A Framework for Integrating Gesture Generation Models into Interactive Conversational Agents

Demonstration Track

Rajmund Nagy∗
KTH, Stockholm, Sweden

Taras Kucherenko∗
KTH, Stockholm, Sweden

Birger Moell
KTH, Stockholm, Sweden

André Pereira
KTH, Stockholm, Sweden

Hedvig Kjellström
KTH, Stockholm, Sweden

Ulysses Bernardet
Aston University, Birmingham, UK

Figure 1: Architecture of the framework for integrating gesture generation models.

ABSTRACT
Embodied conversational agents (ECAs) benefit from non-verbal behavior for natural and efficient interaction with users. Gesticulation – hand and arm movements accompanying speech – is an essential part of non-verbal behavior. Gesture generation models have been developed for several decades: starting with rule-based and ending with mainly data-driven methods. To date, recent end-to-end gesture generation methods have not been evaluated in a real-time interaction with users. We present a proof-of-concept framework, which is intended to facilitate evaluation of modern gesture generation models in interaction.

We demonstrate an extensible open-source framework that contains three components: 1) a 3D interactive agent; 2) a chatbot back-end; 3) a gesticulating system. Each component can be replaced, making the proposed framework applicable for investigating the effect of different gesturing models in real-time interactions with different communication modalities, chatbot backends, or different agent appearances. The code and video are available at the project page https://nagyrajmund.github.io/project/gesturebot.

KEYWORDS
conversational embodied agents; non-verbal behavior synthesis

∗Both authors contributed equally to the paper.

ACM Reference Format:

1 INTRODUCTION
Humans use non-verbal behavior to signal their intent, emotions and attitudes [9, 16]. Similarly, embodied conversational agents (ECAs) can be more engaging when having appropriate nonverbal behavior [21]. It is therefore desirable to enable conversational agents to communicate nonverbally.

Currently existing implementations of ECAs rely primarily on pre-recorded animations or require handcrafted specification of the motion [11, 15, 17], e.g. in XML-based formats [10]. However, recent developments in the field of gesture generation make it possible to produce realistic gestures in a purely data-driven fashion [1, 12, 23]. To date, many of these recent gesture generation methods have not been evaluated in a real-time interaction with users. A potential reason for the lack of evaluation in interaction for the recent models is the difficulty of setting up an interactive conversational agent.

In this work, we outline a framework for embedding data-driven gesture generation models into conversational agents. We envision that this framework with accelerate development of interactive embodied agents with end-to-end gesticulation models.

Our framework is modular, which enables it to be used for a wide range of scientific investigations about intelligent virtual agents, such as experimenting with their voices, gestures, breathing, conversational complexity or gender. For our demonstration, we integrate the speech- and text-driven model developed by Kucherenko et
We provide the virtual environment and the user interface as a platform for researchers to develop and test new modules. Furthermore, the communication between the components consists of sending audio, text or motion file in a message; in our implementation, this is facilitated by the open-source Apache ActiveMQ message broker and the STOMP protocol. Each component is replaceable and is described in the corresponding section below.

2.1 Virtual agent in Unity

We provide the virtual environment and the user interface as a Unity scene. The end user interacts with the conversational agent through voice input or a text field. By using Unity, we ensure that the system can be easily extended with new modules by other researchers in the future. Furthermore, it makes it possible to tailor the environment and the character model according to the requirements of the application (e.g., explore virtual/mixed reality applications).

2.2 Chatbot backend

The user’s input message is sent to the chatbot backend that produces the agent’s response as text and audio. The chatbot backend is comprised of a speech recognition system, a neural conversational model and a text-to-speech synthesizer. For our demonstration, we present two implementations of this component.

In the first configuration, Google’s popular DialogFlow platform is used with its automatic speech recognition module to enable voice-based interaction with the agent. The interfacing to DialogFlow is implemented in a C# module inside Unity.

In the second configuration, we adapt the open-domain chatbot BlenderBot [18] and a text-to-speech model called Glow-TTS [8] to build a virtual agent with free-form conversation capabilities. We leverage open-source implementations (provided by HuggingFace [22] and Mozilla TTS) to seamlessly integrate the two models into the Python backend.

2.3 Gesture generation model

Based on the output of the chatbot backend, the gesture generation model synthesizes the corresponding motion sequence. We adapt a recent gesture generation model called Gesticulator [13], which is an autoregressive neural network that takes acoustic features combined with semantic information as its input, and generates the corresponding gesticulation as a sequence of upper-body joint angles, which is a widely used representation in computer animation and robotics.

In the original paper [13], the network was trained on the Trinity Speech-Gesture dataset [6], consisting of 244 minutes of speech and motion capture recordings of spontaneous monologues acted out by a male actor. The input features – log-power spectrograms for audio and BERT [4] word embeddings for text – are extracted and concatenated at the frame level, and a 1.5 s (30 frames at 20 FPS) sliding window of input features is used for predicting every motion frame, motivated by gesture-speech alignment research.

We tailor the base Gesticulator model to our interactive agent with the following adjustments:

1. We replace the audio features from spectrograms to the extended Geneva Minimalistic Acoustic Parameter Set [5], normalized to zero mean and unit variance. We qualitatively found that it results in better motion for synthesized voice.
2. The text transcriptions that are used for training the model contain precise word timing information, which is usually not available in real-time settings. Therefore, when interacting with a user, we approximate it with speech utterance lengths that are proportional to the syllable count.
3. Finally, we replace the BERT word embedding with FastText [3] (which has significantly lower dimensionality) in order to reduce the feature extraction time.

3 LIMITATIONS

At the current stage of development, each of the components has some important limitations:

1. Both available chatbot backends introduce several seconds of processing time before the agent responds to the user, which might currently affect immersion in the interaction.
2. The synthesized voice of the agent yields out-of-distribution audio samples which significantly degrade the quality of the generated motion. This could be improved by replacing audios in the dataset with synthetic audios [20] or by training a TTS model on the audio from a speech-gesture dataset [2].
3. Finger motion is not modelled by the gesture generation model due to poor data quality in the dataset.

However, the system’s modular design of replaceable components allows addressing these limitations in the future. For instance, it is straightforward to replace Gesticulator with a model that generates full-body motion or to change the chatbot backend as shown in our two distinct examples with DialogFlow and Blenderbot.

4 CONCLUSIONS AND FUTURE WORK

We have presented a framework for integrating state-of-the-art data-driven gesture generation models with embodied conversational agents. As highlighted by the GENEA gesture generation challenge [14], the gesture generation field needs a reliable benchmark. The proposed framework provides such a possibility; it can be used to compare gesture generation models in real-time interactions.

There are many directions in which this work can be extended in the future. More diverse gesticulation can be achieved by choosing a probabilistic gesture generation model instead of a deterministic one like Gesticulator. Incorporating stylistic control [1, 7] in ECA to allow expression of different emotions is a promising direction. Moreover, the proposed framework can be used in user studies to investigate the human perception of different gesticulation.

---

2. https://stomp.github.io/
REFERENCES


