The FineMotion entry to the GENEA Challenge 2020

Vladislav Korzun
korzun@phystech.edu
Moscow Institute of Physics and Technology (National Research University)
Moscow, Russia

Ilya Dimov
iliyadimov@icloud.com
Lomonosov Moscow State University
Moscow, Russia

Andrey Zharkov
andrey.zharkov@phystech.edu
Moscow Institute of Physics and Technology (National Research University)
Moscow, Russia

ABSTRACT

This paper describes FineMotion's gesture generating system entry for the GENEA Challange 2020. We start by using simple baselines and expand them by using context and combining both audio and textual features. Among the participating systems, our entry attained the highest median score in the human-likeness evaluation and second highest median score in appropriateness.

KEYWORDS

embodied agents, neural networks, gesture generation, social robotics, deep learning, word embeddings

1 INTRODUCTION

Gestures are often underrated in human communication. They may contribute a lot to a speech going as far as to change what is being said to the opposite: a simple shrug can make audience question the credibility of the speech. Humans actively use co-speech gestures to convey their emotions or visualize their attitude [6, 10].

The task of generating conversational motions can be used for social robots [12], conversational agents, and even automatic animation of virtual characters. Both rule-based and deep learning approaches have been employed to varying degrees of success. In this work, we propose several models to solve this problem as well as analyze what makes movement seem appropriate and indistinguishable from humans and which features are essential for such a task.

The GENEA Challenge was conducted to explore what kind of models can produce human-like behavior for motion generation. The challenge organizers shared a 3.5-hour long dataset of audio, transcripts, and corresponding motions for body movement as well as several strong baselines and an evaluation phase, consisting of 250 experts.

Our systems were initially built upon baselines[7, 15] provided by organizers. We made several architectural adjustments, but conceptually the core of our systems was not dissimilar from previous work. Our main contributions are adding contextual information and combining both textual and audio information in one model.

Our paper is organized in the following way: section 2 describes data processing, which is shared between all experiments; section 3 describes our models; section 4 contains the discussion of our results; and section 5 is for conclusion. Our code is publicly available[1] to help other researchers reproduce our systems.

A complete task description can be accessed in [8]. Our team was labeled SD in the challenge evaluation results, which will be released later. The dataset used in all experiments is described in [3].

2 DATA PROCESSING

The challenge organizers provided 23 recordings with an overall length of 3 hours and 40 minutes for training. Each recording consisted of an audio file with speech recording, text transcripts, and BVH file with the motion data. The initial motion was captured by 60 frames per second; however the generated motions for evaluation were rendered at 20 frames per second. The motion skeleton contained 71 joints, but we used only 15 points corresponding to the upper body without hands and fingers.

We split the dataset for training and validation in the following way: the first recording *Recording_001* was used for validation (12 minutes), while the rest of the recordings were used for training (3 hours 28 minutes total). As the evaluation process is rather long, we used only 1 minute of *Recording_001* for human evaluation, and the remaining part of the sample was used to calculate mean average error on joints as a sanity check.

For all our models we used the same audio and motion data preparation pipeline provided in one of the baselines [7]. For audio representation we used 26 Mel-frequency cepstral coefficients [2]. We then averaged every five consequent Mel features to align audio features with motions (so that they have 20 FPS each). We represent motion data by 3 dimensional axis-angle rotation vectors for 15 joints. Thus each motion frame has 45 float features. This values are normalized over the mean value on train dataset. All aforementioned transformations of data result in input audio feature matrices to have size (N, 26) and output motion matrices to have size (N, 45), where N represents the number of frames in the sample.

Some of our experiments described further will mention a context window. The context window consists of 61 frames centered around a certain point in time, represented by a frame. We also use a "mean pose" calculated from the training dataset to use it as a starting value in recurrent models.

For paddings we used the MFCCs of silence recording. In text-based models we also used text features in form of GloVe [11] embedding for words in context window.

For all proposed models we smoothed generated motions by applying the Savitzky-Golay filter [13] to them. The length of the filter window and the order of the polynomial are 9 and 3, respectively. We did not use any external data.

[1] https://github.com/FineMotion/GENEA_2020

3 PROPOSED MODELS

3.1 Sequence to sequence model

Our first described model is a sequence to sequence model [14] inspired by [15]. The model in the aforementioned paper used words to generate corresponding motions. The competition dataset provides audio, motions and words. Three seconds of speech correspond to 60 poses and usually contain less than 10 words. We have decided to build our system on audio features and use textual information to further improve the quality of the models. Aside from difference in density between the two sets of features, speech obviously conveys more information like emotions, pauses, voice crackling, which are usually lost in text-to-speech systems.

As motions and audio features are mapped on a one-to-one basis, our first model is a simple seq2seq [14] consisting of GRU encoder and decoder over audio and motions. This baseline system is illustrated on Figure 1, with the exception of word encoder, which is described further. The encoder takes several audio features from frames $A_{i-k}...A_i$, encodes them into a higher dimensional space represented by $AE_{i-k}..AE_i$ and passes it to the decoder, which predicts the following motions labeled $M_{i+1}...M_m$. Decoder's final layer combines decoder hidden state and encoder-decoder dot-product attention [9] to make a motion prediction. As the decoder requires a pose as the first input, we supply it a previously predicted pose or the "mean pose" if no previous poses are available.

We tried to further improve this model by adding a word encoder, which is illustrated in the dotted box in Figure 1. Words are embedded using GloVe [11] and passed to another GRU. The hidden state of word-level encoder is not directly passed to the decoder, but a second encoder-decoder attention vector is calculated, which is supplied to the final layer of the decoder, to make prediction based both on audio, previous poses and words. The words are taken from a 2-second window.

We tuned several hyperparameters and training strategies. As the authors in [15] we employed continuity loss and variance loss to make the generated motions more fluid and natural. The addition of variance loss significantly improved co-speech gesture quality. We trained model with learning rate of 0.001 using Adam optimizer; audio encoder was a 2-layered bidirectional GRU with the hidden dimension of 150 units; word encoder was a single-layered GRU, both input and output dimensions were set to 100 units; decoder was a single-layered GRU with hidden dimension of 150 units. The model was trained for 100 epochs with a batch size of 512, where each sample contained 10 previous poses and 20 poses for prediction.

We also explored various combinations of windows sizes for encoder and decoder. We did not find larger windows to be beneficial to the quality of our predictions and we kept the same window sizes as in the original paper: we use 10 previous frames to predict the following 20 frames.

Another strategy we tried to employ is teacher forcing [1]: during training we mixed poses supplied in the decoder. We used true motions as well as poses generated by the decoder itself. The main idea behind it is to help the model to explore the error space and become more robust. In the end we found out that not supplying real poses at all was the best option and the rest of our models are using their own predictions during training, just as it would happen during inference. This may be attributed to variance loss: the model was rewarded for making different poses, which likely resulted in a pretty constant deviation from true poses.

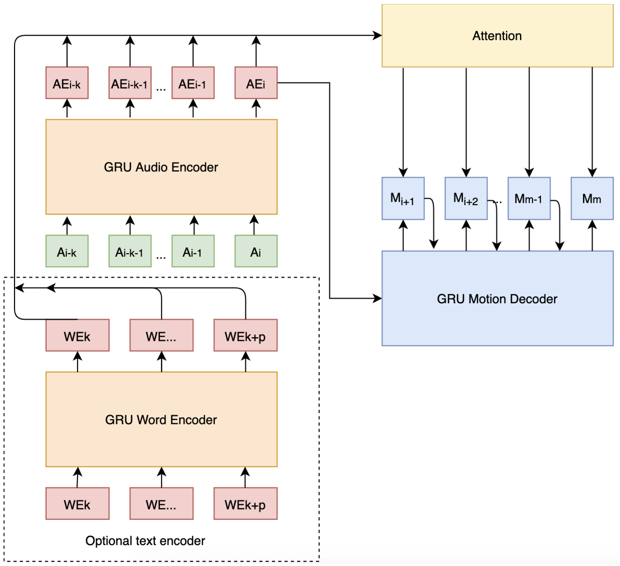

Figure 1: Scheme of baseline seq2seq model on audio features with optional word-level encoder.

We also do not save hidden state of encoder and decoder between batches during training. Each training sample is processed individually without knowledge of previous time period, but during inference the model always supplies it's state for the next segment. This may be the reason behind choppiness in predicted movement, but we never explored it, as smoothing during postprocessing helped us to eliminate this shortcoming.

We'd like to state that our evaluation of hyperparameters is rather subjective: all the changes were judged by a small group of 2-3 people on a one-minute sample from the validation recording. It is quite possible that we misjudged some of our experiments because of an unsuitable time sector or a simple human error.

3.2 Contextual encoder

The second model is inspired by [7]. We have decided to keep sequence to sequence model and enhance it with contextual representations. In our basic sequence to sequence encoder each input corresponds to a single frame.

We decided to represent each frame as a 3-second window around it, which resulted in 61 frames. We used two additional GRU encoders to encode the audio and textual context window as displayed on Figure 2. The audio encoder consists of 3 linear layers with batch normalization to project audio features and one-layer one-directional GRU. All audio encoder layers have hidden size 150. The textual encoder is bidirectional one-layer GRU over GloVe embeddings and hidden size of encoder is similar to the embeddings size which is 100. The outputs of both context encoders are concatenated and projected to be passed as inputs to the seq2seq encoder with hidden dimension of 150 units. The rest of the model

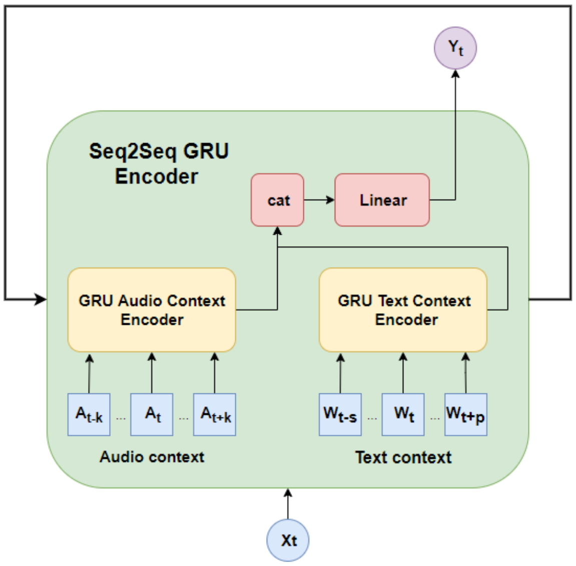

Figure 2: Scheme of contextual encoder.

is a simple sequence to sequence architecture with attention, which was described earlier.

We train this model with Adam optimizer with the learning rate of 0.001 and the batch size of 50. The final model was trained for the 100 epochs, however the target loss stabilized after 80th epoch. Furthermore, motions generated after 80th and 100th epochs were virtually identical.

3.3 Adversarial training

Even a single speaker has a significant variation of his movements even in extremely similar situations, same phrases and contexts. However, so far we described only models which tried to recreate the same movements as the ground truth, even if it was not the only correct behaviour, but one of the many possible motions. To try to overcome this problem we used adversarial training (as done, i.e. in [4, 5]).

The generator model produces motions from audio, while discriminator model tries to classify real and generated motions. The Generator loss is

$$L_G = L_{base}(G) + \lambda L_{adv}(G, D), \qquad (1)$$

where L_{base} contains whatever non-adversarial components of generator loss and L_{adv} represents adversarial loss with weight λ. In all of our experiments we used non-saturating GAN loss.

We tried several discriminator models based on blocks of (1D convolution, 1D batch normalization, LeakyRelu(0.2)). After series of that blocks we flatten the outputs of convolutional block and apply two more linear layers. We varied total number of blocks from 2 to 6 with at least two of them reducing spatial dimension (stride > 1).

Unfortunately, the training with adversarial loss was not stable (especially for relatively high λ values around 10.0). Sometimes we got interesting and diverse results (mostly for small λ values around 0.1), however the quality was still lacking in comparison with our best model so in the final system adversarial training was not used.

4 RESULTS AND DISCUSSION

The challenge organizers used two human-evaluation metrics for evaluation:

- **Human-likeness** - the generated motion should be realistic for human. The evaluation participants should score the motion file without audio by this criterion.
- **Appropriateness** - the generated motion should match the corresponding audio. So participants score motion with audio.

The challenge organizers also provided the results for baselines, which we built our systems upon in our submission. This allows us to find out the importance of our modifications.

The baseline systems from [7] and [15] had 46 and 55 median score in human likeness and 40 and 38 in appropriateness. Our final submission scored 60 and 49. By human-likeness it has the highest median score among the participating systems and baselines. It also has the second highest score by appropriateness. Although our system showed strong improvement over baselines, it is still far behind human generated motions (72 human-likeness and 81 appropriateness). A special sample of mismatched real motion and real audio was also present at the evaluation. It was not surpassed by any of the teams. That means that our synthetic generated motions are significantly less appropriate than random human movement.

To select the best model we compared them on validation data using human evaluation among the three members of our team. The seq2seq model with contextual encoder was unanimously chosen as the best model, however seq2seq with attention over text and audio was a close second.

We found out that our team was looking for specific sorts of movements during the motion evaluation: we generally were looking for correspondence between motions and verbal pauses. We were more inclined to vivid movements, even if they were choppy, and last but not least - we were always looking for fast and sharp movements coinciding with loud and aggressive speech patterns.

Our humble human evaluation has come to a conclusion, that the approach with context encoder helps to make generated motions smoother, because is uses more information, especially for the last frames in a sequence, while basic seq2seq heavily relies on smoothing.

5 CONCLUSION

In our approach we combined text and audio features and thus were able to outperform text- and audio-only baselines. However, the lack of a clear movement quality metric did not allow us to thoughtfully and adequately explore our design choices, thus our architecture is certainly only suboptimal. Computable metrics serve only as a sanity check: approaches using GANs or variance loss could produce a fitting motion, which would be quite different from the original.

Compared with real data (human gestures) there is a striking gap in our system's performance and real motions, meaning that there is still a lot to be improved.

We believe that in the future it is worth exploring text and audio feature fusion more thoughtfully. We also think that generative models have a lot of potential. Our team also did not explore various sound preprocessing techniques, which could result in a more high-dimensional vector input representation, which would allow models to extract a more rich set of features.

REFERENCES

[1] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. 1171–1179.

[2] Steven Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing* 28, 4 (1980), 357–366.

[3] Ylva Ferstl and Rachel McDonnell. 2018. Investigating the use of recurrent motion modelling for speech gesture generation. In *IVA '18 Proceedings of the 18th International Conference on Intelligent Virtual Agents*. https://trinityspeechgesture.scss.tcd.ie

[4] 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.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[6] Mark L Knapp, Judith A Hall, and Terrence G Horgan. 2013. *Nonverbal communication in human interaction*. Cengage Learning.

[7] 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 ACM International Conference on Intelligent Virtual Agents (IVA '19)*. 97–104. https://doi.org/10.1145/3308532.3329472

[8] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. 2020. The GENEA Challenge 2020: Benchmarking gesture-generation systems on common data. In *Proceedings of the International Workshop on Generation and Evaluation of Non-Verbal Behaviour for Embodied Agents (GENEA '20)*. https://genea-workshop.github.io/2020/

[9] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[10] David Matsumoto, Mark G Frank, and Hyi Sung Hwang. 2012. *Nonverbal communication: Science and applications*. Sage Publications.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[12] 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.

[13] Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[15] 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