ReCell: replicating recurrent cell for auto-regressive pose generation

ANONYMOUS AUTHOR(S)

This paper describes Anonymous gesture generating system entry for the GENEA Challenge 2022. Our system is based on auto-regressive approach imitating recurrent cell. Combined with special windowed auto-encoder and training approach this system generates plausible gestures appropriate to input speech.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: embodied agents, neural networks, gesture generation, social robotics, deep learning

ACM Reference Format:
Anonymous Author(s). 2018. ReCell: replicating recurrent cell for auto-regressive pose generation . In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Automatic animation generation for humanoid characters is becoming increasingly popular. Most 3D characters are still animated using expensive motion capture or a small set of predefined animations.

Recent advances in facial animation generation[1, 4] provide a realistic facial expression from audio. Moreover, JALI Viseme Model[2] was already used in recent projects like Cyberpunk2077. Automatic gesture generation, on the other hand, has several existing approaches, including [3, 6, 7, 10], but no standard, ready-for-production solution. One possible explanation for that is the lack of a unified method for reliable comparison. Multiple standards for humanoid rigs and skeletons are also present making it even more challenging to use open-source solutions. Furthermore, some existing approaches generate motions in 3D, while others generate motions in 2D. As a result, comparing them is difficult.

The GENEA Challenge 2020 [8] was held to address this issue. The main goal of this challenge was to determine which models for automatic gesture generation based on speech recording performed better on the same data. The main result of this challenge was that some models outperformed the previous year's best methods results while remaining far from real motion. Even though the target domain was relatively simple: the dataset only contains recordings for one person, and the generated motion was examined for the upper body without fingers. Because there are numerous ways to improve the challenge target, the organizers decided to hold another challenge.

In comparison to the previous one, the task of GENEA Challenge 2022 is more difficult. The proposed models should generate motions for more than one person during the conversation. A person's behavior during a dialogue differs greatly from that of a single performance. The training and testing datasets included examples of the target person gesticulating while listening to the companion. However, these gestures are independent of speech. This behavior could have a significant impact on motion generation systems based on input speech.

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.
© 2018 Association for Computing Machinery.
Manuscript submitted to ACM

We provide a solution that is based on both the speech input and the history of previously generated motions. This system generates plausible gestures from speech using a special windowed auto-encoder and training approach. We also tried to explain the difference between challenges and resulting models by testing one of the previous approaches that performed well on the previous challenge.

Our paper is organized as follows: Section 2 describes data processing, which is shared by all experiments; Section 3 describes our models; Section 4 discusses our results; and Section 5 is for the conclusion.

2 DATA PROCESSING

This time, the organizers of the challenge provided a large dataset based on the "Talking With Hands 16.2M"[9] dataset. It includes motion capture recordings as well as audio and textual transcripts for several persons having a conversation. Mocap data includes motion data for the entire body, including the fingers.

Exploring the data we found several peculiarities which could affect the data-processing pipeline:

- Mocap skeleton differs from common industry formats (for example, Blender's Meta-Rig): the spine has a discontinuity, shoulders have extra bones and hands have a different wrist structure.
- Hand motion is inconsistent: some recordings lack data for finger motion, while others contain glitches and jerks.
- Audio recordings contain speech for both speakers. Some audio cuts are applied to the track in order to conceal confidential information, which greatly complicates speech processing.

To overcome the aforementioned difficulties, we attempted to construct the following data processing pipeline. The pose is represented by 3-dimensional axis-angle rotation vectors for all bone joints and also the position of the root bone. We also leave only non-constant values and normalize them over the training dataset's mean and maximum values. As the result, pose representations with 164 real value features are produced. We tried different representations for audio (MFCC, mel-spectrogram, log mel-spectrogram) with different framerate but found no significant difference during training. We chose MFCC extracted at the same framerate as motion (30fps). To distinguish the target person's speech from the other, we added the textual information as follows. First, we built a vocabulary of symbols on the training dataset. Second, we use one-hot vectors to represent symbols. Finally, we concatenate audio features with obtained vectors that are uniformly distributed within word spans. Figure 1 shows an example of such alignment.

3 MODELS

3.1 Seq2seq

We started to design our architecture taking inspiration from a solution that performed well in the previous competition. The seq2seq model by [5] based on both audio and text features achieved the highest median score on human-likeness and the second highest on appropriateness. Unlike the previous work, we propose eliminating separate encoders for audio and text features. We also decided to use a simple seq2seq instead of a context encoder. To keep the context encoder's main feature of using more context, we decided to use a wider input features window than the length of the output predicting sequence. Other components, such as the loss function and decoder initialization, remained unchanged. Figure 2. depicts the final seq2seq model scheme.

Here $a_{i-p}, \ldots, a_{i+f}$ is an input features window for output motion window $m_i, \ldots, m_{i+k}$, $f >= k$, $m_{i-p}, \ldots, m_{i-1}$ motions from the previous step.

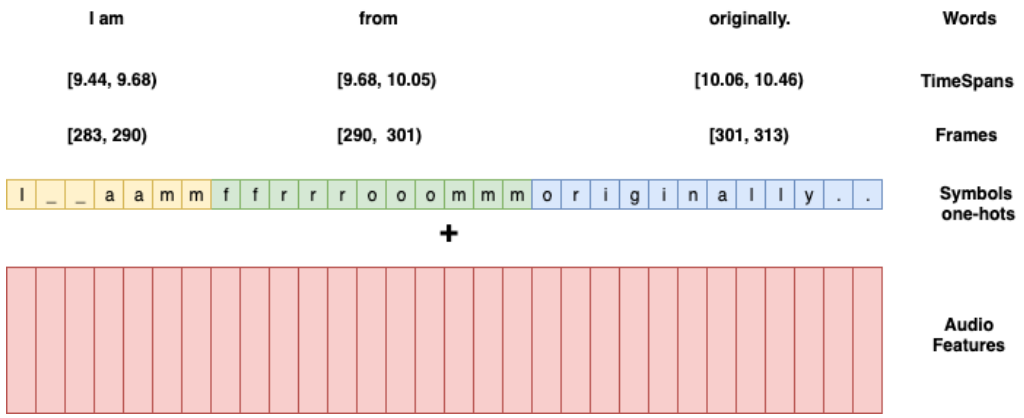

Fig. 1. Example of audio and text features alignment

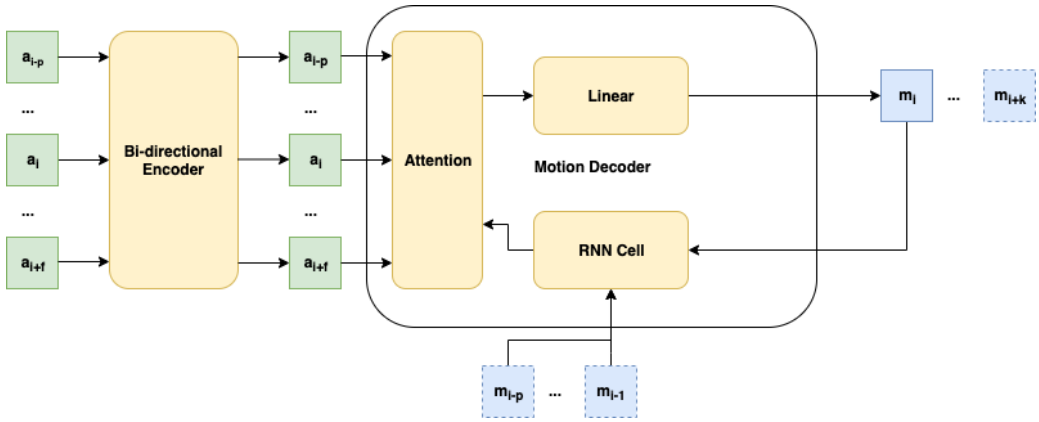

Fig. 2. Scheme of Seq2Seq model

During our tests, this model produced good loss function values, but the resulting motions converged to minor movements around the rest pose. We also tried some training strategies, such as feeding zeros instead of previous poses and adding additional dropouts inside the motion decoder to force the model to focus on input features. However, this shows only minor improvements.

We believe this defect in the method manifested itself due to the significant differences in the datasets used in this and previous challenges. The last year's challenge dataset only included one actor's performance. There were no long pauses in monologue, in contrast to the dialogue speech, and no audio artifacts, allowing the model to create a consistent mapping between speech and gestures. As a result, we require a new model to address the following issues:

(1) Predict expressive motions from actual speech
(2) Realistically continue motion in the absence of speech

We tried to solve the aforementioned problems with the following architecture.

3.2 ReCell

We discovered that the model provided in [6] generates a variety of poses but they are unstable. This model creates a single frame of animation from a sliding window of audio features. We used auto-regressive input to stabilize predicted motions. We also tweaked the feature window encoder. The original model for frame i takes input a i-k/2,. . . ,a $i + k/2$, passes it through a single-direction GRU, and takes the last output. We decided to take the $k/2$-th output from bidirectional GRU, which corresponds to the current frame. During our tests, this modification produced slightly better results. We called this model "ReCell" because it replicates recurrent cell behavior. Figure 3 depicts the model's architecture.

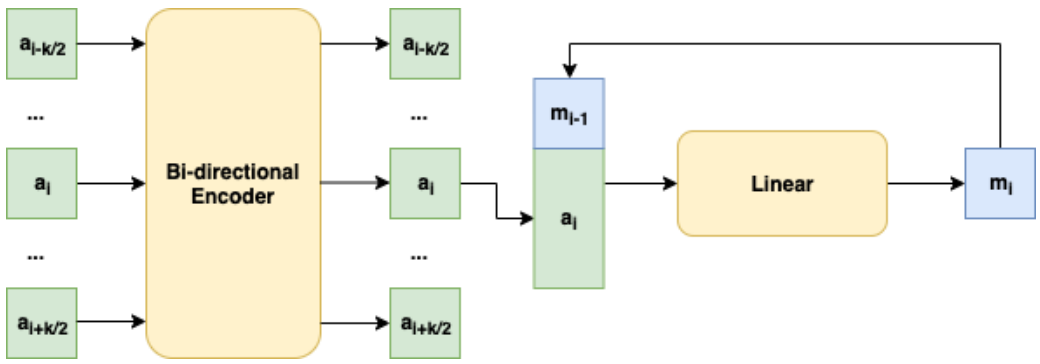

Fig. 3. ReCell model architecture

To avoid the seq2seq model's convergence to a rest pose, we should strike a balance between auto-regressive input and speech features. The authors of [7] faced the same problem. To overcome this they forced the model to extract useful features from the speech input by pretraining the model without auto-regression. Then they also used the teacher forcing technique to let the model properly integrate auto-regressive input. As the result, we attempted to implement these ideas in the following manner.

Our validation step uses a short sequence of successive frames as well as the previous step's output frame. Then we iterate through this sequence, passing the output from the previous iteration to our model speech features. In the training step we only change auto-regressive inputs:

- During the first 10 epochs, we zero the last frame from the previous training step as well as some frames between iterations.
- After the first 10 epochs, we do not zero the last frame from the previous training step, but instead, replace some intermediate frames with real ones.

The precise strategy for replacing frames between iterations along with model parameters will be provided later. On prediction, we go through the entire recording frame by frame, beginning with zero frame as auto-regressive input.

3.3 Windowed auto-encoder

To overcome motion instability we trained a simple linear auto-encoder to reduce the dimension of the output vectors. It has two major components, as usual: encoder and decoder. Input poses are projected into lower-dimensional space by the encoder. In turn, the decoder must reconstruct the input pose from its lower-dimensional representation.

The encoder has one hidden layer, that converts the input vector to a size of 512. After that, we apply ReLU as an activation function. Then we apply another linear layer and Tanh function to normalize. The final size of the output vector is 60.

We use one additional hidden linear layer to restore the vector from the encoded state, which translates the vector into dimension 512. Next, a ReLU and an output linear layer were used, which translates the vector into dimension 164. Since normalized features from the value range [0, 1] were fed to the auto-encoder input, the decoder output was also normalized by Tanh and shifted to [0, 1].

We discovered that when such an auto-encoder is trained, the reconstruction is mostly similar to the input pose, but it occasionally provides unrealistic poses for the left shoulder. We believe this behavior is related to rare jerks in training data. To address this issue, we decided to improve our auto-encoder by reconstructing a short sliding window of poses rather than the single one. As a result, a configured vector of three frames was fed to the auto-encoder, implying that the initial and output states were of dimension 492. Otherwise, the auto-encoder architecture was similar to that described above. We average the outputs in which the current frame participates during inference.

3.4 Final pipeline

Our final approach is as follows. As the main model, we used the ReCell model with the following parameters.

First of all, our Bi-directional encoder consists of linear highway and GRU. Linear highway, in turn, consists of three sequential blocks with linear layer, batch normalization, and dropout with ReLU as the activation function to encode audio features. As mentioned before, we use GRU to extract contextual features from the window of audio features. The size of this window is 61. All hidden sizes are 150.

At the same time, we put the pose from previous step through the another highway with bottleneck of size 40. It contains only two linear layers with ReLU between them and dropout to prepare model output for auto-regression. It enables us to obtain more complex and informative features, as well as translate vectors to the required size. This operation we call Hidden Highway.

Finally, outputs of Bi-directional encoder and Hidden Highway are concatenated together to create final feature vector. Then, we apply another batch normalization, ReLU and Dropout to the resulting feature vector and push it through the last linear layer to produce the next pose.

During the teacher-forcing, we zeroed frames between iterations with a probability of 0.5 in the first ten epochs, and we also replaced some frames with real ones with a probability of 0.5 in the following epochs.

To train the network, we use MSELoss and the Adam optimizer with default parameters from PyTorch.

We also train the main model to predict audio-encoder representations rather than poses. Then, using the Savitsky Golay filter, we reconstruct the poses and smooth the predictions. We use 26 MFCC coefficients concatenated with one-hot encodings of text transcripts as input speech representation.

4 RESULTS AND DISCUSSION

As in the previous challenge, the organizers provide [11] human-evaluation results for comparing the system. As before there are two main metrics: human-likeness and appropriateness. The first evaluates how predicted motions are realistic, and the second estimates the correspondence between speech and predicted motions from it. While human-likeness evaluation follows the previous challenge, the approach to estimating appropriateness is different.

Investigating the results of the previous challenge organizers found that calculated appropriateness was highly correlate with the quality of generation motion. Therefore, this time they decided to calculate appropriateness in

a different way. They ask evaluation study participants not to estimate the result videos alone but compare the corresponding audio-video pair with the mismatched one. It could help to find out if generated motion really appropriate to speech or just random.

Our system achieved slightly poor results for both metrics for the full-body motion: 34 median scores for human-likeness and 51.5 of matched percent for appropriateness. Nevertheless, it outperforms the provided baseline by human-likeness. Although, there are models that showed significantly better results. Moreover, there is a model that outperforms even real motion by human-likeness. This leads us to the fact that there is a huge room to improvement.

The low score in human-likeness could be connected with insufficient fluidity of movement and periodic jerks. It could be smoothed with an even stronger filter but it could make movements «like in jelly». Although the lowest score of appropriateness means that generated motions do not have any semantics. It is only hand movements supporting the speech rhythm. To add semantics the different features could be used, like Glove or BERT embeddings for textual transcripts. It is also could be improved with extensive data cleanup. For example, only motions corresponding to actual speech could be used for training. Thus, even the seq2seq approach could be improved and maybe shows better results.

5 CONCLUSION

The proposed challenge raised issues that are difficult to solve directly with a tried-and-true method. However, with additional data processing and training techniques, it may be possible to solve the problem relatively successfully. Although our approach is far from ideal, it has the potential to be improved. We also believe that, with some modifications, the sequence-to-sequence approach could produce better results in the future.

REFERENCES

[1] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. 2019. Capture, Learning, and Synthesis of 3D Speaking Styles. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10101–10111.

[2] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. 2016. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on graphics (TOG)* 35, 4 (2016), 1–11.

[3] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. 2019. Learning individual styles of conversational gesture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3497–3506.

[4] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. 2017. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

[5] Vladislav Korzun, Dimov Ilya, and Zharkov Andrew. 2021. Audio and Text-Driven approach for Conversational Gestures Generation. *Computational Linguistics and Intellectual Technologies* 20 (2021), 425–432.

[6] 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*. 97–104.

[7] Taras Kucherenko, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexandersson, Iolanda Leite, and Hedvig Kjellström. 2020. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. 242–250.

[8] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. 2021. A large, crowdsourced evaluation of gesture generation systems on common data: The GENEA Challenge 2020. In *26th international conference on intelligent user interfaces*. 11–21.

[9] Gilwoo Lee, Zhiwei Deng, Shugao Ma, Takaaki Shiratori, Siddhartha S Srinivasa, and Yaser Sheikh. 2019. Talking with hands 16.2 m: 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*. 763–772.

[10] 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 IEEE International Conference on Robotics and Automation (ICRA '19)*. 4303–4309. https://doi.org/10.1109/ICRA.2019.8793720

[11] 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.