

# Discriminative Phonetic Recognition with Conditional Random Fields

**Jeremy Morris & Eric Fosler-Lussier**  
Dept. of Computer Science and Engineering  
The Ohio State University  
Columbus, OH 43210  
{morrijer, fosler}@cse.ohio-state.edu

## Abstract

A Conditional Random Field is a mathematical model for sequences that is similar in many ways to a Hidden Markov Model, but is discriminative rather than generative in nature. In this paper, we explore the application of the CRF model to ASR processing of discriminative phonetic features by building a system that performs first-pass phonetic recognition using discriminatively trained phonetic features. With this system, we show that this CRF model trained on only monophone labels achieves an accuracy level in a phone recognition task that is close to that of an HMM model that has been trained on tri-phone labels.

## 1 Introduction

In traditional Automatic Speech Recognition (ASR) processing, a Hidden Markov Model (HMM) is used to model the probability that a given input speech signal is the result of a given utterance. First, the speech signal is transformed through a process of feature extraction into a vector of features. Next, the feature vector is associated with a particular acoustic model of speech through the use of a Gaussian Mixture Model (GMM). The output of these GMMs provide information to the HMM which has been previously trained on labelled acoustic model output to allow it to label these outputs.

Recently, attempts have been made to incorporate features determined via discriminative meth-

ods into the ASR process. These attempts have led to systems like Tandem systems (Hermansky et al., 2000), where individual features are extracted from the speech signal using a neural network, and then used to train the GMMs for the acoustic models. These systems have been proven to be effective for speech recognition.

Due to the nature of HMMs, however, individual features are expected to be decorrelated from each other. Features extracted by phonetic class discriminations are inherently correlated, however, both across time and within a single frame. The output of a discriminative classifier outputting posterior probabilities will also be highly correlated across a single frame - a high probability of one particular phone necessarily leads to low probabilities for other phones. Tandem systems attempt to solve this problem by decorrelating the features through methods such as principle components analysis.

Sequential Conditional Random Fields (CRFs) (Lafferty et al., 2001) are a mathematical model of sequences like HMMs, but unlike an HMM, a CRF makes no assumptions about the independence of its observed features. Where an HMM attempts to have a model of a distribution of the likelihood of the observed features, a CRF takes these features as a given and instead directly models the posterior probability of a label sequence given the observations.

In this paper, we propose a simple CRF system that uses the output of a discriminatively trained neural network directly as input features. Prior work in ASR using CRFs has been made in language modelling (Roark et al., 2004) and in phone classification (Gunawardana et al., 2005). Our work differs

from the work on phone classification in two major ways: First, the work in (Gunawardana et al., 2005) uses a set of observed features derived from traditionally extracted features of the speech signal, while our system uses features derived by running these extracted speech features through a neural network trained to associate speech features and phonetic labels. Second, the work on phone classification involves finding the identity of a phone between two known boundary points, while this work examines phone recognition – positing a sequence of phones from the observed speech data including transitions between phones. In our model, the transitions between phones (boundaries) are not explicitly modelled as features, which means that the system must be able to have a means of hypothesizing these hidden boundaries from those features that are observed. Here we propose a simple method for finding these hidden boundaries, and discuss some future ideas on finding features that might help us make these hidden transitions more explicit.

In the next section we discuss how our model is defined in terms of extracted features and CRFs. In the following section we discuss the experimental setup used in our testing and some results from our initial experiments. Finally, we discuss some directions we are proposing for future investigation.

## 2 CRFs and Discriminative Features

For our model, we begin by training a multi-layer perceptron (MLP) neural network on speech data from the TIMIT acoustic phonetic corpus of speech data (Garofolo et al., 1993). The TIMIT corpus is a collection of speech data that has been labelled at the phonetic level. Our MLP network is trained on individual frames of speech data labelled with the appropriate labels for the features we are trying to examine. These labels might be as basic as the phonetic labels for each frame of data, or they might be something more complex. Regardless of the labels, we end up with a vector of posterior probabilities – with one output for each possible label. These vectors and their labels are used as input for training our CRF model.

A CRF defines a posterior probability  $P(\mathbf{y}|\mathbf{x})$  of a label sequence  $\mathbf{y}$  for a given input sequence  $\mathbf{x}$ . For our purposes, the input sequence  $\mathbf{x}$  corresponds to a

series of frames of speech data, while the label sequence  $\mathbf{y}$  is the phone label sequence assigned to the input sequence. Each frame of  $\mathbf{x}$  is assigned exactly one label in  $\mathbf{y}$ .

A CRF is described by a series of *state feature functions*  $s(y, \mathbf{x}, i)$  with corresponding weights  $\lambda$  and a series of *transition feature functions*  $t(y, y', \mathbf{x}, i)$  with corresponding weights  $\mu$ . Here  $y$  and  $y'$  are labels,  $\mathbf{x}$  is a sequence of observations, and  $i$  an index pointing to a position in the sequence  $\mathbf{x}$ .

A *state feature function* is only non-zero if the label  $y$  matches the label that the feature function is defined for at time  $i$  and the observation sequence  $\mathbf{x}$  at time  $i$  shows evidence of a particular attribute that the feature function is defined for. For our model, the output of the state feature function is tied to the result of the MLP output on the frame of speech that we are observing. For example, we might have a state feature function that is defined for the particular phone label /t/ that looks like this:

$$s(y, \mathbf{x}, i) = \begin{cases} NN_t(x_i), & \text{if } y_i = t \\ 0, & \text{otherwise} \end{cases}$$

Where  $NN_t(x_i)$  is the value output by the neural network for the phone label /t/ on the speech frame  $i$  used as input. We can see that this state feature function will evaluate to some non-zero value when our label is /t/ and the output of the MLP for that frame of speech is itself non-zero.

Transition feature functions are defined in a similar manner, but with a dependency on two labels (the previous label and the current label) rather than just the current label. The transition feature function evaluates to a non-zero value only when the labels in the sequence ( $y_i$  and  $y_{i-1}$ ) match the labels defined for the transition function ( $y$  and  $y'$  respectively) and some attribute in the data exists. In addition, in our current model the transition feature functions do not depend on the observed data. Instead, the transition feature functions are binary, evaluating to 1 when the transition between the frames is the transition defined for the function, and 0 when it is not the transition defined for the function.

Given these feature functions, the form of the conditional probability of a label sequence  $\mathbf{y}$  over the observed sequence  $\mathbf{x}$  takes the form:

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

where

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

and

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

At testing time, we do not know whether a particular transition has occurred or not between a given pair of frames. To deal with this, we postulate all possible transitions and use the Viterbi algorithm to find the transition path that maximizes equation (1).

### 3 Experimental Setup & Results

As described above, we used the TIMIT acoustic phonetic speech corpus for all of our training and testing. Phonetic features were extracted by training a set of multi-layer perceptron neural networks (MLPs) using the ICSI QuickNet MLP neural network software (et al., 2004). These neural networks were trained on individual phone labels from the TIMIT corpus, and for a given input produce an output of 61 posterior probabilities. 12th order PLP features plus delta coefficients centered on the current frame were used as input to these MLPs.

In addition to phonetic labels, we have also looked at the results of using phonological features as input to our CRF system (Morris and Fosler-Lussier, 2006). Phonological features break an individual phone down into its component parts based on the manner and place of articulation the mouth uses when forming the phone, whether the phone is a vowel or a consonant, etc. These phonological features are based on the definition of the International Phonetic Association (IPA) phonetic chart and learned via the same MLP process as for the phone labels. The breakdown of features used to describe phones is given in Table 1.

As a baseline for comparison purposes, we compared phone-level accuracies of the system to the results given by a system built using the Tandem model described in (Hermansky et al., 2000): a principal components analysis is performed to decorrelate the linear outputs of the MLP attribute detectors,

and the results are used as input to train a Gaussian-based HMM. For these experiments, the Tandem system was built using HTK (Young et al., 2002) and trained on a modified version of the outputs from our MLP system described above. For both of the systems we used a reduced phoneme labelling for TIMIT of 39 possible outputs instead of the full 61 phone labels as described in (Lee and Hon, 1989).

To build our conditional random fields models, we used software derived from the Java CRF package on Sourceforge (Sarawagi, 2004). This package (and the code that we derived from it) uses a quasi-Newton LBFSG algorithm to perform the gradient minimization used to train the maximum entropy models. The training process as implemented was based on the work done in (Sha and Pereira, 2003), using their version of the forward-backward algorithm to compute the gradient of the log-likelihood for minimization.

Table 2 shows the results of our initial experiments. We can see that the CRF system trained on phone outputs from the MLP system achieves a result close to the result of the Tandem system trained on the same set of outputs. It is important to note that the Tandem system was trained and tested using a set of triphones as its labelled data instead of the simple monophones that the CRF system was trained on. The CRF achieves accuracy results that are close to the results obtained by the Tandem system, but with no benefit of the explicit triphone context given to the Tandem system.

The CRF model does show lower results in the ‘‘Phone Correct’’ column than the Tandem systems do. This is due to the fact that the CRF model as implemented is much more conservative in its generation of output phones than the Tandem system is. The Tandem model generates many more phones than the CRF system does, causing it to get more of these correct, while negatively impacting the accuracy due to the overgeneration. The CRF system, in contrast, is noticeably undergenerating phones at the moment – when the CRF postulates a phone, it is more often right than the Tandem system, but the Tandem system generates many phones in areas where the CRF does not even propose a hypothesis. This situation is controlled in the Tandem system by a controllable penalty weight on the phone transitions, but it is not yet clear to us how to make the

Table 1: *Phonological features.*

attribute	possible output values
SONORITY	vowel, obstruent, sonorant, syllabic, silence
VOICE	voiced, unvoiced, n/a
MANNER	fricative, stop, flap, nasal, approximant, nasalflap, n/a
PLACE	lab., dent., alveolar, pal., vel., glot., lat., rhotic, n/a
HEIGHT	high, mid, low, lowhigh, midhigh, n/a
FRONT	front, back, central, backfront, n/a
ROUND	round, nonround, roundnonround, nonroundround, n/a
TENSE	tense, lax, n/a

Table 2: *Phone accuracy comparisons.*

Model	Label Space	Phone Accuracy	Phone Correct
Tandem (phones)	triphones	67.32	73.81
CRF (phones)	monophones	66.89	68.49
Tandem (features)	triphones	66.85	72.42
CRF (features)	monophones	63.84	65.45
Tandem (phones & features)	triphones	66.80	73.54
CRF (phones & features)	monophones	67.87	69.47
Tandem (phones & features) [top 39]	triphones	67.85	73.34

analogous control for the CRF system, or if it might be better to introduce more features in the observation to control for this problem instead.

Although the results for using the phonological features alone are not quite as good, we note a noticeable improvement in the overall CRF result when the phonological features are combined with the phonetic outputs. The improvement appears to come primarily in the fewer deletions that the system makes when recognizing short vowels, though the system improves somewhat for almost all phones. Interestingly, the system deletes more fricatives when both the phone classifiers and phonetic feature classifiers are used. More analysis and testing of the CRF model needs to be performed to determine what might be causing this. The Tandem system, in contrast, does not improve when the phonological and phonetic feature classifiers are combined together – in fact, directly using all of these features causes the system accuracy to degrade. If the top 39 dimensions of the principal components analysis are used as input to the Tandem system the results improve, but it is notable that the

CRF can make use of all of the features directly – without a need to perform this kind of manipulation.

#### 4 Discussion & Future Extensions

These results show some interesting capabilities for using Conditional Random Field models for ASR. With a simple model, we achieve results that are comparable to those of an HMM using similar input features. In addition, we have made no attempt to impose extra parameters that the HMM makes use of onto the CRF model like phone transition penalties or a language model scaling factor. While parameters like these could be imposed on the model, we feel that adding more information to the learning process itself may be a better way of tuning this model. This is something we would like to explore further.

Also, the model we have build implements only a simple transition model – we are not making use of any observed evidence to decide if a transition has occurred or not. Despite this, our accuracy results are still reasonable. We feel that finding a set of features that provide transition evidence rather than just

state occupancy evidence is going to be important to improving the capability of this model to generate correct phone hypotheses. One idea we are currently exploring is to incorporate delta values between observed frames as transition features between those two frames, giving the model both evidence that a transition has occurred, as well as evidence of what kind of transition has occurred. Another idea involves incorporating the output of a classifier built as a boundary detector into our system to give it a direct observation of boundaries in the signal. We have seen some success in incorporating this type of boundary feature into an HMM model (Wang and Fosler-Lussier, 2006), and we hope to get this implemented and tested soon.

Finally, one of the strengths of the CRF model is its ability to incorporate many different types of features, many of which may be dependent on one another. We would like to move beyond using just phonetic features and add in many different types of features into our model. Speaker characteristics such as speaking rate, gender, or dialect region have been shown to improve pronunciation modeling (Fosler-Lussier, 1999) *inter alia*, and might add information to the model to allow us to better estimate the state of a current speech frame, or the possibility of a transition between frames. A measure of speaking rate for the speaker, for example, that is used as part of the transition feature functions to help us determine when transitions occur would be a useful addition to our model. As another example, an indication of dialect region or gender may help us identify subtle distinctions in how different phones are realized, allowing us to better estimate the identity of the phone. We are also interested in seeing if higher-level features might make an impact on our CRF model. For example, if we had a good detector for syllabic boundaries, could we improve results by adding this boundary information to our system? Would adding stress or pitch measurements provide any extra useful information to our system? These are some of the questions we have started asking with regards to the capabilities of this model.

Along these lines, we have performed some preliminary naive experiments with adding a hidden gender attribute to the models above, with no real change in the results. We suspect that this is due to the influence of the MLP feature extraction – the

MLPs are trained to be “gender neutral” and so may be abstracting away the influence of gender from the resulting detected features. More work needs to be done here to see if creating gender-specific feature detectors will enhance the system in any way, or if the gender neutral MLPs are already performing as well as we can expect.

## 5 Acknowledgments

The authors would like to thank Keith Johnson, Anton Rytting, 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; the opinions and conclusions expressed in this work are those of the authors and not of any funding agency.

## References

- D. Johnson et al. 2004. ICSI quicknet software package. <http://www.icsi.berkeley.edu/Speech/qn.html>.
- J. E. Fosler-Lussier. 1999. *Dynamic Pronunciation Models for Automatic Speech Recognition*. Ph.D. thesis, University of California, Berkeley.
- J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. 1993. Darpa timit acoustic phonetic continuous speech corpus cdrom.
- A. Gunawardana, M. Mahajan, A. Acero, and J. Platt. 2005. Hidden conditional random fields for phone classification. In *Proc. Interspeech*.
- H. Hermansky, D. Ellis, and S. Sharma. 2000. Tandem connectionist feature stream extraction for conventional HMM systems. In *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*.
- J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.
- K. Lee and H. Hon. 1989. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 1641–1648.
- J. Morris and E. Fosler-Lussier. 2006. Combining phonetic attributes using conditional random fields. In *Submitted to Interspeech 2006, in review*.

- B. Roark, M. Saraclar, M. Collins, and M. Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of ACL*, pages 48–55.
- S. Sarawagi. 2004. CRF package for java. <http://crf.sourceforge.net/>.
- F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology, NAACL*.
- Y. Wang and E. Fosler-Lussier. 2006. Integrating phonetic boundary discrimination explicitly into hmm systems. In *Submitted to Interspeech 2006, in review*.
- S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, 2002. *The HTK Book*. Cambridge University Engineering Department. <http://htk.eng.cam.ac.uk>.