LANGUAGE RECOGNITION USING DEEP-STRUCTURED CONDITIONAL RANDOM FIELDS

Dong Yu [a], *Shizhen Wang* *[b], *Zahi Karam* *[c], *Li Deng* [a]

[a] Microsoft Research, One Microsoft Way, Redmond, WA 98034, USA
[b] University of California, Los Angeles, CA 90095, USA
[c] Massachusetts Institute of Technology, Cambridge, MA 02420, USA
dongyu@microsoft.com, szwang@ee.ucla.edu, zahi@mit.edu, deng@microsoft.com

ABSTRACT

We present a novel language identification technique using our recently developed deep-structured conditional random fields (CRFs). The deep-structured CRF is a multi-layer CRF model in which each higher layer's input observation sequence consists of the lower layer's observation sequence and the resulting lower layer's frame-level marginal probabilities. In this paper we extend the original deep-structured CRF by allowing for distinct state representations at different layers and demonstrate its benefits. We propose an unsupervised algorithm to pre-train the intermediate layers by casting it as a multi-objective programming problem that is aimed at minimizing the average frame-level conditional entropy while maximizing the state occupation entropy. Empirical evaluation on a seven-language/dialect voice mail routing task showed that our approach can achieve a routing accuracy (RA) of 86.4% and average equal error rate (EER) of 6.6%. These results are significantly better than the 82.5% RA and 7.5% average EER obtained using the Gaussian mixture model trained with the maximum mutual information criterion but slightly worse than the 87.7% RA and 6.4% EER achieved using the support vector machine with model pushing on the Gaussian super vector (GSV).

Index Terms — language identification, deep-structure, conditional random field, deep learning, unsupervised learning

1. INTRODUCTION

Significant performance improvement has been achieved in automatic language recognition in the past several years due to the introduction of discriminative classifier methods using shifted-delta cepstral coefficients (SDCCs) as the features [7]. These discriminative methods can be classified into three categories: the SVM techniques using polynomial kernels [2], the Gaussian mixture models (GMMs) trained with the maximum mutual information (MMI) criterion [1], and the SVM techniques using GMM super-vectors as features [4] [3].

In this paper we introduce a new category of discriminative classifier named deep-structured conditional random fields (CRFs) for automatic language recognition. The deep-structured CRF is a multi-layer discriminative model in which the output of the lower layers, together with the original features, is fed into the higher layers as input. The training supervision is provided only at the final layer and the intermediate layers are pre-trained with unsupervised criteria and then fine-tuned with a supervised back-propagation step as explained in Sections 2 and 3.

The deep-structured CRF differs from the GMM-MMI approach in that the GMM-MMI is discriminatively trained generative model while the deep-structured CRF is a discriminative direct model optimized to maximize the conditional likelihood. The deep-structured CRF is also different from the SVM super-vector based approach in that the deep-structured CRF operates on the SDCC features directly.

We have evaluated our approach on a specific language identification task --- a seven-language/dialect voice mail routing task, in which the goal is to dispatch each voice mail to the right automatic speech recognition (ASR) engine for transcribing. The experiments showed that our approach can achieve a routing accuracy (RA) of 86.4% or an average equal error rate (EER) of 6.6%, significantly better than the 82.5% RA and 7.5% average EER obtained using the GMM-MMI approach but slightly worse than the 87.7% RA and 6.4% average EER achieved using the support vector machine (SVM) with model pushing on the Gaussian super vector (GSV).

The rest of the paper is organized as follows. In Section 2, we describe the deep-structured CRF with a focus on its architecture and core ideas. In Section 3 we present an unsupervised pre-training algorithm to learn the intermediate layers by casting it as a multi-objective programming problem aimed at minimizing the average frame-level conditional entropy and maximizing the state occupation entropy at the same time. We report the experimental results in Section 4 and conclude the paper in Section 5.

2. CRF AND DEEP-STRUCTURED CRF

CRFs are *discriminative* models that estimate the class label sequence conditional probabilities directly. The most popular CRF is the linear-chain CRF due to its simplicity and efficiency. If we denote by $\boldsymbol{x} = (x_1, x_2, \cdots, x_T)$ the T-frame observation sequence, and by $\boldsymbol{y} = (y_1, y_2, \cdots, y_T)$ the corresponding state (label) sequence, which may be augmented with a special start (y_0) and end (y_{T+1}) state, the conditional probability of a state (label) sequence $\boldsymbol{y}$ given the observation sequence $\boldsymbol{x}$ is given by

$$p(\boldsymbol{y}|\boldsymbol{x};\Lambda) = \frac{exp\left(\sum_{t,i}\lambda_i f_i(y_t, \boldsymbol{x}, t) + \sum_{t,j}\lambda_j f_j(y_t, y_{t-1}, t)\right)}{Z(\boldsymbol{x};\Lambda)} \quad (1)$$

* Shizhen Wang and Zahi Karam contributed to this work when they were interns at Microsoft Research.

where we have used $f_i(y_t, \boldsymbol{x}, t)$ to represent the observation features and $f_j(y_t, y_{t-1}, t)$ to represent the state transition features. $Z(\boldsymbol{x}; \Lambda)$ is the partition function to normalize the exponential form so that it becomes a valid probability measure. $\Lambda = \{\lambda_i, \lambda_j\}$ are the model parameters that are typically optimized to maximize the L_2 regularized state sequence log-likelihood

$$J(\Lambda, X) = \sum_k \log p\left(\boldsymbol{y^{(k)}} | \boldsymbol{x^{(k)}}; \Lambda\right) - \frac{\|\Lambda\|^2}{2\sigma^2} \quad (2)$$

over the entire training set $\left\{\left(\boldsymbol{x^{(k)}}, \boldsymbol{y^{(k)}}\right) | k = 1, \cdots, K\right\}$, where σ^2 is a parameter that balances the log-likelihood and the regularization term and can be tuned using a development set.

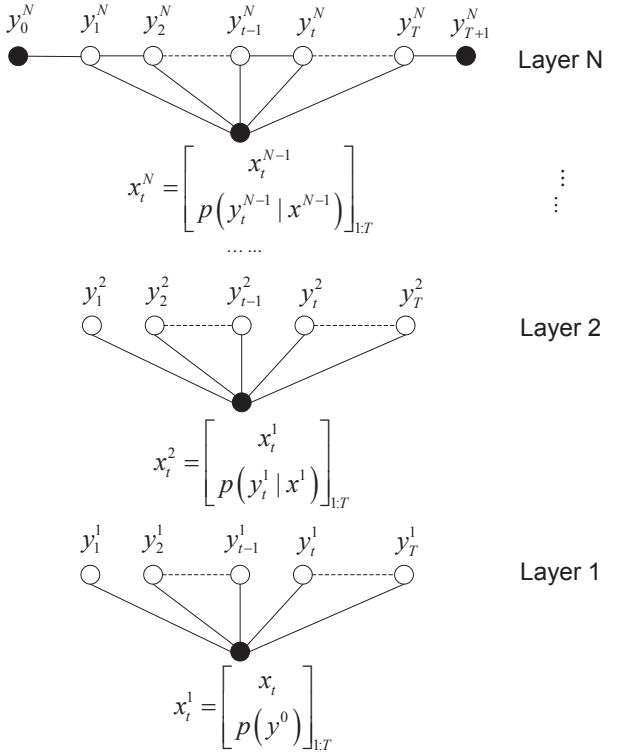

Fig. 1. The graphical representation of the deep-structured CRF. The solid and empty nodes denote the observed and unobserved variables, respectively.

We call the CRF without using the state transition features a zero-th-order CRF, where optimizing the state sequence conditional likelihood becomes equivalent to optimizing the frame-level one. In this case, the frame-level conditional probability can be more efficiently evaluated as

$$p(y_t | \boldsymbol{x}; \Lambda) = \frac{exp(\sum_i \lambda_i f_i(y_t, \boldsymbol{x}, t))}{Z(\boldsymbol{x}; \Lambda)} \quad (3)$$

Note that in the language recognition task typically $y_1 = y_2 = \cdots = y_T$ since most utterances contain only one language. In other words, we use the CRF for a classification problem. However, the CRF-based approach can be naturally used for mixed-lingual language recognition, which is beyond the scope of this paper.

The deep-structured CRF developed and evaluated in this work is a hierarchical model as shown in Fig. 1, where the final layer is a linear-chain CRF and the lower layers are zero-th-order CRFs that do not use state transition features. Note that linear-chain CRFs can also be used in the lower layers. However, we have observed [1] that using zero-th-order CRFs in the lower layers only slightly degrades the accuracy while gaining the benefit of much less computation. In the deep-structured CRF, the observation sequence at each layer is constructed in a way similar to the tandem structure used in some automatic speech recognition systems [5]. Specifically, the observation sequence at layer j consists of two parts: the previous layer's observation sequence $\boldsymbol{x^{j-1}}$ and the frame-level marginal posterior probabilities $p\left(y_t^{j-1} | \boldsymbol{x^{j-1}}\right)$ from the preceding layer $j-1$.

In the deep-structured CRF, both model parameter estimation and state sequence inference are carried out layer-by-layer in a bottom-up manner so that the computational complexity is limited to at most linear to the number of layers used [13][14].

Since we use continuous-valued SDCCs as the features in the language identification task we can achieve better performance by imposing constraints on the distribution of the features, which is equivalent to expanding each continuous feature into several features as discussed in [10][11][12]. Specifically, each continuous feature $f_i(y_t, \boldsymbol{x}, t)$ in the CRF can be expanded to L features

$$f_{il}(y_t, \boldsymbol{x}, t) = a_l\left(f_i(y_t, \boldsymbol{x}, t)\right) f_i(y_t, \boldsymbol{x}, t), \quad (4)$$

where $a_l(.)$ is a weight function whose definition and calculation method can be found in [9][10][11] and the number L needs to be determined based on the amount of training data available.

3. LEARNING OF INTERMEDIATE LAYERS

The number of states at the final layer in the deep-structured CRF is directly determined by the problem to be solved. For example, for a seven-language language recognition task, the final layer would have seven different states, one for each language. Parameter estimation at the final layer is carried out in a supervised manner since the desired output (the true language) is available from the training data. This is not the case, however, for the intermediate layers, which can be considered as abstract internal representations of the original observation with different granularities and can be estimated using either unsupervised or supervised approaches. For example, in [1] we assumed that the number of states at intermediate layers be the same as that in the final layer and that the same label used to train the final layer be used to train the intermediate layers. Although this approach is simple and effective, further performance gain is expected if we allow for a more flexible number of states at the intermediate layers. In this more general case, an unsupervised approach is desired to learning intermediate representations. Development of unsupervised learning constitutes one major innovation of this work, which we describe in detail in this section.

The key idea of our approach is to cast the intermediate layer learning problem into a multi-objective programming (MOP) problem in which we minimize the average frame-level conditional entropy and maximize the state occupation entropy at the same time. Minimizing the average frame-level conditional entropy forces the intermediate layers to be sharp indicators of subclasses (or clusters) for each input vector, while maximizing the occupation entropy guarantees that the input vectors be represented distinctly by different intermediate states.

3.1. Maximize the state occupation entropy

Let us denote by $\boldsymbol{x}$, $\boldsymbol{h}$, and $\Lambda^{\mathrm{h}} = \{\lambda_i^h\}$ the input, output, and parameters of an intermediate layer, respectively. The intermediate layer state occupation entropy is defined as

$$H(h) = -\sum_h p(h) \log p(h) \tag{5}$$

where

$$p(h) = \frac{1}{K} \sum_k \sum_t p\left(h_t = h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right). \tag{6}$$

The derivative of $H(h)$ with respect to λ_i^h can be calculated as

$$\begin{aligned} \frac{\partial H(h)}{\partial \lambda_i^h} &= -\frac{\partial p(h)}{\partial \lambda_i^h} \log p(h) - \frac{\partial \log p(h)}{\partial \lambda_i^h} p(h) \\ &= -[\log p(h) + 1] \frac{\partial p(h)}{\partial \lambda_i^h} \\ &= -\frac{1}{K} [\log p(h) + 1] \sum_k \sum_t \frac{\partial p\left(h_t = h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)}{\partial \lambda_i^h}. \end{aligned} \tag{7}$$

Since

$$\frac{\partial p\left(h_t = h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)}{\partial \lambda_i^h} = \left[p\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right) - p^2\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)\right] f_i\left(h_t, \boldsymbol{x}^{(\boldsymbol{k})}, t\right) \tag{8}$$

we obtain the final gradient

$$\begin{aligned} \frac{\partial H(h)}{\partial \lambda_i^h} = &-\frac{1}{K} [\log p(h) + 1] \\ &\sum_k \sum_t \left[p\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right) - p^2\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)\right] f_i\left(h_t, \boldsymbol{x}^{(\boldsymbol{k})}, t\right) \end{aligned} \tag{9}$$

which is used in a gradient-based optimization procedure to be described shortly.

3.2. Minimize the frame-level conditional entropy

The frame-level conditional entropy at the intermediate layer can be written as

$$H\left(h | \boldsymbol{x}, \Lambda^{\mathrm{h}}\right) = -\sum_k \sum_h p\left(h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right) \log p\left(h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right). \tag{10}$$

Following the similar procedure we compute the derivative of $H\left(h | \boldsymbol{x}, \Lambda^{\mathrm{h}}\right)$ with respect to λ_i^h as

$$\begin{aligned} \frac{\partial H\left(h | \boldsymbol{x}, \Lambda^{\mathrm{h}}\right)}{\partial \lambda_i^h} = &-\sum_k \sum_t \left[\log p\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right) + 1\right] \\ &\frac{\partial p\left(h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)}{\partial \lambda_i^h} \\ = &-\sum_k \sum_t \left[\log p\left(h_t | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right) + 1\right] \left[p\left(h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)\right. \\ &\left. - p^2\left(h | \boldsymbol{x}^{(\boldsymbol{k})}, \Lambda^{\mathrm{h}}\right)\right] f_i\left(h_t, \boldsymbol{x}^{(\boldsymbol{k})}, t\right) \end{aligned} \tag{11}$$

The training of this MOP problem is carried out in a similar way to [8]. Specifically, we start from maximizing the state occupation entropy with the initial parameters set to zero. We then update the parameters by alternating between minimizing the frame-level conditional entropy and maximizing the average state occupation entropy. At each epoch we optimize one objective by allowing the other one to become slightly worse within a limited range. This range is gradually tightened epoch by epoch. The parameter update is carried out by gradient descent using RPROP algorithm [6].

After the intermediate layers are pre-trained layer by layer using the unsupervised approach we just described, the model parameters are jointly fine-tuned using the back propagation to optimize the sequential conditional log-likelihood $J(\Lambda, X)$ [14].

4. EMPIRICAL EVALUATION

The language identification experiments reported in this section have been performed on a Microsoft-internal close-set voice mail routing task, in which the goal is to route each voice mail to the right ASR engine for transcribing. The dataset contains voice mails from seven languages/dialects: Germany German (DE-DE), Australia English (EN-AU), British English (EN-GB), Indian English (EN-IN), Mexico Spanish (ES-MX), Canadian French (FR-CA), and Italian Italian (IT-IT). Each voice mail was recorded with different channels. As shown in Table 1, the average length of the voice mail utterances is 15 seconds with silence included and 11 seconds with silence removed. The average number of speakers for each language is over 500. The training, development and test sets contain 300 voice mails for each language. The training and test sets do not overlap with speakers.

TABLE I
DATA SETS USED IN THE EXPERIMENTS

Seconds	DE-DE	EN-AU	EN-GB	EN-IN	ES-MX	FR-CA	IT-IT	Avg
Mean	13	16	14	19	14	14	16	**15**
Std	7	9	7	10	8	7	9	**8**
Mean VAD	9	12	10	14	10	11	12	**11**
Std VAD	5	7	5	7	5	6	6	**6**
#Speakers	447	447	441	629	439	472	744	**517**

The front end processing steps are summarized in Fig. 2. SDCC features are used with the 7-1-3-7 parameterization as reported in detail in [7]. This corresponds to seven delta cepstral coefficients stacked from seven different time locations. The complete feature vector contains 56 coefficients with seven cepstral coefficients and 49 SDCCs per frame. These features are normalized using the utterance mean subtraction. The silence frames are then removed through statistical voice activation detection (VAD) module.

Fig. 2. Feature extraction pipeline in the experiments.

To evaluate the deep-structured CRF (DCRF), we have conducted a series of experiments using a range of configurations. Table II summaries the experimental results, where each continuous feature is expanded to four using the approach described in [10] when the distribution constraint is used. From Table II, we observe that the single-layer linear-chain CRF gives

low performance -- a routing accuracy (RA) of 44.6% and 34.3% with and without the distribution constraint, respectively. The two-layer DCRF significantly improves the accuracy. As shown in Table II, if a 128-state low layer is learned using the unsupervised approach (described in Section 3), 79.5% RA is achieved without the distribution constraint and without using the original SDCC feature in the second layer. The RA is improved to 83.6% when the distribution constraint is used. It is further increased to 85.1% when the tandem structure is used and to 86.4% after the back-propagation fine-tuning.

To compare our DCRF-based techniques with existing state-of-the-art techniques, we have conducted experiments using the GMM-MMI and GSV with model pushing (GSV-MP) on the same dataset. For the GMM-MMI system, the best configuration contains 256 Gaussians and was initialized using ten iterations of EM algorithm with the maximum likelihood (ML) criterion. For the GSV-MP system, we first built a language and gender independent GMM universal background model (UBM). The best configuration contains 1024 Gaussians in the UBM, and the UBMs were trained with ten iterations of EM adapting all parameters -- mixture weights, Gaussian means, and diagonal covariances.

From Table II we can observe that the best GMM-MMI system can obtain 82.5% RA which is 3.9% lower than that achieved using the best two-layer DCRF model. Since each feature is expanded to four in the DCRF, the number of parameters used in the two-layer DCRF with a 128-state intermediate layer is the same as that used in the GMM-MMI system with 256 Gaussian mixtures. Note that our model still underperforms the best GSV-MP system with the high 87.7% RA on the task at hand.

The same conclusion can be drawn using the equal error rate (EER) metric as shown in Table III. The best two-layer DCRF produces a 6.6% average EER, which is 0.9% better than the 7.5% EER obtained using the GMM-MMI system, and 0.2% worse than that obtained using the GSV-MP system.

TABLE II
EXPERIMENTAL RESULTS (ROUTING ACCURACY)

Model	# States /Mixtures	Distribution Constraint	Tandem	RA(%)
CRF	-	no	-	34.3
CRF	-	yes	-	44.6
DCRF+pretrain	128	no	no	79.5
DCRF+pretrain	128	yes	no	83.6
DCRF+pretrain	128	yes	yes	85.1
DCRF+back-prop	128	yes	yes	86.4
GMM-MMI	256	-	-	82.5
GSV-MP	1024	-	-	87.7

TABLE III
EXPERIMENTAL RESULTS (EQUAL ERROR RATE)

Model	# States /Mixtures	Distribution Constraint	Tandem	Average EER(%)
DCRF+back-prop	128	yes	yes	6.6
GMM-MMI	256	-	-	7.5
GSV-MP	1024	-	-	6.4

5. CONCLUSIONS

We have developed a novel discriminative model and classifier, the deep-structured CRF with flexible intermediate layers, for language recognition. Experiments on the voice mail routing task demonstrate its superiority in performance over the popular GMM-MMI approach.

The deep-structured CRF in its current form still underperforms the state-of-the-art GSV-MP system. Our approach can be improved by adding more layers and using better pre-training approach.

6. ACKNOWLEDGEMENTS

We would like to thank Dr. Geoffrey Zweig in Microsoft Research for providing the GMM-MMI training and testing tools. Thanks also go to Dr. Yifan Gong and his team in Microsoft Corporation for providing the experimental data.

7. REFERENCES

[1] L. Burget, P. Matejka, and J. Cernocky, “Discriminative training techniques for acoustic language identification,” in Proc. *ICASSP*, 2006, pp. 209–212.

[2] W. M. Campbell, P. A. Torres-Carrasquillo, and D. A. Reynolds, “Language recognition with support vector machines,” in Proc. of Odyssey 04, 2004, pp. 285–288.

[3] W. M. Campbell, "a covariance kernel for SVM language recognition,” in Proc. *ICASSP*, pp. 4141-4444, 2008.

[4] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, “Acoustic language identification using fast discriminative training,” in Proc. *Interspeech*, 2007.

[5] H. Hermansky, D. P. W. Ellis, S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems,” in Proc. *ICASSP*, vol.3, pp. 1635-1638, 2000.

[6] M. Riedmiller and H. Braun, “A direct adaptive method for faster back-propagation learning: The RPROP algorithm,” in Proc. *IEEE ICNN*, vol. 1, pp. 586-591, 1993.

[7] P. A. Torres-Carrasquillo1, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, Jr, “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” in Proc. *ICSLP*, pp. 89-92, 2002.

[8] S. Yaman and C.-H. Lee, “A flexible classifier design framework based on multi-objective programming,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 16, no. 4, pp. 779-789, 2008.

[9] D. Yu, L. Deng, Y. Gong, and A. Acero. “A novel framework and training algorithm for variable-parameter hidden Markov models,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 7, pp. 1348-1360, September 2009.

[10] D. Yu, L. Deng, and A. Acero, "Using continuous features in the maximum entropy model,” *Pattern Recognition Letters*. vol. 30, no. 8, pp.1295-1300, June, 2009.

[11] D. Yu and L. Deng, "Solving nonlinear estimation problems using Splines," *IEEE Signal Processing Magazine*, vol. 26, no. 4, pp.86-90, July, 2009.

[12] D. Yu, L. Deng, and A. Acero. “Hidden conditional random field with distribution constraints for phonetic classification,” in Proc. *Interspeech,* 2009.

[13] D. Yu, S. Wang, and L. Deng, "Sequential labeling using deep-structured conditional random fields,” submitted to IEEE *journal of selected topics in signal processing*.

[14] D. Yu, L. Deng, and S. Wang, "Learning in the deep-structured conditional random fields", *NIPS* 2009 Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.