FURTHER EXPERIMENTS WITH DETECTOR-BASED CONDITIONAL RANDOM FIELDS IN PHONETIC RECOGNITION

Jeremy Morris and Eric Fosler-Lussier

Department of Computer Science and Engineering
The Ohio State University, Columbus, OH, USA
{morrijer,fosler}@cse.ohio-state.edu

ABSTRACT

In our prior work with Conditional Random Fields (CRFs), we have shown that it is possible to achieve results in the phonetic recognition task with a CRF that approach the results of a similarly trained HMM system (but with many fewer parameters), and we have shown that using two different feature sets that are supposedly redundant gives an improvement in the performance of the CRF. In this paper, we explore two new areas with our CRF model. First, we show that by using two feature sets that are just transforms of each other, we achieve an improvement of results in the CRF model. Second, we show that by adding a single pass of realignment to our CRF model training, we achieve an accuracy result in the phone recognition task that is superior to that of an HMM system trained with triphone labels, despite only training the CRF on monophone labels with no explicit triphonic context.

Index Terms— Speech recognition, Stochastic fields

1. INTRODUCTION

Sequential Conditional Random Fields (CRFs) [1] are a mathematical model of sequences much like Hidden Markov Models (HMMs), but with some properties that are different from HMMs that make them useful to examine for Automatic Speech Recognition (ASR) applications. Unlike an HMM, no assumptions of independence among input features are made by a CRF. This allows many possibly redundant and overlapping features to be used as input to the model without the need to worry about decorrelating the inputs.

In the ASR domain, CRFs have been used for building language models [2] and for phone classification [3]. In our own prior work with CRFs, we have shown results for phone recognition using neural net-derived phonological features as inputs [4] as well as combinations of neural net-derived phonetic class posterior features and phonological posterior features as inputs [5]. Our prior work showed that these CRF models, trained only with monophone contexts, could achieve results that were superior to HMM-based models that also used only a monophone context and approached the results of an HMM-based model trained using a triphone context.

In this paper, we have chosen to look at two areas of interest that expand upon prior work. In our previous experiments, our CRF models and HMM-based models were trained with two related but different types of outputs from our neural networks. While the CRF models were trained directly with the posterior outputs of the trained neural networks, the HMM models were trained with a version of these same outputs that were subjected to a non-linear transformation to decorrelate the outputs from one another. While the claim for CRFs is that this decorrelation is unnecessary, we test this claim by examining the effect of using the decorrelated features as inputs instead of the posteriors. In addition, in our prior comparisons to the HMM-based model, the HMM models were allowed to realign their training data during training, while the CRF training data was fixed to be the hand-transcribed phonetic transcripts through the training process. Here we investigate the gains we can achieve by allowing realignment during the training process.

We start with a brief overview of our CRF model. We then discuss our experimental setup and our results from experiments in feature combinations and in embedded training of the CRF model. We end with a discussion of these results and of our planned future work.

2. THE CRF MODEL

In this section we present a summary of our implementation of the CRF model for these experiments. More detailed information on our implementation of this model can be found in [4] and [5].

As described in [1], a Conditional Random Field defines a posterior probability $P(\mathbf{y}|\mathbf{x})$ of a label sequence $\mathbf{y}$ for a given input sequence $\mathbf{x}$. We use the following form of the CRF as our model:

$$P(\mathbf{y}|\mathbf{x}) \propto \exp \sum_i (S(\mathbf{x}, y, i) + T(\mathbf{x}, y, i)) \quad (1)$$

where

$$S(x, y, i) = \sum_j \lambda_j s_j(y, \mathbf{x}, i) \quad (2)$$

and

$$T(x, y, i) = \sum_k \mu_k t_k(y_{i-1}, y_i, \mathbf{x}, i) \quad (3)$$

The CRF is described as a series of *state feature functions* S and *transition feature functions* T. State feature functions are of the form $s(y, x, i)$, where y is an input label, $\mathbf{x}$ is an input observation sequence, and i is an index pointing to a position in the input sequence $\mathbf{x}$. Each state feature function has an associated weight λ, which indicates the importance of this feature/label combination to the overall probability of the label sequence. Transition feature functions are similar to state feature functions, but take the form $t(y_{i-1}, y_i, \mathbf{x}, i)$, which includes the value y_{i-1} of the label immediately preceding the current label y_i. Each transition feature function likewise has an associated weight μ.

For our model, the individual state feature functions use the output of a multi-layer perceptron (MLP) neural network that has been trained to output posterior probabilities for either individual phone classes or phonological feature classes. Our state feature functions are defined as:

$$s_{/t/,f}(y, \mathbf{x}, i) = \begin{cases} NN_f(x_i), & \text{if } y_i = /t/ \\ 0, & \text{otherwise} \end{cases}$$

where NN_f is the output of the MLP for the feature f on the speech frame i used as input. This sample state feature function will evaluate to some non-zero value only when the label on frame i is /t/ and the MLP outputs some non-zero value for feature f when the frame x_i is used as input. Note that this gives us one state feature function for each label/input feature pair.

In this model transition feature functions are binary, evaluating to 1 when the prior label and current label match the values for the defined function and 0 when the labels do not match. Since at testing time we do not know which transitions have occurred between a given pair of frames, we postulate all possible transitions and use the Viterbi algorithm to find the transition path that maximizes equation (1).

3. EXPERIMENTAL SETUP

For these experiments, we make use of the TIMIT acoustic phonetic corpus for all training and testing [6]. Phonetic features and phone classes are extracted through the use of neural networks built using the ICSI QuickNet neural network software package [7]. These neural networks were trained by using 12th order PLP cepstral features plus delta coefficients derived from the TIMIT training set as inputs. The neural networks were given a nine-frame window of these coefficients centered on the middle frame as their inputs. One neural network was trained to identify phone classes, with one output for each of the possible 61 TIMIT phone labels. A set of n-ary neural networks was also trained to classify individual phonological features as shown in Table 1. These features are derived from the International Phonetics Association (IPA) phonetic chart. Labels for these features were derived from the TIMIT phonetic labels via a one-to-one mapping for each feature class represented by the labelled phone.

Table 1. *Phonological features.*

attribute	possible output values
SONORITY	vowel, obstruent, sonorant, syllabic, sil.
VOICE	voiced, unvoiced, n/a
MANNER	fricative, stop, flap, nasal, approx., nas.flap, n/a
PLACE	lab., dent., alv., pal., vel., glot., lat., rhot., n/a
HEIGHT	high, mid, low, lowhigh, midhigh, n/a
FRONT	front, back, central, backfront, n/a
ROUND	round, nonrnd, rndnonrnd, nonrndrnd, n/a
TENSE	tense, lax, n/a

We then use these networks to derive phone class and phonological feature class posteriors for the TIMIT training set. These posterior outputs are used to train the CRF models. To build our models, we use software derived from the Java CRF package found on Sourceforge [8]. This package (and our code) uses a quasi-Newton LBFGS algorithm to perform the gradient minimization used to train the weights for the CRF model. The training process is based on the work done in [9] and uses their version of the forward-backward algorithm to compute the gradient for log-likelihood minimization. Training was performed using the training partition of the TIMIT corpus. A small (17 speaker/136 utterance) development set was split off from the TIMIT test partition and used to determine when the CRF training should be stopped.

After the CRF models have been trained, the remainder of the TIMIT test partition is used for evaluation. Features derived from the test partition are fed into the CRF models and lattices built from these CRF models are decoded using the AT&T FSM toolkit [10]. The single best pass through the lattice is determined and used to compare against the hand labelled master label file. Results from the CRF are mapped from the 61 TIMIT phone labels down to 39 phone labels following [11]. Note that no external duration modelling, phone insertion/deletion penalties, or external language models are applied to the CRF lattice to determine the best path through the lattice.

4. FEATURE COMBINATIONS

In [4], we claimed that, unlike an HMM model, a CRF model should not require us to perform a decorrelation of the input feature vectors into the model.[1] Our comparison of the CRF and the HMM showed similar results for the CRF and the HMM when the HMM was using linear outputs from the neural network that had been decorrelated through a Karhunen-Loeve (KL) transform, while the CRF used the posterior prob-

[1] Rather than decorrelation, one can use full or semi-tied covariance matrices at an additional parameter cost.

ability outputs from the same network. Therefore, we chose to see if the same linear outputs processed with a KL transform would give us the same results as the posterior outputs for the CRF.

Table 2 shows the results. Here we see that for the phone classification output, there is a slight but significant ($p \leq 0.05$) drop in accuracy when we use the transformed linear outputs instead of phone posteriors. In addition, there is a small but significant improvement in using the transformed phonological feature linear outputs in place of the posterior outputs. As an additional experiment, the linear, non-transformed phonological feature outputs were used to try to determine whether the improvement came from the decorrelation, or from the use of the linear outputs themselves. The difference between the non-transformed linear output and the posterior output is significant, but the difference between transformed linear output and untransformed linear output is not significant. This indicates that the gain achieved from changing from the posterior outputs to transformed linear outputs occurs without the transformation and would suggest that it is primarily the linear outputs and not the decorrelation that is giving the improvement.

Table 2. *Phone accuracy comparisons.*

Model	Feature Space	Phone Accuracy	Phone Correct
Phone posteriors	61	67.32	68.81
Phone linear KL	61	66.80	68.45
Phone post. + linear KL	122	68.13	69.77
Phono. posteriors	44	65.45	66.86
Phono. linear KL	44	66.37	67.97
Phono. linear only	44	65.91	68.44
Phono. post. + linear KL	88	67.36	68.94

As described in [5], we have seen gains in our system accuracy by adding in features that are supposedly redundant features. Would using both the transformed linear outputs and the posterior outputs of the same input features give us any additional gain in our system accuracy? As shown in Table 2, both the phone classification output and the feature classification output show significant gains in accuracy when their respective CRFs are trained on both posterior and linear outputs, despite the fact that there was no significant difference between the phone classification CRF performance with either of the two sets of features by themselves.

5. VITERBI TRAINING

In [4], we compared a CRF system for phone recognition against a Tandem HMM system that had been trained with the same inputs. Tandem HMM systems, described in [12], are HMM-based models that use neural network outputs as feature vector inputs. Our accuracy results for the CRF system trained with monophone labels beat the HMM system that used only monophone labels and compared favorably to an HMM system using triphone labels. The CRF system, however, had a number of disadvantages in its training that the HMM system did not have. One of these disadvantages was that the CRF training data was never realigned during the training process, while the HMM training effectively allowed for realignment. As such, the CRF was trained only with labels as they were transcribed by the human transcribers, while the HMM system was allowed to realign this data to make a better fit for training. We wanted to see if realignment of the training data might have an impact on the CRF training.

To test this hypothesis, we first trained CRFs for the data using the hand-transcribed labels derived from the TIMIT transcriptions. The training data was then passed through the CRF recognizer and force aligned to the training data phone label sequence. The best result of the forced alignment was used as the labels for retraining the CRF, using the previously learned weights as seed weights and allowing more iterations of gradient minimization against the new labels to modify the weights.

These results are shown in Table 3. A comparison to the results of a Tandem HMM system trained with triphone labels and with both 4 and 16 gaussian mixture models per state as described in [4] is also shown. The CRF trained with 61 phone class feature outputs and one pass of realignment achieves an accuracy result significantly better ($p \leq 0.05$) than either of the 4 or 16 gaussian mixture Tandem systems. The CRF achieves these accuracy results with far fewer parameters than either Tandem HMM (the CRF has roughly 5200 parameters while the 16 mixture Tandem system uses 1.7 million parameters). For the phonological feature CRF, realignment achieves a result that is slightly worse than the 4 mixture HMM Tandem system, but not significantly worse.

In addition, we show results for the CRF system trained with both posterior and linear features for completeness with the previous section. A comparable version of the HMM model has not been trained as it is not clear how the combination of these features together for the HMM model should be performed – the purpose of using the linear, KL-transformed features for the HMM in the first place was to provide the system with decorrelated features for processing. We do note, however, that the HMM system shown here uses over two million parameters, not including the MLP neural network weights. In contrast, the CRF uses only slightly more than 8200 parameters in addition to the neural network weights and is trained using only monophone labels, yet achieves a significant ($p \leq 0.05$) improvement in performance over three of the four HMM models and achieves the same performance as the 16 mixture, phonological feature model.

Finally as a comparison, we show results for a CRF trained using a set of labels aligned using the 16 mixture model HMM trained for the features. The HMM-aligned system shows results comparable to the CRF trained with TIMIT hand labels and no realignment, indicating that our improve-

Table 3. *Phone accuracy comparisons with realignment.*

Model	Labels	Features	Realign?	Phone Acc.	Phone Corr.
Tandem HMM Phone linear KL (4gmms)	triphones	61	yes	68.07	73.88
Tandem HMM Phone linear KL (16gmms)	triphones	61	yes	69.34	75.62
CRF Phone posteriors	monophones	61	no	67.32	68.81
CRF Phone posteriors	monophones	61	yes	69.92	72.74
CRF Phone post. + linear KL	monophones	122	yes	70.10	73.25
CRF Phone posteriors / HTK alignment	monophones	61	no	66.76	67.54
Tandem HMM Phonological linear KL (4gmms)	triphones	44	yes	68.30	73.58
Tandem HMM Phonological linear KL (16gmms)	triphones	44	yes	69.13	75.00
CRF Phonological posteriors	monophones	44	no	65.45	66.86
CRF Phonological posteriors	monophones	44	yes	67.81	70.97
CRF Phono. post. + linear KL	monophones	88	yes	69.13	72.07
CRF Phonological post. / HTK alignment	monophones	44	no	65.99	66.89

ment is not just due to the change from using hand-aligned TIMIT labels.

6. DISCUSSION AND FUTURE WORK

In this work, we have shown some simple extensions of our CRF model for phone recognition that make use of transformations of the input features. These features are redundant and highly correlated, yet we have shown some significant improvements in recognition accuracy by making use of this redundant data without performing any decorrelation. This supports the work that we showed in [5], where phonological feature posteriors and phone class posteriors were combined to show improved results for the CRF model without applying any form of decorrelation. These two experiments give us some level of confidence that we will be able to add other features extracted from the signal, even if those features provide information that is redundant with the information we already have and not hurt the overall performance.

In addition, we have shown that by allowing realignment of the training data for this system, we can improve our phone recognition accuracy to be superior to that of a triphone-labelled HMM model but without explicitly incorporating any triphone context. Another possible enhancement would be to allow the MLP neural networks to be retrained according to the best alignment of the CRF model, and to run cycles of alternating neural network retraining and CRF retraining.

7. ACKNOWLEDGMENTS

The authors would like to thank Keith Johnson, Anton Rytting, Ilana Bromberg, and Yu Wang for useful discussions of this work and the International Computer Science Institute for providing the neural network software. This work was supported by NSF ITR grant IIS-0427413 and a fellowship from the AFRL/Dayton Area Graduate Studies Institute; the opinions and conclusions expressed in this work are those of the authors and not of any funding agency.

8. REFERENCES

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

[2] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proc. of ACL*, 2004, pp. 48–55.

[3] A. Gunawardana, M. Mahajan, A. Acero, and J. Platt, "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, 2005.

[4] J. Morris and E. Fosler-Lussier, "Combining phonetic attributes using conditional random fields," in *Proc. Interspeech*, 2006.

[5] J. Morris and E. Fosler-Lussier, "Discriminative phonetic recognition with conditional random fields," in *HLT-NAACL Workshop on Compuationally Hard Problems and Joint Inference*, 2006.

[6] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

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

[8] S. Sarawagi, "CRF package for java," `http://crf.sourceforge.net/`, 2004.

[9] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proc. of HLT, NAACL*, 2003.

[10] F. Pereira M. Mohri and M. Riley, "AT&T finite-state machine library," `http://www.research.att.com/sw/tools/fsm/`, 2003.

[11] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. on Acoustics, Speech and Signal Processing*, pp. 1641–1648, 1989.

[12] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2000.