HIDDEN-ARTICULATOR MARKOV MODELS FOR SPEECH RECOGNITION

Matthew Richardson, Jeff Bilmes and Chris Diorio

University of Washington
{mattr@cs, bilmes@ee, diorio@cs}.washington.edu

Correspondence:
Matthew Richardson
Department of Computer Science
Box 352350
University of Washington
Seattle, WA 98195-2350
U.S.A.

mattr@cs.washington.edu
(206)616-1842

Number of pages: 28 + title page + this page
Number of Tables: 10 + Appendix
Number of Figures: 8

Keywords: speech recognition, articulatory models, noise robustness, factorial HMM

ABSTRACT

Most existing automatic speech recognition systems today do not explicitly use knowledge about human speech production. We show that the incorporation of articulatory knowledge into these systems is a promising direction for speech recognition, with the potential for lower error rates and more robust performance. To this end, we introduce the Hidden-Articulator Markov Model (HAMM), a model which directly integrates articulatory information into speech recognition.

The HAMM is an extension of the articulatory-feature model introduced by Erler in 1996. We extend the model by using diphone units, developing a new technique for model initialization, and constructing a novel articulatory feature mapping. We also introduce a method to decrease the number of parameters, making the HAMM comparable in size to standard HMMs. We demonstrate that the HAMM can reasonably predict the movement of articulators, which results in a decreased word error rate. The articulatory knowledge also proves useful in noisy acoustic conditions. When combined with a standard model, the HAMM reduces word error rate 28-35% relative to the standard model alone.

1 INTRODUCTION

Hidden Markov Models (HMMs) are the most successful technique used in automatic speech-recognition (ASR) systems. At the hidden level, however, ASR systems most commonly represent only phonetic information about the underlying speech signal. Although there has been much success using this methodology, the approach does not explicitly incorporate knowledge of certain important aspects of human speech production.

We know, for example, that speech is formed by the glottal excitement of a human vocal tract comprised of articulators, which shape and modify the sound in complex ways. These articulators, being part of a physical system, are limited by certain physical constraints, both statically and temporally. Our hypothesis, for which we find support in this paper, is that we can improve ASR by using a statistical model with characteristics and constraints that are analogous to the true human articulatory system.

Explicitly incorporating articulatory information into an ASR system provides a number of potential advantages. For example, an articulatory system should be better able to predict coarticulatory effects. This is because coarticulation is due to physical limitations and anticipatory and residual energy-saving shortcuts in articulator movement (Hardcastle 1999). Furthermore, by modeling articulators explicitly, an ASR system can exploit the inherent asynchrony that exists among (quasi-dependent) articulatory features. This, in turn, might more accurately model the production of speech (Deng 1994). Although speech production does not necessarily have a strong influence on speech recognition, our belief is that exploring articulatory-based ASR in tandem with other statistical methodologies will ultimately lead to better ASR technology.

Finally, articulatory models allow using articulatory states in multiple contexts. Most speech recognition systems are based on phoneme recognition, which allows them to share phoneme training across multiple contexts (i.e. the same phone appearing in different words). Similarly, the articulatory model is even finer-grained than phonemes, allowing the same articulatory state to be used across multiple contexts (i.e. the same mouth position being used as part of the production of two different phonemes).

There has been much interest in incorporating articulatory knowledge into speech recognition. In (Kirchhoff 1998) Kirchhoff demonstrates a system that uses artificial neural networks to estimate articulatory features from acoustic features. When used in combination with an acoustic-based HMM, the system achieves a lower word error rate in both clean and noisy speech. Frankel (Frankel 2000) (Frankel 2001) also uses neural networks to estimate articulatory motion, which is then incorporated into a speech recognition system. Some early work on incorporating articulatory knowledge can be found in (Schmidbauer 1989), (Blomberg 1991), (Elenius 1992), and (Eide 1993). Other cases of articulatory based speech recognition are included in the following: (Blackburn 1995), (Deng 1997b), (Rose 1997), and (Picone, 1999).

One well-known difficulty with articulatory based systems is the inverse mapping problem, which is that many different articulatory configurations can produce an identical acoustic realization; this difficulty is commonly referred to as the "many-to-one" problem. Although this limits its effectiveness, inverse mapping may still be used to provide additional constraints to an ASR system, increasing the system’s accuracy. A detailed discussion of the inverse mapping problem can be found in (Bailly 1992).

We incorporate articulatory information into an ASR system in a number of ways. We extend the articulatory feature model introduced by Erler (Erler 1996) by using diphone units, developing a new technique for model initialization, and constructing a novel articulatory feature mapping. We provide the model with articulatory information in the form of mappings from phonemes to articulatory configurations, and in static and temporal constraints designed to inform the system about the limitations of the human vocal apparatus. The resulting model yields a reduction in word error rate and estimates articulator motion.

We organize the rest of the paper as follows: Section 2 presents our model in detail, describing the phonetic mapping and constraints we used. Section 3 describes how we initialize and train the model.

Section 4 presents experimental results showing that, when the model uses articulatory knowledge, improvements in speech recognition performance are obtained. We also show in Section 4 that the articulatory sequences estimated by the model correlate well with real-world articulatory sequences.

2 THE MODEL

To incorporate articulatory knowledge into speech recognition, we use a Hidden-Articulator Markov Model (HAMM). A HAMM is simply an HMM in which each state represents an articulatory configuration. The state transition matrix is governed by constraints on articulator motion. Therefore, this model makes the assumption that the probability distribution of articulatory features is determined by the previous articulatory configuration, and is independent of any earlier articulatory configuration.

The Hidden-Articulator Markov Model (HAMM) is based on the articulatory feature model presented in (Erler 1996). We introduced the HAMM in (Richardson 2000a) and extended it in (Richardson 2000b). In a HAMM, each articulator, i, can be in one of M_i positions. An *articulatory configuration* is an N-element vector $C=\{c_1,c_2,\ldots,c_N\}$, where c_i is an integer $0 \le c_i < M_i$ and N is the number of articulators in the model.

We can cast the HAMM as a *factorial HMM* (Saul 1999), with additional dependencies between separate Markov chains (see Figure 1). The dependency from one time slice to another is governed by the *dynamic constraints* and the dependencies within a time slice are governed by the *static constraints* (see Section 2.2). Factorial HMMs have been applied to speech recognition (Logan 1998) but without the use of an articulatory feature space. The HAMM is an instance of a more general family of models called *dynamic Bayesian networks*

(Ghahramani 98), which, in turn, are a specific case of *graphical models* (Lauritzen 96). (Zweig 98) is an excellent example of using dynamic Bayesian networks for speech recognition that shows how they can be modified to allow the addition of information such as articulatory context. There are standard algorithms for inference and training on graphical models, but we chose to implement the HAMM as an HMM with a large state space which is the Cartesian product of the components; each state is associated with an articulatory configuration. This approach allows us to use the comparatively efficient standard HMM algorithms.

There are many potential advantages of a HAMM over a traditional HMM for speech recognition. The HAMM has prior knowledge about speech production, incorporated via its state space, transition matrices, and phoneme-to-articulator mappings. By using a representation that has a physical basis, we can more easily incorporate knowledge such as co-articulation effects. For example, the production of the English phoneme /k/ depends on the tongue position of the subsequent vowel; the tongue is further forward when followed by a front vowel ("key"), and is further back when followed by a back vowel ("caw") (Hardcastle 1999). Our model allows explicit representation of this knowledge, in this case by adjusting the forward/backward position of the tongue when mapping the phoneme /k/ into the articulatory space, based on the placement of the subsequent vowel. We have not yet incorporated coarticulation knowledge into our model, but this shows promise for future work.

The following subsections provide more detail about how we construct the HAMM, and how we use mappings and constraints to provide the model with articulatory knowledge.

2.1 Phoneme Mapping

To use the HAMM, we must first define how a spoken word traverses through the articulatory state space. We consider a word to be defined by a sequence of articulator targets; in producing the word, the mouth traces out a continuous path through the articulatory state space, reaching each target in sequence. To map words to a sequence of articulator configuration targets, we make the simplifying assumption that we can model words as a sequence of phonemes, each of which is mapped to one or more articulatory configurations.

Using Edwards (Edwards 1997) as a guide to phonetics and speech production, we devised an articulatory feature space that is described by eight features – the position of the jaw, the separation of the lips, positioning of the tongue, etc… (see Figure 2). Each feature can be in one of a number of discrete positions. For example, in our model we have quantized the separation of the lips (“Lip Sep”) into four possible positions, ranging from “closed” to “wide apart”. We manually examined each phoneme’s articulatory characteristics to determine the best mapping into our articulatory feature space. This mapping is given in the Appendix.

For some phonemes, an articulator may be in one of multiple configurations. In such a case, the phoneme is mapped into a vector of articulator ranges; each articulator can be in any of the positions specified by the range. For example, when pronouncing the phoneme /h/, we allow a lip separation of either “apart” or “wide apart”, but do not allow the lips to be “closed” or “slightly apart”.

Some phonemes require a specification of articulator motion rather than static positioning. This occurs with the stops (/t/, /b/, etc..) and diphthongs (such as the /ay/ in “bite”). In these cases, a phoneme is produced by the movement from one articulatory state to another. Thus, we constructed the model to allow phonemes to be mapped to a sequence of articulatory configurations.

Our model calculates on the state space formed by the Cartesian product of the articulatory state space (hence, each state in the model is a particular articulatory configuration). For the features we chose, this state space is enormous (over 25,000 states), resulting in slow runtimes and the potential for severe under-training. Thus, we reduce this space a prori by imposing both static and dynamic constraints on the set of possible hidden articulatory configurations; static constraints eliminate unlikely articulatory configurations, and dynamic constraints restrict the transitions between states. These are described further in the next section.

2.2 Constraints

The static constraints limit the possible set of articulatory configurations. They do this by disallowing unrealistic combinations of articulatory features. These constraints are described using the following rules:

1. If the lips are widely separated then don't allow rounded or wide lip width.
2. If the lips are closed then don't allow rounded or wide lip width.
3. If the jaw is lowered, don't allow the lips to be closed or almost closed.
4. If the tongue tip is near or is touching the alveolar ridge, then the tongue body must be mid-high or high, and the tongue body cannot be back or slightly back.

5. If the velic aperture is open then voicing must be on.
6. If the velic aperture is open then tongue cannot be forward or slightly forward.
7. The velic aperture may only be open in a given articulatory configuration X if there is a transition directly from X to a nasal phoneme articulatory configuration.

Some of these constraints, such as (1), (3), and (4), are physical constraints, imposed by the limitations of the articulation system. Other constraints, such as (2), disallow states that are physically possible but would not normally be used while speaking naturally in American English. This set of static constraints reduces the number of states in the HAMM from 25,600 to 6,676.

We also impose dynamic constraints on the model to prevent physically impossible articulatory movements. We only allow the model to contain a transition from some configuration C to some configuration D if $\forall i$: $-1 \leq d_i - c_i \leq 1$, where c_i is the (integer) position of articulator i in the articulatory configuration C. This imposes a continuity and maximum velocity constraint on the articulators whereby in one time step each articulator may move by at most one position[1].

2.3 Diphones

The basic unit in the HAMM is a diphone. To construct a diphone, we list the sequence of articulatory targets from the last target of the first phoneme to the last target of the second phoneme. In this way, a chain of diphones will properly sequence through each phonetic target (see Figure 3a). The states between the targets are filled in and allowable transitions are added.

We constrain the model so that the only allowable state vectors between any two target phoneme vectors, P and Q, are those C which satisfy:

$$\forall i: \min(\{p_i, q_i\}) \leq c_i \leq \max(\{p_i, q_i\})^2 \qquad (1)$$

Thus, in traversing from one target articulatory configuration to another, the model may only pass through states which fall "between" those target vectors.

For example, suppose for N=2 we are constructing a graph from phoneme P={ [3 2] } to Q={ [1 1] → [0 2] }. Then the resulting graph (assuming none of these states are removed by static constraints) is shown in Figure

3b. Note that we only allow transitions which move closer to the next target state (we also allow self-transitions, which are not shown in the figure). Also, because an articulation target could consist of a range of positions for some articulator, we take additional steps to prevent cycles in the transition graph by requiring that at least one of the articulators that changed position was originally outside of its target range.

Notice that the HAMM allows for asynchrony, whereby one articulator may move with or without other articulators moving, thus more accurately representing speech production. In addition, many different diphones may contain the same intermediate articulatory configuration. Since our acoustic probability distributions are dependent on the articulatory configuration, not the diphone using it, having the same intermediate configurations leads to a large amount of sharing between diphones.

3 TRAINING

We train our HAMM using the Baum-Welch algorithm. We construct an HMM for each diphone using the static and dynamic constraints from Section 2.2. We construct words by concatenating diphone models. For instance, the model for the word "meatball" is the concatenation of the diphone models /m/-/i/, /i/-/t/, /t/-/b/, /b/-/a/, /a/-/l/. Thus, the model learns transition probabilities on per-diphone basis.

To reduce the model size, we removed states that, during training, had very low state-occupation probabilities. Training reduced the number of parameters in the HAMM from 2 million to 522 thousand.

3.1 Initial Model Construction

Training requires an initial model, which is iteratively improved until it converges to a local optimum. The quality of the initial model can have a large effect on the performance of the trained model, and on its convergence. The states (articulatory configurations) in our model fall into two categories: (1) states which correspond to a phoneme, and (2) all other allowable states. There is no single obviously best way to initialize the parameters for states in category (2). We chose a simple interpolation method based on an assumption about the geometry of the articulatory states. We felt that this would be sufficient to produce sensible starting values, which is crucial for EM.

We used segmental k-means to determine an initial setting for the Gaussian parameters for states which fell into category (1) above. Each category (2) state was initialized by a weighted interpolation of the category (1) states. The weighting was given by the inverse Euclidean distance between the state being initialized, and the states from which we were interpolating (see Figure 4 for a diagram of this for a fictitious two-articulator system).

In equation (2) we show the desired probability distribution for the state being initialized. S is the set of all possible category (1) states, and w_i are inversely proportional to the Euclidean distance in our N-dimensional discrete articulatory feature space (where N=8 in our case).

$$p(x) = \sum_{i \in S} w_i N\left(x; \mu_i, \sigma_i^{\,2}\right) \text{ with } \sum_{i \in S} w_i = 1 \tag{2}$$

The mean and variance of equation (2) is given by:

$$\hat{\mu} = \sum_{i \in S} w_i \mu_i \quad \text{and} \quad \hat{\sigma}^2 = \left[\sum_{i \in S} w_i \left(\sigma_i^2 + \mu_i^2 \right) \right] - \hat{\mu}^2 \tag{3}$$

For category (2) states, we used a diagonal Gaussian with these means and variances. In the multi-component case, where each state is a mixture of Gaussian components, we do the same, using a random assignment of components from the states being interpolated to the component being initialized.

State transition probabilities were initially set to 0.9 for self-loops, with the remaining 0.1 probability evenly distributed among all outgoing transitions.

3.2 Untrained Diphones

Frequently, in speech recognition systems, untrained diphones (diphones which appear in the test set but not in the training set) are mapped to trained diphones using decision-tree based state tying (Young 1996). Rather than implementing this mapping, we depended on the shared nature of the articulatory models to predict untrained diphones. Different diphones represent different trajectories in a shared articulatory state space. Thus, an untrained diphone may still be considered trained as it traverses through articulatory states which have been trained as portions of other diphones. The only untrained portions of such a diphone are the state transition probabilities, and states which did not appear in *any* trained diphone. The values of state transition probabilities are known to be of less importance to word error rate than the means and variances in Gaussian mixture HMM systems, so we left them fixed to their initial values. States which did not appear in any trained diphone were removed from the model. In Figure 5, we show this diagrammatically. Suppose diphones B-E and M-A are trained. It is apparent that diphone M-E, although untrained, primarily uses states which have previously been trained (light gray circles). If there were no other trained diphones, then four states (dark gray circles) would be removed from the M-E diphone. Note that this can result in oddly shaped diphones, with missing “corners” or narrow transition graphs. An alternative would be to initialize the untrained states with the interpolation method described in Section 3.1.

We can see that the use of the articulatory knowledge allows us to construct models for diphones which did not exist in the training set. Though we chose to remove untrained states from the diphones, we could alternatively have constructed sensible Gaussian probability distributions for them using the interpolation method describe earlier. This is an example of why it is that articulatory models require less data, since traditional models would require the training data to contain all diphones which appear in the test set.

As an analogy, consider phoneme-based HMMs and word-based HMMs. To train a word-based HMM requires a training set in which every possible word is uttered at least once (and hopefully multiple times). It is nearly impossible to get such data, so ASR systems instead use phoneme based models. With a phoneme based model, the training set needs only to contain at least a few instances of each phoneme, a much simpler task. Phoneme-based speech recognition systems are able to recognize words which never appeared in the training set, a task that would be impossible for word-based recognizers.

In traditional phoneme-based systems, a diphone may only be modeled if it exists in the training set. By using a finer-grained model, the HAMM is able to model diphones which were not encountered in the training data. As a phoneme-based model is able to construct unseen words out of trained phonemes, the HAMM is able to construct unseen diphones out of trained articulatory states.

4 EXPERIMENTS AND RESULTS

We obtained speech recognition results using PHONEBOOK, a large-vocabulary, phonetically-rich, isolated-word, telephone-speech database (Pitrelli 1995). All data was represented using 12 MFCCs plus c_0 and deltas resulting in a 26 element feature vector sampled every 10ms. In the HAMM, each state used a mixture of two diagonal covariance Gaussians. While it is true that ASR systems typically use more mixtures, we chose to use two mixtures so that the number of parameters was somewhat more comparable to our phonetic HMM baseline.

Additionally, we generated two baseline models, *3state* and *4state*, which were standard left to right, diagonal Gaussian HMMs with 3 and 4 states per phoneme and with 16 and 24 mixtures per state respectively.

The training, development, and test sets were as defined in (Dupont 1997), and consisted of approximately 20000, 7300, and 6600, utterances, respectively. Test words did not occur in the training vocabulary, so test word models were constructed using diphones learned during training. Training was considered complete when the training data log-likelihood difference between successive iterations fell below 0.2%.

4.1 Comparison with Random

To verify that the HAMM uses the articulatory knowledge to its advantage, we compared its performance to that of a HAMM with no articulatory knowledge. To construct such a model, we used a random mapping of phonemes to articulatory features. To ensure a fair comparison, we used the same feature space, static constraints, and dynamic constraints that were introduced in Section 2.

We used two methods for producing random mappings. In the first, referred to as *arbitrary*, we simply selected a random value within the given feature range for all features across all phonemes. In the second, referred to as *permutation*, we randomly rearranged the original mapping. In other words, each phoneme was mapped in the same way as some randomly selected phoneme in the original mapping without duplication. Table 1 demonstrates the difference between the random mappings.

The arbitrary mapping was "more" random since it was drawn from a uniformly distributed state space. The permutation method produced a mapping that was still fairly random, yet retained the same distribution over features as the original mapping. For instance, in the original mapping, the *velic aperture* was *open* for only three phonemes. In a permutation mapping, this would still be the case, while in an arbitrary mapping, it would be *open* for approximately half of the phonemes.

Table 2 shows the results of this experiment on the test set. The arbitrary and permutation mappings both resulted in significantly worse ($p < 0.01$ using two-tailed z-test) word error rates than our knowledge-based original mapping. Furthermore, the arbitrary mapping required significantly more parameters[3]. From these results, we conclude that the articulatory knowledge does indeed contribute to the better performance of the HAMM.

4.2 Model Combination

The HAMM performs worse than the *3state* and *4state* models (see Table 3). We hypothesized, however, that since it is based on articulatory knowledge, the HAMM would make different mistakes than the standard models. Therefore, there might be a benefit to combining the HAMM with the other models. In certain cases, the success of combining two systems has been shown to rely on those two systems making different mistakes.

There are a variety of techniques for combining models. One simple way is by a weighted sum of the models' log-likelihoods. The weighting of each model is based on the prior confidence in its accuracy. Under certain modeling assumptions, if the errors are independent this can result in a higher accuracy (Bishop 1995). We used this technique for our model combination experiments.

We measured the performance of the HAMM when combined with the *4state* model in this way. We used a weight of 5.0 for the *4state* model's likelihoods, and a weight of 1.0 for the HAMM's, which were the optimal weights *based on the development set*. Figure 6 shows the results of this combination *on the test set* across a variety of likelihood weights.

We verified that the log-likelihoods for the two models vary over the same range of values. This implies that the reason that the performance of the combined model is best when *4state* is given a higher weight than the HAMM is likely due to the fact that the *4state* model alone has a lower word error rate (WER) than the HAMM alone. For comparison purposes, we also measured the performance of a combination of the *4state* and *3state* models, whose likelihoods were both given a weight of 1.0 (also the optimal weights based on the development set).

In Table 3 we show the results of performing model combination. The HAMM performed significantly worse than the *4state* model, but the combination of the two performed significantly better (12-22% relative decrease in WER versus *4state* alone), but at the expense of many more parameters. Also note that combining the *3state* model with the *4state* model had much less effect on the WER.

To understand these results, we analyzed the mistakes made by each system. On the 600 word test set, the HAMM chose the correct hypothesis 47% of the times that the *4state* model made a mistake. The two models made the same mistake only 15% of the times that the *4state* model made a mistake. More details about the differences in mistakes between the two models can be found in Table 4. The probable reason that the combination of the HAMM and *4state* model does so well is that they make different mistakes, as our analysis has shown.

4.3 Reducing the number of Parameters – State Vanishing Ratio

One disadvantage of the HAMM is its large state space and therefore number of parameters. We thus removed states during training that had low state occupation probabilities. During each training iteration, a state *i* was removed from a diphone if equation (4) held for that state.

$$\gamma_i < \sum_{j=1}^{N} \frac{\gamma_j}{N\tau} \quad \text{where} \quad \gamma_i = \sum_t \gamma_i(t) \quad \text{and} \quad \gamma_i(t) = p(Q_t = i \mid X) \tag{4}$$

where N is the number of states in the diphone, i represents a state, Q_t is the hidden state random variable, and X is the entire observation set. τ is what we call the *state vanishing ratio (SVR)*. When SVR is very high, few states are removed; a low SVR results in the removal of many states. When a state was removed, any transitions to it were proportionately re-directed to all of its direct successors.

Models were trained initially using a large SVR, $\tau=10^{20}$. After training converged, the SVR was decreased and models were re-trained until convergence. As a final step, states were removed if they existed only in untrained diphones.

Figure 7 shows the effect of various SVRs on the number of model parameters, as well as on the word error rates. As expected, when SVR decreases so do the number of parameters, but somewhat surprisingly we also found a WER improvement. After determining the ideal SVR on the development set ($\tau=10^5$) , we tested the pruned model on the test set. As Table 5 shows, the pruned model has 51% fewer parameters, but shows a 16-24% relative WER reduction. Later experiments use this reduced model.

We verified that the pruned HAMM still outperforms a random model after both the HAMM and the random models have been pruned using the SVR technique. The results are summarized in Table 5. Each of the models (one HAMM, five random) was pruned with a SVR of 10^2, 10^3, 10^5, 10^{10}, and 10^{20}. The SVR which achieved the lowest WER on the 75 and 150 word development sets was then used for the test set. The HAMM significantly out-performed the random models ($p<0.01$). The HAMM also had significantly fewer parameters than the random models ($p<0.01$).

4.4 Model Combination on Reduced model

We gave the HAMM model a weight of 1, and found the optimal *4state* model weight (searching in increments of 0.5) based on the development set to be 2.5. On the test set, the combined model achieved a 28-35% WER improvement over the *4state* model alone (see Table 5). This demonstrates that a HAMM can give practical gains when used in combination with a standard model.

4.5 Noise

A potential advantage of articulatory based HMMs is robustness to noise. Table 6 compares the performance of the models in a noisy environment[4] (the models were trained, as earlier, with clean speech; only the test utterances had noise added to them). We used stationary white Gaussian noise at 15dB SNR. Interestingly, the HAMM and the *4state* model achieved comparable WER in this case (recall that in the noise-free experiments, the HAMM performed significantly worse than the *4state* model). We believe the articulatory knowledge assists the HAMM by being more attuned to the speech-like information contained in the signals. Again, we combined the two models, using a weight of 1 for both (the optimum on the development set), and obtained a 23-26% relative WER improvement over the 4state model alone.

4.6 Diphone Models

Because we did not implement decision tree state tying (see Section 4.3), it was necessary to demonstrate that such a procedure would be unlikely to have changed our results much. Also, the HAMM is diphone-based and the *4state* model is monophone-based; as a result, our experiments may exhibit a bias against the *4state* model due to the fact that the HAMM has the opportunity to learn context-dependent models while the *4state* model does not. In what follows, we attempted to normalize for both of these issues, in order to ensure that our experiments were fair to both *4state* and the HAMM.

First, we built diphone *4state* models, called *4state-d1*, and *4state-d2* with 1 and 2 diagonal Gaussian components per state, respectively. We also constructed a new reduced test set which is the full test set minus any words which contain at least one diphone that appeared in the training set less than 10 times. On average, the reduced test set was 12% smaller than the full test set, both in utterances and lexicon size. The reduced set was necessary for testing the *4state-d* models. By comparing the results between *4state* on the full and reduced test sets, we found that the reduced test set is simpler, in that the models have less errors on it than on the full test set (See Table 7). We have verified that the words which were removed were no greater than average in having errors, and thus the error reduction in the reduced test set was due to the reduction in lexicon size.

Note that the relative WER increase in going from the reduced to the full test set is lower for the HAMM than it is for the *4state* monophone model (13% increase vs. 24% increase, on average), which implies the HAMM does not have a disproportionately larger number of errors in the words containing untrained diphones. This suggests that the HAMM's articulatory-based methods do a reasonable job at estimating parameters for unseen diphones. Also note that the performance of the 4state-d models is similar to the 4state model. This suggests that we have not been unfair in our comparison of the HAMM to the 4state model, even though the 4state model is only a monophone model while the HAMM is a diphone model.

It is also interesting since it appears that monophone phone models are not improved upon with diphone models, as is typically the case. It appears that monophone HMM models might be sufficient for this database.

4.7 Real Articulatory Data

A Viterbi path using our HAMM is an estimation of articulatory feature values throughout an utterance. To show that our model reasonably predicts articulator movements, we compared the Viterbi path with recordings of articulator motion. The articulator data comes from the MOCHA (Wrench 2000) database, which contains both speech and the measured time-aligned articulator trajectories. Data for two speakers, a female (fsew0) and a male (msak0), is currently available.

The MOCHA database contains recorded trajectories (in both X and Y dimensions) for 9 Electromagnetic Articulograph (EMA) coils attached to various parts of the mouth of the speaker. Note that in the MOCHA database, positive x-direction is toward the back of the vocal tract, away from the teeth, and positive y-direction is up, toward the roof of the mouth.

The formulae for converting from the X-Y space of the MOCHA data to our articulator feature space are given in Table 8. Table 9 explains the MOCHA abbreviations. For instance, to calculate the *Jaw Separation*, we took the difference between the upper incisor Y position (UI_Y) and the lower incisor Y position (LI_Y). This gave us a continuous value, which is at a minimum when the jaw is closed and at a maximum when the jaw is fully open. This corresponds to the *Jaw Separation* feature, which has a value of 0 when the jaw is closed and 3 when the jaw is open. Voicing was determined by measuring the c_0 energy in the laryngograph recordings which are also part of the database.

Using the HAMM, we calculated the optimal Viterbi path through the articulatory state space for the phrases in the MOCHA database, and then compared the estimated articulatory feature values with the actual measured feature values (after they had been converted as described above) using a correlation coefficient (See Table 10). All values greater than 0.01 are statistically significant ($p<0.01$). As can be seen, the diagonal entries tend to have the highest correlation. Table 10 also presents the correlation of the measured MOCHA features with themselves. This table demonstrates which correlations between features are expected, due to the physical behavior of the articulators. For instance, the strong negative correlation between the estimated jaw opening parameter with the measured lowness of the tongue is normal, as it also occurs within the measured data. The estimated and measured feature correlations generally agree.

There are a multitude of reasons why these correlations are not higher. First, the MOCHA data is recorded at 16kHz but PHONEBOOK is telephone-quality (8Khz, μ-law encoding). Second, our model was trained using isolated word speech but MOCHA is continuous speech. Third, our quantization of articulatory features as represented in the hidden state space is not necessarily linear, but is assumed to be by the correlation coefficient calculation. Also, MOCHA contains British English whereas PHONEBOOK contains only American utterances. Nevertheless, the correlations indicate that the HAMM is indeed representing articulatory information, and that the Baum-Welch algorithm has not re-assigned the state meanings during training.

4.8 Viterbi Path Through The Articulatory State Space

A Viterbi path decoding using our HAMM results in an estimation of articulatory feature values for an utterance. In Figure 8, we show a comparison of the spectrogram and the HAMM's automatically estimated articulatory features for the word "accumulation".

As can be seen, it is difficult to precisely compare the two figures. One feature which is easy to see in the spectrogram is voicing (feature 8), which seems to align very well with the HAMM's voicing feature. Another positive item to note is that the states evolve somewhat asynchronously, which is what we expect to find if the HAMM is indeed modeling the articulator movements (Deng 1994b). Other work on modeling the asynchronous evolution of articulators can be found in (Deng 1994a), (Deng 1997a), and (Deng 1998).

Recall that the mapping from phonemes to articulatory configurations used in these experiments was manually derived. We believe a data-driven technique for determining the articulatory mapping would provide better results. To this end, we used the Viterbi path, which allowed us to determine which states were most used by a particular phoneme. We generated a new mapping from phonemes to articulatory configurations by mapping each phoneme to the most common articulatory configuration(s) for it in the Viterbi paths across all training utterances. Theoretically, one could iterate this process indefinitely, using the model to estimate phonetic mappings, and using the resulting phonetic mappings to create a new model. There are many difficulties in doing this properly. For instance, some phonemes map to multiple configurations, or a sequence of configurations, both of which are lost when choosing only the most common configuration for the new mapping. We tried to solve the first problem by considering when the most and second-most common articulatory configurations for a phoneme occurred with similar frequency. In this case, we mapped the phoneme to a range which covered both configurations. For the second problem, we used the original mapping to determine which phonemes are diphthongs, and mapped these phonemes to a sequence of two articulatory configurations by dividing the Viterbi paths in half and finding a mapping for each. Empirically, we found that this data driven method for automatically mapping phonemes to articulatory configurations resulted in minor improvement in one iteration, followed by degraded performance in future iterations. Although our results in this area were not promising, we believe it may be a useful direction for future research.

5 DISCUSSION

In this work, we have presented the hidden articulatory Markov model as an alternative or companion to standard phone-based HMM models for speech recognition. We have found that either in noisy conditions, or when used in tandem with a traditional HMM, a hidden articulatory model can yield improved WER results. We have also shown that the HAMM is able to reasonably estimate articulator motion from speech.

There are a number of avenues to improve this work. In the future, we plan to add more articulatory knowledge, with rules for phoneme modification that arise as a result of physical limitations and shortcuts in speech production, as was done in (Erler 1996) (for example, vowel nasalization). Such rules may help speech recognition systems in the presence of strong coarticulation, such as in conversational speech.

While this work focused on diphone modeling, we would like to verify that the results apply for more context-dependent models as well. Diphone modeling limits the context dependency which the HAMM is able to model. This limitation can be circumvented by replacing the simple diphones in the HAMM with context-dependent diphones, in which each endpoint of the diphone is context-dependent (e.g. a triphone).

We would also like to use the MOCHA database in the training process. We believe it could be used to improve model initialization, determine better articulatory feature mappings, and find more realistic constraints on the articulator dynamics. We have done some preliminary work in using a combination of the MOCHA data and state interpolation (as introduced in Section 3.1) to create a better initial model. This work has been unsuccessful to date, which we believe is partially due to the mismatch between MOCHA and PHONEBOOK, and partially due to the difficulty in accurately quantizing the continuous-valued features given in MOCHA into meaningful discrete-valued features as required by the HAMM.

One remaining question is why has the use of articulatory information alone, without the use of phonetic information, neither helped to improve WER nor has decreased the number of parameters. We believe that it is because it is important to model the distinctive articulatory attributes in each word, and to structure the model discriminatively (Bilmes 2000). In the future, we plan to produce structurally discriminative HAMMs, both in the hidden level, and at the observation level, in what could be called a Buried Articulatory Markov Model (BAMM) (Bilmes 1999).

We have presented results demonstrating the practical usefulness of a HAMM. We accomplished a reduction in model size by 51%, while achieving a reduction in WER of 16-24%. By combining with a standard HMM model, we accomplish a 28-35% WER reduction relative to the HMM model alone, resulting in the lowest WER for PHONEBOOK that we are aware of, other than the recent work by Livescu (Livescu 2001). In the presence of noise, we improved on recognition over a standard HMM by 23-26%.

[1] Bailly, G., et al., 1992. Inversion and Speech Recognition. Signal Processing VI: Proceedings of EUSIPCO-92, vol.1 pp.159-64

[2] Bilmes, J.A., 1999. Buried Markov Models for Speech Recognition. ICASSP 1999, vol. 2 pp. 713-16

[3] Bilmes, J.A., 2000. Dynamic Bayesian Multinets. Proceedings of the 16^{th} conference on Uncertainty in Artificial Intelligence, pp. 38-45

[4] Bishop, C., 1995, *Neural Networks for Pattern Recognition*. (Oxford University Press)

[5] Blackburn, C. and Young, S., 1995. Towards improved speech recognition using a speech production model. Proc. Eurospeech, 1995, vol. 2, pp. 1623-1626

[6] Blomberg, M., 1991. Modelling articulatory inter-timing variation in a speech recognition system based on synthetic references. Proceedings of Eurospeech 1991

[7] Coker, C., 1976. A model of articulatory dynamics and control. Proc. IEEE 64, pp.452-60

[8] Deng, L., 1998. A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. Speech Communication, vol. 24:4, pp. 299-323

[9] Deng, L., 1997(a). Autosegmental representation of phonological units of speech and its phonetic interface. Speech Communication, vol. 23:3, pp. 211-222

[10] Deng, L., Ramsay, G., and Sun, D., 1997(b). Production models as a structural basis for automatic speech recognition. Speech Communication, vol. 22:2, pp. 93-111

[11] Deng, L. and Sun, D., 1994(a). A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. Journal of the Acoustical Society of America, vol. 95:5, pp. 2702-2719

[12] Deng, L. and Sun, D., 1994(b). Phonetic Classification and Recognition Using HMM Representation of Overlapping Articulatory Features for all classes of English sounds. ICASSP, 1994, pp.45-8

[13] Dupont, S., et al., 1997. Hybrid HMM/ANN systems for training independent tasks: Experiments on PHONEBOOK and related improvements. ICASSP, 1997, pp.1767-70

[14] Edwards, H.T., 1997, *Applied Phonetics: The Sounds of American English* second edition. (Singular, San Diego)

[15] Eide, E., Rohlicek, J.R., Gish, H., and Mitter, S., 1993. “A linguistic feature representation of the speech waveform,” Proceedings of ICASSP, 1993, pp. 483-6

[16] Elenius, K. and Blomberg, M., 1992. Comparing phoneme and feature based speech recognition using artificial neural networks. Proceedings of ICSLP, 1992, pp. 1279-82

[17] Erler, K. and Freeman, G.H., 1996. An HMM-based speech recognizer using overlapping articulatory features. J. Acoust. Soc. Am. 100, pp.2500-13

[18] Frankel, J., Richmond, K., King, and Taylor, P., 2000. An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces. Proceedings of ICSLP, 2000.

[19] Frankel, J., King, S., 2001. ASR – Articulatory Speech Recognition. Proceedings of Eurospeech 2001.

[20] Ghahramani, Z., 1998. Learning Dynamic Bayesian Networks. in *Lecture Notes in Artificial Intelligence*. (Springer-Verlag)

[21] Gupta, S. and Schroeter, J., 1993. Pitch-synchronous frame-by-frame and segment-based articulatory analysis by synthesis. J. Acoust. Soc. Am. 94:5, Nov. 1993, pp.2517-30

[22] Hardcastle, W.J. and Hewlett, N. (eds.), 1999. *Coarticulation. Theory, Data and Techniques*. (Cambridge)

[23] Kirchhoff, K., 1998. Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments. Proceedings of ICSLP, 1998, pp.891-4

[24] Lauritzen, S.L., 1996, *Graphical Models*. (Oxford Science Publications)

[25] Livescu, K., 2001, Segment-Based Recognition on the PhoneBook Task: Initial Results and Observations on Durational Modeling. Proc. Eurospeech, 2001.

[26] Logan, B. and Moreno, P., 1998. Factorial HMMs for acoustic modeling. ICASSP, 1998 , pp.813-6

[27] Picone, J., Pike, S., Regan, R., Kamm, T., Bridle, J., Deng, L., Ma, Z., Richards, H., and Schuster, M., 1999. Initial evaluation of hidden dynamic models on conversational speech. Proceedings of ICASSP, 1999, pp. 109-112
[28] Pitrelli, J., Fong, C., Wong, S.H., Spitz, J.R. and Lueng, H.C., 1995. PhoneBook: A phonetically-rich isolated-word telephone speech database. ICASSP, 1995 pp.101-4
[29] Richardson, M., Bilmes, J., and Diorio, C., 2000a. Hidden-Articulator Markov Models for Speech Recognition. ASR2000, pp.133-9
[30] Richardson, M., Bilmes, J. and Diorio, C., 2000b. Hidden-Articulator Markov Models: Performance Improvements and Robustness to Noise. ICSLP 2000, vol. 3, pp.131-4
[31] Rose, R., Schroeter, J., and Sondhi, M., 1996. The potential role of speech production models in automatic speech recognition. Journal of the Acoustical Society of America, vol. 99, pp. 1699-1709
[32] Saul, L. and Jordan,M., 1999. Mixed Memory Markov models: decomposing complex stochastic processes as mixtures of simpler ones. Machine-Learning 37:1, Oct 1999, pp.75-87
[33] Schmidbauer, O., 1989. Robust statistic modelling of systematic variabilities in continuous speech incorporating acoustic-articulatory relations. Proceedings of ICASSP, 1989, pp. 616-9
[34] Wrench, A., 2000. A Multichannel/Multispeaker Articulatory Database for Continuous Speech Recognition Research. Workshop on Phonetics and Phonology in ASR, Saarbruecken, Germany, 2000.
[35] Young, S., 1996. A Review of Large-vocabulary Continuous-speech Recognition. IEEE Signal Processing Magazine, 13:5, Sept. 1996, pp.45-57
[36] Zweig, G. and Russell, S., 1998. Speech Recognition with Dynamic Bayesian Networks. AAAI-98, pp.173-80

Footnotes:

(1) Each time step is one frame of speech, 10ms in our experiments

(2) Recall, p_a or q_a may be a range of values, see Section 2.1.

(3) The arbitrary model begins with more parameters as well. In the arbitrary mapping, the beginning and ending phones of a diphone are more likely to contain different values for each feature since the entropy of each feature is higher than in the original or permuted mappings. This results in larger diphone models. Many of these states, however, were not removed by the state elimination algorithm, implying that they were being used by the model.

(4) Note that because PHONEBOOK is telephone quality speech, it is already somewhat noisy, so even the clean-speech case isn't really clean.

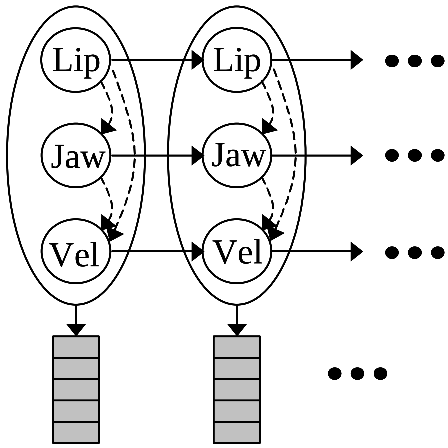
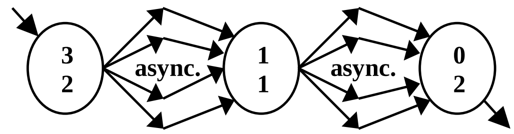


Figure 1: HAMM cast as a factorial HMM

Jaw	Lip Sep	Lip Round	Tongue Back/Fwd	Tongue Low/High	Tongue Tip	Velic Aper.	Voicing
nearly closed	closed	rounded and or protruded	back	low and flat	low	closed	off
neutral	slightly apart	slightly rounded or tensed corners	slightly back	mid or central	neutral	open	on
slightly lowered	apart	neutral	neutral	mid-high	between or touching top teeth		
lowered	wide apart	wide	slightly forward	high	near alveolar ridge		
			forward		touching alveolar or upper ridge		

Figure 2: Articulatory feature space.



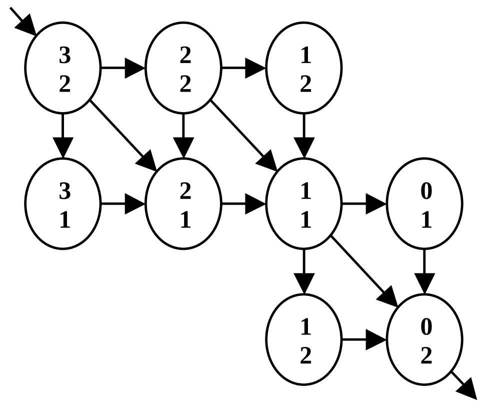


Figure 3a: (upper) A diphone model is a sequence of articulator configuration targets, with asynchronous articulatory movement in between.

Figure 3b: (lower) Example HMM transition graph for a diphone. Note, each state also has a transition back to itself, which was omitted for clarity.

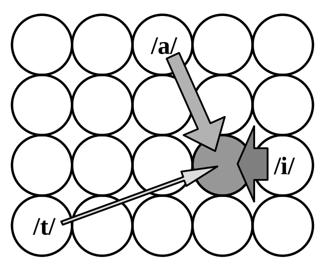


Figure 4: Sample state initialization. The shaded circle is a state being initialized. It is interpolated from states which are mapped to directly by a phoneme. The width of the arrow represents the weight given to each factor in the interpolation, which is proportional to the inverse Euclidean distance between them.

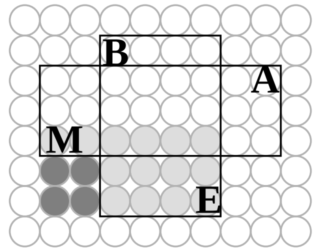


Figure 5: Demonstrates the shared nature of states (circles) across trained diphones /B/-/E/ and /M/-/A/ (boxes) and untrained diphone /M/-/E/ (grey circles). Dark grey circles are untrained states which are removed from the model.

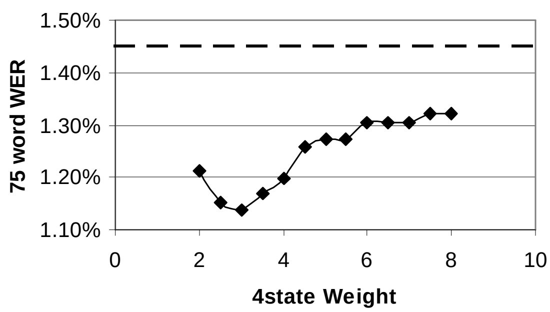


Figure 6: WER of the combined (*4state* + HAMM) model across various *4state* weightings (the HAMM has a weight of 1.0) for the 75 word lexicon test (the y-axis range has been scaled to improve resolution). The black dashed line shows the performance of the *4state* model alone.

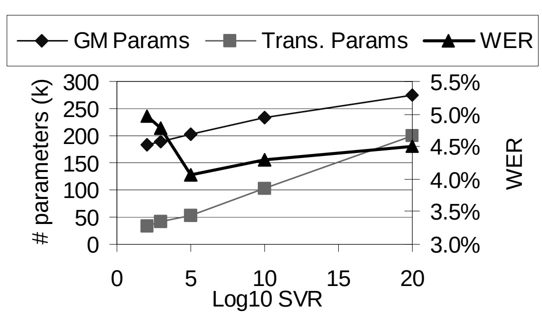


Figure 7: Effect of varying SVR. Shown are the number of Gaussian mixture (GM) parameters, transition parameters, and word error rate on the development set.

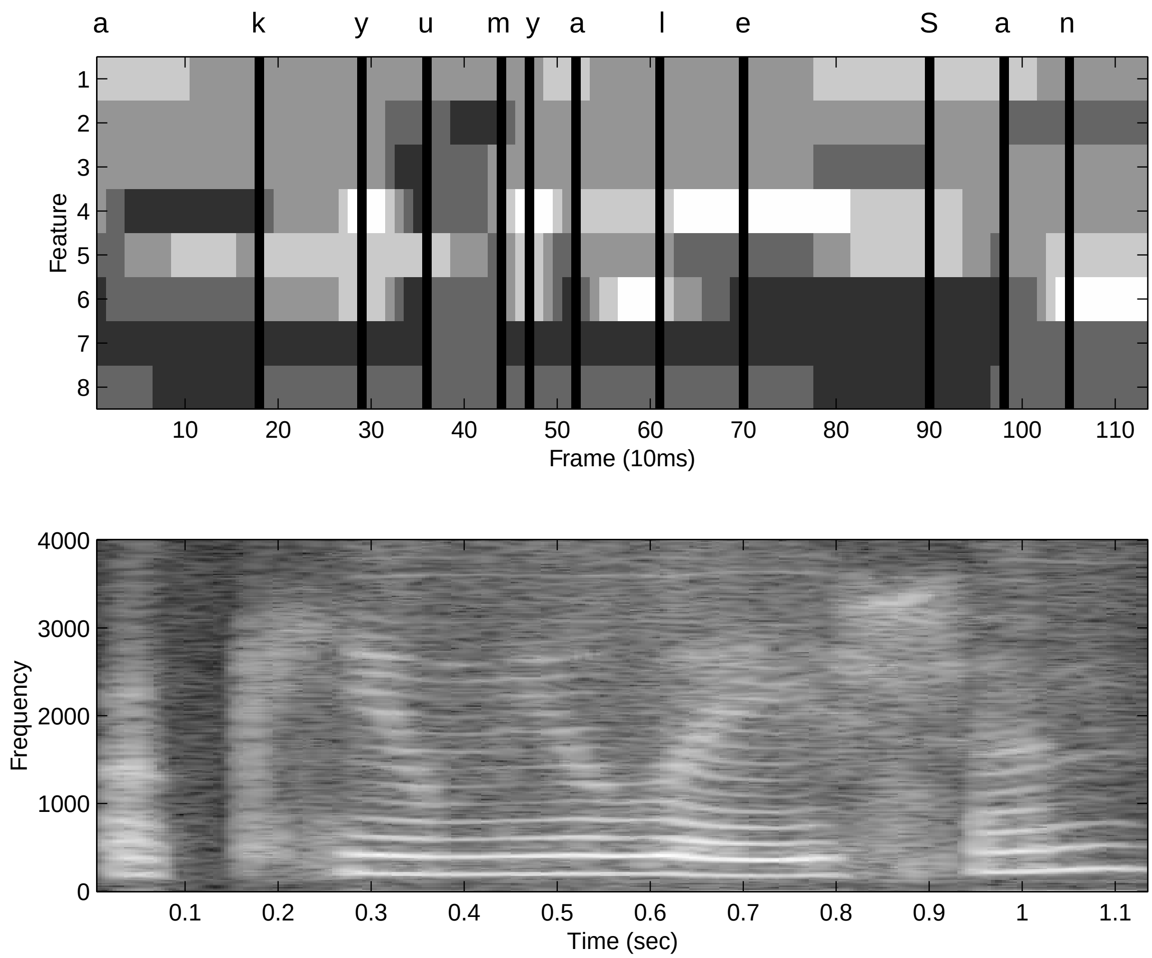


Figure 8: HAMM Viterbi path decoding for the word "accumulation." The lower half of the figure is a spectrogram of the speech. The upper half shows the estimated articulatory configurations over time (note: features are numbered 1-8 with 1=Jaw and 8=Voicing). The black vertical lines denote the estimated diphone boundaries.

Phone	Normal		Permutation		Arbitrary	
	Jaw	Nasal	Jaw	Nasal	Jaw	Nasal
a	0	1	1	0	0	0
b	2	0	0	1	1	1
c	1	0	0	0	2	1
d	0	0	2	0	1	0

Table 1: Sample phoneme mapping, highlighting the difference between *permutation* and *arbitrary* random mappings. *Permutation* is a reordering of the rows, while *arbitrary* is purely random. Notice how the *permutation* mapping retains the distribution of values for a given feature.

Lex. Size	75	150	300	600	params
original	3.23%	4.67%	6.69%	9.03%	522k
arbitrary	3.72 ± 0.08%	5.18 ± 0.06%	7.19 ± 0.20%	9.81 ± 0.22%	661k ± 10k
permutation	4.76 ± 0.24%	6.77 ± 0.40%	9.11 ± 0.43%	12.35 ± 0.35%	462k ± 13k

Table 2: Word Error Rate comparison of original phone mapping versus random mappings for various lexicon sizes. Random model results are given as mean ± standard error (we tested 5 arbitrary models and 2 permutation models). The original mapping is significantly better than either of the random mappings. (Note that the number of parameters varies due to pruning).

Lexicon Size	75	150	300	600	params
HAMM	3.23%	4.67%	6.69%	9.03%	522k
3state	1.88%	2.91%	4.20%	6.14%	105k
4state	1.45%	2.79%	4.04%	5.76%	203k
4state+3state	1.42%	2.49%	3.71%	5.46%	308k
4state+HAMM	1.27%	2.18%	3.29%	4.56%	725k

Table 3: Word Error Rate comparison showing the advantage of combining models. The best combination is the standard *4state* HMM with the HAMM.

HAMM	***4state***	***Occurrences***
correct	correct	5825
correct	wrong	177
wrong	correct	393
wrong	wrong (same)	57
wrong	wrong (different)	146
total		6598

Table 4: The *HAMM* and *4state* models make different mistakes on the 600 word task, making model combination likely to be beneficial.

Model	75	150	300	600	params
unpruned HAMM	3.23%	4.67%	6.69%	9.03%	520k
pruned HAMM	2.46%	3.77%	5.47%	7.56%	255k
pruned random models	3.18% ± 0.08%	4.48% ± 0.11%	6.53% ± 0.15%	8.83% ± 0.17%	388k ± 27k
4state	1.45%	2.79%	4.04%	5.76%	203k
pruned HAMM + 4state	0.99%	1.80%	2.79%	4.17%	458k

Table 5: WER Results on the test set for various lexicon sizes. Random model results are given as mean ± standard error (over 5 models). The pruned HAMM does better in both WER and number of parameters than before pruning, as well as in comparison with random models. The last entry is the combined model, which out-performs all other models tested.

Model	75	150	300	600
HAMM	15.40%	20.63%	26.16%	32.43%
4state	14.65%	20.70%	26.76%	33.68%
combined	10.91%	15.60%	20.61%	25.86%

Table 6: WER results on the test set in the presence of 15db SNR additive noise for various lexicon sizes.

Model	test set	75	150	300	600	param
4state	full	1.45%	2.79%	4.04%	5.76%	203k
4state	reduced	1.08%	2.18%	3.31%	5.08%	203k
4state-d1	reduced	1.39%	2.29%	3.48%	4.79%	217k
4state-d2	reduced	1.13%	1.91%	2.86%	4.10%	425k
HAMM	full	2.46%	3.77%	5.47%	7.56%	255k
HAMM	reduced	2.08%	3.25%	4.92%	7.02%	255k

Table 7: Comparison of diphone and non-diphone systems on full and reduced test sets. The reduced test set contains no words with untrained diphones.

Feature	Abbr.	M	Low →	High	Formula
Jaw Separation	Jaw	4	closed	open	UI_Y – LI_Y
Lip Separation	Lip	4	closed	open	UL_Y – LL_Y
Lip Rounding	Rnd	4	round	wide	none
Tongue Body	BF	5	back	fwd.	-TB_X – BN_X
Tongue Body	LH	4	low	high	TB_Y – BN_X
Tongue Tip	Tip	5	low	high	TT_Y – BN_Y
Velic Aperture	Vel	2	closed	open	-V_Y – BN_Y
Voicing	Voic	2	off	on	laryn. c_0 energy

Table 8: Articulatory dimensions. M denotes the number of quantization levels. Formulas are given for translating from recorded MOCHA (Wrench 2000) data to our articulatory space (see Section 4.7). All values except laryngograph energy come from the EMA data.

Feature	Description
UI	Upper Incisors
LI	Lower Incisors
UL	Upper Lip
LL	Lower Lip
TT	Tongue Tip (5-10mm from extended tip)
TB	Tongue Blade (approx. 2-3cm beyond TT)
TD	Tongue Dorsum (approx. 1-2cm beyond TB)
V	Velum (approx. 1-2 cm beyond hard palate)
BN	Bridge of nose reference

Table 9: Description of MOCHA features

		Measured Feature								Measured Feature						
		Jaw	Lip	BF	LH	Tip	Vel	Vce		Jaw	Lip	BF	LH	Tip	Vel	Vce
Estimated Feature	Jaw	.36	.21	-.22	-.29	-.31	.18	.20		.21	.15	-.14	-.18	-.21	.03	.15
	Lip	.14	.36	-.12	-.08	-.06	-.06	-.03		.07	.27	-.08	-.07	-.01	-.11	-.08
	BF	-.17	.15	.22	-.02	.23	-.10	-.12		-.22	.03	.03	.04	.28	.08	-.13
	LH	-.44	-.07	.14	.36	.43	-.19	-.22		-.32	-.01	.05	.23	.31	-.02	-.14
	Tip	-.18	-.11	-.06	.11	.36	.03	-.04		-.06	-.02	.02	.02	.20	.11	.04
	Vel	-.08	-.12	.09	.08	.08	.29	.22		.01	-.06	.10	.06	.02	.23	.28
	Vce	.21	.09	-.09	.00	-.16	.16	.61		.23	.14	-.05	-.08	-.13	.16	.60

		Measured Feature								Measured Feature						
		Jaw	Lip	BF	LH	Tip	Vel	Vce		Jaw	Lip	BF	LH	Tip	Vel	Vce
Measured Feature	Jaw	1.0	.40	-.23	-.31	-.62	.24	.35		1.0	.50	.08	-.40	-.65	.01	.33
	Lip	.40	1.0	.09	.08	-.17	.06	.19		.50	1.0	.12	.02	-.18	-.05	.25
	BF	-.23	.09	1.0	.13	.01	-.23	-.14		.08	.12	1.0	.08	-.10	.07	-.08
	LH	-.31	.08	.13	1.0	.45	-.19	.00		-.40	.02	.08	1.0	.55	-.12	-.09
	Tip	-.62	-.17	.01	.45	1.0	-.13	-.27		-.65	-.18	-.10	.55	1.0	.06	-.19
	Vel	.24	.06	-.23	-.19	-.13	1.0	.23		.01	-.05	.07	-.12	.06	1.0	.16
	Vce	.35	.19	-.14	.00	-.27	.23	1.0		.33	.25	-.08	-.09	-.19	.16	1.0

Table 10: Correlations of estimated vs. measured articulator positions of female (upper-left) and male (upper-right) data. Correlations of measured articulator positions vs. themselves in female (lower-left) and male (lower-right) data. Measurements are from MOCHA, estimates are from the pruned HAMM Viterbi path.

APPENDIX

Using Edwards (Edwards 1997) as a guide to phonetics, we constructed the mapping from phonemes to articulatory configurations (given below). Note that some phonemes have multiple values for a given feature, such as the tongue tip position in phoneme /R/. Some phonemes also are defined as a sequence of configurations, such as the phoneme /p/, which is formed by bringing the lips together (lip separation=0, “closed”) to temporarily stop the flow of air, and then separating them (lip separation=2, “apart”).

phoneme	sample word	jaw	lip separation	lip width	tongue body (back/fwd.)	tongue body (low/high)	tongue tip	velic aper.	voiced
i	b**EA**t	0	1	2	4	3	0	0	1
I	b**I**t	3	2	2	4	2	0	0	1
e	b**AI**t	1	2	2	4	1	0	0	1
E	b**E**t	3	2	2	4	1	0	0	1
@	b**A**t	3	3	1	3	0	0	0	1
a	b**O**b	3	2	2	2	0	0	0	1
c	b**OUGH**t	3	2	0	1-2	3	0	0	1
o	b**OA**t	3	2	0	1	1	0	0	1
^	b**U**t	2	2	2	2	1	0	0	1
u	b**OO**t	1	1	0	0	3	0	0	1
U	b**OO**k	1	2	1	0	3	0	0	1
Y	b**I**te								
	onset	3	2	2	3	0	0	0	1
	offset	1-2	2	2	4	3	0	0	1
O	b**OY**								
	onset	2	2	0	1	0-1	0-1	0	1
	offset	0-1	2	1-2	4	3	1	0	1
W	b**OU**t								
	onset	3	2	2	3	0	0	0	1
	offset	1-2	2	0	0	3	0	0	1
R	bI**R**d	2	2	0	2-3	2	0-1	0	1
x	sof**A**	2	2	2	2	1	0	0	1
X	butt**ER**	2	2	1	2	2	0-1	0	1
l	**L**et	1	2	2	3	2	4	0	1
w	**W**et	1	2	0	0	3	1	0	1
r	**R**ed	1	2	1	2	2	3	0	1
y	**Y**et	1	2	2	4	3	3	0	1
n	**N**eat	1	1	2	2	3	4	1	1
m	**M**eet	1	0	2	2	1	1	1	1
G	si**NG**	1	2	2	0	3	1	1	1
h	**H**eat	2	2-3	2	2	1	1	0	0

phoneme	sample word	jaw	lip separation	lip width	tongue body (back/fwd.)	tongue body (low/high)	tongue tip	velic aper.	voiced
s	**S**ee	1	2	1-2	3	2-3	0-1	0	0
S	**Sh**e	2	2	1-2	3	3	0	0	0
f	**F**ee	2	0	2	2	1	1	0	0
T	**Th**igh	2	2	2	4	2	2	0	0
z	**Z**oo	1	2	1-2	3	3	0-1	0	1
Z	mea**S**ure	2	2	1-2	3	3	0	0	1
v	**V**an	2	0	2	2	1	1	0	1
D	**Th**y	2	2	2	4	0	2	0	1
p	**P**ea								
	setup	1	0	2	2	1	1	0	0
	release	1	2	2	2	1	1	0	0
t	**T**ea								
	setup	1	1	2	4	3	4	0	0
	release	1	2	2	4	2	3	0	0
k	**K**ey								
	setup	1	2	2	0	3	1	0	0
	release	1	2	2	0	2	1	0	0
b	**B**ee								
	setup	1	0	2	2	1	1	0	1
	release	1	2	2	2	1	1	0	1
d	**D**ay								
	setup	1	1	2	4	3	4	0	1
	release	1	2	2	4	2	3	0	1
g	**G**eese								
	setup	1	2	2	0	3	1	0	1
	release	1	2	2	0	2	1	0	1
C	**Ch**ur**CH**								
	start	2	2	1-2	4	3	4	0	0
	end	1	2	2	3	3	0	0	0
J	**J**u**DG**e								
	start	2	2	1-2	4	3	4	0	1
	end	1	2	2	3	3	0	0	1