sequences of symbols or the deletion symbol ε. The automata are linked by a set (typically sparse) of arcs and there is a probability associated with each arc. The usage proposed here complements and extends the work presented in [6], in which WFSTs were used to correct phone recognition errors. Here, we extend the technique to convert noisy phone strings into word sequences.

2. Structure of the WFST network

As shown in, for instance, [7, 4], the speech recognition process can be realised as a cascade of WFSTs. In our study, we define the following transducers:

1. S_{out}, the phoneme sequence to be decoded into words W_{in}^{*}
2. C, the confusion matrix transducer, which models the probabilities of phoneme insertions, deletions and substitutions.

Accepted after peer review of full paper

3. D, the dictionary transducer, which maps sequences of decoded phonemes from $S_{out} \circ C$ into legal words.
4. G, the language model transducer, which allows valid sequences of words from D.

Thus, the process of estimating the most probable sequence of words W^*_{in} given S_{out} can be expressed as:

$$W^*_{in} = \mathcal{T}^*(S_{out} \circ C \circ D \circ G). \quad (1)$$

where $\mathcal{T}^*$ denotes the operation of finding the most likely path through a transducer and $\circ$ denotes composition of transducers [7]. Details of each transducer used will be presented in the following sections.

2.1. Confusion Matrix Transducer C

In this section, we describe the formation of the confusion matrix transducer C. Defining p^i_{out} as the i'th phone in S_{out} and p^j_{in} as the j'th phone in S_{in}, $Pr(p^j_{in}|p^i_{out})$ is estimated from the speaker's confusion matrix, which is obtained from an accurate alignment of many sequences of S_{in} and S_{out} [3].

However, C can also map multiple phone insertions and deletions. Consider Table 1, which shows an alignment from one of our experiments. The top row of phone symbols represents the transcription of the word sequence and the bottom row the output from the phone recogniser. It can be seen that the phoneme sequence *b aa* is deleted after *ax*, and this can be represented in the transducer as a multiple substitution/insertion: $ax \rightarrow ax\ b\ aa$. Similarly the insertion of *ng dh* after *ih* is modelled as $ih\ ng\ dh \rightarrow ih$. The probabilities of these multiple substitutions/insertions/deletions are estimated again by counting. In cases where a multiple insertion or deletion is made of the form $A \rightarrow B\ C$, the appropriate fraction of the unigram probability mass $\Pr(A \rightarrow B)$ is subtracted and given to the probability $\Pr(A \rightarrow B\ C)$, and the same process is used for higher order insertions or deletions.

A fragment of the confusion transducer that represents the alignment of Table 1 is presented in Figure 1. For convenience, the weight for each confusion in the transducer is estimated as $-log(\Pr(p^j_{in}|p^i_{out}))$. In practice, we have found it convenient to build an initial set of transducers directly from the speaker's "unigram" confusion matrix, which is estimated using each transcription/output alignment pair available from that speaker, and then to add extra transducers that represent multiple substitution/insertion/deletions. The complete set of transducers are then *determinized* and *minimized*, as described in [7]. The result of these operations is a single transducer for the speaker.

One problem encountered when limited training data is available from speakers is that some phonemes are never decoded during the training phase, and therefore it is not possible to make any estimate of $Pr(p^j_{in}|p^i_{out})$. This is shown in Figure 2, which shows a confusion matrix estimated from a single talker. Note that the columns are the stimulus and the rows are the response in this matrix, and so blank rows are phonemes that have never been decoded. We used two techniques to smooth the missing probabilities.

2.1.1. Base Smoothing

It is essential to have a non-zero value for every diagonal element of a confusion matrix to enable the decoding process to work using an arbitrary language model. One possibility is to set all diagonal elements for which no data exists to 1.0, i.e, to assume that the associated phone is always correctly decoded. However, if the estimate of the overall probability of error of the recogniser on this speaker is p, a more robust estimate is to set any unseen diagonal elements to p, and we begin by doing this. We then need to decide how to assign non-diagonal probabilities for unseen confusions. We do this by "stealing" a small proportion of the probability mass on the diagonal and re-distributing it along the associated row. This is equivalent to assigning a proportion of the probability of correctly decoded phonemes to as yet unseen confusions. The proportion of the diagonal probability that is used to estimate these unseen confusions depends on the amount of data from the speaker: clearly, as the data increases, the confusion probability estimates become more accurate and it is not appropriate to use a large proportion. Some experimentation on our data revealed that re-distributing approximately 20% of the diagonal probability to unseen confusions worked well.

Figure 1: Example of the Confusion Matrix Transducer C.

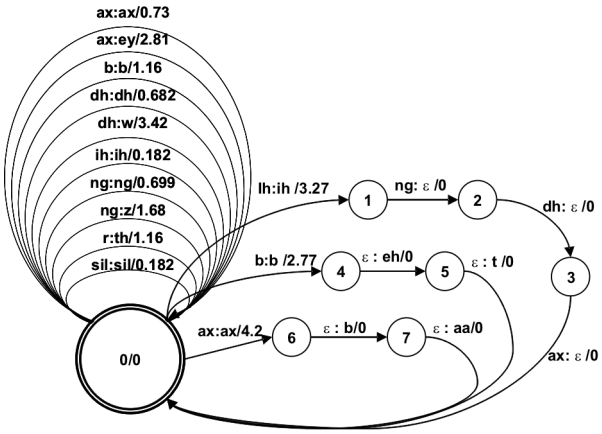
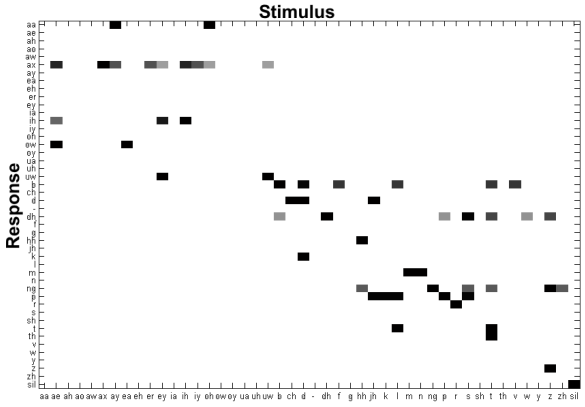


Figure 2: Confusion Matrix for C



2.1.2. SI Smoothing

The base smoothing described in section 2.1.1 could be regarded as "speaker dependent" in that it uses the (sparse) confusion estimates made from the speaker's own data to smooth the

Table 1: *Alignment of transcription S_{in} and recognised output S_{out}.*

TR:	sil	ax	b	aa	th	ih		ax	z	w	ey		ih	ng	dh	ax	b	eh	t	sil
REC:	sil	ax			r	ih	ng	dh	ax	ng	dh	ax	l	ih	ng	dh	ax	b		sil

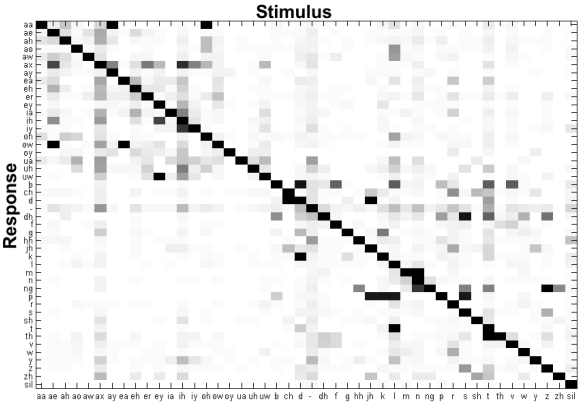
unseen confusions. However, these estimates are likely to be noisy, so we add another layer of smoothing using the speaker-independent (SI) confusion-matrix whose elements are well-estimated from the training-data described in Section 3. The influence of this confusion-matrix on the speaker-dependent matrix is controlled by a mixing factor λ. Defining the elements of the SI confusion matrix as q^j_{in} and q^i_{out}, the resulting joint confusion-matrix can be expressed as:

$$C_{joint} = \lambda SI + (1-\lambda)SD. \quad (2)$$

$$C_{joint} = \lambda \Pr(q^j_{in}|q^i_{out}) + (1-\lambda)\Pr(p^j_{in}|p^i_{out}). \quad (3)$$

The effect of both the base smoothing and the SI smoothing can be seen by comparing Figures 2 and 3.

Figure 3: SI Smoothing of C, with $\lambda = 0.25$.

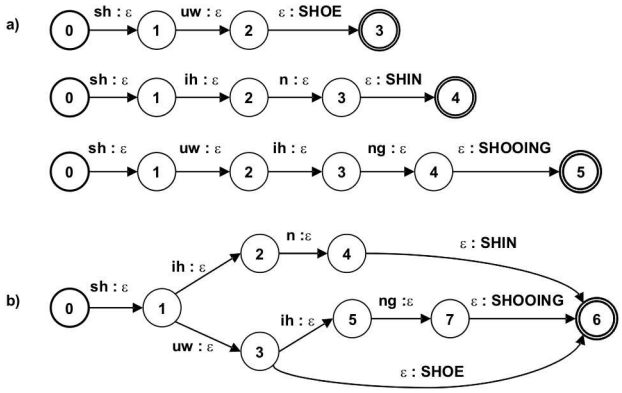


2.2. Dictionary *D* and Language Model *G* Transducers

The transducer D maps sequences of phonemes into valid words. Although other work has investigated the possibility of using WFSTs to model pronunciation in this component [1], in our study the pronunciation modelling is done by the transducer C. A small fragment of the dictionary entries is shown in Figure 4 a), where each sequence of phonemes that forms a word is listed as an FST. The minimized union of all these word entries is shown in Figure 4 b). The single and multiple pronunciations of each word were taken from the British English BEEP pronouncing dictionary [9].

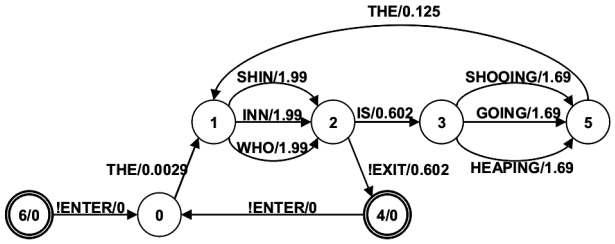
The language model transducer consisted of a word bigram, as used in our previous work [3], but now represented as a WFST. HLStats [9] was used to estimate these bigrams and a script was designed to do the convertion into WFST format. A fragment of the word bigram FST G is shown in Figure 5. The NEMOURS sentences (see Section 3) are nonsense phrases that have a simple syntax of the form "the X is Y the Z", where X and Z are usually nouns and Y is a verb in present participle form [2] (for instance, the phrases "The shin is going the who", "The inn is

Figure 4: Example of the Dictionary Transducer *D*.



heaping the shin", etc.). The network of Figure 5 allows sequences of this kind to be recognized explicitly, but an arbitrary word bigram grammar can be represented using one of these transducers.

Figure 5: Example of the Language Model Transducer *G*.



All three transducers used in these experiments were determinized and minimized in order to make execution more efficient.

3. Speech Data, Recogniser, and Results

The Wall Street Journal (WSJ) database was used to build the SI speech recogniser. The training set consisted of the WSJ data from 92 speakers in set si_tr. This was used to construct 45 monophone acoustic models. The models were a standard three state left-right topology with eight mixture components per state. The front-end used 12 MFCCs plus energy plus delta and acceleration coefficients. From this system, the SI confusion-matrix described in Section 2.1.2 was estimated using a phoneme-bigram model with a grammar scale factor of 10. In order to keep both systems independent for confusion-matrix estimation, this language model was estimated from the corresponding WSJ speech transcriptions, and not from the

dysarthric database.

The dysarthric speech data was provided by the NEMOURS database [2]. We used the data from 10 speakers (74 sentences per speaker) with varying degrees of dysarthria. Note that although each of the 740 sentences in this set is different, the vocabulary is shared. A subset of the first 34 sentences from each speaker was used for confusion-matrix estimation, and the remaining 40 were used for testing after adaptation [3].

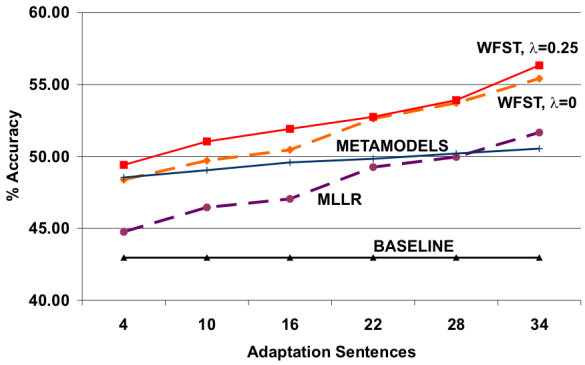
The HTK package [9] was used throughout for the experiments. The FSM Library [7] from AT&T was used for the experiments with WFSTs. For comparison purposes, a standard speaker adaptation technique, MLLR (maximum likelihood linear regression [5] [9]) was applied, always using the same set of adaptation sentences as were used for estimating the metamodels in our previous work [3] and training of the transducer C. The word language model for all systems was a bigram model estimated from the (pooled) 74 sentences provided by each speaker (113 different words).

In all the experiments reported here, MLLR adaptation was performed using different numbers of adaptation utterances. The adapted acoustic models were then used in the phone recogniser which supplied the output phone string for both the metamodels and the WFSTs. This recogniser used a bigram phoneme language model estimated from the training-data transcriptions of each speaker. The results marked "MLLR" in Figure 6 are for a recogniser that used the adapted acoustic models but with a word-level language model. This language model was the same for all three techniques used.

3.1. Results on dysarthric speakers

Figure 6 shows the mean word accuracies (i.e. (Hits −Insertions)/ Nwords) across all the NEMOURS database speakers for different amounts of adaptation data and using different decoding techniques. The baseline is the performance with no adaptation (horizontal line). The figure shows clearly the gain in performance given by the WFSTs over both MLLR and our previous technique (metamodels).

Figure 6: Mean across all dysarthric speakers: comparison of % accuracy for different techniques



In previous experiments using the metamodels on MLLR-adapted models, we found that the word recognition accuracy was much higher than that provided by MLLR alone when the amount of training data was small (four sentences). However, this advantage decreased as the number of sentences reached the maximum (34) [3]. Using WFSTs on MLLR-adapted models, performance on four utterances of adaptation data is slightly higher compared with metamodels, and continues to increase as the amount of training-data increases. The SI Smoothing increases the WFSTs performance over the Base Smoothing when the training data is small (four to 16 sentences): thereafter, the two smoothing schemes do not give significantly different performance.

Figure 6 shows results for only two values of λ: $\lambda = 0$ (Base Smoothing only) and SI Smoothing with $\lambda = 0.25$. Performance decreases as λ increases above 0.25. This is probably because the confusion pattern of dysarthric speech is different from normal speech, and higher values of λ move the estimated confusion matrices too much towards normal speech confusion patterns.

4. Discussion and Future Work

In this paper, we have shown how a set of weighted finite state transducers (WFSTs) at the confusion-matrix, word and language levels can be cascaded in order to correct errors made by dysarthric speakers, whose pattern of errors is markedly different from normal speakers because of their condition. The results obtained using this technique are significantly better than those obtained using the standard speaker adaptation technique, MLLR, and also better than our previous approach using metamodels. Future work will concentrate on integrating better the confusion-matrix transducer with the speech recogniser and also on robust estimation of confusion matrices from sparse data.

5. References

[1] Bodenstab, Nathan and Fanty, Mark. Multi-pass pronunciation adaptation. *ICASSP 2007*, 2007.

[2] Bunnel, H.T. and Polikoff, J.B. The nemours database of dysarthric speech. *Proceedings of ICSLP*, 1996.

[3] Caballero, Omar and Cox, Stephen. Modelling confusion matrices to improve speech recognition accuracy, with an application to dysarthric speech. *Interspeech 2007*, 2007.

[4] Fosler-Lussier, Eric, Amdal, Ingunn, and Jeff Kuo, Hong-Kwang. On the road to improved lexical confusability metrics. *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation* ., 2002.

[5] Leggetter, C.J. and Woodland, P.C. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9(2):171–85, 1995.

[6] Levit, M., Alshawi, H., Gorin, A., and Nöth, E. Context-Sensitive Evaluation and Correction of Phone Recognition Output. In *Proc. Eighth European Conference on Speech Communication and Technology*. ISCA, 2003.

[7] Mohri, M., Pereira, F.C.N., and Riley, M. Weighted finite-state transducers in speech recognition. *Computer Speech and Language.*, 2002.

[8] Rosen, K.and Yampolsky, S. Automatic speech recognition and a review of its functioning with dysarthric speech. *Augmentative and Alternative Communication*, 16:48–60, 2000.

[9] Young, Steve and Woodland, Phil. *The HTK Book (for HTK Version 3.3)*. Cambridge University Engineering Department, 2005.