The TransGesture entry to the GENEA Challenge 2022

ANONYMOUS AUTHOR(S)

This paper presents a gesture generation model based on an RNN-transducer consisting of three neural networks: Encoder, Prediction Network, and Joint Network. We also introduce new loss functions, namely statistical losses, as the additional term to the standard MSE loss. We show the subjective evaluation results and discuss the results and takeaways from the challenge.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**; • **Computing methodologies → Neural networks**.

Additional Key Words and Phrases: gesture generation, speech audio, neural networks, deep learning

ACM Reference Format:
Anonymous Author(s). 2022. The TransGesture entry to the GENEA Challenge 2022. In *GENEA Challenge '22: Generation and Evaluation of Non-verbal Behaviour for Embodied Agents, November 07–11, 2022, Bengaluru, India.* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Non-verbal information, especially gestures, play an important role in human communication by emphasizing, supporting, or complementing speech [9, 14]. Researchers have discovered that introducing non-verbal expressions into robots or embodied agents has positive effects on human-agent interaction [13, 15]. Thus, enabling such robots/agents to accompany their speech with gestures is important to facilitate natural human-agent interaction.

As the manual implementation of gestures is time- and labor-consuming, automatic gesture generation has attracted the community's attention. Nowadays, using data-driven approaches has been the mainstream of the research. Researchers have explored several input modalities, including audio [11], text [19], and speaker identities [18]. From another perspective, a long-time trend in automatic gesture generation is adopting architectures of speech recognition and machine translation. For example, DeepSpeech [6], Seq2Seq [16], and Transformers [17] are used in many gesture generation studies [1, 7, 19]. These models are broadly categorized into *encoder-based models* (e.g., DeepSpeech) and *attention-based encoder-decoder models* (e.g., attention Seq2Seq and Transformers). However, in the speech recognition field, the third type of model, namely *RNN-transducers* [5], has attracted recent attention from the community [8].

An RNN-transducer is composed of three networks, namely Encoder, Prediction Network, and Joint Network, as depicted in Figure 1. Encoder is similar to the ones used in encoder-based models. It takes input signals (e.g., audio or text) and maps them to latent representation. Prediction Network is autoregressive; it takes the transducer's previous output and maps it to another latent representation, which is useful to predict the next output. Joint Network takes the two latent representations, joins them, and produces outputs. RNN-transducers have several good properties for gesture generation over encoder-based models and attention-based encoder-decoder models. Unlike encoder-based models, RNN-transducers are autoregressive; their successive outputs are assumed to be not independent. Attention-based encoder-decoder models require whole input at once to compute attention, whereas RNN-transducers can be streamable.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
Manuscript submitted to ACM

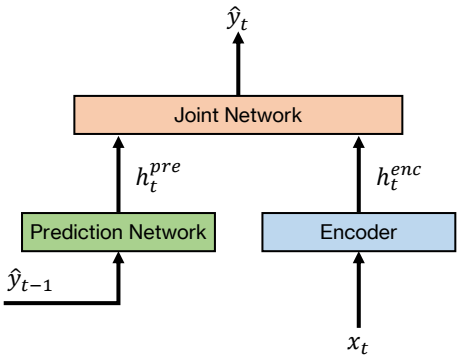

Fig. 1. A schematic diagram of RNN-transducer.

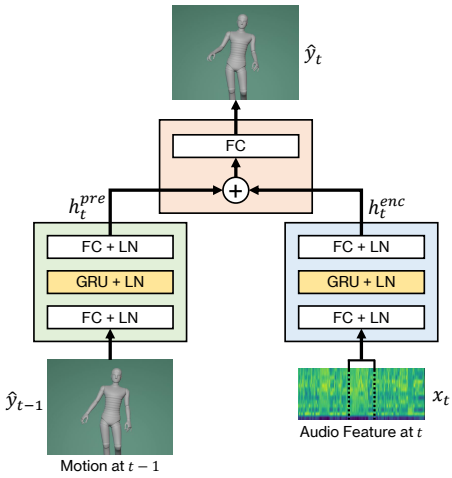

Fig. 2. Architecture of our gesture generation model.

To the best of our knowledge, RNN-transducers have not been adopted for gesture generation despite these preferable properties. In this challenge submission, we examine an RNN-transducer-based model for gesture generation. We introduce statistical loss functions to put statistics of generated gestures close to the ground truths'.

2 METHOD

2.1 Input and Output Representations

Following the previous studies of audio-driven gesture generation methods, we choose Mel frequency cepstral coefficients (MFCC) as input audio features. First, we downsample the raw audio files from 44.1 kHz to 16kHz. We extract 13 MFCC features from Mel spectrograms of 26 Mel filter banks, extracted from the raw audio files at 30 frames per second (FPS) using Torchaudio 0.10.1. We set a window length to 0.025 seconds and set the hop length to the reciprocal of the FPS (i.e., 1/30 seconds).

The output motion representation is joint angles relative to a T-pose, which are parameterised using the exponential map [4] at 30 FPS. Each dimension of the joint angles is normalized by subtracting the mean and dividing by the absolute maximum over the training set, resulting in the range of $[-1, 1]$. For ease of training, we consider the "upper body" (excluding fingers) joint set.

2.2 Network Architecture

Our gesture generation model is an RNN-transducer, as shown in Figure 2. The model is composed of three networks, namely Audio Encoder, Prediction Network, and Joint Network. Each network has 256 channels in its hidden layers.

Audio Encoder takes MFCC features at time t, namely x_t, and processes it through a set of a fully connected layer, a two-layered Gated Recurrent Unit (GRU) [3], and a fully connected layer, resulting in a latent feature vector h_t^{enc}. We apply Layer Normalization [2] to each layer.

Prediction Network is autoregressive. It takes the previous output of Joint Network, $\hat{y}_{t-1}$, and produces a latent feature h_t^{pre}, which is taken by Joint Network to estimate the output of current time $\hat{y}_t$. We use the same architecture for Prediction Network and Audio Encoder.

Joint Network takes the two networks' outputs, h_t^{enc} and h_t^{pre}, to outputs motion of current time $\hat{y}_t$. h_t^{enc} and h_t^{pre} are integrated by element-wise summation followed by a fully connected layer. The output from the last fully connected layer is activated by a hyperbolic tangent (tanh) function to have the value range of $[-1, 1]$.

2.3 Loss Function and Optimizer

Our first loss function is the Mean Squared Error (MSE) between the model outputs and ground truth motions in the exponential map representation. The MSE loss is given by

$$L_{mse} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2, \tag{1}$$

where m is the mini-batch size, i is the index of each sample in the mini-batch, y is the ground truth exponential map, and $\hat{y}$ is the model output.

While MSE is the standard loss function used in gesture generation, we observe that our network outputs tend to collapse to the mean pose. One way to prevent such collapse is adding adversarial discriminator loss, as done in []. However, we empirically found that such adversarial losses tend to unstabilize the training and produce "messy", inhuman-like gestures. Thus, we introduce new loss functions, namely statistical losses. The statistical losses ensure that the statistical characteristics of the generated and ground truth gestures are close. Specifically, our system uses the mean and variance of the gestures, i.e.,

$$L_{mean} = \frac{1}{m}\sum_{i=1}^{m}(Mean(\hat{y}_i) - Mean(y_i))^2, \tag{2}$$

$$L_{var} = \frac{1}{m}\sum_{i=1}^{m}(Var(\hat{y}_i) - Var(y_i))^2. \tag{3}$$

The final loss function is the weighted sum of the above three terms,

$$L = L_{mse} + \lambda_1 L_{mean} + \lambda_2 L_{var}, \tag{4}$$

where we empirically set to $\lambda_1 = 2$ and $\lambda_2 = 5$. We trained the model by Adam optimizer [10] with a learning rate of 0.001, and a mini-batch size of 32. We ran the model training for 10 epochs, where the improvement in validation loss had stopped. We tuned the hyperparameters by manually watching the generated gestures in the validation set.

3 EXPERIMENT AND RESULT

Dataset. The challenge dataset is Talking With Hands 16.2M [12]. We used the official training, validation, and test splits provided by the challenge organizers. Our system was trained on the training split only. We did not perform sample selection, i.e., we used all the samples in the training set. For the details of the challenge dataset, please refer to the challenge paper [20].

Implementation. We implemented our system using Python 3.9.7, PyTorch 1.10.1, and Torchaudio 0.10.1. For BVH file processing and motion parameterization, we used PyMO library provided by the challenge organizers. Before converting the generated exponential maps into BVH files, we applied a Savitzky-Golay filter, as the audio-only baseline system suggested.

Evaluation. As described in the challenge paper [20], this year's challenge evaluated the submitted gestures from two aspects: human-likeness and appropriateness. The label indicating our challenge entry is "USL".

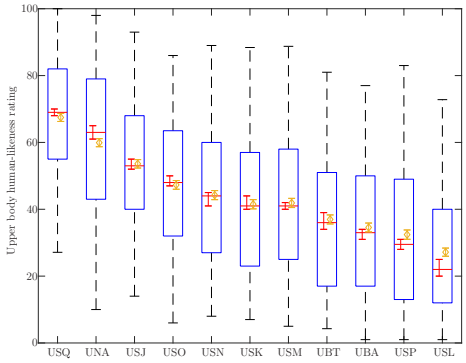

Fig. 3. Human-likeness evaluation result.

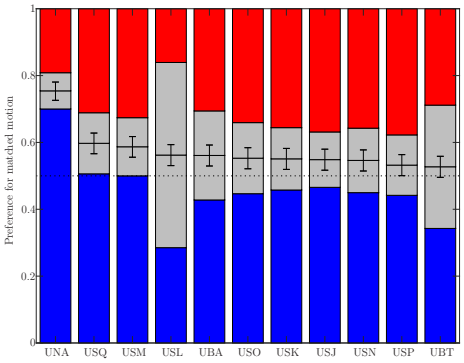

Fig. 4. Appropriateness evaluation result.

3.1 Human-Likeness Evaluation

In this evaluation, participants were asked to watch the *silent* videos of gesture motions and evaluate the human-likeness of the gesture motions with 0-100 score. The evaluation result is presented in Figure 3. Our entry, USL, attained the lowest median score among the participating systems. There are several possible reasons for the low score. First, our entry did not contain finger movements as described in Section 2.1.[1] Second, in some of the evaluation videos, our entry contained no clear gesticulation, i.e., only body sway was performed. Third, we observed that some of our generated gestures were somewhat too symmetric; both arms did similar movements, with similar height and speed. Thus, the evaluation participants might have given low scores to such videos.

3.2 Appropriateness Evaluation

In this evaluation, participants were asked to watch videos of gesture motions *with speech audio*. The participants watched two videos side-by-side: one generated from the matched speech and the other one generated from the mismatched speech, and selected which one was more appropriate (or they were equal) for the given speech. The result is shown in Figure 4, where the blue bar, grey bar, and red bar represent the ratio of responses preferring the matched, tied, and mismatched motions, respectively. Our entry USL has the largest grey bar among all the entries, implying its gesticulation was somewhat obscured.

4 CONCLUSIONS AND TAKEAWAYS

This paper has presented our entry for the GENEA challenge 2022. Honestly speaking, we had had difficulty in generating diverse, meaningful gestures from the challenge dataset. We had spent too much time implementing and debugging the prototype system and had not had much time to tune up and improve it. We found that the dataset contains a non-negligible amount of non-gesticulated frames; exploring a filtering strategy for such samples would be useful. Also, we found that normalization on the output representation affected the gesture quality; our system could not learn gesticulation from unnormalized exponential maps. Since the text-only baseline and audio-only baseline employed different preprocessing steps and output representations (joint coordinates + rotation matrix vs. exponential maps), we took some time to compare and analyze the baseline codes. We have learned and experienced much on this challenge.

[1]The text-only baseline (labelled UBT) also did not contain the finger motion.

REFERENCES

[1] Eiichi Asakawa, Naoshi Kaneko, Dai Hasegawa, and Shinichi Shirakawa. 2022. Evaluation of text-to-gesture generation model using convolutional neural network. *Neural Networks* 151 (2022), 365–375.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

[3] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

[4] F. Sebastian Grassia. 1998. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools* 3, 3 (1998), 29–48.

[5] Alex Graves. 2012. Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711.

[6] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

[7] Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. 2018. Evaluation of speech-to-gesture generation using bi-directional LSTM network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA '18)*. 79–86.

[8] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. 2019. Streaming end-to-end speech recognition for mobile devices. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '19)*. IEEE, 6381–6385.

[9] Adam Kendon. 1980. Gesticulation and speech: Two aspects of the process of utterance. *The Relationship of Verbal and Nonverbal Communication* (1980), 207–227.

[10] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR '15)*.

[11] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. 2019. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents (IVA '19)*. 97–104.

[12] Gilwoo Lee, Zhiwei Deng, Shugao Ma, Takaaki Shiratori, Siddhartha S Srinivasa, and Yaser Sheikh. 2019. Talking With Hands 16.2M: A large-scale dataset of synchronized body-finger motion and audio for conversational motion analysis and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV '19)*. 763–772.

[13] Richard E. Mayer and C. Scott DaPra. 2012. An embodiment effect in computer-based learning with animated pedagogical agents. *Journal of Experimental Psychology: Applied* 18, 3 (2012), 239–252.

[14] David McNeill. 1992. *Hand and mind: What gestures reveal about thought.* University of Chicago press.

[15] Maha Salem, Friederike Eyssel, Katharina Rohlfing, Stefan Kopp, and Frank Joublin. 2013. To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. *International Journal of Social Robotics* 5, 3 (2013), 313–323.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems* 27.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30.

[18] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2020. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–16.

[19] Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2019. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA '19)*. IEEE, 4303–4309.

[20] Youngwoo Yoon, Pieter Wolfert, Taras Kucherenko, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. 2022. The GENEA Challenge 2022: A large evaluation of data-driven co-speech gesture generation. In *Proceedings of the ACM International Conference on Multimodal Interaction (ICMI '22)*. ACM.