STRUCTURED DISCRIMINATIVE MODELS USING DEEP NEURAL-NETWORK FEATURES

R. C. van Dalen, J. Yang, H. Wang, A. Ragni, C. Zhang, M. J. F. Gales

Department of Engineering, University of Cambridge, United Kingdom

ABSTRACT

State-of-the-art speech recognisers employ neural networks in various configurations. A standard (hybrid) speech recogniser computes the likelihood for one time frame and state, using only one out of thousands of possible neural-network outputs. However, the whole output vector carries information. In this paper, features from state-of-the-art speech recognisers are collected per phone given a particular context, and input to a discriminative log-linear model. The log-linear model is trained with conditional maximum likelihood or a large-margin criterion. A key element is the prior on the parameters of the log-linear model. The mean of the prior is set to the point where the performance of the original systems is attained. The log-linear model then provides an additional increase over the state-of-the-art performance of the individual systems.

Index Terms— automatic speech recognition, tandem HMM, hybrid HMM, discriminative log-linear models, structured support vector machines

1. INTRODUCTION

State-of-the-art speech recognisers employ neural networks in various configurations, but always connected to hidden Markov models (HMMs). HMMs make two assumptions. The Markov assumption limits the time horizon of the distribution over states to only one time frame. The conditional independence assumption models the acoustics for time frames as independent given the state sequence. Though these assumptions makes it feasible to train and decode with HMMs, they also limit the power of the model.

This paper therefore uses a structured model, a segmental conditional random field [1, 2], which models whole segments of audio (for words or phones) at once. There is a choice of features for the segments [1, 3, 4, 5, 6]. Good performance can be achieved with features in *generative score-spaces*, extracted from generative models, in particular, HMMs. It is counter-intuitive to extract features from HMMs, the exact model that the argument is to overcome the limitations of. However, it turns out that those features can be more powerful than the models itself, while the structure of HMMs can still be exploited for efficiency while extracting the features [4].

In this paper, features will be extracted from two types of neural-network-based systems: tandem and hybrid systems. This combination will be shown to have interesting properties. The features for single phones are also extracted from competitor phones. This uses more outputs from the neural network than the one that traditional HMMs use. This paper will show the features to be in a relevant subspace. Also, since they are related to the HMM systems, it is possible to set an informative prior for the discriminative model centred around the point where performance of the underlying systems can be achieved. The parameters can then be trained to further improve performance from that point.

This paper is structured as follows. Section 2 will introduce the system architecture that this paper will use and compare it with existing systems. Section 3 will discuss the form of structured discriminative models and where the segmental features enter the system. Section 4 will construct interesting features and analyse their properties. Section 5 will report on the experimental results.

2. SYSTEM ARCHITECTURE

A technique that has improved speech recognisers for a long time is system combination. The combination can be performed at various stages. For example, combination of the output word sequences of various speech recognisers [7] uses no knowledge of the workings of the individual systems. Here, however, the combination of components is chosen to reap the advantages that each of them offers.

Figure 1 shows an overview of the system used in this work. The top line illustrates the neural network used to extract *bottleneck features* [8]. This takes one frame of features extracted directly from the audio, and is trained on a target vector that is zero except for one 1 (one-out-of-K coding) indicating the frame label, a context-dependent state. The architecture of the network is such that one hidden layer has a small number of nodes, which forces the representation of the input to be parsimonious. The output from this "bottleneck" layer, not the output layer, is therefore used in the rest of the system.

The second line houses a *tandem* hidden Markov model (HMM) system [9], so called because its inputs are traditional PLP and pitch features as well as features from the bottleneck layer of the neural network. Since Gaussian distributions are straightforward to manipulate, they allow adaptation to speaker (or acoustic environment), in the form of a linear transformation (constrained maximum-likelihood linear regression or CMLLR) [10]. After being transformed, the features are input to a pool of Gaussian mixture models attached to a hidden Markov model, which are trained together using standard extended Baum–Welch estimation.

The third line illustrates a *hybrid* HMM system [11], which attaches a neural network to an HMM. Hybrid HMMs often take filter-bank features as their inputs. In this work, on the other hand, it takes the same transformed input as the tandem system. It is therefore referred to as a *stacked hybrid* system. The same speaker-dependent

This work was supported in part by EPSRC Project EP/I006583/1 (Generative Kernels and Score Spaces for Classification of Speech) within the Global Uncertainties Programme, in part by a Google Research Award, and in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. The authors would like to thank the LORELEI team for providing the multilingual deep neural network features.

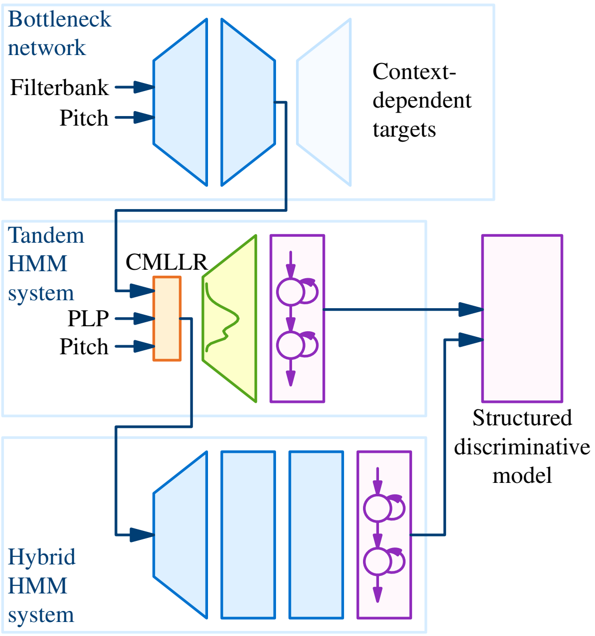


Fig. 1: The architecture of the system.

transformation from the tandem system can be applied to the features for the hybrid system to normalise them.

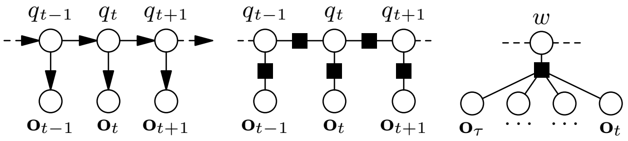
In the final part of the diagram, scores from the tandem and hybrid HMMs are combined in a log-linear model. Unlike earlier work [12, 13, 14] which combined scores for individual frames, this paper will combine scores for whole context-dependent phones. This allows more interesting feature-spaces (see section 4) that use information about all phones in a specific context.

3. STRUCTURED DISCRIMINATIVE MODELS

Speech recognition is a sequence-to-sequence classification task: a variable-length sequence of audio comes in, and a variable-length sequence of words must be inferred. The word sequence is asynchronous with the audio frame sequence, so along with the word sequence, the segmentation of the audio must be inferred. This conundrum is traditionally sidestepped by inferring a symbol sequence that is synchronous with the sequence of audio frames, and mapping deterministically it to a sequence of words. The symbol sequence contains states of a hidden Markov model, and the state space is made very large, to distinguish between word sequences.

An alternative option is to explicitly introduce structure, in this case by explicitly segmenting the audio into words (or phones). This work will use a *structured discriminative model* (see [15] for an in-depth discussion), which models the word sequence and the segmentation of the audio into words, conditional on the audio. The conditional distribution will be a log-linear model. This type of model is sometimes called a segmental conditional random field [2].

Figure 2 illustrates the difference in terms of graphical models in two steps. Figure 2a shows a model of states q_t over time and observations $\mathbf{o}_t$. The graph shows that the distribution factorises into transition probabilities $P(q_t|q_{t-1})$ and observation probabilities $P(\mathbf{o}_t|q_t)$ It is a directed model, which indicates that these distributions are normalised.



(a) $p(\mathbf{q},\mathbf{O})$ modelled with a hidden Markov model.

(b) $P(\mathbf{q}|\mathbf{O})$ modelled with a linear-chain conditional random field.

(c) $P(\mathbf{w},\mathbf{s}|\mathbf{O})$ modelled with a segmental conditional random field.

Fig. 2: Probabilistic models for speech recognition.

Figure 2b shows a linear-chain conditional random field (CRF). One of the two difference with an HMM is visible: the arrows have been replaced with undirected edges to factors, the black boxes. These factors replace the conditional distributions, but are not required to be normalised. The other difference is not visible. Though it is certainly possible for Figure 2b to illustrate a *Markov random field*, a distribution $p(\mathbf{q},\mathbf{O})$ over all variables, here it is a conditional random field [16], which gives the conditional distribution $P(\mathbf{q}|\mathbf{O})$.

The rightmost graph, in Figure 2c, illustrates a segmental conditional random field. This is again a conditional model, which gives the distribution over words $\mathbf{w}$ as well as the segmentation $\mathbf{s}$. The graphical model here is merely one instantiation: as the segmentation changes, the graphical model changes too.

In this work, the conditional distribution will be a log-linear model. The most important aspect of this is the feature function $\boldsymbol{\phi}$, which produces values in a *joint* feature space. This means that it takes both the observations $\mathbf{O}$ and the class $(\mathbf{w},\mathbf{s})$ as arguments, and returns a fixed-length vector describing the match between the two. It is only feasible to use such a model if the feature function is defined so that the structure of the sentence can be exploited. Given observation $\mathbf{O}$ and parameter vector $\boldsymbol{\alpha}$, the probability of the word sequence $\mathbf{w}$ and the segmentation $\mathbf{s}$ is given by a log-linear model:

$$P(\mathbf{w},\mathbf{s}|\mathbf{O};\boldsymbol{\alpha}) \triangleq \frac{1}{Z(\mathbf{O},\boldsymbol{\alpha})}\exp\Big(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\mathbf{s})\Big), \tag{1}$$

where $Z(\mathbf{O},\boldsymbol{\alpha}) \triangleq \sum_{\mathbf{w},\mathbf{s}}\exp\big(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\mathbf{s})\big)$ is a constant that ensures that the conditional distribution is normalised. Since $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\cdot)$ is a dot product, the parameter vector $\boldsymbol{\alpha}$ contains one parameter for each element of the feature vector. This one-to-one mapping will make it possible to set an informative prior, as section 4 will detail.

The feature vector should be constructed to make decoding and training feasible. Additionally, in this work the feature vector will be set so that log-linear model can be related to standard systems. It is therefore a concatenation of a feature vector for the acoustic model $\boldsymbol{\phi}_{\mathsf{am}}(\cdot)$ and one for the language model $\boldsymbol{\phi}_{\mathsf{lm}}(\cdot)$:

$$\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\mathbf{s}) \triangleq \begin{bmatrix} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O},\mathbf{w},\mathbf{s}) \\ \phi_{\mathsf{lm}}(\mathbf{w}) \end{bmatrix}. \tag{2}$$

The parameter vector $\boldsymbol{\alpha}$ can be split similarly into $\boldsymbol{\alpha}_{\mathsf{am}}$ and α_{lm}.

The language model feature here has only one dimension, and is set to the logarithm of the probability that an n-gram language assigns to the word sequence, $P(\mathbf{w})$. The contribution of the language model to the conditional probability is therefore $\exp(\alpha_{\mathsf{lm}}\cdot\phi_{\mathsf{lm}}(\mathbf{w})) = \exp(\alpha_{\mathsf{lm}}\cdot\log P(\mathbf{w})) = P(\mathbf{w})^{\alpha_{\mathsf{lm}}}$.

The acoustic feature is defined as a sum of segment features, each in the joint feature space, for each word w_i and corresponding audio segment $\mathbf{O}_{s_i}$:

$$\boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}, \mathbf{w}, \mathbf{s}) \triangleq \sum_{i}^{|\mathbf{w}|} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}_{s_i}, w_i). \tag{3}$$

Since the summation in this equation is substituted into the log-linear model in (1), the contribution of each of the terms is multiplied in the conditional distribution:

$$\begin{aligned} \exp\left(\boldsymbol{\alpha}_{\mathsf{am}}^{\mathsf{T}} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}, \mathbf{w}, \mathbf{s})\right) &= \exp\Big(\boldsymbol{\alpha}_{\mathsf{am}}^{\mathsf{T}} \textstyle\sum_{i}^{|\mathbf{w}|} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}_{s_i}, w_i)\Big) \\ &= \prod_{i}^{|\mathbf{w}|} \underbrace{\exp\Big(\boldsymbol{\alpha}_{\mathsf{am}}^{\mathsf{T}} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}_{s_i}, w_i)\Big)}_{\text{factor in graphical model}}, \end{aligned} \tag{4}$$

where each of the terms is a factor in Figure 2c. Section 4 will discuss how the segment-dependent acoustic features are constructed.

For decoding, it is in theory possible to marginalise out the segmentation. However, this is infeasible, so instead the segmentation and word sequence that maximise the posterior in (1) will be found:

$$\begin{aligned} \arg\max_{\mathbf{w},\mathbf{s}} P(\mathbf{w}, \mathbf{s} | \mathbf{O}; \boldsymbol{\alpha}) &= \arg\max_{\mathbf{w},\mathbf{s}} \frac{1}{Z(\mathbf{O}, \boldsymbol{\alpha})} \exp\Big(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}, \mathbf{s})\Big) \\ &= \arg\max_{\mathbf{w},\mathbf{s}} \Big(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}, \mathbf{s})\Big) \\ &= \arg\max_{\mathbf{w},\mathbf{s}} \left[\alpha_{\mathsf{lm}} \phi_{\mathsf{lm}}(\mathbf{w}) + \sum_{i} \boldsymbol{\alpha}_{\mathsf{am}}^{\mathsf{T}} \boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O}_{s_i}, w_i)\right]. \end{aligned} \tag{5}$$

This maximisation can be performed exactly (as in [3, 4]), or by constraining the hypotheses to those found in a lattice, which will be done in this work.

3.1. Training criteria

There are a number of ways in which a log-linear model can be trained. The training criterion used on standard HMMs, the likelihood of observations and transcription, is unavailable since the probability of the observations is not modelled. However, it is possible to optimise the likelihood of the correct word sequence and segmentation $(\mathbf{w}_{\text{ref}}, \mathbf{s}_{\text{ref}})$ given the observations, the conditional likelihood. This is possible for HMMs as well, when it is often called "maximum mutual information". The criterion can be written, summing over all utterances r,

$$\begin{aligned} \boldsymbol{\alpha}^{*} &= \arg\max_{\boldsymbol{\alpha}} \sum_{r} \log P\big(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{s}_{\text{ref}}^{(r)} \big| \mathbf{O}^{(r)}; \boldsymbol{\alpha}\big) \\ &= \arg\max_{\boldsymbol{\alpha}} \sum_{r} \Bigg[\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\text{ref}}^{(r)}, \mathbf{s}_{\text{ref}}^{(r)}) \\ &\qquad - \log\Bigg(\sum_{\mathbf{w},\mathbf{s}} \exp\Big(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}, \mathbf{s})\Big)\Bigg)\Bigg]. \end{aligned} \tag{6}$$

This criterion can be maximised with a form of expectation–maximisation.

Another criterion that is frequently used for speech recognition is minimum Bayes risk (MBR). This uses a loss function $\mathcal{L}(\mathbf{w}_{\text{ref}}, \mathbf{w})$ between the reference word sequence and segmentation and the competitors:

$$\boldsymbol{\alpha}^{*} = \arg\min_{\boldsymbol{\alpha}} \sum_{r} \sum_{\mathbf{w}} \mathcal{L}(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{w}) P\big(\mathbf{w}, \mathbf{s} \big| \mathbf{O}^{(r)}; \boldsymbol{\alpha}\big). \tag{7}$$

Though it is possible to use this criterion for log-linear models, in this paper it will only be used to train HMM systems, in section 5.

A third training criterion, which will be used in the next two sections, is the maximum margin criterion. This aims to improve the margin between the reference transcription and the most competing sequence $\mathbf{w}$. This margin gives a trade-off between the cost and the logarithm of the likelihood ratio between the reference and the competitors:

$$\begin{aligned} \boldsymbol{\alpha}^{*} = \arg\min_{\boldsymbol{\alpha}} \sum_{r} \Bigg[& \max_{\mathbf{w} \neq \mathbf{w}_{\text{ref}}^{(r)}, \mathbf{s}} \mathcal{L}(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{w}) \\ & - \log\left(\frac{P\big(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{s}_{\text{ref}}^{(r)} \big| \mathbf{O}^{(r)}; \boldsymbol{\alpha}\big)}{P\big(\mathbf{w}, \mathbf{s} \big| \mathbf{O}^{(r)}; \boldsymbol{\alpha}\big)}\right)\Bigg]_{+}. \end{aligned} \tag{8}$$

Here, $[\cdot]_{+}$ is the hinge-loss function, and the margin is defined with a loss function and the log-posterior ratio [17].

This criterion is the same as for structured SVMs, as is decoding as per (5). Known algorithms for structured SVMs can therefore be applied. A Gaussian prior $p(\boldsymbol{\alpha})$ is usually introduced into the training criterion [17] by adding a term $\log p(\boldsymbol{\alpha}) = \log \mathcal{N}(\boldsymbol{\mu}_{\alpha}, C\mathbf{I}) = K - \frac{1}{2C}\|\boldsymbol{\alpha} - \boldsymbol{\mu}_{\alpha}\|_2^2$ (with K constant) to the criterion. Substituting (1) into (8) and cancelling out the normalisation term yields the following convex optimisation:

$$\begin{aligned} & \arg\min_{\boldsymbol{\alpha},\xi} \frac{1}{2}\|\boldsymbol{\alpha} - \boldsymbol{\mu}_{\alpha}\|_2^2 + \frac{C}{R}\xi \\ & \text{s.t. } \forall \text{ competing hypotheses } \mathbf{w}: \\ & \boldsymbol{\alpha}^{\mathsf{T}} \sum_{r} \Big[\boldsymbol{\phi}\big(\mathbf{O}^{(r)}, \mathbf{w}_{\text{ref}}^{(r)}, \mathbf{s}_{\text{ref}}^{(r)}\big) - \boldsymbol{\phi}\big(\mathbf{O}^{(r)}, \mathbf{w}, \mathbf{s}\big)\Big] \\ & \qquad \geq \sum_{r} \mathcal{L}(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{w}^{(r)}) - \xi, \end{aligned} \tag{9}$$

where $\xi \geq 0$ is the "slack variable", introduced to replace the hinge loss. (9) can be solved using the cutting-plane algorithm [18].

4. ACOUSTIC FEATURE SPACE

The previous section has explained how a segmental conditional random field can be parametrised as a log-linear model, where features are extracted for each word or phone. These segment features are summed in feature space, or equivalently, the contribution to the probability multiplied. This section will discuss how this work will compute the features per segment. In this work, the features will be derived from HMM models, which means they are in a generative score-space. The exact features will be described in three steps. The first step is to define the feature space and the mean of the prior in such a way that the scores approximate the original HMM's. The second step is to include features derived from competing phones instead of only the ones in the hypothesis. The third step is to combine the features extracted from two different systems (here, tandem and hybrid systems).

So far, the classes for the log-linear model have been assumed to be sequences of words w, but from this section onward, sequences of context-dependent phones (u, c) will be used, where u is the phone and c the context, e.g. the identity of left and right phone.

To allow each parameter to apply to only one phone, for each segment the feature vector contains mostly zeroes. Only the portion of the vector dedicated to the hypothesised phone is non-zero. For phone with context (u, c) and segment $\mathbf{O}$, $\phi'(\mathbf{O}, (u, c))$ returns that non-zero part of the feature vector. The whole feature vector can be constructed using the Kronecker delta $\delta_v(u)$, which returns 1 if $u = v$ and otherwise 0:

$$\boldsymbol{\phi}_{\mathsf{am}}(\mathbf{O},(u,c)) \triangleq \begin{bmatrix} \delta_1(u)\,\phi'(\mathbf{O},(1,c)) \\ \vdots \\ \delta_V(u)\,\phi'(\mathbf{O},(V,c)) \end{bmatrix}; \quad \boldsymbol{\alpha}_{\mathsf{am}} \triangleq \begin{bmatrix} \boldsymbol{\alpha}_{\mathsf{am},1} \\ \vdots \\ \boldsymbol{\alpha}_{\mathsf{am},V} \end{bmatrix}. \tag{10}$$

The parameter vector $\boldsymbol{\alpha}_{\mathsf{am}}$ is split up in parallel with feature vector $\boldsymbol{\phi}_{\mathsf{am}}(\cdot)$, with $\boldsymbol{\alpha}_{\mathsf{am},u}$ the parameters specific to phone u.

This paper uses generative score-spaces, which means that the feature vectors are derived from the log-likelihoods of generative models. In speech recognition, the standard generative model is the hidden Markov model. Denote the likelihood for phone u with $l(\mathbf{O}; u, c)$. The first type of feature vector to be discussed makes the score closely related to that of the underlying HMM. Each phone-specific feature is just one-dimensional, with the log-likelihood of that phone given by the phone HMM:

$$\phi'(\mathbf{O},(u,c)) \triangleq \big[\log l(\mathbf{O}; u, c)\big]; \qquad \boldsymbol{\mu}_{\alpha_{\mathsf{am},u}} \triangleq \big[\,1\,\big]. \tag{11}$$

Here, $\boldsymbol{\mu}_{\alpha_{\mathsf{am},u}}$ is the slice of the mean of the prior (as in section 3.1) that applies to $\boldsymbol{\alpha}_{\mathsf{am},u}$. If $\boldsymbol{\alpha}_{\mathsf{am},u}$ is set equal to the prior mean, i.e. set to 1, the score that the log-linear model assigns is related to that of the underlying HMM with Viterbi. The difference is that inside the phone segment, the weights of all paths through the states are summed, whereas the Viterbi algorithm applied to an HMM just counts the single highest-likelihood path within phone HMMs.

4.1. Features from competing phones

Additional features that can be extracted from an HMM system are the log-likelihoods that it would assign to each of the competitor phones. Here, the competitors v are compared in the same phone context c. All competitors' log-likelihoods are added to the phone-specific part of the feature vector:

$$\phi'(\mathbf{O},(u,c)) \triangleq \begin{bmatrix} \log l(\mathbf{O}; 1, c) \\ \vdots \\ \log l(\mathbf{O}; V, c) \end{bmatrix}; \quad \boldsymbol{\mu}_{\alpha_{\mathsf{am},u}} \triangleq \begin{bmatrix} \delta_u(1) \\ \vdots \\ \delta_u(V) \end{bmatrix}. \tag{12}$$

The mean of the prior $\boldsymbol{\mu}_{\alpha_{\mathsf{am},u}}$ is all zeroes, except for a 1 for $\log l(\mathbf{O}; u, c)$. If the parameters are set equal to the prior mean, therefore, the resulting score is exactly the same as in (11), and again related to the underlying HMM through the same argument. However, there is more opportunity for optimisation. Note that the division of the whole feature vector into phones in (10) has not changed. The length of the complete feature vector is therefore V^2. Thus, the parameters of the log-linear model are tied for the same phone across different contexts. This tying structure is different from the tying structure of the HMM.

4.2. Features from multiple systems

One of the properties that makes log-linear models attractive is that features can be straightforwardly added, by appending them to the feature vector. In this paper, the interest is in combining features extracted from a tandem HMM and a hybrid HMM. The likelihood given by the tandem HMM is written $l_{\mathsf{T}}(\mathbf{O}; u, c)$, and the one given by the hybrid HMM $l_{\mathsf{H}}(\mathbf{O}; u, c)$. The phone-specific features for the log-linear model then is formed by (12) applied to both systems:

$$\phi'(\mathbf{O},(u,c)) \triangleq \begin{bmatrix} \log l_{\mathsf{T}}(\mathbf{O}; 1, c) \\ \vdots \\ \log l_{\mathsf{T}}(\mathbf{O}; V, c) \\ \log l_{\mathsf{H}}(\mathbf{O}; 1, c) \\ \vdots \\ \log l_{\mathsf{H}}(\mathbf{O}; V, c) \end{bmatrix}; \quad \boldsymbol{\mu}_{\alpha_{\mathsf{am},u}} \triangleq \begin{bmatrix} \frac{1}{4}\cdot\delta_u(1) \\ \vdots \\ \frac{1}{4}\cdot\delta_u(V) \\ 1\cdot\delta_u(1) \\ \vdots \\ 1\cdot\delta_u(V) \end{bmatrix}. \tag{13}$$

Here, the mean of the prior $\boldsymbol{\mu}_{\alpha_{\mathsf{am},u}}$ is found by stacking means like in (12), but scaled. The scaling factors $\frac{1}{4}$ and 1 are examples taken from [14], which uses these values for frame-level combination.

4.3. Analysis of the feature space

It is interesting to relate the feature space of the log-linear model to the acoustics. Like the well-known support vector machine (SVM), a log-linear model has linear decision boundaries, and like with the SVM one way of getting around this is to introduce a non-linear feature space. The standard way of doing this for SVMs is using kernels. This is possible for speech recognition [19] but not straightforward. Instead, this paper uses the log-likelihood feature space discussed in the previous section. The previous section has also pointed out that the log-likelihood score-space allows for meaningful priors. This section will analyse the nature of the features further.

First, features extracted from the tandem system. To aid the analysis of the relationship between the frame-level acoustics, the contribution of only one frame will be considered. Additionally, the assumption will be made that in the likelihood for each phone, for each time step only one Gaussian dominates. For each frame $\mathbf{o}_t$, the log-likelihood of a Gaussian (with index 1, say) with mean $\boldsymbol{\mu}_1$ and diagonal covariance $\boldsymbol{\Sigma}_1$ can be written as a dot product. The left-hand vector has values dependent on the parameters of the Gaussian, and the right-hand vector depends on the observation $\mathbf{o}_t$:

$$\log \mathcal{N}(\mathbf{o}_t;\, \boldsymbol{\mu}_1,\, \boldsymbol{\Sigma}_1) = \begin{bmatrix} k_1 \\ \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 \\ -\frac{1}{2}\operatorname{diag}\big(\boldsymbol{\Sigma}_1^{-1}\big) \end{bmatrix}^{\mathsf{T}} \cdot \begin{bmatrix} 1 \\ \mathbf{o}_t \\ \operatorname{diag}\big(\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}}\big) \end{bmatrix}, \tag{14}$$

where k_1 is a constant, which is a function of $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$, and $\operatorname{diag}(\cdot)$ gives the diagonal of a matrix as a vector. $\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1^{-1}$ are known as the "natural parameters" of the Gaussian. The space that is produced has zero-, first- and second-order statistics from $\mathbf{o}_t$. The feature space in (12) contains the log-likelihoods of all competitor phone HMMs. The different Gaussians then span a subspace which the log-linear parameters $\boldsymbol{\alpha}_{\mathsf{am},u}$ apply to:

$$\boldsymbol{\alpha}_{\mathsf{am},u}^{\mathsf{T}} \cdot \begin{bmatrix} k_1 & \big[\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\big]^{\mathsf{T}} & -\frac{1}{2}\operatorname{diag}\big(\boldsymbol{\Sigma}_1^{-1}\big)^{\mathsf{T}} \\ k_2 & \big[\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\big]^{\mathsf{T}} & -\frac{1}{2}\operatorname{diag}\big(\boldsymbol{\Sigma}_2^{-1}\big)^{\mathsf{T}} \\ \vdots & \vdots & \vdots \\ k_V & \big[\boldsymbol{\Sigma}_V^{-1}\boldsymbol{\mu}_V\big]^{\mathsf{T}} & -\frac{1}{2}\operatorname{diag}\big(\boldsymbol{\Sigma}_V^{-1}\big)^{\mathsf{T}} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \mathbf{o}_t \\ \operatorname{diag}\big(\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}}\big) \end{bmatrix}. \tag{15}$$

However, it must be noted that the feature space is not merely a subspace of a polynomial of the acoustic features. In reality, mixtures of Gaussians are used, which sum probabilities: a non-linear transformation in feature space. Also, the states of the HMMs can each focus on specific parts of the audio. The log-likelihoods of competitors within the same context gives the model access to relevant contrasts. Additionally, and importantly, the prior with the mean in (12) forces the model into the known-good area of parameter space.

A similar analysis can be performed for features extracted from the hybrid system. Again, assume that one state dominates at each time step, and again consider the contribution of a single time frame. The last layer of a hybrid system, as in this paper, is often a *softmax* layer. This means that after the outputs $\mathbf{y}_t$ of the last hidden layer are multiplied with weights $\mathbf{A}$, their exponent is taken.[1] The contribution to one frame, the logarithm of this value, is therefore determined by the rows $\mathbf{a}_u$ that correspond to the context-dependent phone targets connected with the HMM for phone (u, c). The log-linear parameters $\boldsymbol{\alpha}_{\mathsf{am},u}$ apply to this:

$$\boldsymbol{\alpha}_{\mathsf{am},u}^{\mathsf{T}} \cdot \left[\mathbf{a}_1^{\mathsf{T}} \, \dots \, \mathbf{a}_V^{\mathsf{T}} \right]^{\mathsf{T}} \cdot \mathbf{y}. \tag{16}$$

Given the assumptions, the features extracted from the hybrid system, like those extracted from the tandem system, form a subspace projection of a non-linear transformation. For the tandem features, the transformation is fixed; for the hybrid system, on the other hand, the transformation is learnt, and has a much higher dimensionality. Similarly to the features from the tandem system, the use of trained HMMs of all competitors help find an interesting projection.

5. EXPERIMENTS

The structured discriminative models with features derived from neural-network HMMs are tested on two types of corpora.[2] The first is the well-known noise-corrupted AURORA 4 corpus. The second is a selection of languages from the IARPA-funded Babel program, with spontaneous telephone speech.

5.1. AURORA 4

AURORA 4 is a medium-to-large noise-corrupted speech recognition task [20]. The multi-style training data is the WSJ0 subset of WSJ SI284 data [21], for 14 hours of speech, artificially corrupted using 6 types of noise and two microphone conditions at signal-to-noise ratios (SNR) ranging between 10–20 dB. The test set is an artificially corrupted subset of the development set of 1992 November NIST evaluation using 6 types of noise under two microphone conditions with SNRs in the range 5–15 dB. It is split into 4 sets: set A with clean data, set B with data corrupted by 6 types of noise, set C with data corrupted by channel distortion and set D with data corrupted by noise and channel distortion. Evaluation is performed using the standard 5000-word WSJ0 bigram model.

The tandem system uses context-dependent triphone HMMs with 3 emitting states. For the tandem system, the input features are PLP and bottleneck features, 65 dimensions in total. The bottleneck features are also based on PLP features. There are 24k Gaussians in 3033 tied states. The system is trained first with maximum likelihood, and then with the minimum phone error (MPE) criterion.

For the hybrid system, the input features to the neural network is 72 filterbank features, with 11 consecutive frames are concatenated as the input of the DNN. For this corpus, the features for the hybrid system, unlike in figure 1, are separate from those of the tandem system. The neural network layers have size $792 \times 2000^5 \times 3033$ and are trained layer-by-layer with a frame-level cross-entropy criterion, and then with the sequence-level MPE criterion.

HMM	Log-linear	Test set				Avg.
criterion	model criterion	A	B	C	D	
	—	6.78	11.73	15.39	26.21	17.85
ML	CML	7.17	11.59	15.00	26.01	17.69
	large-margin	6.61	11.49	14.93	25.85	17.54
	—	7.15	11.06	14.37	24.54	16.79
MPE	CML	6.95	11.00	14.29	24.39	16.68
	large-margin	7.02	10.92	14.16	24.28	16.60

Table 1: AURORA 4: performance (word error rate) with structured discriminative models on a log-likelihood score-space from a tandem HMM.

HMM	Log-linear	Test set				Avg.
criterion	model criterion	A	B	C	D	
cross-entropy	—	4.09	8.17	8.07	19.41	12.69
	CML	4.02	8.16	7.94	18.64	12.34
	large-margin	3.96	8.03	8.05	18.6	12.27
	—	3.96	7.64	7.79	18.51	12.05
MPE	CML	3.94	7.53	7.36	18.38	11.91
	large-margin	3.66	7.59	7.47	17.99	11.76

Table 2: AURORA 4: performance (word error rate) with log-linear models on a log-likelihood score-space from a hybrid HMM.

The log-linear models are based on the feature space in equation (12) and trained on multi-style data. Lattices are generated using the appropriate system, and each arc of the lattice is annotated with the log-linear features. Training of the parameters of the log-linear models then uses gradient descent on either of two criteria: the conditional maximum likelihood criterion, and a large-margin criterion.

Table 1 shows word error rates with the tandem system and the log-linear model with features from that system. Whether the underlying HMM system is trained with maximum likelihood or with the MPE criterion, the power of the additional features allows the log-linear model to improve performance. This indicates that the feature space for the log-linear model contains information that the HMM systems have no access to.

The same trends occur, but then enlarged, for the hybrid system, in Table 2. This is surprising. The baseline here is particularly strong at 12.05 %. Compared to the tandem system, however, the effective subspace that the parameters of the log-linear model operate in allows more improvement. The improvement over the MPE-trained hybrid is 0.3, which is larger than over the tandem system.

Language	Id	Release
Swahili	202	IARPA-babel202b-v1.0d
Tok Pisin	207	IARPA-babel207b-v1.0b
Lithuanian	304	IARPA-babel304b-v1.0b

Table 3: Babel OP2 languages used in this paper.

[1] The result is also normalised so that the entries add to 1. However, this affects all paths equally so it is left out for the analysis.

[2] Details about the data are available at `https://www.repository.cam.ac.uk/handle/1810/251276`.

5.2. Babel languages

The other type of corpus is the "very limited language packs" (VLLP) of three languages from the IARPA Babel program, Option Period (OP) 2. The main task of the Babel program is keyword spotting, and speech recognition is merely an intermediate process, but here the performance of speech recognisers will be examined. Table 3 details the exact releases used.

The core tool for ASR development is an extended version of the HTK toolkit [22]. The extension mainly includes a complete integration of support for neural networks into HTK [23].

According to the rules of the Babel OP2 program rule, no phonetic lexica may be used. Therefore the systems use graphemic lexica generated using an approach which is applicable to all Unicode characters [24]. For each language, the training data (in the "very limited language pack") is only 3 hours of conversational telephone speech; the test data is 10 hours. The language models are estimated only on the transcripts of the acoustic data.

The HMM systems are exactly those in [14]. The front-end is an MRASTA based neural network [25, 26], which is initially trained with the data from 11 Babel "full language packs", generating 62-dimensional bottleneck features. The input features contain the bottleneck features, 13 PLP coefficients with dynamics of orders 1, 2, and 3, and pitch and probability-of-voicing features (estimated with the Kaldi toolkit [27]) with dynamic coefficients of orders 1 and 2.

Two sets of acoustic models are constructed. One is a speaker-independent (SI) model, which is based on the tandem features. The other is estimated using speaker adaptive training (SAT) [28]. SAT is performed using global constrained maximum-likelihood linear regression (CMLLR) [10] on the maximum-likelihood trained models, followed by MPE. During training, the supervision for CMLLR is the reference. During testing, the SI model with a trigram LM is used to produce hypotheses. The resulting CMLLR transforms are used to obtain speaker-normalised features, which are then input to the Tandem-SAT model. The number of context-dependent states is 1000; each state has an average of 16 components.

As illustrated in Figure 1, the stacked hybrid system use the same features as the tandem system, derived from the CMLLR transforms generated by the tandem SAT system. The input to the hybrid DNN is a concatenation of 9 consecutive feature vectors. The network has layer sizes of $963 \times 1000^4 \times 1000$ and is initialised by layer-wise pre-training with context-independent targets. Fine-tuning is done using the frame-level cross-entropy criterion with context-dependent targets. The number of context-dependent states is the same as in the tandem system. Then, sequence training using the MPE criterion is applied for further improvement.

As a comparison, a "joint" [14] system, which applies log-linear combination of the tandem and hybrid systems at the frame level, is used. The frame-level log-likelihoods from the tandem system are multiplied by $\frac{1}{4}$, and those from the hybrid system by 1. They are then added at the frame level and used instead of the normal HMM log-likelihoods. The language model is kept the same.

The segmental conditional random field is trained as follows. Lattices are produced by decoding with the joint system, and then annotated with features from both tandem and hybrid systems, as in (13). Gradient descent is then used to train the log-linear parameters using the large-margin criterion in (8). At decoding time, lattices from the joint system are rescored, and then the best path is selected.

Table 4 contains word error rates on a few systems and languages. The first two blocks examine the effect of speaker-dependent transformations of the acoustic features for a hybrid system on the performance of the log-linear model. The top block has results of the speaker-independent system. The log-linear model improves performance by 0.6 % absolute. The second block has the same contrast, but now based on a speaker-dependent HMM. The increase of performance from the log-linear model is also 0.6 %. Speaker-dependent transformations therefore do not appear to decrease the usefulness of the features derived from the HMM.

Language	System	Criterion	Word error rate
Swahili	hybrid SI	MPE	61.3
	+ log-linear	large-margin	60.7
	hybrid-SAT	MPE	60.5
	+ log-linear	large-margin	59.9
Tok Pisin	hybrid-SAT	MPE	52.7
	+ log-linear	large-margin	52.5
Lithuanian	hybrid-SAT	MPE	63.2
	+ log-linear	large-margin	62.9

Table 4: Babel languages: performance with structured discriminative models.

System	Criterion	Word error rate
Tandem	MPE	62.5
Hybrid	MPE	60.5
→ Joint	manual	59.4
→ log-linear	manual	59.1
	large-margin	57.9

Table 5: Babel program, Swahili: performance with structured discriminative models from SAT systems.

The rest of the table examines how performance improvement varies for different languages. For Tok Pisin, the improvement is 0.2, and for Lithuanian 0.3. Though the performance increase does vary with languages, there is consistently an increase.

Results of experiments with combining tandem and hybrid systems are in Table 5. The top block repeats the tandem and hybrid HMM baselines. The next block shows the performance of the "joint" system from [14], which performs frame-level combination. The weights are fixed to $\frac{1}{4}$ for the tandem system and 1 for the hybrid. The next line shows the result of the log-linear model that uses the same parameters, as illustrated in (13). Performance improves by 0.3 % compared to the joint system, probably caused by the difference in the assignment of the underlying HMM states to time frames. Firstly, the likelihoods are used, so the sum over all paths instead of the one best path is used, and secondly, those paths can be different between the tandem and hybrid systems, allowing a more optimal alignment for both. When the parameters of the log-linear model are trained, for the bottom line of Table 5, performance increases further by 1.2 %.[3] The absolute improvement over the joint system is 1.5 %.

6. CONCLUSION

This paper has introduced a method for using a structured discriminative model with features extracted from neural-network systems in tandem and hybrid configurations. The features are computed per audio segment. This leverages adaptation with the tandem system, the performance of the hybrid system, and exploits information about competing HMMs. The features are in an interesting space, in which an informative prior can be defined. The overall model shows consistent performance increases over the underlying systems.

[3] This is better than an older version of this paper, due to a bugfix.

7. REFERENCES

[1] Martin Layton, *Augmented Statistical Models for Classifying Sequence Data*, Ph.D. thesis, Cambridge University, 2006.

[2] Geoffrey Zweig and Patrick Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, 2009.

[3] A. Ragni and M. J. F. Gales, "Inference algorithms for generative score-spaces," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2012, pp. 4149–4152.

[4] R. C. van Dalen, A. Ragni, and M. J. F. Gales, "Efficient decoding with generative score-spaces using the expectation semiring," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, May 2013.

[5] Kris Demuynck, Dino Seppi, Dirk Van Compernolle, Patrick Nguyen, and Geoffrey Zweig, "Integrating meta-information into exemplar-based speech recognition with segmental conditional random fields," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[6] Yanzhang He and Eric Fosler-Lussier, "Efficient segmental conditional random fields for one-pass phone recognition," in *Proceedings of Interspeech*, 2012.

[7] Jonathan G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, 1997.

[8] František Grézl, Martin Karafiát, Stanislav Kontár, and Jan Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[9] Hynek Hermansky, Daniel P.W. Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[10] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[11] Hervé A. Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, Springer Science & Business Media, 1994.

[12] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Revisiting hybrid and GMM-HMM system combination techniques," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[13] S. Rath, K. M. Knill, A. Ragni, and M. J. F. Gales, "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," in *Proceedings of Interspeech*, 2014.

[14] H. Wang, A. Ragni, M. J. F. Gales, K. M. Knill, P. Woodland, and C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Proceedings of Interspeech*, 2015.

[15] M. J. F. Gales, S. Watanabe, and E. Fosler-Lussier, "Structured discriminative models for speech recognition: An overview," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 70–81, Nov 2012.

[16] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the International Conference on Machine Learning*, 2001.

[17] S.-X. Zhang, Anton Ragni, and M. J. F. Gales, "Structured log linear models for noise robust speech recognition," *IEEE Signal Processing Letters*, vol. 17, pp. 945–948, 2010.

[18] T. Joachims, T. Finley, and Chun-Nam Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[19] S.-X. Zhang and M. J. F. Gales, "Kernelized log linear models for continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6950–6954.

[20] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR evaluation," Tech. Rep. AU/384/02, Mississippi State University, 2002.

[21] Douglas B. Paul and Janet M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1992.

[22] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland, "The HTK book (for HTK version 3.4)," 2006.

[23] C. Zhang and P. Woodland, "A general artificial neural network extension for HTK," in *Proceedings of Interspeech*, 2015.

[24] M. J. F. Gales, K.M. Knill, and A. Ragni, "Unicode-based graphemic systems for limited resource languages," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2015.

[25] Zoltán Tüske, Joel Pinto, Daniel Willett, and Ralf Schlüter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[26] Zoltán Tüske, David Nolden, Ralf Schlüter, and Hermann Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, 2011.

[28] Tasos Anastasakos, John McDonough, Richard Schwartz, and John Makhoul, "A compact model for speaker-adaptive training," in *Proceedings of the International Conference on Spoken Language Processing*, 1996, pp. 1137–1140.