INTERSPEECH 2009 BRIGHTON

CRANDEM: Conditional Random Fields for Word Recognition

Jeremy Morris, Eric Fosler-Lussier

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

Abstract

To date, the use of Conditional Random Fields (CRFs) in automatic speech recognition has been limited to the tasks of phone classification and phone recognition. In this paper, we present a framework for using CRF models in a word recognition task that extends the well-known Tandem HMM framework to CRFs. We show results that compare favorably to a set of standard baselines, and discuss some of the benefits and potential pitfalls of this method.

Index Terms: Random fields, automatic speech recognition

1. Introduction

Over the past few years, Conditional Random Fields (CRFs) [1] have been examined as a potential model for automatic speech recognition. CRFs have a number of useful features that make them attractive for use in a speech recognition task. Notably, they lack the conditional independence assumptions that HMMs require between frames of speech, which is useful as the signal processing model for extracting features from a speech signal creates frames of speech that actually are not independent.

A number of different approaches to using CRFs for ASR have been explored in recent years. In [2], the authors present a formulation of the CRF model called a Hidden Conditional Random Field that derives a CRF model from the sufficient statistics of a comparable HMM model over MFCC features for phone classification. In [3], the author makes use of CRFs to perform phone recognition over a high-dimensional but sparse representation of the MFCC feature space using likelihood scores from a large number of Gaussian models. In our own work [4], we make use of CRFs to integrate the output of a variety of phone classifiers and phonological feature classifiers for the phone recognition task.

This previous work with CRFs in ASR has concentrated on tasks at the phonetic level rather than at the word level. There are good reasons for this — since the CRF model is a posterior probability model rather than a likelihood model like an HMM it is not simply a matter of transferring the techniques that work for HMMs directly over to the CRF. In [5] we presented a model for speech recognition that we dubbed a *Crandem* system to attempt to bridge the gap between the work on phone recognition and a true word recognition system. The Crandem system is an extension of the well-known Tandem HMM system for ASR [6]. Where Tandem systems use the outputs of a discriminative MLP classifier as inputs to an HMM, a Crandem system uses the output of a CRF in a similar fashion. Our results showed that a Crandem system could outperform a similar Tandem system on the TIMIT phone recognition task.

In this paper, we describe the first use of CRF models in a word recognition framework. In the next section, we review the structure of the Crandem system and how we use it to move from our initial experiments in phone recognition to word recognition. In the sections that follow, we discuss the design of our experiments on the Wall Street Journal corpus to test the word recognition capabilities of this system, and present our results and an analysis of those results. Finally, we discuss briefly areas where we may be able to extend and expand upon the work presented here.

2. Model

In a standard Tandem system [6], acoustic features (such as PLP coefficients) are used to train a multi-layer perceptron (MLP) classifier to output a phone classification for a particular frame (or, more accurately, a window of frames) of acoustic feature data. The MLP classifier is then used to generate a vector of values, each corresponding to the probability that the frame should be classified with a particular phone label. These vectors are then used as inputs to a standard, generative mixture of Gaussians HMM model.

Our extension of this model [5] works much like the Tandem model outlined above; rather than use the outputs from an MLP as inputs to an HMM we use a CRF to generate our vectors of posterior probabilities. Unlike an MLP classifier, which can take in a single frame of speech and output a probability of a phone label given that frame of speech, a CRF classifier evaluates the probability of an entire sequence of phone labels given the entire sequence of input speech features to provide a global estimate for the probability of an entire utterance.

However, in [7], a variant of the forward-backward algorithm is derived that allows us to use a CRF model to compute a vector of local posterior probabilities for any frame in our utterance. While this variant was initially derived to compute local posterior probabilities for the purposes of training the CRF model, it can easily be repurposed to generate posterior probabilities for our Crandem system. The conditional probability of an utterance given by the CRF model can be defined by:

$$P(Q|X) = \frac{\exp \sum_t (\sum_i \lambda_i s_i(q_t, X, t) + \sum_j \mu_j f_j(q_{t-1}, q_t, X, t)}{Z(X)} \quad (1)$$

Where each s_i (with associated weight λ_i) is a state feature function that associates an input vector X with a phone label q. Additionally, each f_j (with associated weight μ_j) is a transition feature function that associates the vector X with a phone label transition between a pair of states q and q', and the $Z(X)$ term is a normalization constant over all possible paths over the input X. As previously shown, this allows us to write the formula for the local posterior of a given label q_i at time point t for a given input utterance signal X as:

10.21437/Interspeech.2009-569

$$P(q_{i,t}|X) = \frac{\alpha_{i,t}\beta_{i,t}}{Z(X)}, Z(X) = \sum_j \alpha_{j,t}\beta_{j,t} \quad (2)$$

where α and β are defined as a collection of potentials leading up to a particular time step (α) and from that time step to the end of the utterance (β) similar to the alpha-beta recurrence in standard E-M training for HMMs.

Using this recurrence, we can now use our CRF model trained for phone recognition to generate a vector of posterior probabilities suitable for use in a Tandem-like Crandem system. Our prior work showed that this model led to an improvement in TIMIT phone recognition over a standard MLP-Tandem system trained on the same MLP outputs as the CRF model.

3. Experimental Setup

The experiments described here follow the same general structure as those in our previous work using the Crandem structure for phone recognition, and a diagram of the structure of a Crandem system is shown in Figure 1. We use the CSR-I (WSJ0) corpus [8] broken into training, development and test sections to train all of our models. As in our previous work, the inputs to our CRF models are the outputs of a set of MLP ANNs trained to do frame-level phone classification. Local posterior vectors are obtained from the CRF models and fed as input features to a standard HMM system.

Since the WSJ0 corpus does not have phone-level transcriptions for each utterance provided, we first obtain frame-level phone class targets for training our MLPs and our CRF models by aligning an HMM model over the training portion of the WSJ0 corpus using the HTK toolkit [9]. This HMM model is trained using a standard 39-dimensional input vector of 12 MFCCs + energy coefficients along with first and second-order deltas.

MLP ANNs are built using the Quicknet MLP framework [10]. Our MLP networks are trained using a nine-frame window of 12 PLP + energy coefficients along with first and second-order deltas as inputs and the frame label determined by our alignment above as the target. The MLPs are trained over 6488 training utterances across 75 different speakers and using a cross-validation set of 650 utterances from 8 other speakers to estimate MLP convergence. The MLPs have a 4000 hidden unit layer and have a 54 unit output layer.

Our CRF models are trained using the output layer of our MLP ANNs which are used as the s functions in Equation 1. Following our previous work [5], we use linear outputs from the MLPs to train the CRF. The models are trained using a stochastic gradient training method [2]. We train our CRFs using the same labels and the same breakdown of utterances for training and cross-validation used for MLP training, and stop the CRF training when the improvment in the phone-level accuracy of the cross-validation set ceases. Once the CRF models are sufficiently trained, we use the models to generate a vector of local

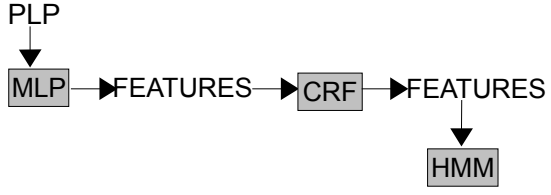

Figure 1: Crandem system overview

Table 1: *WER comparisons across models.*

Model	Training Iterations	Dev WER	Eval WER
MFCC Baseline	NA	9.3%	8.7%
MLP Tandem	NA	9.1%	8.4%
Crandem	1	8.9%	9.4%
Crandem	10	10.2%	10.4%
Crandem	20	10.3%	10.5%

posteriors for each frame of input data. We generate posteriors for the entire training set as well as the development and evaluation sets.

Finally, we train Crandem models using HTK by using the local posteriors generated by our CRF models (per Equation 2) as inputs to an HMM built using HTK. These models are tuned on a development set of 368 utterances from 10 different speakers. We have found that our best results are obtained if we use the log posterior outputs from the CRF models and if we follow the common practice of Tandem HMM models and perform a Karhunen-Loeve transformation and reduce the dimensionality of our input vectors before training our HMM models.

We compare our Crandem system to two other systems as baselines. The first is the standard HMM system built using MFCCs that we used to generate our label files for MLP training above. The second baseline is a Tandem HMM system built using the same linear style outputs of our MLP ANNs used to train our CRF models. Our evaluation is performed over a set of 330 utterances from 8 different speakers for the 5000 word vocabulary WSJ recognition task, and all of our systems use the same bigram language model and the same lexicon for this evaluation - only the input features vary from system to system.

4. Results & Analysis

Table 1 compares our two baseline models to the results of our Crandem system after 1, 10 and 20 iterations of CRF training. Each of the above models has 16 Gaussians per mixture. The MLP Tandem model had its best performance on the development set when the 54 dimensional output of the MLP was reduced to 39 dimensions, while the Crandem systems all had their best performance on the development set when the 54 dimensional output of the CRF local posterior calculations were reduced to 21 dimensions.

As the results show, a single iteration of CRF training using the MLP posteriors as inputs produced an statistically insignificant ($p \geq 0.05$) degradation in the WER over the baseline MFCC system and significant ($p \geq 0.05$) degradation in the WER over the baseline MLP system. Surprisingly, further iterations of CRF training lead to an increase in the error rate rather than a reduction. To check the possibility that our system is behaving in a radically different manner on WSJ than our previous system trained on TIMIT, we examined phone recognition results. Table 2 shows the phone accuracy for each of our above systems on our development set, and makes it clear that the above degradation of word error comes despite a (non-significant) increase in the phone accuracy of the models. Additionally, Table 2 shows that as with our previous work, our Crandem models show an improvement in phone accuracy over decoding directly off of the CRF itself, though unlike our previous work in these experiments our basic MLP Tandem model performs significantly ($p \geq 0.001$) better than our best Crandem model for phone recognition. We have found that tuning our CRF to optimize phone recognition accuracy leads to de-

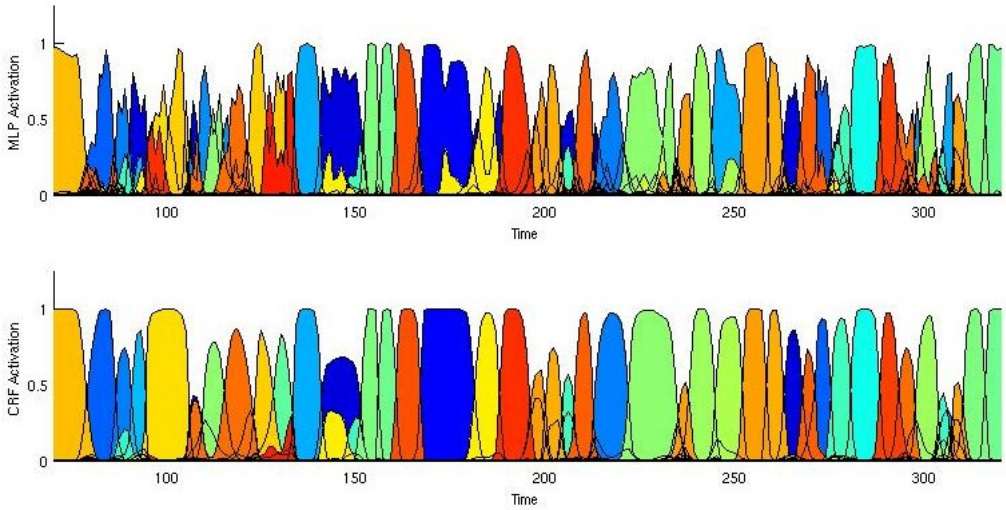

Figure 2: MLP activation vs. CRF activation

Table 2: *Phone accuracy comparisons across models.*

Model	Training Iterations	Dev Phone Accuracy
MFCC Baseline	NA	70.1%
MLP Tandem	NA	75.6%
Crandem	1	72.8%
Crandem	10	72.8%
Crandem	20	72.9%
CRF	1	69.5%
CRF	10	70.6%
CRF	20	71.0%

graded performance for word recognition, and as such we have tuned our CRF to optimize performance for word recognition rather than phone recognition accuracy.

Is it possible that there is some characteristic of the Crandem-style features that make them behave differently for word recognition than for phone recognition? Figure 2 shows an utterance from our development set that compares the initial MLP activation value per frame to the activation value per frame of a set of posterior features from a CRF after one training iteration. We can see from this example that the CRF produces a smoother set of activations than the initial MLP outputs – more of the activations from the CRF produce outputs close to a value of 1.0 and sustain this value over multiple frames of speech. Conversely, the MLP outputs, though smooth in some places, show a much stronger tendency toward jagged peaks – representing areas where the MLP scored a much higher value for a particular phone in a single frame than in surrounding frames. This behavior is observed consistently within our CRF features in the development set as well as within the training set.

The transition features of the CRF model provide an explanation for the smoother graphs of the CRF posterior outputs. In these experiments, only a bias feature is used for each possible transition. However, this single feature is enough to introduce a Markov dependency in the CRF outputs that is not explicitly defined in the MLP outputs. These transition features cause the CRF model to prefer a more gradual change in the magnitude of the various phone output values than even the MLP model with a context window of 9 frames produces.

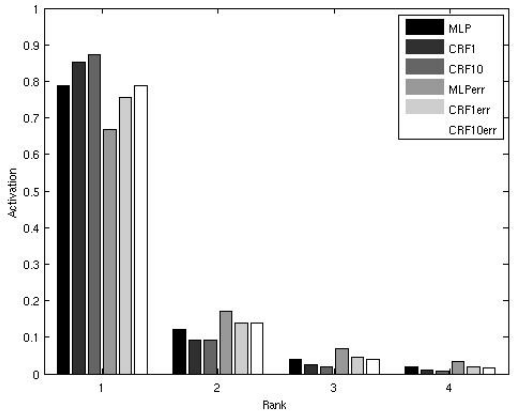

Figure 3: Ranked Average Per Frame activation MLP vs. CRF

Another factor in the smoothing of the output space for the CRF posteriors is that the CRF on average produces higher values for the phone class with the highest score in a single frame than the MLP classifier on the same frame, pushing the peaks of the scores higher on the CRF relative to the MLP. Figure 3 shows the average value of the top five highest valued classes per frame computed over the development set (results from the training set show a similar pattern). Note that the average score of the top ranked class in each frame is higher in the results produced by the CRF than in the results produced by the MLP (0.853 vs 0.788). Conversely, for the lower ranking classes the CRF produces smaller values on average than the MLP (0.092 vs. 0.120 for the average second highest per frame value). This is another factor that leads to the smoothing seen in Figure 2 — the value of the highest scoring class is pushed closer to one while the values of the nearest competitors are pushed closer to zero relative to the MLP outputs. This behavior holds for the top 12 classes in the development set (14 in the training set). The lower ranking classes receive values very close to one another between the MLP and the CRF features and very close to zero overall.

Recall that our Crandem system required a much larger dimensionality reduction on the input features than the Tandem

Table 3: *WER comparisons with MFCCs.*

Model	Training Iterations	WER
MFCC Baseline	NA	8.7%
MLP+MFCC Tandem	NA	7.1%
Crandem+MFCC	1	7.1%
Crandem+MLP	1	8.8%

system. These smoother outputs help to explain this more extreme dimensionality reduction — the overall space being described by the CRF outputs is much less complex in nature, with reduced variation overall, and so fewer dimensions are needed to perform recognition over this new space. In addition, this smoothing effect may help to explain our degraded performance on word recognition after multiple iterations of CRF training. Figure 3 also shows a comparison of the ranked average class values of frames marked as phone errors by the phone recognition process over our development set. The gap between the average value of the top ranked class and the second or lower ranked classes is much larger for the CRF than for the MLP, and gets larger with more iterations of CRF training. This behavior in the features is not surprising – this separation of classes is what is expected from a discriminative model like a CRF. But this behavior suggests a reason for our degraded performance in word recognition. When a phone error is made by the CRF (i.e. when the highest scoring class is not the correct class), these larger distances between the classes make it harder for the system to fit the observation to the Gaussians for the correct class, making it more difficult for the system to choose between alternatives and leading to a word error. Analysis of the development set suggests that at least in some cases this is likely occurring even between the MLP-Tandem system and single iteration Crandem system, though it does not explain all of the differences in word error between the MLP-Tandem system and the Crandem systems.

Tandem systems are often implemented with both MLP and MFCC features concatenated together as inputs. Table 3 compares the results of a Crandem system with MFCC features appended to a similar Tandem system. Here we can see that the MLP-Tandem system and the Crandem system perform comparably, with the difference between the two systems being statistically insignificant and both systems performing significantly ($p \geq 0.005$) better than the baseline system trained only on MFCCs. Table 3 also includes a system trained on both the MLP and CRF outputs concatenated together, which performs insignificantly worse on the evaluation set than the MLP-Tandem system shown in Table 1.

5. Future Work

These results are to our knowledge the first attempt at the use of CRFs for the task of word recognition, and these initial results show a working system for incorporating CRF results into a word recognition system. These results also make clear some of the challenges for using features like these in an HMM-based word recognition system. These features have characteristics different from the MLP features that have been used in Tandem systems, and so although our work is based on extending Tandem systems to CRF features, we may need to explore different approaches to transforming and incorporating our features into an HMM-based system.

Given the nature of the features as described in the previous section, one area we are exploring is the use of pronunciation modeling to improve our results. Better pronunciation modeling could help us overcome the difficulties the model is having when the wrong phone is ranked highest by the CRF model. Another extension worth pursuing is the use of discriminative training for the HMM model itself. Finally we have only done some preliminary exploration of using the CRF model to combine multiple feature types and using these as inputs for the Crandem model – this needs a deeper examination as the ability to combine multiple correlated features together is one of the major attractions of the CRF model.

The mismatch between what CRF models provide the best phone recognition and what CRF models provide the best word recognition is another area that suggests itself for examination. This effect suggests that perhaps there is a mismatch between the goal of maximizing the frame-level accuracy (as the base CRF model attempts to do) and maximizing the word-level accuracy that we actually are seeking to improve. Incorporating word-level training criteria into our CRF training may lead to improved results and we are currently examining ways to undertake this task.

6. Acknowledgment

The authors would like to thank Ilana Bromberg, Chris Brew, Preethi Jyothi, Prateeti Mohapatra, Rohit Prabhavalkar, and Tim Weale 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 NSF CAREER grant IIS-0643901; the opinions and conclusions expressed in this work are those of the authors and not of any funding agency.

7. References

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

[2] A. Gunawardana, M. Mahajan, A. Acero, and J. Platt, "Hidden Conditional Random Fields for phone classification," in *Proceedings of Interspeech*, 2005.

[3] Y. Abdel-Haleem, "Conditional random fields for continuous speech recognition," Ph.D. dissertation, Unifersity of Sheffield, Sheffield, UK, 2006.

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

[5] E. Fosler-Lussier and J. Morris, "Crandem systems: Conditional random field acoustic models for hidden markov models," in *Proceedings of the ICASSP*, 2008.

[6] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *Proceedings of the ICASSP*, 2000.

[7] F. Sha and F. Pereira, "Shallow parsing with Conditional Random Fields," in *Proceedings of HLT, NAACL*, 2003.

[8] J. Garafalo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete," Linguistic Data Consortium, 2007.

[9] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, Cambridge Unveristy Engineering Department, 2002, http://htk.eng.cam.ac.uk.

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