Attentional Heterogeneous Transfer Learning with Copula Models in Speech Recognition

Shan Zhang, Saikiran Bulusu, Manish Gupta

May 11, 2018

1 Introduction

It has been shown that a well-trained model can be used as a teacher to guide the training of other child models in [1–3]. The basic idea is to transfer the compressed *knowledge* in a complex model to a simple model so that the output distributions or high level features, referred to as knowledge sources, generated by the former can be utilized to achieve significant inference performance.

Current speech recognizers consists of various components: acoustic models, language models, pronunciation models and text normalization. Each of these components make assumptions about the underlying probability distributions they model. In [4], a Listen, Attend and Spell (LAS) model was proposed for large vocabulary conversational speech recognition. LAS does not make any independence assumptions about the nature of the probability distribution of the output character sequence, given the input acoustic sequence. The LAS model has two components: a listener and a speller. The listener is a pyramidal recurrent neural network encoder that accepts filter bank spectra as inputs and gives compressed high level features (knowledge sources extracted from the source model). The speller is an attention-based recurrent network decoder that emits each character conditioned on all previous characters by consuming these higher level features. The attention mechanism in [4] assumes that the high level features are independent of each other.

However, these knowledge sources can be heterogeneous; hence, they are dependent. To ground heterogeneous sources into a common feature space or to characterize their dependence structure, we apply a copula-based approach to capture the statistical dependence. Different from [4], we propose a copula-based Listen and Spell (CBLA) model which transcribes speech utterances directly to characters without pronunciation, Hidden Markov Models (HMMs) or other components of traditional speech recognizers.

2 Background

In this section, we give a brief introduction to Copula Theory. Dependence modeling with copulas provides a flexible and powerful approach for continuous multivariate distributions since it separates modeling univariate marginals from modeling the multivariate (dependence) structure. A copula, specified independently from marginals, is a multivariate distribution with uniform marginal distributions. The unique correspondence between the copula and any multivariate distribution is stated in Sklar's theorem [5] which is a fundamental theorem of the copula theory.

Theorem 1 (Sklar's Theorem) *The joint distribution function F of random variables $x_1, \ldots, x_d$ with continuous marginal distribution functions $F_1, \ldots, F_d$ can be cast as*

$$F(x_1, x_2, \ldots, x_d) = C(F_1(x_1), F_2(x_2), \ldots, F_d(x_d)), \tag{1}$$

where C is an unique d-dimensional copula. Conversely, given a copula C and univariate Cumulative Distribution Functions (CDFs) $F_1, \ldots, F_d$, F in Eqn. (1) is a valid multivariate CDF with marginals $F_1, \ldots, F_d$.

For absolutely continuous distributions F and $F_1, \ldots, F_d$, the joint Probability Density Function (PDF) of random variables $x_1, \ldots, x_d$ can be obtained by differentiating both sides of Eqn. (1):

$$f(x_1, \ldots, x_d) = \Big(\prod_{m=1}^{d} f_m(x_m) \Big) c(F_1(x_1), \ldots, F_d(x_d)), \tag{2}$$

where $f_1, \ldots, f_d$ are the marginal densities and c is referred to the density of copula C that is given by

$$c(\mathbf{u}) = \frac{\partial^L (C(u_1, \ldots, u_d))}{\partial u_1, \ldots, \partial u_d}, \tag{3}$$

where $u_m = F_m(x_m), m = 1, 2, \ldots, d$ and $\mathbf{u} = [u_1, \ldots, u_d]$.

Thus, given specified univariate marginal distributions $F_1, \ldots, F_d$ and copula model C, the joint distribution function F can be constructed by

$$F(F_1^{-1}(u_1), F_2^{-1}(u_2), \ldots, F_d^{-1}(u_d)) = C(u_1, u_2, \ldots, u_d), \tag{4}$$

where $u_m = F_m(x_m)$ and $F_m^{-1}(u_m), m = 1, 2, \ldots, d$ are the inverse distribution functions of the marginals.

Note that $C(\cdot)$ is a valid CDF and $c(\cdot)$ is a valid PDF for uniformly distributed random variables u_m, $m = 1, 2, \ldots, d$. Since the random variable u_m represents the CDF of x_m, the CDF of u_m naturally follows a uniform distribution.

Since different copula functions may model different types of dependence, selection of copula functions to characterize the joint statistics of random variables is a key problem. Various families of copula functions are described in [5]. For Gaussian copula,

$$c_{\mathbf{z}}(\mathbf{u}; \mathbf{\Sigma_z}) = \frac{1}{|\mathbf{\Sigma_z}|^{1/2}} \exp\{-\frac{1}{2}\mathbf{u}^T(\mathbf{\Sigma_z}^{-1} - I)\mathbf{u}\}$$

Moreover, the *dependence parameter* denoted by $\boldsymbol{\phi}$, contained in a copula function, is used to characterize the amount of dependence among d random variables. Typically, $\boldsymbol{\phi}$ is unknown a *priori* and needs to be estimated, e.g., using Maximum Likelihood Estimation (MLE) or Kendall's τ [6].

3 Copula-based Listen and Spell Model

In the following, we will formally describe the CBLA. Let $\mathbf{x} = [x_1, x_2, \cdots, x_T]$ be the input sequence and $\mathbf{y} = [y_1, y_2, \cdots, y_M]$ be the output sequence. We model each character output y_i as a conditional distribution over the previous characters $y_{<i}$ and the input signal $\mathbf{x}$ using the chain rule for probabilities:

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i}). \tag{5}$$

Copula-based LAS model consists of two sub-modules: the listener and the speller. The listener (source model) performs the role of encoding and transforms the original signal $\mathbf{x}$ into a high level representation $\mathbf{h} = [h_1, h_2, \cdots, h_U]$ with $U \leq T$. The speller (target model) is a copula-based attentional character decoder. It consumes the extracted feature $\mathbf{h}$ and produces a probability distribution over character sequences:

$$\begin{aligned} \mathbf{h} &= \text{Listen}(\mathbf{x}) \\ P(y_i|\mathbf{x}, y_{<i}) &= \text{Spell}(y_{<i}, \mathbf{h}) \end{aligned} \tag{6}$$

The details of the two sub-modules are provided in the following subsections.

3.1 Listen

The Listener does a encoding process using a Bidirectional Long Short Term Memory RNN (BLSTM) [7]. A direct application of BLSTM for encoding converged slowly. Since the input speech signals can be hundreds to thousands of frames long, we use the pyramidal BLSTM (pBLSTM) in [4] which is equipped to reduce the length U of $\mathbf{h}$, from T, the length of the input $\mathbf{x}$. The pBLSTM model concatenates the outputs at consecutive steps of each layer before feeding it to the next layer, i.e.,

$$h_i^j = \text{pBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}]). \tag{7}$$

We employ 3 pBLSTMs on top of the bottom BLSTM layer to reduce the time dimension 8 times. This allows the speller model to extract the relevant information from a smaller number of times steps. In addition to reducing the resolution, the deep architecture allows the model to learn nonlinear feature representations of the data.

3.2 Spell

The Speller consists of a attentional copula-based LSTM transducer. At every output step, the transducer produces a probability distribution over the next character conditioned on all the characters seen previously. The distribution for y_i is a function of the decoder state s_i and context a_i. The decoder state s_i is a function of the previous state $s_{i?1}$, the previously emitted character $y_{i?1}$ and context $a_{i?1}$. The context vector a_i is produced by an attention mechanism. Specifically

$$\begin{aligned} \mathbf{a}_i &= \text{Copula-AttentionContext}(\mathbf{s}_i, \mathbf{h}) \\ \mathbf{s}_i &= \text{RNN}(\mathbf{s}_{i-1}, y_{i-1}, \mathbf{a}_{i-1}) \\ P(y_i|\mathbf{x}, y_{<i}) &= g(\mathbf{s}_i, \mathbf{a}_i) \end{aligned} \tag{8}$$

where $g(\cdot)$ is an MLP with softmax outputs over characters, and RNN is a 1 layer LSTM.

At each time step, i, the attention mechanism, Copula-AttentionContext generates a context vector, a_i encapsulating the information in the acoustic signal needed to generate the next character. The attention model consists of two main steps. First, the contents of the decoder state $\mathbf{s}_i$ are matched to the contents of $\mathbf{h}_u$ representing time step u of $\mathbf{h}$. We refer the matched values as energy denoted by $\mathbf{r}_{i,u}$. Since the features $\mathbf{h} = [h_1, h_2, \cdots, h_U]$ are dependent, the energies $\mathbf{r}_i$ are also dependent. Second, copula-based dependence modeling is applied on the energy $\mathbf{r}_i$ to generate an attention vector $\boldsymbol{\alpha}_i$. The vectors $\mathbf{h}_u$ are linearly blended using $\boldsymbol{\alpha}_i$ to create $\mathbf{a}_i$. We propose two copula-based modeling methodologies for energy $\mathbf{r}_i$.

3.2.1 Copula-AttentionContext Formulation 1

For the energy $\mathbf{r}_i \in \mathbb{R}^{N_{\text{batch}} \times U}$ in a batch, consider them a multivariate time series. Using Gaussian copula to characterize the multivariate dependence, we have

$$\mathbf{c}_{\mathbf{r}_i}(\mathbf{e}; \boldsymbol{\Sigma}_{\mathbf{r}_i}) = \frac{1}{|\boldsymbol{\Sigma}_{\mathbf{r}_i}|^{1/2}} \exp\{-\frac{1}{2}\mathbf{e}^T(\boldsymbol{\Sigma}_{\mathbf{r}_i}^{-1} - I)\mathbf{e}\}$$

where $\mathbf{e}_i = [\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \cdots, \mathbf{e}_{i,N_{\text{batch}}}]$, $\mathbf{e}_{i,j} = F_j(\mathbf{r}_{i,j}), j \in [1, 2, \cdots, N_{\text{batch}}]$ and $\boldsymbol{\Sigma}_{\mathbf{r}_i} = \text{cov}(\mathbf{r}_i) \in \mathbb{R}^{N_{\text{batch}} \times N_{\text{batch}}}$. Note that $\mathbf{c}_{\mathbf{r}_i} \in \mathbb{R}^{U \times 1}$.

The attention mechanism is given by

$$\begin{aligned}
\mathbf{r}_{i,u} &= \langle \phi(\mathbf{s}_i), \psi(\mathbf{h}_u) \rangle \\
\mathbf{v}_i &= \mathbf{r}_i \otimes \mathbf{c}_{\mathbf{r}_i}^T \\
\boldsymbol{\alpha}_{i,u} &= \frac{\exp(\mathbf{v}_{i,u})}{\sum_{u'} \exp(\mathbf{v}_{i,u'})} \\
\mathbf{a}_i &= \sum_u \alpha_{i,u} h_u,
\end{aligned}$$

where $\otimes$ denotes element-wise production, ϕ and ψ are MLP networks.

3.2.2 Copula-AttentionContext Formulation 2

In this case, we assume the multivariate time series in a batch are independent. We use bivariate (distributed) Gaussian copula to model the dependence across time. Bivariate Gaussian copula is given as

$$\mathbf{c}_{\mathbf{r}_{i,j},\mathbf{r}_{i,k}}(\mathbf{e}; \boldsymbol{\Sigma}_{\mathbf{r}_{i,j},\mathbf{r}_{i,k}}) = \frac{1}{|\boldsymbol{\Sigma}_{\mathbf{r}_{i,j},\mathbf{r}_{i,k}}|^{1/2}} \exp\{-\frac{1}{2}\mathbf{e}^T(\boldsymbol{\Sigma}^{-1}_{\mathbf{r}_{i,j},\mathbf{r}_{i,k}} - I)\mathbf{e}\}$$

where $\mathbf{e} = [\mathbf{e}_{i,j}, \mathbf{e}_{i,k}]$, $\mathbf{e}_{i,j} = F_j(\mathbf{r}_{i,j})$.

The attention mechanism is given by

$$\begin{aligned}
\mathbf{r}_{i,u} &= \langle \phi(\mathbf{s}_i), \psi(\mathbf{h}_u) \rangle \\
\mathbf{v}_{i,1} &= \mathbf{r}_{i,1} \\
\mathbf{v}_{i,u} &= \mathbf{r}_{i,u} \times \mathbf{c}_{\mathbf{r}_{i,u},\mathbf{r}_{i,u-1}}, \text{ for } u > 1 \\
\boldsymbol{\alpha}_{i,u} &= \frac{\exp(\mathbf{v}_{i,u})}{\sum_{u'} \exp(\mathbf{v}_{i,u'})} \\
\mathbf{a}_i &= \sum_u \boldsymbol{\alpha}_{i,u} \mathbf{h}_u,
\end{aligned}$$

where ϕ and ψ are MLP networks.

3.3 Learning and Decoding

We train the parameters of our model to maximize the log probability of the correct sequences. Specifically,

$$\tilde{\theta} = \max_{\theta} \sum_i \log P(y_i | \mathbf{x}, \tilde{y}_{<i}; \theta). \tag{9}$$

During the inference, we estimate the most likely character sequence given the input acoustics:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{argmax}} \log P(\mathbf{y}|\mathbf{x}). \tag{10}$$

4 Experiments

For the experiments, we used TIMIT dataset. We first preprocess the acoustic signals and extract MFCC features with dimension 39. The input sequence has the length 784. We used three layers of 512 pBLSTM nodes for Listener model and one LSTM with 512 nodes for the Speller model. After three pBLSTM layers, the time steps are reduced to 98. The learning rate for our simulation is 0.0001 and batch size is 4.

The performance of our models are compared with benchmark case of [4], which doesn't use copula-based dependencies.

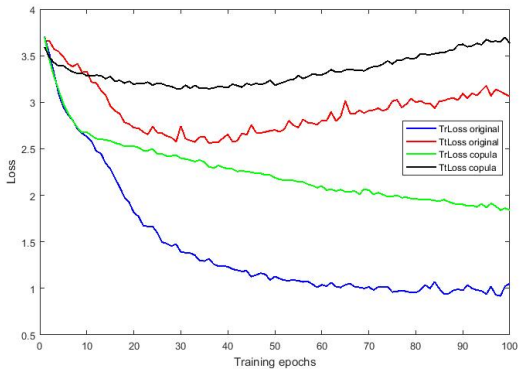

Figure 1: Learning Curve for Copula-AttentionContext Formulation 1.

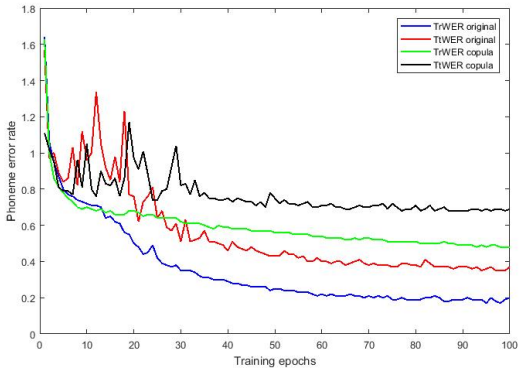

Figure 2: Performance curve for Copula-AttentionContext Formulation 1.

In Fig. 1 and Fig. 2, we present the learning and performance curves for Copula-AttentionContext formulation 1. We considered energy of a batch as multivariate time series. Our results follows the similar trends but convergence is slower than expected compared to the benchmark model.

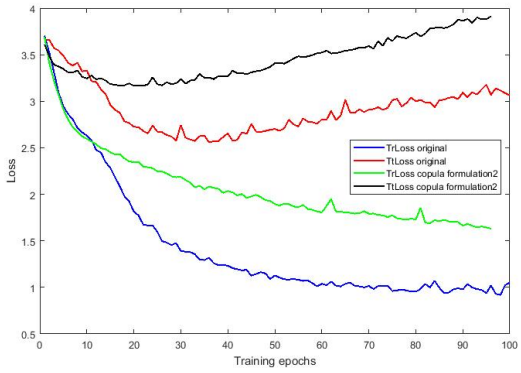

Figure 3: Learning Curve for Copula-AttentionContext Formulation 2.

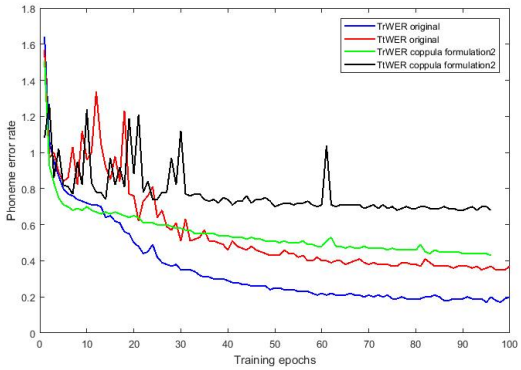

Figure 4: Performance curve for Copula-AttentionContext Formulation 2.

In Fig. 3 and Fig. 4, we show the learning and performance curves for Copula-AttentionContext formulation 2. The similar results are obtained for formulation 2, where multivariate time series in batch are independent. We believe that slow convergences in both cases could be because of overfitting of data. We also believe that our approach could perform better than benchmark model, if we use complex dataset than TIMIT.

In Fig. 5 and Fig. 6, the comparison of our two approaches are shown. We found that formulation 2 works better, which suggests that assuming dependencies along the time step is better in speech recognition.

5 Conclusions and Discussions

We implemented copula based Listen and Spell model for speech recognition to decode the output sequence of characters from acoustic features of TIMIT

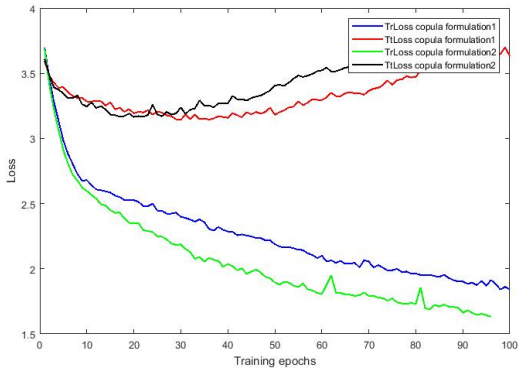

Figure 5: Learning Curve for comparison of Copula-AttentionContext Formulation 1 and 2.

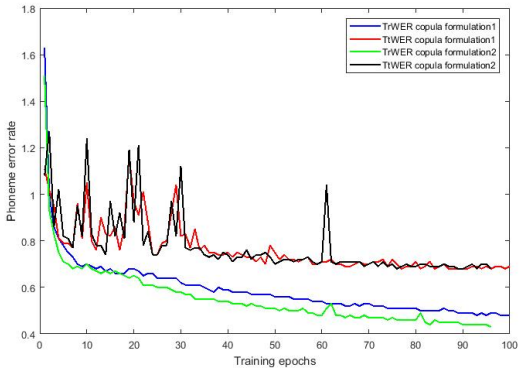

Figure 6: Performance curve for comparison of Copula-AttentionContext Formulation 1 and 2.

dataset. We proposed two different types of attention context formulation to evaluate appropriate weights for the higher level features. Although, we did not achieve performance gains compared to the benchmark model, our results follow the same trends and training loss converges. We believe the reason for slow convergence is overfitting as TIMIT dataset is very simple for our proposed model. Hence, future investigations need to be conducted on more complex dataset like CHIME or TED-LIUM. We also observed that exploiting the dependencies across time steps is more rewarding compared to dependencies across multiple user inputs.

References

[1] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

[2] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[3] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[5] R. B. Nelsen, *An introduction to copulas*. Springer Science & Business Media, 2013, vol. 139.

[6] H. He, "Heterogeneous sensor signal processing for inference with nonlinear dependence," Ph.D. dissertation, Syracuse University, 2015.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.