A Factored Conditional Random Field Model for Articulatory Feature Forced Transcription

Rohit Prabhavalkar [#], Eric Fosler-Lussier [#], Karen Livescu [*]

[#] *Department of Computer Science and Engineering, The Ohio State University*
Columbus, Ohio, USA
prabhava@cse.ohio-state.edu
fosler@cse.ohio-state.edu

[*] *Toyota Technological Institute at Chicago*
Chicago, Illinois, USA
klivescu@ttic.edu

***Abstract*—We investigate joint models of articulatory features and apply these models to the problem of automatically generating articulatory transcriptions of spoken utterances given their word transcriptions. The task is motivated by the need for larger amounts of labeled articulatory data for both speech recognition and linguistics research, which is costly and difficult to obtain through manual transcription or physical measurement. Unlike phonetic transcription, in our task it is important to account for the fact that the articulatory features can desynchronize. We consider factored models of the articulatory state space with an explicit model of articulator asynchrony. We compare two types of graphical models: a dynamic Bayesian network (DBN), based on previously proposed models; and a conditional random field (CRF), which we develop here. We demonstrate how task-specific constraints can be leveraged to allow for efficient exact inference in the CRF. On the transcription task, the CRF outperforms the DBN, with relative improvements of 2.2% to 10.0%.**

I. Introduction

Models of articulatory features have been investigated in both speech recognition research [1], [2], [3], [4], [5] and linguistics [6]. By "articulatory features", we refer to discrete-valued, speaker-independent parameters of speech production such as voicing, nasality, and constriction degrees and locations. For speech recognition, there is evidence that articulatory feature-based models may help to better handle noise [7], multilinguality [8], and pronunciation variation [9], [10].[1]

Articulatory models of word pronunciation are more complex than traditional phone-based ones, because they must account for the ability of articulators to desynchronize (that is, to reach their positions for a given sub-word state at different times) or fail to reach their target positions. In prior work, this has been addressed using dynamic Bayesian networks (DBNs) explicitly modeling the multiple articulatory streams [9] or finite-state models (hidden Markov models, finite-state transducers) in which the multiple state variables are "collapsed" into a single one [1], [11], [4]. In this paper, we present a factored conditional random field [12], [13] (CRF), which includes different factors for different articulatory feature streams.[2] The factorization allows us to explicitly represent articulatory asynchrony and avoid the large number of parameters involved in models with collapsed state spaces. In this paper, we discuss how task-specific constraints can be exploited to perform exact inference efficiently as if the model were a single linear chain, while maintaining the factorization in the original distributed model.

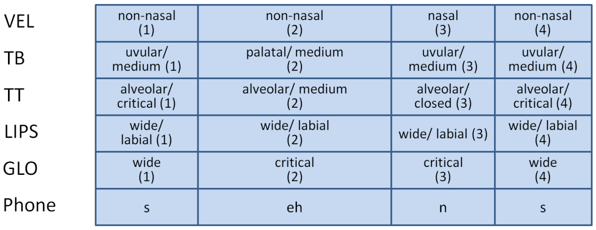
One difficulty associated with articulatory models is the lack of data transcribed with articulatory labels. Articulatory labels have been variously obtained through conversion from phonetic labels [7], [8], manual transcription [14], and conversion from physical measurements to speaker-independent quantities [15]. All of these suffer from the difficulty of obtaining the labels or appropriate source data. In this work, we address this in a similar manner to which phonetic transcriptions are often obtained: by forced transcription given the acoustics and word labels, along with a model of the relationship between the words, articulatory features, and acoustics. This necessitates a detailed pronunciation model in terms of articulatory features. Additional motivation for the CRF models developed here is the desire for a discriminative articulatory feature-based end-to-end speech recognizer. Our models can be straightforwardly adapted for this purpose, although in this paper we focus on forced transcription.

The important aspects of the models we present are that:

1) They are *conditional models*, unlike previously proposed DBN or finite-state models of articulatory features. This should improve their discriminative ability.
2) They are *undirected graphical models*, making it easy to incorporate a wide variety of classifiers as feature functions (as in [16]).
3) They are *factored*, encoding linguistic knowledge about articulator configurations into the model parameterization.
4) They are *highly constrained*, so that it is possible to perform fast exact inference.

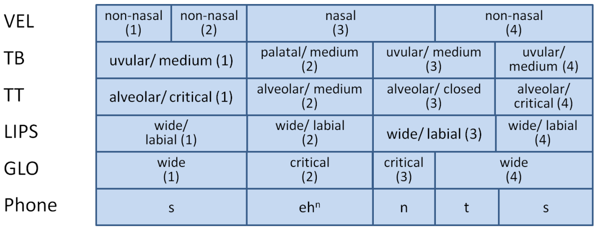
[1] By the term "articulatory feature-based models", we include what are sometimes referred to as "production models", "phonological feature models", or "gestural models".

[2] Factored CRFs have also been referred to as dynamic CRFs [13]. We prefer the term "factored" since "dynamic" is typically used for referring to models with repeating structure. While our models are dynamic, it is the factorization of the state space that is important.



VEL	non-nasal (1)	non-nasal (2)	nasal (3)	non-nasal (4)
TB	uvular/ medium (1)	palatal/ medium (2)	uvular/ medium (3)	uvular/ medium (4)
TT	alveolar/ critical (1)	alveolar/ medium (2)	alveolar/ closed (3)	alveolar/ critical (4)
LIPS	wide/ labial (1)	wide/ labial (2)	wide/ labial (3)	wide/ labial (4)
GLO	wide (1)	critical (2)	critical (3)	wide (4)
Phone	s	eh	n	s

Fig. 1. Canonical pronunciation for *sense*, corresponding to a production where the features velum (VEL), tongue body (TB), tongue tip (TT), lip position (LIPS) and glottis (GLO) are synchronized. Values in parentheses are the sub-word state indices (SubwordState^i) corresponding to each sub-word unit in the word (see Section III).



VEL	non-nasal (1)	non-nasal (2)	nasal (3)		non-nasal (4)	
TB	uvular/ medium (1)		palatal/ medium (2)	uvular/ medium (3)		uvular/ medium (4)
TT	alveolar/ critical (1)		alveolar/ medium (2)	alveolar/ closed (3)		alveolar/ critical (4)
LIPS	wide/ labial (1)		wide/ labial (2)	wide/ labial (3)		wide/ labial (4)
GLO	wide (1)		critical (2)	critical (3)	wide (4)	
Phone	s		eh^n	n	t	s

Fig. 2. A non-canonical variant pronunciation for *sense* (with an epenthetic [t] insertion), produced when the velum and tongue features desynchronize.

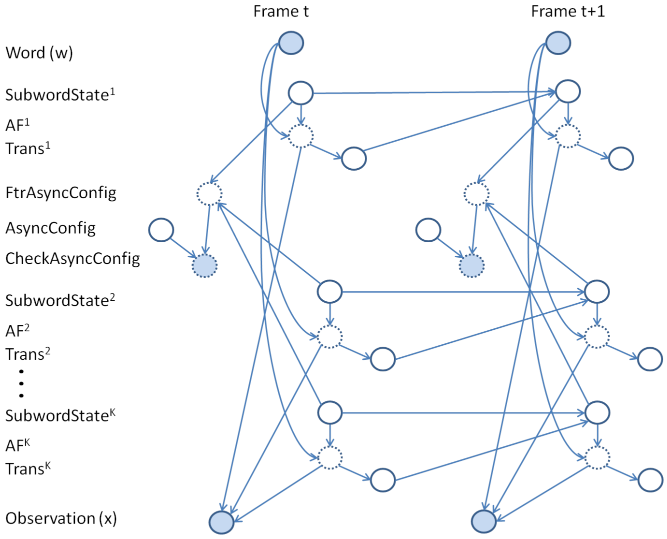


Fig. 3. Two frames of a DBN model for articulatory feature alignment. Shaded nodes correspond to observed variables, while unshaded nodes correspond to hidden variables. Variables that are deterministic given the values of their parents appear as dashed circles.

In experiments, we demonstrate the superiority of the proposed approach over a DBN model using a similar factorization.

II. Articulatory Feature-based Models of Pronunciation

Following previous work by Livescu et al. [3], [9], we use a set of articulatory features based on the tract variables of articulatory phonology [6]. These include the position and constriction degree of the lips (3 and 4 values, respectively), the tongue tip (4 and 6 values), and tongue body (4 and 6 values), and the states of the velum (nasality, 2 values) and glottis (voicing state, 3 values).

We assume that we have a standard dictionary of phone-based canonical pronunciations for the words in our vocabulary. We further assume that each phone can be deterministically mapped to a set of articulatory feature target values (in our case, using the mapping described in [9]). The pronunciation of each word can therefore be re-written in terms of articulatory feature targets. The articulatory features may move asynchronously from one target to the next; if the transitions are completely synchronized, then the resulting pronunciation corresponds (by construction) to the canonical pronunciation. Figures 1 and 2 show the canonical pronunciation and one non-canonical variant of the word *sense*. Note that the sequence of targets for each articulatory feature is identical in these two pronunciations; the pronunciations differ only in the relative timing of the targets in different feature tiers.

The models developed in this work are applied to the task of articulatory feature forced alignment: Given a parameterization of the acoustics for the entire utterance ($\mathbf{x}$) (e.g., PLP coefficients) corresponding to word(s) w, predict the most likely values of the articulatory features in each time frame. In this paper we will assume that the word boundaries are also given, and we will operate on one word at a time. (Our models can, however, be easily extended to multi-word sequences.) That is, we seek to solve the following:

$$\mathbf{AF}^{1*}, \mathbf{AF}^{2*}, \ldots, \mathbf{AF}^{K*} = \operatorname*{argmax}_{\mathbf{AF}^1, \mathbf{AF}^2, \ldots, \mathbf{AF}^K} p(\mathbf{AF}^1, \mathbf{AF}^2, \ldots, \mathbf{AF}^K | w, \mathbf{x}) \quad (1)$$

where K is the number of articulatory features and $\mathbf{AF}^i = (\mathrm{AF}^i_1, \ldots, \mathrm{AF}^i_t, \ldots, \mathrm{AF}^i_T)$ is the sequence of values of articulatory feature i, where T is the number of frames in the utterance.

III. Dynamic Bayesian Network-based model

A dynamic Bayesian network (DBN) model for this task, based on the models used previously by Livescu et al. [3] (with minor modifications), is shown in Figure 3.

Each articulatory feature AF^i has a corresponding sub-word state variable (SubwordState^i), whose value ranges from 1 to the number of states (in our case, phones) in the word. For each frame, the value of the sub-word state variable is either incremented (if the transition variable Trans^i $= 1$) or remains the same in the next frame (if Trans^i $= 0$). The feature variables AF^i depend on the word and SubwordState^i. Using the example in Figure 1, if AF^1 is velum (VEL), then when $w =$ *sense* and SubwordState^1 $= 1$, AF^1 $=$ “non-nasal”, while when SubwordState^1 $= 3$, AF^1 $=$ “nasal”.

The model constrains asynchrony as follows: For each pair (i, j) of articulatory features, define the degree of asynchrony between the two streams ($d^{i,j}_t$) at time frame t as the difference between the sub-word state indices corresponding to the two streams at that frame:

$$d_t^{i,j} = \text{SubwordState}_t^i - \text{SubwordState}_t^j \qquad (2)$$

with $d_t^{i,j} = 0$ indicating that streams i and j are synchronized at time t. We constrain the degree of asynchrony to no more than M for any pair of streams; that is,

$$-M \leq d_t^{i,j} \leq M, \qquad 1 \leq i,j \leq K \text{ and } 1 \leq t \leq T \qquad (3)$$

where T is the length of the utterance. In the DBN, these constraints are imposed using the FtrAsyncConfig, AsyncConfig and CheckAsyncConfig variables. The variable FtrAsyncConfig_t is a vector representing the current configuration of asynchrony amongst all of the streams, consisting of the degrees of asynchrony of streams $2, \ldots, K$ with respect to the first stream: $(d_t^{2,1}, d_t^{3,1}, \cdots, d_t^{K,1})$. For example, $\text{FtrAsyncConfig}_t = (1, 0, \cdots, 0)$ indicates that the second articulatory stream is one state ahead of the first stream, while the remaining streams are synchronized with the first. AsyncConfig is a vector variable with no parents that can take on any value corresponding to an allowable asynchrony configuration. CheckAsyncConfig is a dummy observed variable with value 1 (say) which has a non-zero probability for only those configurations in which AsyncConfig and FtrAsyncConfig are assigned the same value. Therefore, the distribution of AsyncConfig, which is learned during DBN training, represents the probability of each asynchrony configuration.[3]

IV. Conditional Random Field-based model

The proposed CRF, shown in Figure 4 as a factor graph [18], incorporates a number of aspects of the DBN. Each factor (red or blue square) represents a (non-negative) function over the set of variables that are connected to it. As before, $\mathbf{x} = (x_1, \ldots, x_t, \ldots, x_T)$ denotes the sequence of observations and w denotes the word spoken in the utterance. The sequence of all other variables in the graph (which are all hidden) is denoted by $\mathbf{y} = (y_1, \ldots, y_t, \ldots, y_T)$, where y_t is all of these variables at time t. Each factor node is associated with a potential function, which may depend on the time t and is indexed by the set of variable nodes (or clique) c that it is connected to in the graph. In the factor graph presented in Figure 4, for example, one factor is associated with the set of variable nodes $c = \{\text{SubwordState}^1, \text{AF}^1, w\}$ at each time frame.

Let y_t^c represent the values of the hidden variables in c at time t. The probability distribution over $\mathbf{y}$, conditioned on the observations $\mathbf{x}$ and the word w, can then be expressed as a normalized product of potentials corresponding to the factors:

$$p(\mathbf{y}|\mathbf{x}, w) = \frac{1}{Z(\mathbf{x}, w)} \prod_t \prod_{c \in \mathcal{C}} \phi_c(y_t^c, \mathbf{x}, w, t) \qquad (4)$$

where $Z(\mathbf{x}, w)$ is a normalization term, the potentials $\phi_c(y_t^c, \mathbf{x}, w, t)$ are any non-negative functions, and $\mathcal{C}$ is the set of variable subsets corresponding to all of the factors.

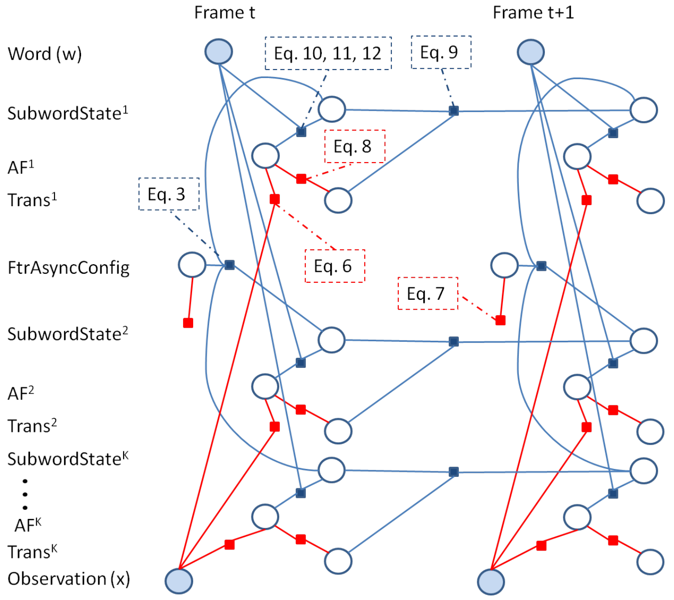


Fig. 4. Factor graph [18] representing the proposed CRF model for articulatory feature forced transcription. The shaded nodes represent variables that we condition on. The red and blue square nodes represent factors, each of which is a non-negative function defined over the configuration of the set of variables connected to it. The number in the dashed box corresponds to the equation(s) in the text that implements the corresponding factor.

A. Deterministic vs. Trainable Factors

We distinguish between factors that are associated with learnable parameters of the model, denoted in red in Figure 4, from those that enforce deterministic constraints, denoted in blue. Each trainable factor is associated with a vector of *feature functions*, $\mathbf{f}^c(y_t^c, \mathbf{x}, w, t) = [f_i^c(y_t^c, \mathbf{x}, w, t)]^T, \quad 1 \leq i \leq N_c$ for some integer N_c. We represent the potential associated with this factor as

$$\phi_c(y_t^c, \mathbf{x}, w, t) = e^{\lambda_c \cdot \mathbf{f}^c(y_t^c, \mathbf{x}, w, t)} \qquad (5)$$

Where λ_c is a vector of weights to be learned. In equation 5, we have implicitly assumed that weights are tied across corresponding cliques in different time frames t. The specific parameterization of each feature function depends on the particular clique.

In our model the trainable factors are associated with configurations of feature asynchrony, individual articulatory features (the "acoustic model"), and articulatory feature transitions (the "transition model"). The feature functions associated with each articulatory feature variable AF_t^i are constructed by first computing a set of statistics $g_{l,m}(x_t)$ from the acoustics (e.g. $g_{l,m}(x_t)$ could be the l^{th} output of a particular multilayer perceptron (MLP) indexed by m, as in Section V). These statistics are then used to construct individual components in the vector of feature functions associated with the articulatory feature variable (AF^i),

$$f_{i,j,l,m}(\text{AF}_t^i, \mathbf{x}, t) = g_{l,m}(x_t) \delta(\text{AF}_t^i = a_j^i) \qquad (6)$$

where a_j^i is one value that AF^i can take and $\delta(z = z') = 1$

[3] The structure using both the deterministic FtrAsyncConfig and non-deterministic AsyncConfig allows the asynchrony distribution to be learned as part of the DBN training via Expectation-Maximization. See [17] for more details.

if $z = z'$ and 0 otherwise. The feature functions associated with the feature asynchrony configuration (FtrAsyncConfig) and articulatory feature transitions ($\text{AF}^i, \text{Trans}^i$) are

$$f_r(\text{FtrAsyncConfig}_t, t) = \delta(\text{FtrAsyncConfig}_t = r) \quad (7)$$

$$f_{i,j,v}(\text{AF}^i_t, \text{Trans}^i_t, t) = \delta(\text{AF}^i_t = a^i_j)\delta(\text{Trans}^i_t = v) \quad (8)$$

where r is an asynchrony configuration vector as defined in Section III and v is a boolean value indicating the presence or absence of a state transition.

The deterministic factors, on the other hand, are binary (zero-one) functions, whose only purpose is to ensure that invalid values of the variables $\mathbf{y}$ are assigned zero probability. For the model in Figure 4, since the sub-word state indices index into the pronunciation of the word, any valid assignment must satisfy the condition that sub-word state indices increment by at most 1 from one frame to the next (equation 9 below); that in the first time frame, the articulators are in the first state of the word (equation 10); that the last frame corresponds to the last state in $\text{pron}(w)$, the pronunciation of the word (equation 11); and, finally, that the articulatory feature value AF^i is the correct value at the current sub-word state (equation 12), where $\text{pron}(w)^{i,s}$ is the value of AF^i in sub-word state s in the pronunciation of the word:

$$0 \leq \text{SubwordState}^i_{t+1} - \text{SubwordState}^i_t \leq 1 \quad (9)$$

$$\text{SubwordState}^i_1 = 1 \quad (10)$$

$$\text{SubwordState}^i_T = |\text{pron}(w)| \quad (11)$$

$$\text{AF}^i_t = \text{pron}(w)^{i,s} \text{ if } \text{SubwordState}^i_t = s \quad (12)$$

For example, the condition in equation 10 would be expressed with the deterministic factor

$$\phi(\text{SubwordState}^i_1, \mathbf{x}, w, 1) = \begin{cases} 1 & \text{SubwordState}^i_1 = 1 \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

B. Simplifying the Model

In pilot experiments, we implemented the CRF of Figure 4 using the GRMM toolkit [19]. However, exact inference in this CRF was prohibitively slow, whereas approximate inference algorithms resulted in low accuracies. We believe that this is due to the fact that the toolkit does not automatically exploit the sparsity that results from deterministic constraints in our model. In this section, we describe how we take advantage of this sparsity to allow us to do fast exact inference in our CRF.

We first observe that a number of deterministic variables (variables that are deterministic given other variables) can be eliminated if we include more general feature functions. Exploiting the fact that AF^i_t and FtrAsyncConfig_t are deterministic given SubwordState^i_t and w, we can restate the feature function in equations 6–8 as

$$f_{i,j,l,m}(\text{SubwordState}^i_t = s, \mathbf{x}, w, t) = g_{l,m}(x_t)\delta(\text{pron}(w)^{i,s} = a^i_j) \quad (6a)$$

$$f_r(\text{SubwordState}^{1:K}_t, t) = \delta((d^{2,1}_t, \ldots, d^{K,1}_t) = r) \quad (7a)$$

$$f_{i,j,v}(\text{SubwordState}^i_t = s, \text{SubwordState}^i_{t+1}, w, t) = \delta(\text{pron}(w)^{i,s} = a^i_j)\delta(\text{SubwordState}^i_{t+1} = s + v) \quad (8a)$$

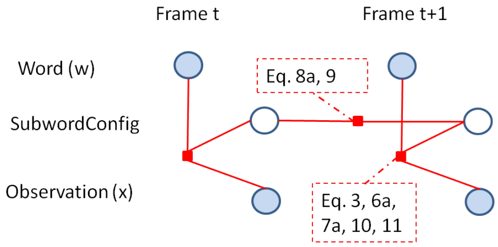


Fig. 5. Factor graph representation of final simplified model after sub-word state variables have been collapsed to obtain a linear chain. The numbers in each dashed box correspond to the equations in the text that implement the corresponding factor.

where $d^{i,j}_t$ is as defined in equation 2. Thus, an equivalent model can be obtained in which pronunciations are represented only through the configurations of the K sub-word state variables. The state variables can then be collapsed into a single variable SubwordConfig_t, whose domain is the cross-product of the individual SubwordState^i_t variables. We stress here that this process of collapsing the articulatory variables into a single variable is purely for convenience; the transformed model is *exactly equivalent* to the original factored CRF model since we ensure that the feature functions (both deterministic and trainable) constructed on this variable in the resulting model are exactly the same as those in the previous model. Finally, we obtain a linear chain CRF, shown in Figure 5, that captures exactly the same dependencies as in the original model in Figure 4. Now the state variable is $\text{SubwordConfig}_t = (\text{SubwordState}^1_t, \ldots, \text{SubwordState}^K_t)$. The factor corresponding to the state transition includes all potentials involving only these variables (eq. 8a, 9). The second factor includes the acoustic statistic-based features (eq. 6a), the asynchrony features (eq. 7a) and the deterministic potentials (eq. 3, 10-11) encoding the dictionary.

C. Efficient Exact Inference

We can perform inference on the factor graph to obtain the marginal distributions over SubwordConfig_t using the standard sum-product algorithm [18], which in this case is equivalent to the standard alpha-beta recursions for linear-chain CRFs [12]. If the pronunciation of word w has W states, $W = |\text{pron}(w)|$, and the maximum allowable asynchrony between any pair of feature streams is M, then it follows from equation 3 that SubwordConfig_t can take on at most $W(2M+1)^{K-1}$ values. Additionally, note that the set of sub-word state configurations s' that can appear in a frame following a given configuration s is much smaller than the full state-space of $W(2M+1)^{K-1}$ allowable configurations: Since each sub-word state can either increment or else retain the same value between adjacent frames, the set of next configurations is of size at most 2^K. A similar analysis holds for the set of previous configurations. Thus, by only summing over valid previous and next configurations in the alpha-beta recursions [12] the overall complexity of inference in the model can be reduced to $O(T2^K W(2M+1)^{K-1})$. Note that collapsing all of the variables at each frame in Figure 4 directly and performing standard inference algorithms for the corresponding linear chain model would incur quadratic complexity in the size of the cross-product of the cardinalities of the variables at each frame $O(T(|W|^K 2^K (2M+1)^{K-1}|AF^1|\cdots|AF^K|)^2)$. Exploiting task-specific constraints allows us to reduce training time by many orders of magnitude: In our experiments, each pass through the training data takes less than a minute.

V. Experiments

Our experiments are performed on a subset of the Switchboard Transcription Project (STP) [20] data. The STP data consists of subsets of the Switchboard telephone conversational speech corpus [21] that was transcribed manually at a detailed phonetic level including diacritics for nasalization, frication, etc.

We use identical data sets to those of [9]: We include all words from the "train-ws96-i" subset if they are one of the 3500 most likely words in Switchboard, excluding partial words and filled pauses. We divide these words into a training set (sets 24-49, consisting of 2941 words and 89,748 frames), development set (set 20, with 165 words and 5365 frames), and test set (sets 21-22, with 236 words and 7037 frames). Following [9], the STP phone transcriptions are stripped of diacritics other than nasalization and converted to articulatory feature labels, which serve as the ground truth. (This is, of course, not ideal since the phone labels are not necessarily an accurate representation of the articulatory configurations.) We parameterize the acoustics by computing 12th-order speaker-normalized PLPs with energy, deltas and double-deltas to obtain a 39-dimensional input representation.

In all of our experiments, we assume that all four tongue features (tongue tip and tongue body location and opening degree) are completely synchronized, the two lip features (location and opening) are synchronized, and the glottis and velum features are synchronized. The models therefore effectively have $K = 3$ articulatory feature streams, which we refer to as L (lips), T (tongue), and G (glottis/velum). We allow a maximum asynchrony of one sub-word state ($M = 1$) between any pair of streams. Considering these constraints, the numbers of distinct L, T and G labels in the data are 8, 25, and 4, respectively. Some example values of these "features" are L = (protruded/ narrow), T = (alveolar/ closed/ uvular/ wide), G = (voiceless/ non-nasal). Additional details of the features and phone-to-feature mappings can be found in [9].

A. DBN and CRF systems

The DBN systems (described in Section III) were implemented using the Graphical Models Toolkit (GMTK) [22]. The distribution of $p(x|L,T,G)$ was modeled as a Gaussian mixture, with the number of Gaussians for each (L,T,G) configuration tuned on the development set. These models were trained with Expectation-Maximization using only acoustics and corresponding words. As a baseline, we implemented a DBN with no asynchrony ($M = 0$), which is almost identical to a phone-based system (DBN-noasync). We also trained a system that allowed asynchrony of up to one sub-word state ($M = 1$): a system identical to the baseline DBN but with one state of allowed asynchrony (DBN-async).

System	L Err Rate (%)	T Err Rate (%)	G Err Rate (%)	Joint Err Rate (%)
DBN-async-Train	11.2	34.7	15.9	47.2
DBN-noasync-Test	9.3	35.2	16.0	40.6
DBN-async-Test	9.6	35.2	16.8	44.0
Tandem-async-Test	9.9	35.4	17.7	43.7
CRF-Lin-Test	9.2	32.8*	14.4*	40.0
CRF-LogPost-Test	9.6	33.4*	14.8*	39.7

TABLE I

Frame error rates (in %) of forced transcription using various models. (*) indicates a statistically significant improvement over the DBN-noasync-Test result at the ($p \leq 0.05$) level using a one-tailed z-test. The first row shows the training error of the system used to generate training data; the remaining rows give test results.

After training, the asynchronous DBN system (DBN-async) was used in forced-alignment mode to produce sub-word state labels for the training set given the identity of each word; these were used as training labels for the CRF-based systems. (The ground-truth labels derived directly from the STP phones cannot be used, as they do not conform to the constraints of the model, such as limited asynchrony.) The CRF-based systems were trained to optimize conditional log-likelihood $p(\mathbf{y}|\mathbf{x}, w)$ using stochastic gradient descent with 'weight averaging': we maintain a running average of the weight vectors resulting from all of the updates up to a given descent step. In our experience and that of others [23], this weight averaging improves performance over using unaveraged weights.

The feature functions for the CRFs can be constructed using arbitrary classifiers; for this study we use multilayer perceptrons trained with the QuickNet toolkit [24]. We train four MLPs, one classifying each of the three articulatory feature streams (L, T, G) and one classifying phones. Each MLP has one hidden layer with a sigmoid activation function and a softmax activation function on the output layer, with the number of units in the hidden layer determined by tuning the MLP frame accuracy on the development set. We consider three statistics derived from the MLPs for constructing CRF feature functions: (a) *posteriors*, the outputs of the MLPs, (b) the *log posteriors* (CRF-LogPost), and (c) *linear outputs*, obtained by removing the final softmax output layer (CRF-Lin). Using MLP posteriors directly produced consistently (slightly) worse performance than the other two cases. To determine whether the use of discriminative classifiers (MLPs) alone accounts for the performance improvement, we also tested a "tandem" style system [25] consisting of the asynchronous DBN in which the acoustic features were projections of the linear outputs of the MLPs onto the top 39 principal components.

B. Results and Discussion

Table I shows the frame error rates for all models measured against the phone-derived articulatory feature labels for each of the three articulatory feature streams, as well as for joint classification of the entire 3-stream configuration. Since the CRFs are trained on labels produced by the asynchronous DBN model on the training set, we present results of the asynchronous DBN system for the training and test set; for all other systems, only test results are given.

The asynchronous DBN's performance is insignificantly different from either the Tandem or DBN-noasync performance in predicting individual L, T, and G labels (first three result columns in Table I). Similarly, the performance of CRF-Lin and CRF-LogPost is comparable. Both CRFs, however, show significant ($p \leq 0.05$) improvement in classifying the T and G streams, compared to each of the baseline models. This is encouraging for two reasons. First, recall that the training labels for the CRF are derived using the baseline DBN system. The training error of the DBN is quite high, so it is encouraging that the CRF trained on these labels can still outperform the DBN. Second, in these pilot experiments we explored a very limited set of CRF feature functions, both in terms of the MLP-derived statistics and in terms of their association with single articulatory feature streams. In previous work, CRFs have been shown to be effective combiners of MLP-based feature detectors [16], so similar improvements may be possible here if we incorporate additional MLP classifiers. Additionally, it is possible to create feature functions that capture higher-order interactions (pairwise or three-way) between the articulatory features. Incorporating such features within our model is straightforward.

In terms of joint classification performance, the CRF-Lin, CRF-LogPost and DBN-noasync systems are all comparable. The tandem system and the asynchronous DBN system, however, performed significantly worse than the DBN-noasync model. One reason for the worse performance of the asynchronous DBN relative to DBN-noasync may be the difficulty of training Gaussian mixtures for the rarer asynchronous states.

VI. Conclusions

We have presented a new CRF that jointly models sequences of several articulatory features, with results on a forced transcription task showing the superiority of our model over a baseline DBN. In future work, we intend to extend the model to account not only for articulatory asynchrony but also for reductions (or, more generally, substitutions) in articulatory positions, as in [10]. Ultimately, we would like to use such a CRF as an end-to-end word recognizer, and we are currently exploring this extension.

Acknowledgment

This research was supported by NSF grants IIS-0905420 and IIS-0905633. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Communication*, vol. 22, pp. 93–111, 1997.

[2] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, pp. 723–742, 2007.

[3] K. Livescu, O. Çetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss, and K. Saenko, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop," in *Proc. ICASSP*, 2007.

[4] C. Hu, X. Zhuang, and M. Hasegawa-Johnson, "FSM-based pronunciation modeling using articulatory phonological code," in *Proc. Interspeech*, 2010.

[5] V. Mitra, H. Nam, C. Espy-Wilson, E. Saltzman, and L. Goldstein, "Robust word recognition using articulatory trajectories and gestures," in *Proc. Interspeech*, 2010.

[6] C. P. Browman and L. Goldstein, "Articulatory phonology: an overview," *Phonetica*, vol. 49, pp. 155–180, 1992.

[7] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, pp. 303–319, 2002.

[8] S. Stüker, F. Metze, T. Schultz, and A. Waibel, "Integrating multilingual articulatory features into speech recognition," in *Proc. Eurospeech*, 2003.

[9] K. Livescu, "Feature-based pronunciation modeling for automatic speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2005.

[10] P. Jyothi, K. Livescu, and E. Fosler-Lussier, "Lexical access experiments with context-dependent articulatory feature-based models," in *Proc. ICASSP*, 2011.

[11] M. Richardson, J. Bilmes, and C. Diorio, "Hidden-articulator Markov models for speech recognition," *Speech Communication*, vol. 41, pp. 511–529, 2003.

[12] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001.

[13] C. Sutton, K. Rohanimanesh, and A. McCallum, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," in *Proc. ICML*, 2004.

[14] K. Livescu, A. Bezman, N. Borges, L. Yung, O. Çetin, J. Frankel, S. King, M. Magimai-Doss, X. Chi, and L. Lavoie, "Manual transcription of conversational speech at the articulatory feature level," in *Proc. ICASSP*, 2007.

[15] P. Ghosh and S. Narayanan, "A subject-independent acoustic-to-articulatory inversion," in *Proc. ICASSP*, 2011.

[16] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 617–628, 2008.

[17] K. Livescu and J. Glass, "Feature-based pronunciation modeling with trainable asynchrony probabilities," in *Proc. ICSLP*, 2004.

[18] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, pp. 498–519, 2001.

[19] C. Sutton, "GRMM: GRaphical Models in Mallet," http://mallet.cs.umass.edu/grmm/, 2006.

[20] S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus," in *Proc. ICSLP*, 1996.

[21] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proc. ICASSP*, 1992.

[22] J. Bilmes and G. Zweig, "The graphical models toolkit: An open source software system for speech and time-series processing," in *Proc. ICASSP*, 2002.

[23] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010.

[24] D. Johnson et al., "ICSI QuickNet software package," http://www.icsi.berkeley.edu/Speech/qn.html, 2004.

[25] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.