EmoNet: Reconstruction Of Emotion As People Read Using Deep Neural Network With Attention

Zhengxuan Wu
Department of Management Science and Engineering
Stanford University
Stanford, CA 94304
wuzhengx@stanford.edu

Xiyu Zhang
Department of Computer Science
Stanford University
Stanford, CA 94304
sherinez@stanford.edu

Xuan Zhang
Department of Statistics
Stanford University
Stanford, CA 94304
kayleez@stanford.edu

Abstract

Modeling the emotional cognition process from text remains an unsolved building block for developing human-centered Artificial Intelligence (AI). Though neural machine translator, QA agent built by deep learning models could help goal-driven language modelings, but could not easily interpret the emotion cognition process underlying the language. A majority of existing literature in understanding emotion cognition process focuses on modeling using acoustic and visual features rather than linguistic ones. Few study exist in the domain of understanding the continuous emotion cognition process of text. Here, we want to close the gap. In this paper, we implement four different deep learning models with the Stanford Emotional Narratives Dataset (SENDv1) [1] to predict emotion valence from linguistic features and to reconstruct emotion from text. Our transformer-based model achieves a CCC score of **0.47** on the Test set, which is higher than the CCC score of 0.46 of human's ratings with audio, visual and linguistic features.

1 Introduction

"Hi Alexa, I'm tired today since I had to take care of my mom in hospital. Please play me some music." - Imagine this is the conversation you have with your personal home assistant Alexa, would you like to see her playing rock music as you usually like, or some country music to relax? When John comes back home, he complains about his tedious works at the construction site, and asks his home robot to cook for him instead. Probably what he wants is a piece of well-cooked steak instead of a cold salad with ranch sauce. Understanding human emotions can help robots or any other AIs make more user-friendly decisions.

Emotion cognition from linguistic cues is a crucial part of Human-centered AIs. Without emotion understandings, AIs could not successfully co-exist with people [1, 2]. Language is the carrier when conveying your emotion to other people [3]. Various research showed that data from social platforms could potentially predict one's psychological traits [4]. Users' emotion valence is also

[1]This dataset is not released from its original paper. Contact $desmond\_ong@ihpc.astar.edu.sg$ for the dataset.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

highly correlated with their daily updates in those platforms throughout a year. For example, the national happiness score could be predicted by Facebook data [5].

Using human ratings of emotion valence of videos to train models helps us understand the emotion cognition process of text. Studies showed that human's cognitive and affective development start at an early stage of one's brain evolution [6]. Using multi-modalities, including visual, audio and linguistic inputs, one is capable of determining other people's emotion valences. Fundamentally, human's capability of emotion cognition is from the training process of daily social communications. For example, when your friend is weeping while telling you a heart-broken story, you are able to associate this anecdote with low emotional valence, which, in turn, means sadness. Similarly, when we are reading the inspiring speech *I Have A Dream* from the American baptist minister Martin Luther King Jr., our mind is picturing the facial expressions of Martin and imagining the pitches of his voice while interpreting his emotion. Thus, the training process of a model should involve the true ratings that are annotated using multi-modalities inputs. The Figure 1 shows the sentiment analysis of our model on the speech *I Have A Dream*.

Figure 1: Left: The sentiment reconstruction of the inspiring speech *I Have A Dream* from the American baptist minister Martin Luther King Jr.. We can see the emotion valence of the speech goes up at the end. This matches with the actual speech scripts, as **dream** is emphasised at the end of the speech. On the left, we overlay three example of windowed words. Negative words are found during the turns of the emotion valence. Right: The changes of emotion valences produced by the indicated windows.

Models for predicting the emotion valence of texts should be trained using the true ratings, which is rated using visual, audio and linguistic features. This simulates the cognition process of our brain in emotion cognition of text. When reading texts, humans interpret the emotion from text only, but their emotion cognition ability is learned by the true communication of emotion with other people. Various deep learning models were developed to predict emotion valence of video inputs which contains visual and audio features [7, 8, 9]. A few studies focused on the linguistic features. However, these models are usually trained by hand-labelled data, which, in fact, is labelled by human and only using text itself. This is not matching the emotion cognition process in human's brain. When reading the text, one tends to reconstruct the facial and audio features and use those features to interpret the emotion. Our study aims to close the gap. Using the SENDv1 dataset, we train our model with only the linguistic inputs but with the true ratings which are rated using visual, audio and linguistic inputs by human.

Only a small portion of researchers in the research community focuses on the time-series modeling of emotion cognition. Few models are capable of capturing the dynamics of emotion as they develop over time, which is referred as **time-series emotion recognition**. Specifically, time-series emotion recognition modeling takes previous outputs with the dynamic context of the current time point. The model considers the emotion propagated through time. For example, when we are annotating videos with texts, annotating each frame is not a separated task. In fact, information about tagging is likely to be passed across frames. Using time-series modelings, the process of information propagating through time could be modelled more effectively [10].

In the rest of this paper, we will provide a overview of existing modeling methodologies in related fields. Then, we will introduce our approach of constructing models for reconstructing emotion valence from linguistic features, includes our baseline, feature parsing and four deep learning models. Next, we will discuss the performance of each model, and interpret our results. At the end, we will

provide the contributions our paper brings that could potentially help the research community to better understand the emotion cognition process of human.

2 Related Works

Deep neural networks could learn to predict dimensional emotional valence and arousal [11, 12]. In the Audio/Visual Emotion Challenge and Workshop (AVEC 2016), studies showed that using machine learning algorithm, computers could automate the process of recognizing emotional arousal and valence [12]. Researchers found that using Convolutional Neural Networks (CNN), they were able to make predictions on valence ratings on videos using visual features. Likewise, other studies demonstrated that using deep neural network, computers were able to recognize emotion in speech [13, 14].

Recurrent neural networks (RNN) were widely used in studies of making continuous predictions. RNN assumes that the predictions at current state depend on the predictions made in the previous states [15, 12]. Studies in natural language processing proved that RNN was a powerful model for sequential data [16]. Similarly, studies proved that RNN, combined with convolutional neural networks, increased the accuracy in video-based emotion recognition [17]. Likewise, researchers developed an automatic emotion recognition framework using Long Short-Term Memory Recurrent Neural Networks (LSTM) with linguistics, audio and visual features. They proved that their model were able to discriminate between high and low levels of arousal, expectation, power, and valence [18].

LSTMs were widely used in making continuous predictions for visual cognition such as natural language descriptions [19, 20], video paragraph captioning [21], video classifications [22] and video-based traffic prediction [23]. Specifically, LSTM had been proved to be successful in making predictions in emotion valence in video-based dataset [24, 18]. Studies claimed that LSTM could learn temporal dynamics and had distinct advantages over traditional deep neural networks [23]. Unlike RNN, LSTM simulated the actual process of human's emotion cognition process by considering how the emotion evolved over time in optimized contextual knowledge. Researchers successfully used LSTM to predict continuous emotion valence and arousal scores using features from audio, facial videos, ECG and EDA [24]. Likewise, previous literature stated that LSTM showed large improvements in grammaticality in the generation of natural language descriptions of videos [19].

Transformer became the state-of-the-art *de facto* to the natural language process (NLP) community [25]. The Transformer model is constructed on attention, which embeds each word or window in a sentence with the words surrounding the center words. This closely models the human interpretation process of languages, as human often pay attention to keywords when interpreting the meaning of text [26]. The Transformer-based models are used in various tasks, including neural text comprehension [27], QA agent [28] and neural machine translators [29]. In our paper, we propose four different models based on Transformer, LSTM and auto-regression models that are the state-of-the-art models in the NLP community.

3 Approach

In this part, we will discuss our approach in detail, including how we construct the models and the reasons behind it. It aims to provide some insights to develop models that could reconstruct continuous emotion valence of the narrator from the linguistic inputs. We provided a detailed structural diagram for all models in the appendix.

3.1 Evaluation Method

We used the Concordance Correlation Coefficient (CCC [30]) as the metric to evaluate our models. It compares the dynamic similarities of two lists of time-series data points. It was widely used in previous studies in modeling time-series data [31]. The CCC for two time-series vectors X and Y is

calculated as:

$$\begin{aligned} \text{CCC}_{XY} &\equiv 1 - \frac{E\left[(X-Y)^2\right]}{E\left[(X-Y)^2\right]|_{\text{setting } \rho_{XY}=0}} \\ &= \frac{2\rho_{XY}\sigma_X\sigma_Y}{\sigma_X^2 + \sigma_Y^2 + (\mu_X - \mu_Y)^2} \end{aligned} \tag{1}$$

where $\rho_{XY} \equiv \text{cov}(X,Y)/(\sigma_X\sigma_Y)$ represents the (Pearson) correlation coefficient, and μ and σ indicate the average and standard deviation respectively. CCC measures agreement between two curves, where 1 means that the two time-series are in perfect agreement and 0 means that they are uncorrelated.

3.2 Human Baseline

In the SENDv1 dataset, we have human ratings of emotional valence for each video. For each of them, we have about 20 samples. Our baseline CCC is the CCC achieved by human annotators. The mean and standard deviation of observer CCC on the **training set** was $.45 \pm .14$, the mean (and SD) observer CCC on the **validation set** was $.47 \pm .120$, and finally, the mean (and SD) observer CCC on the **test set** was $.46 \pm .14$.

3.3 CNN Features Fusion

Most current research fused the word embedding in a window by taking the point-wise average of every vector in the window [32]. However, this is a naive way of representing the sentiment meanings of a window. We introduce a more systematic way for the window level embedding by adding a CNN layer before feeding into the deep neural network. For every time window, we generate the word embedding using GloVe. We padded every window with 0 so they have equal length of words. Then, we used a CNN layer with a Highway network to reduce a window embedding to a single vector. We used 1 dimensional convolution with max pooling for our CNN with one linear projection layer with a gated output layer for the Highway network. ReLu layer is not used as the word embedding from GloVe could result in negative values.

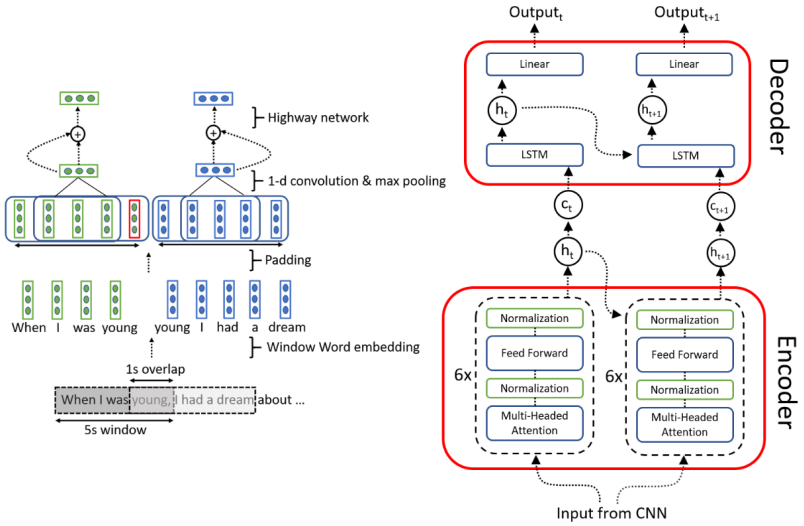


Figure 2: Left: Graphical illustration of overlap-window embedding CNN layer. Right: General model architecture of our LSTM variants. In this figure, we set the encoder to be a Transformer, and the decoder to be a one layer LSTM. This structure is shared among all of our LSTM variants and the Transformer model. The encoder and decorder will be different for these models. See Appendix A for more details.

3.3.1 Long Short-Term Memory Networks

As we described in our previous sections, the Long Short-Term Memory (LSTM) network performs well in language related tasks. It captures the information propagation through time. It simulates the process of emotion expression of human, as human's emotion largely depends on the emotion state from the previous time point.

In this paper, we adopted the framework that is used before in predicting cognition process of empathy [32]. We re-implemented three LSTM variants with the "vanilla" **LSTM**. In addition to the vanilla **LSTM**, we implemented auto-regressive LSTM (**AR-LSTM**) which would explicitly make a prediction on the current time point based on the prediction from the last time point. We also implemented the Encoder-Decoder LSTM (**ED-LSTM**) which is applied in many other fields, including language translations.

We constructed all of our LSTM variants in an encoder-decoder way. These LSTM variants differ in the decoder part. For all of our models, we first feed in the window embedding for every time window. We then use the encoder layer to again embed the window. Next, we compute a local attention layer using a Multi-layer Perceptron with an attention window of length l which has been proved to be effective in building cognitive models [32]. This means that, at time t, we compute a set of l attention weights which are then used to weight the hidden states at the previous timesteps, to give a context vector c_t:

$$h_t = \text{Encoder}(X_{1:t}) \tag{2}$$

$$\{a_{t-l+1}, \ldots, a_t\} = \text{Optimizer}(X_t) \quad \text{(attention weights)} \tag{3}$$

$$c_t = \sum_{j=0}^{l-1} a_{t-j} h_{t-j} \quad \text{(context vector)} \tag{4}$$

These outputs are then fed into the decoder layer which is different across our variants. For LSTM, we used a linear layer as the decoder. For AR-LSTM, we used a linear layer which takes the current network output as well as the prediction from last time point. Finally, for ED-LSTM, we used another LSTM decoder layer which takes the current encoder output and the hidden state output from the last time point to make prediction.

$$\text{LSTM:} \quad \hat{Y}_t = \text{MLP}(c_t) \tag{5}$$

$$\text{AR-LSTM:} \quad \hat{Y}_t = \text{MLP}(c_t, \hat{Y}_{t-1}) \tag{6}$$

$$\text{ED-LSTM:} \quad \hat{Y}_t = \text{LSTM}(c_t, \hat{h}_{t-1}) \tag{7}$$

We used the mean squared error (i.e., $\text{MSE}(\hat{Y}_{1:T}, Y_{1:T}) = \sum_{t=1}^{T} (\hat{Y}_t - Y_t)^2$) as the loss function to be minimized.

3.3.2 Transformer

As stated in previous sections, the Transformer model is the state-of-the-art language modeling methodology in the community. It closely simulate the process of language understanding process of humans. As for interpreting the emotion, one should expect that annotators pay close attention to keywords from the text inputs. In this paper, we built our Transformer model and attach an output header to it. The Transformer model shares the same structure as our LSTM models. However for the encoder layer, we implement a Transformer network which computes the embedding only based on self-attention scores. The Transformer model we implemented is a multi-head transformer. It has 6 identical sub-encoder layers, each of which contains an attention layer and a feed-forward layer. The connection layer between encoder layers is a linear normalization layer with a dropout rate of 0.1. The Transformer model encodes the original window embedding from CNN, which is then fed into an LSTM decoder layer, the same one as the decoder of the ED-LSTM model.

3.3.3 Customized Loss Function

Besides mean squared error, we also implemented a customized loss function using CCC value. Every video outputs a predicted CCC score, which is calculated from predicted ratings. The number of point estimates in each video corresponded to the number of windows. The model was trained

by maximizing the sum of CCC scores over all videos. The CCC result on evaluation set slightly improved compared to MSE loss, which conformed to our expectation as the goal was to directly maximize CCC score.

4 Experiments

4.1 Dataset

The dataset we used contains 193 video clips. We divided the current dataset into Training (60% of the dataset, 117 videos), Test (20% of the dataset, 38 videos) and Evaluation (20%, 38 videos) sets. This dataset contains videos, acoustic sound waves and transcripts for every video.

We are interested only in linguistic features. However, acoustic features are partially used in force alignment[2], which is used to generate corresponding time stamps for words. We used GloVe 300 to embed words. After we got the embedding vector for all the words, we replaced NaN values with 0 to avoid undefined behaviors. We normalized the ratings, originally ranging from 0 to 100, down to 0 to 1. We also implemented overlap windows to generate more features.

4.2 Model Configuration

Based on our hyper-parameter tuning results, our CNN embeds a window, which originally contains a sequence of 300-dimensional GloVe word vectors, into a 256-dimensional vector. Because our main goal is to provide insights in constructing emotion cognition model, parameter tuning is not our priority. To minimize our training time, all our LSTM layer was a single layer LSTM with various local attention length of 3. All of our LSTM had a hidden size of 256. Our Transformer model had 6 encoder layers with hidden size of 128. The feed forward layer in Transformer had a hidden size of 128. We trained all our models with an initial dropout rate (on the input embedding) of 0.1, which helped us regularize the learnt weights and help prevent over-fitting [33]. All of our models support training in CUDA. We trained our models using NV6 virtual machine (1 x M60 GPU (1/2 Physical Card)) provided by Microsoft Azure. The max epoch we set was 1000. The training time for each of our models was less than 1 hour.

4.3 Results And Analysis

4.3.1 Evaluation And Tuning

In this section, we will describe the results from our training process. It includes model performances on the evaluation set using MSE loss. All these results are calculated after hyper-parameters tuning. We used MSE in this stage since the MSE loss is asymptotically the same as CCC loss function. As the MSE loss goes to 0, the CCC will go to its max value, 1. Since we are predicting a continuous

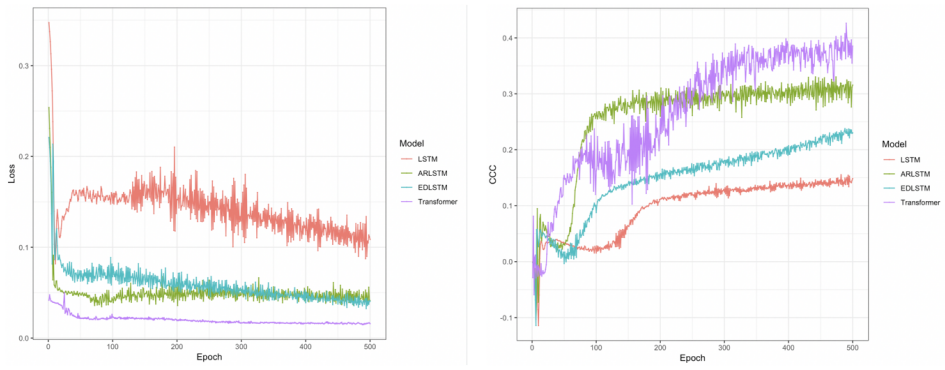


Figure 3: Left: Loss on Evaluation set of all models. Right: CCC on Evaluation set of all models.

[2]https://github.com/ucbvislab/p2fa-vislab

Evaluation Set					
Window Size & Overlap Size	**Statistics**	**LSTM**	**ED-LSTM**	**AR-LSTM**	**Transformer**
Window = 5s, Overlap = 0s	MSE Loss	0.048	0.031	0.025	**0.014**
	Corr	0.365	0.533	0.442	**0.612**
	CCC	0.224	0.351	0.325	**0.44**
	Best CCC	0.801	0.92	0.915	**0.983**
Window = 5s, Overlap = 2.5s	MSE Loss	0.05	0.03	0.027	**0.014**
	Corr	0.346	0.53	0.443	**0.548**
	CCC	0.217	0.355	0.334	**0.434**
	Best CCC	0.8	0.929	0.921	**0.978**
Window = 10s, Overlap = 5s	MSE Loss	0.063	0.029	0.058	**0.015**
	Corr	0.367	0.518	0.398	**0.557**
	CCC	0.218	0.337	0.243	**0.397**
	Best CCC	0.773	0.964	0.769	**0.975**
Window = 5s, Overlap = 0s, with CCC Loss	CCC	-	-	-	**0.493**

*** Human Evaluated Baseline CCC with visual and audio features = **0.47**

Test Set					
Loss Function	**Statistics**	**LSTM**	**ED-LSTM**	**AR-LSTM**	**Transformer**
CCC Loss	CCC Mean	0.291	0.095	0.296	**0.449**
	CCC Std	0.272	0.329	0.31	**0.352**
MSE Loss	CCC Mean	0.149	0.251	0.177	**0.469**
	CCC Std	0.241	0.299	0.269	**0.301**

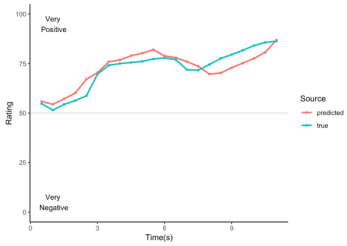


Figure 4: Left: Table that shows the comparison of performances of different models using our tuned parameters. The performance of Transformer is the best. Right: Plot of our predictions against the target value of one video in the evaluation set. The CCC for this is 0.98. These two plots are generated using MSE loss function. We used MSE to tune our hyper-parameters.

rating, we plotted the MSE loss, correlation between prediction and target curves, and CCC score of our prediction against the epoch in our training process. All these were evaluated on the Evaluation set. The performance of the Transformer model was the best. The "vanilla" LSTM performed relatively worse than the others. However, the loss of each model converges during the training process.

Our models, including LSTM, AR-LSTM, ED-LSTM and Transformer, are trained for 1000 epochs and evaluated on the evaluation set. They are all trained with the same random seed and the same training dataset. The training time is approximately 1 hour and is very close for each model.

Based on the results, Transformer outperformed all other models with a highest CCC of 0.49 with MSE loss on the Evaluation set. With partial information from linguistic and audio features, the results obtained from Transformer model is close to human performance in the Test set. For a single video, Transformer can reach a maximum CCC of 0.98. We plotted one example of our predictions against the true values based on the Transformer model (CCC=0.98).

The predicted curve are following the true values closely. This is expected as we are having previous information passed along as making new predictions in the future time points. Transformer model is out-performing other RNN variants. This is expected as Transformer provide a better window embedding at each time point.

4.3.2 Customized Loss Function And Testing

We implemented our own CCC loss function. Based on the results, we found that the CCC loss function tended to work better in the beginning of the training process, But it had much higher variance in the later training process. MSE loss function converged to a lower loss in the training process. However, CCC loss function achieves higher accuracy in Evaluation set in the early stage of the training process. We compared the performance of models with MSE and CCC loss function in the Test set. We found that MSE loss gets better results. The Transformer model is reaching 0.47 in the Test set.

4.4 Sentiment Analysis

In order to interpret the model, we collected a natural emotion story online. We then generated a word embedding file for each word in the file. We also annotated each word with a starting and ending timestamp. We set the story to be at a speed of 0.1 second per character. Between each word, there is a 0.2 second pause. Based on the plot and the actual words in the windows, we conclude that sudden turns in the sentiment changes are well-captured by the model. Turn words like *But* and *Suddenly* tend to change sentiment trends in this example. We can also see that negative words cause turning in the sentiment trends as well.

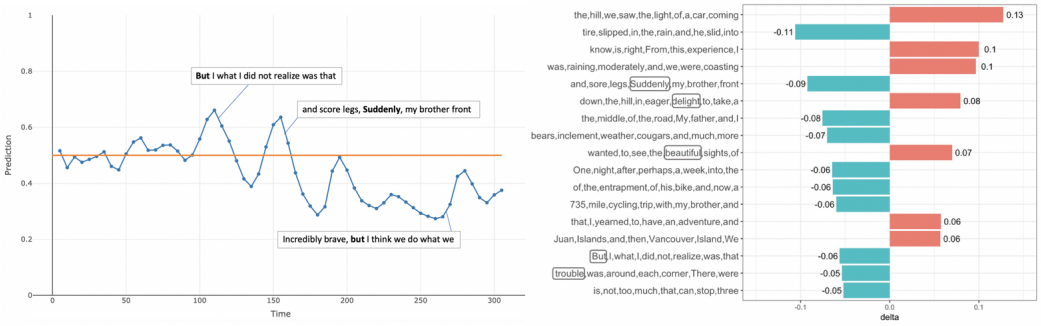


Figure 5: Left: The sentiment reconstruction of a personal anecdotes available online [4]. Right: The changes of emotion valences produced by the indicated windows.

4.5 Conclusion

In order to recover the emotion hidden in the text, researchers need to overcome various challenges of modeling emotion cognition process continuously. In this paper, we first propose four different deep learning models to predict the emotion of linguistic inputs. The performance of the Transformer model is close to our human annotators who rate the emotion based on visual, audio and linguistic inputs. We conclude that our Transformer model could potentially behave the same way as brain does in reconstructing emotion valence from text.

In this paper, we propose a Transformer model making continuous emotion valence prediction on text. Based on our literature reviews, the Transformer model is used in various NLP tasks. In this paper, we also find that attention-based model performs the best in emotion cognition modeling. We propose a new CCC loss function for models that make continuous ratings over time. Based on our models, we then compare the performances of our customized loss function which is based on CCC with the standardized MSE loss function. Though MSE is often used in the research community for optimizing the loss, we find that the CCC loss function performs better in our Evaluation set and is close to MSE in the Test set. In the final part of the paper, we compare human's reading behavior with the model's behavior. We find that the model is catching the turn words in the text and shifting the emotion valence accordingly. We find that turn words are the driver words in emotion cognition with text.

Most importantly, we propose a model that is reconstructing the storytellers' emotion in a way that is the same as human. Trained using the emotion ratings as the target ratings, our model is using lower-dimensional features, linguistic only features, to make predictions on a higher-dimensional space, the ratings that are annotated using visual, audio and linguistic inputs. This is matching human's cognition process of interpreting the emotion changes of a book or a story, which only contains the text inputs. Human's emotion cognition ability is trained or built by millions of short moments in lives, which contains visual, audio and linguistic inputs. Thus, when readers catch the emotion of a story, they are reconstructing the whole story in mind as if they were in the story. By training on a higher-dimensional target, we propose that our model closely match the human's emotion cognition process of text.

In conclusion, modeling the emotion cognition process with linguistic inputs remains a difficult yet important building block of understanding the emotion cognition process of humans. In this paper, we have outlined ways to construct valid and effective models to make continuous emotion valence predictions over large text corpus. We hope this paper will inspire more researchers to accomplish more ambitious and quantitative results in the future.

References

[1] Desmond C Ong, Jamil Zaki, and Noah D Goodman. Affective cognition: Exploring lay theories of emotion. *Cognition*, 143:141–162, 2015.

[2] Desmond C Ong, Jamil Zaki, and Noah D Goodman. Computational models of emotion inference in theory of mind: A review and roadmap. *Topics in Cognitive Science*, 2018.

[3] Kristen A Lindquist, Ajay B Satpute, and Maria Gendron. Does language do more than communicate emotion? *Current Directions in Psychological Science*, 24(2):99–108, 2015.

[4] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[5] Adam DI Kramer. An unobtrusive behavioral model of gross national happiness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 287–290. ACM, 2010.

[6] Laurence Steinberg. Cognitive and affective development in adolescence. *Trends in cognitive sciences*, 9(2):69–74, 2005.

[7] Pooya Khorrami, Thomas Paine, and Thomas Huang. Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 19–27, 2015.

[8] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014.

[9] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Au-aware deep networks for facial expression recognition. In *FG*, pages 1–6, 2013.

[10] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[11] Pooya Khorrami, Tom Le Paine, Kevin Brady, Charlie Dagli, and Thomas S Huang. How deep neural networks can improve emotion recognition on video data. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 619–623. IEEE, 2016.

[12] Michel Valstar, Jonathan Gratch, Björn Schuller, Fabien Ringeval, Denis Lalanne, Mercedes Torres Torres, Stefan Scherer, Giota Stratou, Roddy Cowie, and Maja Pantic. Avec 2016: Depression, mood, and emotion recognition workshop and challenge. In *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM, 2016.

[13] Joy Nicholson, Kazuhiko Takahashi, and Ryohei Nakatsu. Emotion recognition in speech using neural networks. *Neural computing & applications*, 9(4):290–296, 2000.

[14] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth annual conference of the international speech communication association*, 2014.

[15] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[17] Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 467–474. ACM, 2015.

[18] Martin Wöllmer, Moritz Kaiser, Florian Eyben, BjöRn Schuller, and Gerhard Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153–163, 2013.

[19] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. *arXiv preprint arXiv:1604.01729*, 2016.

[20] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[21] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016.

[22] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[23] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[24] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. Long short term memory recurrent neural network based multimodal dimensional emotion recognition. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 65–72. ACM, 2015.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[26] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

[27] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[29] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

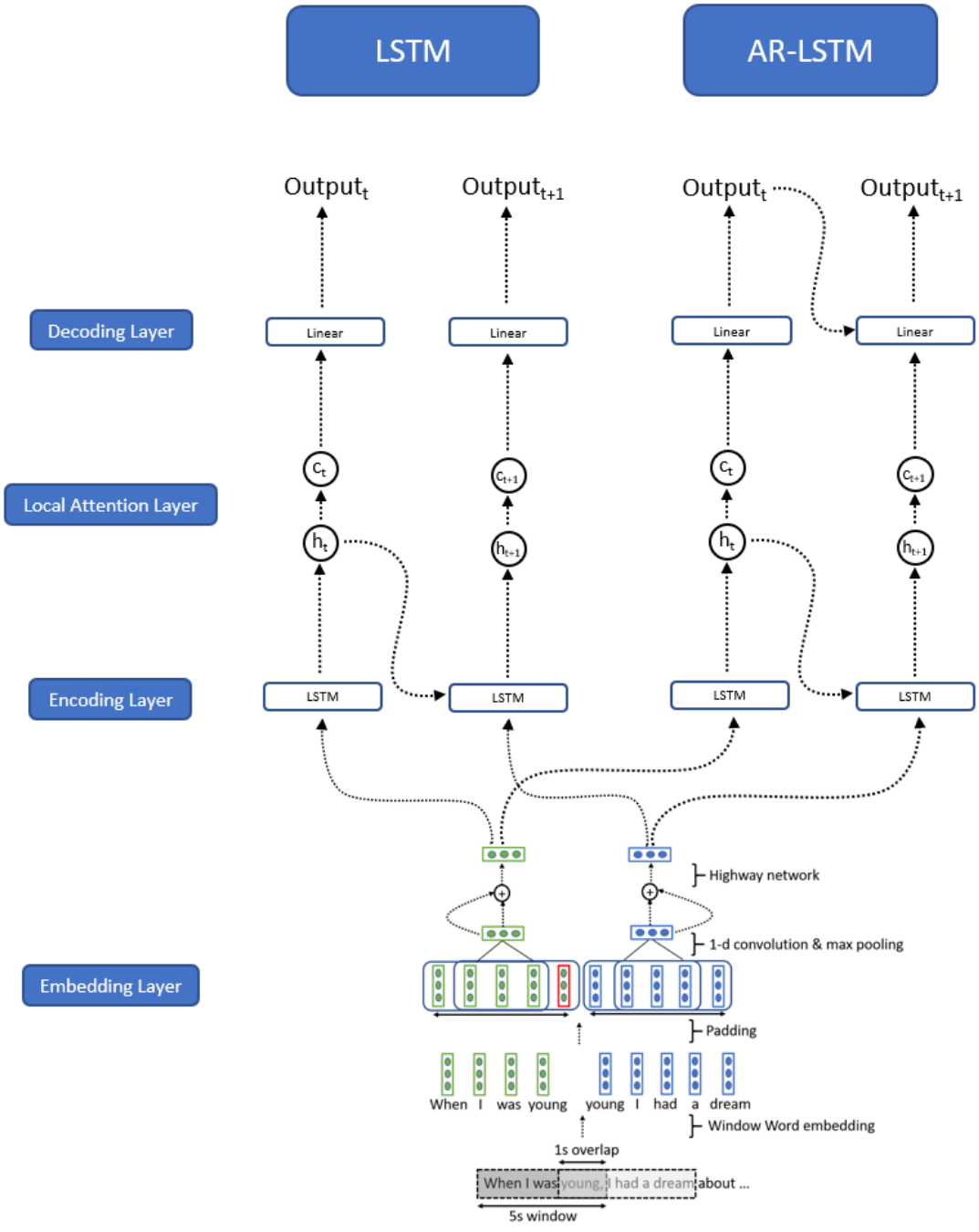
[30] Lawrence I-Kuei Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989.

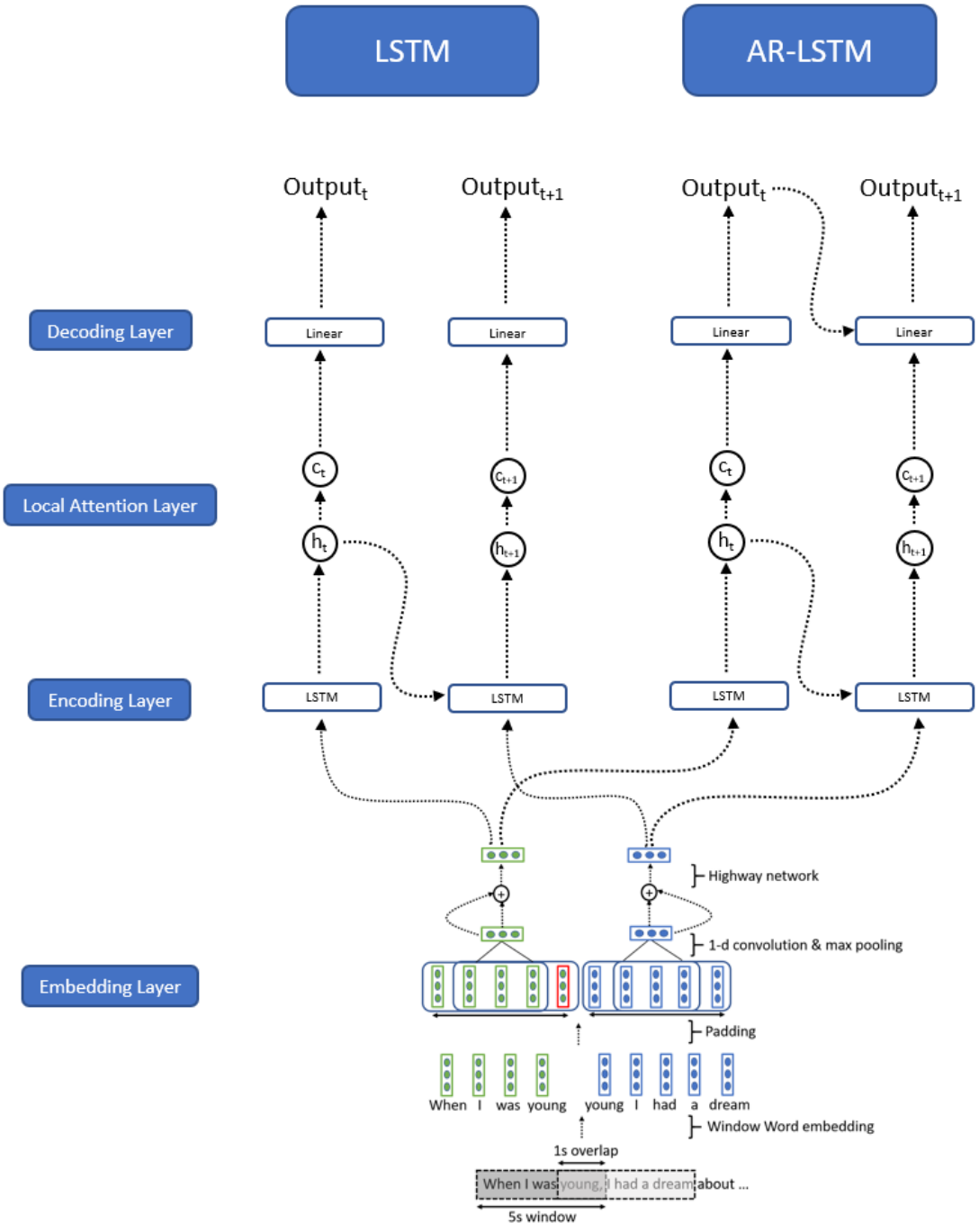
[31] Fabien Ringeval, Björn Schuller, Michel Valstar, Jonathan Gratch, Roddy Cowie, Stefan Scherer, Sharon Mozgai, Nicholas Cummins, Maximilian Schmitt, and Maja Pantic. AVEC 2017: Real-life depression, and affect recognition workshop and challenge. In *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*, pages 3–9, 2017.

[32] Zhi-Xuan Tan, Arushi Goel, Thanh-Son Nguyen, and Desmond C. Ong. A multimodal lstm for predicting listener empathic responses over time, 2018.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Appendix A: Visualizations of Model Structures




LSTM
AR-LSTM
Output$_t$
Output$_{t+1}$
Output$_t$
Output$_{t+1}$
Decoding Layer
Linear
Linear
Linear
Linear
Local Attention Layer
c_t
c_{t+1}
c_t
c_{t+1}
h_t
h_{t+1}
h_t
h_{t+1}
Encoding Layer
LSTM
LSTM
LSTM
LSTM
Highway network
1-d convolution & max pooling
Embedding Layer
Padding
When I was young young I had a dream
Window Word embedding
1s overlap
When I was young, I had a dream about ...
5s window