ADVANCED LSTM: A STUDY ABOUT BETTER TIME DEPENDENCY MODELING IN EMOTION RECOGNITION

Fei Tao[1*] , Gang Liu[2]

1. Multimodal Signal Processing (MSP) Lab, The University of Texas at Dallas, Richardson TX
2. Institute of Data Science and Technology (iDST)-Speech, Alibaba Group (U.S.) Inc.

fxt120230@utdallas.edu,g.liu@alibaba-inc.com

ABSTRACT

Long short-term memory (LSTM) is normally used in *recurrent neural network* (RNN) as basic recurrent unit. However, conventional LSTM assumes that the state at current time step depends on previous time step. This assumption constraints the time dependency modeling capability. In this study, we propose a new variation of LSTM, *advanced LSTM* (A-LSTM), for better temporal context modeling. We employ A-LSTM in weighted pooling RNN for emotion recognition. The A-LSTM outperforms the conventional LSTM by 5.5% relatively. The A-LSTM based weighted pooling RNN can also complement the state-of-the-art emotion classification framework. This shows the advantage of A-LSTM.

Index Terms— multi-task learning, attention model, A-LSTM, recurrent neural network, emotion recognition

1. INTRODUCTION

Recurrent neural network is recently used as a dynamic model for sequential input. *Long short-term memory* (LSTM) is usually adopted as basic units in RNN because it is able to solve the gradients vanishing and exploding problems in RNN training [1]. It uses memory cell and gates to control whether information will be memorized, output or forgotten. The LSTM takes two inputs, output from lower layer and output from previous time step in current layer. This configuration implies an assumption that the current state depends on the state of previous time step. This assumption of time dependency may constraint the modeling capability of RNN. In this paper, we propose a new variation of LSTM, *advanced LSTM* (A-LSTM), to address this issue. In A-LSTM, current state depends on multiple states of different time steps. This releases the constrains in conventional LSTM and provides better time dependency modeling capability.

This paper presents our early study on A-LSTM. We explore the modeling capability of A-LSTM in the application of voice-based emotion recognition. Recognizing emotion will improve the user experience of voice-based *artificial intelligence* (AI) product, like Siri, Alexa. Even though recent research shows combining audiovisual speech processing systems can have better performance than audio only system ([2, 3, 4, 5]), based on audio in real world. The input voice to system in real application may contains long silence (or pause) or non-speech voice filler, the conventional low level statistics feature like *Interspeech 2010 paralinguistic challenge feature set* (IS10) or GeMAPs [6, 7], may fail. Weighted pooling based on attention mechanism is an appealing solution for these cases [8], which relies on RNN. We built an attention based weighted pooling framework with multi-task learning for emotion recognition in this study. When we apply A-LSTM in this framework, it gains 5.5% relative improvement compared with conventional LSTM.

The remaining of the paper is organized as following structure. Section 2 reviews previous work. Section 3 introduces the IEMOCAP corpus. Besides, the acoustic feature extraction is also described. Section 4 describes the details of the proposed approach. Section 5 describes the experiments and analysis of results. Section 6 concludes the work and leads to the future direction of the work.

2. RELATED WORK

[9, 10, 11] show that temporal information is beneficial to emotion identification. [12, 13, 14] show that the performance of the neural network will be improved when higher layers can see more time steps from lower layer. These works rely on DNN rather than RNN. They do not discuss the timing sequence modeling. [15, 16] propose solutions to having alternative connections between layers in DNN. These solutions are different from the conventional connections within network. [17, 18] modify the LSTM architecture relying on residual or highway connection. However, the modifications in these papers are focusing on connecting the memory cells between lower and higher layers. They do not modify the connection within the same layer. [19] modifies the output hidden value to higher layer by a weighted summation. [20] follows similar idea. It uses weighted pooling of the hidden values of multiple historic time steps at each time steps which improves the information richness to higher layer. This is equivalent to allow higher layer see more time steps. But they do not modify the memory cell which means the time dependency is not changed. [21] shows that the combination of near time steps may not improve the system a lot. The combination should contain a long term range. Besides, they do not combine the multiple states at each step, which is different from the high order RNN. For emotion recognition, [8] recently proposes at-

∗This work was done during the author's summer internship at Alibaba Group (U.S.) Inc.

tention based weighted pooling RNN to extract acoustic representation. The work shows the weighted pooling RNN can outperform conventional pooling approach, like mean, maximum, or minimum. It also shows the RNN framework can capture the section of interest. Multi-task learning recently shows its advantage in emotion recognition task [22, 23]. But in these papers, the regression of valence and arousal values are normally set as auxiliary tasks, which is hard to obtain.

3. CORPUS DESCRIPTION AND FEATURE EXTRACTION

We validate A-LSTM in the application of categorical emotion classification. We used IEMOCAP [24] corpus in this study which has 5 sections and 10 actors in total. In each section, there were two actors (one male and one female) involved in scripted or spontaneous scenarios to perform specific emotions. The utterances were segmented and with one categorical label, which is among angry, fear, excited, neutral, disgust, surprised, sad, happy, frustrated, other and XXX. XXX was the case that the annotators were not able to have agreement on the label. The corpus has 10039 utterances with average duration of 4.5 s per utterance (12.55 hr in total). The distribution of emotion classes is not balanced. In this study, we select 4 classes, neutral, happy, angry and sad. The total number of utterances used is 4490.

The corpus has video and audio channels. We only used audios in this study. The audio was collected by high quality microphones (Schoeps CMIT 5U) at the sample rate of 48 kHz. We downsampled them to 16 kHz and extracted a 36D acoustic feature. The acoustic feature includes 13D MFCCs, *zero crossing rate* (ZCR), energy, entropy of energy, spectral centroid, spectral spread, spectral entropy, spectral flux, spectral rolloff, 12D chroma vector, chroma deviation, harmonic ratio and pitch. The extraction was performed within a 25 ms window whose shifting step size was 10 ms (100 fps). The acoustic feature sequence was z-normalized within each utterance.

4. PROPOSED APPROACH

4.1. Attention Based Weighted Pooling RNN

Attention based weighted pooling RNN is a data-driven framework to learn utterance representation from data, which is suitable for practical application [8]. It relies on the attention mechanism [25] to learn the weight of each time step. The weighted summation is then computed as the representation of the whole utterance. Multi-task learning incorporates several aspects of knowledge into training, therefore it can learn better representation.

The system diagram is shown in Figure 1. The diagram has two parts, trunk and branch (two dashed boxes in the diagram). The branch part contains sub-parts for different tasks. In this study, there are three sub-parts, whose tasks are emotion, speaker and gender classification respectively. The trunk part is the shared part of all tasks, which will process input and extract representation for classification tasks. On top of the trunk part, there is a layer of weighted pooling based on attention. The pooling is computed as Equation 1, where h_T is the hidden value output from the LSTM layer at time T, and A_T is a scalar number representing the corresponding weight at time T. A_T is computed in a softmax fashion following Equation 2, where W is a parameter to be learned. $\exp(W \cdot h_T)$ represents the potential energy at time T. This is similar to attention mechanism. If the frame at time T has high potential energy, its weight will be high and therefore gain high "attention"; if the potential energy is low, the weight and "attention" will also be low. By this way, the model can learn to assign weights to different time steps from data.

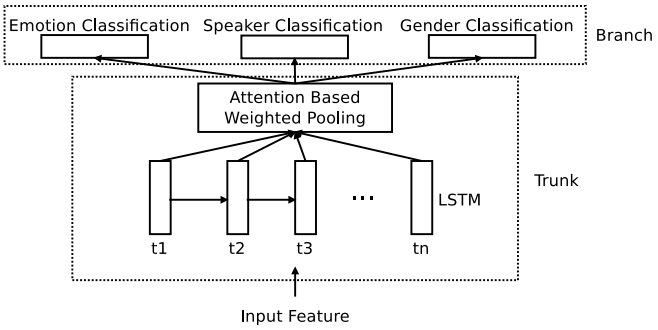

Fig. 1. The attention based weighted pooling RNN. The LSTM layer is unrolled along the time axis (time t1 to tn). The trunk part has the layers that are shared by all the tasks. On top of the trunk part, there is branch part for tasks. The main task is emotion classification. The auxiliary tasks are speaker and gender classifications.

If weights at all time steps are same, the weighted pooling is equal to arithmetic mean.

$$WeightedPooling = \sum_{T=t1}^{tn} A_T \times h_T \tag{1}$$

$$A_T = \frac{\exp(W \cdot h_T)}{\sum_{T=t1}^{tn} \exp(W \cdot h_T)} \tag{2}$$

In this study, we define that trunk part has two hidden layers. The first layer is fully connected layer which has 256 RELU neurons. The second one is a *bidirectional LSTM* (BLSTM) layer with 128 neurons. The hidden values go to weighted pooling layer after the LSTM layer. In the branch part, each task has one fully connected hidden layer with 256 RELU neurons and one softmax layer performing classification.

4.2. Advanced LSTM

Conventional LSTM tasks take the output from lower layer and previous time step as input and feed value to higher layer. The gating mechanism is used to control information flow by point-wise multiplication (denoted as $\odot$ operation in the following contents). There is a cell to memorize information within the unit. The diagram is shown in Figure 2.

The cell is updated as Equation 3, where f_t and i_t are the forgetting and inputting gates at time t. $\widetilde{C}_t$ is new candidate cell values. It is computed as Equation 4, where $tanh$ is the

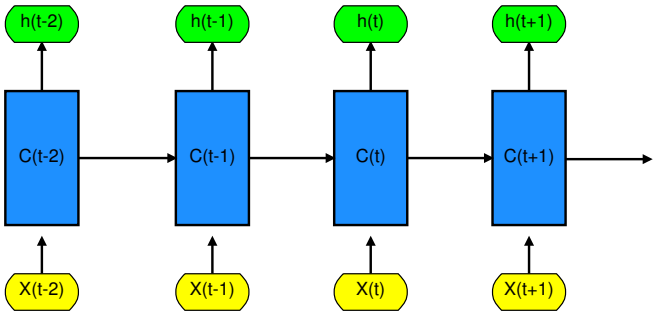

Fig. 2. The unrolled conventional LSTM. Unrolling is along the time axis. The C is the cell memory, X is the values from lower layer, and h is the hidden values to higher layer. State at time t depends on the one at time $t-1$ in conventional LSTM.

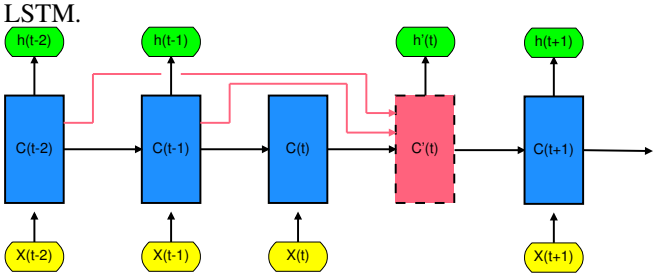

Fig. 3. The unrolled A-LSTM. Unrolling is along the time axis. The C is the cell memory, X is the values from lower layer, and h is the hidden values to higher layer. The dashed box is a weighted summation operation to combine the states at time $t-2$, $t-1$ and t. C' and h' is new cell memory and hidden value after combination. They are passed to compute the states at time $t+1$.

activation function, W_C is a set of weights to be learned, b_C is the bias, and $[h_{t-1}, x_t]$ is the concatenation of the values from previous time step (h value) and lower layer (x value). h value at time t is computed by Equation 5, where o_t is outputting gate. The state at time t depends on the state at time $t-1$, because C_t is computed based on h_{t-1} and C_{t-1}. The computation about controlling gates are omitted for simplification.

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{3}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$h_t = o_t \odot tanh(C_t) \tag{5}$$

The A-LSTM is different from the conventional one. It releases the assumption that time t state depends on time $t-1$ state. It use weighted summation of multiple states at different time steps to compute cell (C value) and hidden (h value) values. The diagram is shown in Figure 3.

In A-LSTM, Equation 3 is modified to Equation 6, and Equation 4 is modified to Equation 7. C' is a weighted combination of states at selected time steps. It is computed following Equation 8, where T is the set of selected time steps. For example, $t-2$, $t-1$ and t are selected to compute $t+1$ state in Figure 3. If we denote the index of time point $(t+1)$ as 0, T is denoted as a set of $\{3,2,1\}$. In the remaining contents, we follow the same naming convention to show our configuration of A-LSTM. W_{C_T} is a scalar number representing the corresponding weight at a time step. It is learned from Equation 9. Hidden value h at time t is computed following Equation 10. It is same as Equation 5 except the cell value now is C'. h' is then computed following Equation 11 and 12. h' is weighted combination of hidden values at selected steps in set T. In Equation 9 and 12, W is shared parameter learned from data. Therefore, C' and h' contains the states and hidden values information in set T. They are computed every $max(T)$ steps in this study. For example, in Figure 3, they are computed every 3 steps ($t-2$,$t-1$,t are used to compute $t+1$; $t+1$,$t+2$,$t+3$ are used to compute $t+4$; and so on).

A-LSTM is able to allow more flexible time dependency modeling capability. It makes the cell to recall far back historic records. Recalling every once in a while will be like the human learning mechanism, which makes learning better. Therefore the cell memory can memorize information better compared with conventional LSTM.

$$C_t = f_t \odot C'_{t-1} + i_t \odot \widetilde{C}_t \tag{6}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h'_{t-1}, x_t] + b_C) \tag{7}$$

$$C' = \sum_T W_{C_T} \times C_T \tag{8}$$

$$W_{C_T} = \frac{\exp(W \cdot C_T)}{\sum_T \exp(W \cdot C_T)} \tag{9}$$

$$h_t = o_t \odot tanh(C'_t) \tag{10}$$

$$h' = \sum_T W_{h_T} \times h_T \tag{11}$$

$$W_{h_T} = \frac{\exp(W \cdot h_T)}{\sum_T \exp(W \cdot h_T)} \tag{12}$$

5. EXPERIMENTS AND RESULTS

We evaluate our proposed A-LSTM on selected utterances from IEMOCAP corpus, which belonged to neutral, happy, angry and sad classes. We run two sets of experiments. In the first one, we compared different types of LSTMs. All the systems were based on weighted pooling RNN framework. In the second one we compared RNN framework with a *deep neural network* (DNN) framework, which represents current state-of-the-art system on IEMOCAP. Multi-task learning was applied during all the systems. The weights for emotion, speaker and gender classification were 1, 0.3, 0.6 respectively. We randomly selected 1 male and 1 female as testing subjects. The data from other subjects were used as training data. 10% of the training data was used as validation data to check whether we need early stopping. The early stopping criteria was that in continuous 3 epochs, the accuracy on the validation data was lower than the highest accuracy.

Table 1. The performance of baseline and proposed systems. MAF is macro average F-score. MAP is macro average precision.

Approach	MAF	MAP	Accuracy (%)
conventional LSTM	43.8	64.3	52.7
mean LSTM	43.5	64.3	52.8
advanced LSTM	**46.2**	**65.8**	**55.3**

Macro average F-score (MAF) (also named as unweighted average F-score) *macro average precision* (MAP) (also named as unweighted average precision) and accuracy were used as performance metrics. The metrics were computed with the open source tool, Scikit-learn [26]. Since the classes were imbalanced, we mainly rely on the MAF for performance evaluation.

5.1. Weighted Pooling RNN Results

We built up two baseline systems for comparison under the RNN framework. The first one used conventional LSTM. The second one used recurrent unit that was similar to the A-LSTM structure except that W_{C_t} and W_{h_t} were fixed. They were determined to same values which made the combination equivalent to arithmetic mean of the states at selected time steps. We therefore name it as "mean LSTM". The proposed framework used A-LSTM as recurrent unit. The parameter details of the neural network has been described in Section 4.1. Dropout was used in all the layers in the network except the attention based weighted pooling layer and the parameter of W in Equation 9 and 12. The dropout rate was 0.5. The set of T for A-LSTM used in this experiment was $\{5, 3, 1\}$. The time steps were selected every 2 time points. It was observed in pilot experiment that the training would be difficult when too many times steps in T, we therefore fixed to 3 selected time steps. Adam [27] was used as optimizer. The batch size for all systems was 32.

The performance of the two baseline systems and the proposed systems are listed in Table 1. Comparing A-LSTM and conventional LSTM shows that the A-LSTM is able to outperform the conventional LSTM by 5.5% in terms of MAF. Since the weighted pooling layer can see the hidden values from all time steps, this improvement is not from the benefit of seeing more time steps in higher layer. It leveraged the advantage of the flexible time dependency modeling capability of A-LSTM. This is especially useful in emotion recognition, because emotion is usually shown a state within a range of time steps rather than at a time step instantly. In this study, we have 256 neurons in the BLSTM (each direction has 128 neurons), so we only need add 256 parameters, which is the W size, to achieve this improvement. This cost can be ignored compared with about 600 k parameters of network.

The results also show that there is no improvement when we fixed the weights. Comparing mean LSTM and A-LSTM implies that learnable weights are better. Learning weights as a framework of data-driven assignment allows the model to make the assignment according to different situations. It is better because time dependency may vary at different time steps.

Table 2. The comparison between DNN and RNN frameworks. "IS10" is Interspeech 2010 feature set. "Seq" is the sequential acoustic feature. "RNN+DNN" is the fusion result.

Approach	feature	MAF	MAP	Accuracy (%)
DNN	IS10	56.9	66.8	58.2
RNN	Seq	46.2	65.8	55.3
RNN+DNN	IS10+Seq	**58.2**	**69.6**	**58.7**

5.2. Comparison between RNN and DNN Frameworks

We also built a DNN with multi-task learning for comparison. The network has two parts, shared part and separate part. The former part is shared by all the tasks, which has 2 fully connected layers with 4096 RELU neurons per layer. The later part has 3 separate sub-networks respectively for 3 tasks. Each sub-network has 1 fully connected layers with 2048 RELU neurons. On top of that, there is a softmax layer for classification. The batch size was 32 and dropout rate was 0.5. The optimizer was *stochastic gradients descending* (SGD). We used IS10 feature set extracted with openSMILE [28] as input because it was suitable for the three tasks. IS10 was z-normalized based on the mean and variance from training part. We also used the tool of Focal [29] to fuse the results from these two frameworks.

The results of the experiment are shown in Table 2. It is shown that the RNN framework is about 23.2 % worse than DNN framework. There are two reasons here. First, we have very limited data, which is only about 3200 training utterances. This amount may not train RNN framework sufficiently, especially training RNN is more difficult than DNN. Second, all the utterances were well segmented in IEMOCAP. It may not have long silence and pause as the situation in real world. The fusion result shows combining the two frameworks is better than either single one. It indicates that RNN framework can complement the DNN even with few training data. Besides, there are about 58 M parameters in DNN which is about 100 times as the one in RNN which means that RNN will have low hardware requirement when it is employed.

6. CONCLUSION AND FUTURE WORK

We proposed a new type of LSTM, A-LSTM, in this paper. This was a early study of A-LSTM. We applied it in the weighted pooling RNN for emotion recognition. It is shown that the A-LSTM can outperform the conventional LSTM under weighted pooling RNN framework with few extra parameters. The improvement leverages the advantage of flexible time dependency modeling capability in A-LSTM. Even though the weighted pooling RNN framework can not beat the state-of-the-art DNN framework on IEMOCAP, it can complement the DNN to achieve better performance. It also has the advantage in practical application in real world.

Future work is necessary to explore A-LSTM in other tasks. The idea of combining states at multiple time steps can also be extended to *gated recurrent unit* (GRU) in the future. More data is also needed for training the RNN framework.

7. REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] F. Tao, J.H.L. Hansen, and C. Busso, "An unsupervised visual-only voice activity detection approach using temporal orofacial features," in *Interspeech 2015*, Dresden, Germany, September 2015, pp. 2302–2306.

[3] F. Tao, J.H. L. Hansen, and C. Busso, "Improving boundary estimation in audiovisual speech activity detection using Bayesian information criterion," in *Interspeech 2016*, San Francisco, CA, USA, September 2016, pp. 2130–2134.

[4] F. Tao and C. Busso, "Gating neural network for large vocabulary audiovisual speech recognition system," *IEEE transactions on acoustics, speech, and signal processing*, submitted.

[5] F. Tao and C. Busso, "Bimodal recurrent neural network for audiovisual voice activity detection," in *Interspeech 2017*, Stockholm, Sweden, Sep. 2017, pp. 1938–1942.

[6] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Interspeech 2010*, Makuhari, Japan, September 2010, pp. 2794–2797.

[7] F. Eyben, K.R. Scherer, B. Schuller, J. Sundberg, E. Andre, C. Busso, L.Y. Devillers, J. Epps, P. Laukka, S.S. Narayanan, and K.P. Truong, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, vol. 7, no. 2, pp. 190–202, 2015.

[8] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, U.S.A., Mar. 2017, IEEE, pp. 2227–2231.

[9] G. Liu and J.H. L. Hansen, "Supra-segmental feature based speaker trait detection," in *Odyssey 2014*, Joensuu, Finland, June 2014.

[10] Y. Lei G. Liu and J.H. L. Hansen, "A novel feature extraction strategy for multi-stream robust emotion identification," in *Interspeech 2010*, Makuhari Messo, Japan, Sep. 2010, pp. 482–485.

[11] T. Rahman, S. Mariooryad, S. Keshavamurthy, G. Liu, J.H.L Hansen, and C. Busso, "Detecting sleepiness by fusing classifiers trained with novel acoustic features," in *Interspeech 2011*, Florence, Italy, Aug. 2011, pp. 3285–3288.

[12] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Arizona, USA, 2015, IEEE, pp. 92–97.

[13] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts.," in *INTERSPEECH 2015*, Dresden, Germany, Sept. 2015, pp. 3214–3218.

[14] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *the IEEE conference on computer vision and pattern recognition*, Washington, USA, Jun. 2016, pp. 770–778.

[16] R. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[17] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, Mar. 2016, IEEE, pp. 5755–5759.

[18] J. Kim, M. El-Khamy, and J. Lee, "Residual lstm: Design of a deep recurrent architecture for distant speech recognition," *arXiv preprint arXiv:1701.03360*, 2017.

[19] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feedforward sequential memory networks: A new structure to learn long-term dependency," *arXiv preprint arXiv:1512.08301*, 2015.

[20] R. Soltani and H. Jiang, "Higher order recurrent neural networks," *arXiv preprint arXiv:1605.00064*, 2016.

[21] Y. Wang and F. Tian, "Recurrent residual learning for sequence classification.," in *EMNLP*, 2016, pp. 938–943.

[22] R. Xia and Y. Liu, "A multi-task learning framework for emotion recognition using 2d continuous space," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 3–14, 2017.

[23] S. Parthasarathy and C. Busso, "Jointly predicting arousal, valence and dominance with multi-task learning," in *INTERSPEECH 2017*, Stockholm, Sweden, Aug. 2017.

[24] C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, and S.S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, December 2008.

[25] Dzmitry D. Bahdanau, J. Chorowski, D. Serdyuk, and Yoshua Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Apr. 2016, IEEE, pp. 4945–4949.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] F. Eyben, M. Wöllmer, and B. Schuller, "OpenSMILE: the Munich versatile and fast open-source audio feature extractor," in *ACM International conference on Multimedia (MM 2010)*, Florence, Italy, October 2010, pp. 1459–1462.

[29] Niko Brummer, "Focal," https://sites.google.com/site/nikobrummer/focal, 2017, Retrieved Aug 1st, 2017.