Automatic Text-to-Gesture Rule Generation for Embodied Conversational Agents

Abstract

Interactions with embodied conversational agents (ECAs) can be enhanced using human-like co-speech gestures. Traditionally rule-based co-speech gesture mapping has been utilized for this purpose. However, the creation of this mapping is laborious and often requires human experts. Moreover, human-created mapping tends to be limited, therefore prone to generate repeated gestures. In this paper, we present an approach to automate the generation of rule-based co-speech gesture mapping from publicly available large video dataset without the intervention of human experts. At runtime, word embedding is utilized for rule searching to get the semantic-aware, meaningful, and accurate rule. The evaluation indicated that our method achieved comparable performance with the manual map generated by human experts, with a more variety of gestures activated. Moreover, synergy effects were observed in users' perception of generated co-speech gestures when combined with the manual map.

Keywords: Gesture generation, Rule-based mapping, Virtual agents, Social agents, computer animation

Introduction

In verbal communication, people often use gestures to convey the oral message clearly or to express their emotional intent [1]. These gestures accompanying speech are called *co-speech gestures*. While head motions and facial expressions can be used, a large portion of the co-speech gestures belongs to hand motions [2].

With the rise of VR/AR, embodied agents have been playing a vital role in diverse virtual scenarios, e.g., entertainment, training, and education [3, 4, 5, 6, 7, 8, 9, 10]. These embodied agents need believable gesture animations to convey the state of conversation [11, 12]. The generation of believable and meaningful gestures is a challenging task. Besides, for conversational agents, the generation of gestures needs to be prompt in response to the users' query.

There are two major techniques to generate co-speech gestures: rule-based and data-driven. The input to these systems can be text, speech, or both. Rule-based techniques [13, 14, 15] are kind of expert systems where human experts create the mapping between the keywords or phrases and gestures, i.e., explicit rules exist. When generating co-speech gestures, these systems search for matched rules based on the input. Since human experts manually generate the rules, the number of rules is limited. Therefore, these techniques often result in repetitive and unnatural co-speech gestures [16]. On the other hand, data-driven techniques exploit a large amount of data to train models. The models are either probabilistic models [17] or deep-learning models [18, 19, 20, 21] (instead of explicit rules) and trained using annotated data [22], motion-captured data [18, 23], or publicly available videos [19, 20]. Although human labor is often required to annotate data, model training itself is done by algorithms. Thus, it can be seen automated compared with the rule-based techniques where human experts explicitly create the rules. The performance of data-driven systems heavily dependent on the input of the systems. Audio-driven systems [23, 18] use acoustic features, tend to perform better then text-driven systems [17, 19]. However, text-driven systems can be more useful if conversational knowledge-base is expandable. For exam-

ple, chatbots often use knowledge-base and the data in the knowledge-base are generally stored in a tree-like structure. This tree-like structure gives the flexibility to add new information easily. In fact, chatbots can serve as the backend system for embodied conversation agents (ECAs) and in such a case Text-to-Speech (TTS) systems can be used to generate speech from text [24].

The literature review indicates that the rule-based system suffers from repetitive gestures and limited mapping [16]. The text input based data-driven approaches suffer from either the small amount of data or laborious manual annotations [17] and also deep-learning related problems, such as over-fitting, vanishing or exploding gradients. In the framework of Ali et al. [24], an embodied conversation agent uses a chatbot backend system. The co-speech gesture system in that framework is rule-based similar to [14]. This paper extends the work of Ali et al. [24], with the focus on the approach to automatically generate co-speech gesture rules from a large scale data for ECAs.

Our approach improved the previous rule-based methods in two ways. First, we automated the rule generation procedure. We exploited a large amount of pose data and timestamped texts extracted from publicly available videos. Using a sliding window algorithm on the pose data, we calculated the similarity between pose sequence (i.e., the gesture performed) and gesture animations from the pre-defined 3D gesture animation set. When high similarity was found, we extracted the phrase from the location of the sliding window and paired the phrase with matched gesture animation, forming a rule entry (see Figure 1). By doing so, we automatically generated rules without the interventions of human experts. The resultant of the process is a large rule-base mapping of text and gestures. Second, we improved the rule searching algorithm to find a semantically sounding entry from the rule-base. In generating co-speech gesture sequence for a given text, the text is tokenized into fixed-length phrases and is searched in the rule-base. Instead of simple text matching, we employed GloVe word embedding technique [25] to convert the phrase into a semantic-aware vector representation and the similarity between the phrase from input text and the phrase from the rule-base was

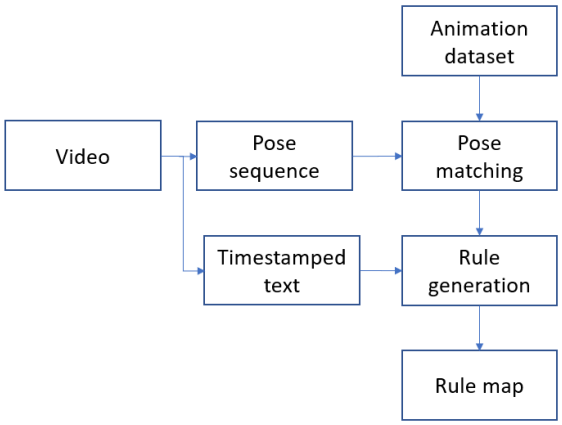

Figure 1: Overview of our approach pipeline for rule-map generation.

checked in this vector space. Thus, phrases with similar meanings but different word choices can induce a high similarity score.

Therefore, our contribution can be summarized as follows:

- An automated rule generation method for ECAs, which takes advantage of a large dataset publicly available.
- An improved rule searching by employing semantic-aware word embedding for text-based co-speech gesture generation.

In sections following the introduction, we present our approach in detail, objective and subjective evaluations of our methods, followed by an in-depth discussion of the evaluation results. Finally, we provide limitations and future work direction, and we conclude the paper.

Our Automatic Rule Generation Approach

We propose a co-speech rule generation approach that automatically extracts rules from a large video dataset without the intervention of human experts. The outcome of our approach is a usable mapping table between text and co-speech gestures. We exploit the fact that in the rule-based system the experts use a set of pre-defined animation clips and manually map them to either word or phrases. The experts usually group those animation clips into different categories, i.e., beat, deictic, iconic and metaphoric.

The mapping is usually based on their experience. Even in some data-driven methods, the data is manually annotated to reflect those categories. With the evolution of pose estimation methods from videos, it becomes very easy to generate pose sequences from RBG videos. These pose estimation methods can generate a 2D or 3D pose sequence. 2D pose sequence estimation is much better than the 3D pose sequence for now. The popular open-source 2D pose estimation system is OpenPose [26]. Although OpenPose is a fast system, it can produce artifacts or inaccurate results in some situations. So the results cannot be directly used. These sequences are however good for pose matching and classification. We used a set of well-defined 3D gesture animations and created the corresponding set of 2D projections. Then we performed 2D sequence to sequence matching. This way we were able to create a comprehensive mapping without human experts.

Dataset

The dataset for our approach consists of two parts. One dataset is of videos from public sources like streaming websites. From this dataset, we extract the pose sequence and speech-to-text for each clip. Yoon et al. [19] provides a fully automated pipeline to extract the required data from a large set of videos. They also provided the dataset as "TedTalk dataset." It contains 106 hours of data. *TedTalk dataset* contains clips. Each clip has pose frames and corresponding text which was uttered by the speaker as shown in Figure 2. The text is aligned with pose sequence by converting the time to frame number. Each word has a starting frame number and ending frame number. Since the text is obtained from the subtitles and further converted to frame interval, some misalignment can occur. The other dataset is the 2D projection of predefined gesture animations. We used the animations provided by ICT Virtual Human Toolkit [14]. This dataset will be referred to as *Gesture dataset*. *Gesture dataset* contains variable-length animations. All animations were processed to be of equal size of max length animation by padding zero frames on both sides of each animation. In the next step, only upper body joints were selected from both datasets. After that, both datasets with selected joints were normalized with neck joint as a reference point.

Automatic rule generation

The core idea in our approach is to use the *Gesture dataset* as strided sliding window on the *TedTalk dataset* to obtain {gesture, text} pair. The algorithm takes a clip from *TedTalk dataset*. *Gesture dataset* is slid over the clip. On each step, we compute the frame-by-frame cosine similarity and then average it for each gesture. After that, we select the closest gesture. We selected the threshold 0.92. The selection was random in case of more than one gesture above a threshold. In the discussion section, we discussed the reason for the threshold. Once the gesture is selected then we extract the text from the corresponding location. The maximum length of the corresponding text is five words. In the case of text larger then five it is trimmed from both sides. After that, the pair is saved in the mapping table. The stride between each step is the length of the gesture sequence. The algorithm is shown in Algorithm 1.

Algorithm 1: Automatic rule generation

```
Result: {gesture,text}
TeldTalk-dataset
Gesture-dataset
stride=Length(Gesture-dataset[0])
pairList=[]
while !EOF TedTalk-dataset do
    clip=nextClip()
    while !EOF clip do
        nGesture=[]
        subclip=clip[i:i+stride]
        while !EOF Gesture-dataset do
            gesture=nextGesture()
            dist=averageCosSimilarity(subclip,gesture)

            if dist ≥ 0.92 then
                nGesture.append(gesture)
            end
        end
        i=i+stride
        if size(nGesture) ≥ 1 then
            gesture=randomSelect(nGesture)
            text=getText(subClip)
            pairList.append(gesture,text)
        end
    end
end
```

Figure 2: Dataset sample with pose and corresponding text

Algorithm 2: Animation sequence generation

```
Result: {animation-sequence}
rule-map //pair-list
text //to be animated
sequence=[]
text-tokens=tokenize(text,5) //5 word token
while !EOF text-tokens do
    token=nextToken()
    vecToken=ConvertToVec(token)
    nSim=[]
    while !EOF rule-map do
        rule=nextRule()
        vecRule=ConvertToVec(rule)
        sim=CosSimilarity(token,rule)
        nSim.append(sim,rule)
    end
    mSim=max(nSim.sim)
    sequence.append(mSim.rule)
end
```

Animation sequence generation

The obtained rule-map is now ready to be used. At runtime, the text which needs to be animated is passed to the rule-map driver algorithm. The algorithm takes the text, tokenizes it into five words chunks. Then each chunk is searched in the rule-map and the corresponding animation is fetched for the best match. Then those animations are played sequentially by the animator. In our case, that was done with Unity3D animator. Since the search for the rule is basically a string search, this can be further improved. In literature, almost all the methods use some sort of direct string matching. Although string matching makes sense it has a flaw. It does not care about the context of the string. In natural language processing (NLP) research, researchers developed the word embedding or word2vec. We used GloVe [25] pre-trained word2vec with 300 dimensions. To use GloVe, at the matching step we obtained the vector representation of both strings i.e five-word chunk from the input text and text from the rule and then measure the cosine similarity. Vector representation of a given string is obtained by summation of all the word vectors of each word of the given string. The resulting vector representation of a string is a vector of 300 float values. The algorithm is shown as Algorithm 2. The comparison of word embedding and string match is discussed in the discussion.

The other part of the Animation sequence generation is preserving time consistency. The gesture animations are of varying lengths between 1 to 3 seconds. We assigned a fixed time slot for each animation. From the empirical analysis, we concluded that the 5-word chunk generates appropriate time slots. We divide the input text in 5-word chunks. Any chunk less then 5 is dropped. If input text is less then 5 words then it is processed as a single slot. For example, a 15-word sentence has 3 animation slots. Time for each slot is calculated at runtime by dividing the length of the audio from the TTS system by the number of animation slots. This way no matter which TTS system is used the gestures will have higher time consistency.

Evaluation and Results

Objective Evaluation

For the evaluation of our method, we used three rule-based mapping tables. The maps are:

- **Manual:** ICT Virtual Human Toolkit's default rule-base available in the toolkit. Each rule in the *Manual map* consists of multiple words and multiple animations.

- **Auto:** Map generated through Automatic rule generation. Each rule in auto map is consists of a sequence of words and a single animation. Word embedding is used to retrieve the rule.

- **Hybrid:** Map generated by the combination of *Manual map* and *Auto map*, where *Manual map* had a high priority.

The hypothesis of the evaluation is defined as *"for fairly large and general text input, gesture generation system should be able to retrieve the maximum number of gestures from a given map."* For the input text, we feed the complete novel "Farm animals" by George Orwell to our system. As shown in Figure 4, the output was the frequency of each animation over the whole text by each map. Statistics show that *Manual map* uses only 18 animations out of 57. Whereas *Auto map* and *Hybrid map* used all of the animations. This measure shows that our method gives a more generalized and less repetitive animation sequence. One key observation is that an application-specific mapping can generate a more reasonable animation sequence. Our method is for general conversation scenarios. In those scenarios, gestures are for the complimentary purpose to increase the effectiveness of the conversation. Figure 3 shows a sample of animation sequences by three maps. From the *Manual map*, the system generated a sequence with one animation. From *Auto map* and *Hybrid map* it generated sequence with 2 animations.

Subjective Evaluation

In this section, we describe the experiment that we conducted to see how the generated gestures of ECA are perceived by human users. Similar to the objective evaluation presented in the previous section, we compared gestures generated by the three methods: *Manual*, *Auto*, and *Hybrid*. However, here we measured users' subjective ratings on three categories: *naturalness*, *time consistency*, and *semantic consistency*. For that, we used the questionnaire used in [18]. Each category consists of three questions and we did not modify the questions.

For this study, we prepared five texts with various lengths and converted them into audio using Google Text-to-Speech. To generate co-speech gestures, we implemented a virtual environment with a female human character standing at the center using the Unity3D game engine (see Figure 3). The lip animations were achieved using the SALSA LipSync [1]. The co-speech gestures were rendered using the three generation methods per each text and recorded, thus a total of 15 videos were generated.

We made an online survey form using the pregenerated videos. A within-subjects design was used in making the survey form, in which participants watched each video and answered to the questions asking their subjective impression on the co-speech gesture of the ECA. Videos of the same text were grouped and the order was randomized. We asked the participant to read the text before watching the video. Participants were asked to rate each question using a 5-Likert scale (1: totally disagree, 5: totally agree). We additionally asked participants to rank the three methods based on their preference.

Participants were recruited from Amazon Mechanical Turk. A total of 29 participants completed the survey, however, data from 9 participants were removed as they did not pass our screening criteria. To identify insincerely completed surveys, we embedded an attention question among the actual questions in the middle of the survey. We also measured the completion time. Participants who failed to pass the attention question or spent less than threshold time—members of our group completed the survey to set the threshold time—were rejected. Finally, there were 20 valid participants (4 females, 16 males, age 22–67, average 36.6). Fourteen were native speakers, four were fluent or proficient users, and two marked their English proficiency as conversational.

To perform statistical analysis, we aggregated the ratings per each category for each generation method and calculated mean ratings per participant. We treated preference rank as ratings—we assigned 1 to the most preferred and 3 to the least preferred method. Due to the ordinal aspect of the original ratings, we performed nonparametric Friedman tests with the 5% significance level on each category. The results are summarized in Figure 5.

There were significant main effects of the gesture generation methods on *time consistency* ($\chi^2 = 8.72$, $p < .05$), *semantic consistency* ($\chi^2 = 10.31$, $p < .01$), *preference* ($\chi^2 = 7.80$, $p < .05$), but not on *naturalness* ($\chi^2 = 2.36$,

[1] https://crazyminnowstudio.com

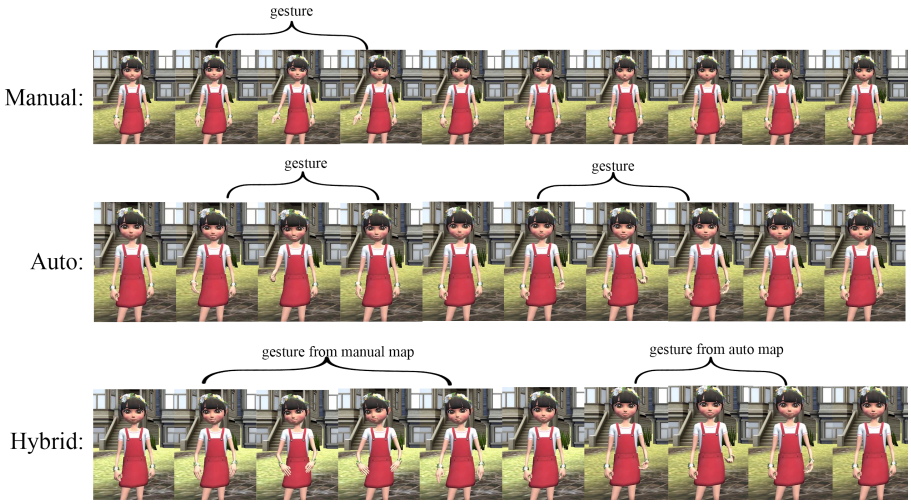

Figure 3: Gestures for a sample text "The lion is not a trained pet. It can hurt you"

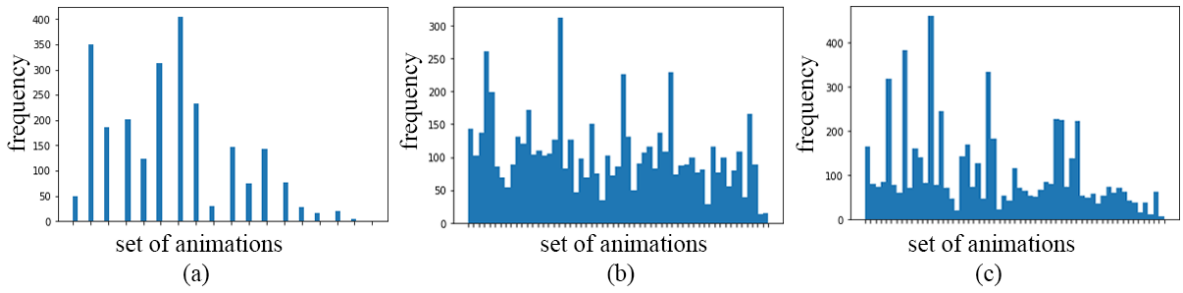

Figure 4: Evaluation of gesture distribution. x-axis is set of animations and y-axis is frequency of each animation. Left to Right: a) Manual b) Auto c) Hybrid

$p = .31$). Pairwise comparisons using Dunn's post-hoc tests with Bonferroni correction revealed that participants gave higher ratings on *time consistency* for Hybrid compared to Auto ($p < .05$). For *semantic consistency*, ratings for Hybrid were higher than those of Manual ($p < .05$) and those of Auto ($p < .05$). Finally, participants favored Hybrid compared to Manual ($p < .05$), but the tendency was insignificant when compared with those of Auto ($p < .08$).

Discussion

Ensuring Gesture Diversity

One of the main issues with rule-based systems is writing down rules for all possibles gestures, which we humans naturally do [18]. To address this issue, we exploited the data-driven approach where models are tuned from a large dataset. However, instead of training models, we extracted rules from the dataset. The rationale behind using a large dataset is that if the dataset covers diverse gestures, so do the trained models. There are two latest types of research available in the domain of deep learning which are doing work similar to our research. Kucherenko et al.[18] used a large set of studio captured data. However, their method mostly generated beat gestures. Similarly, Yoon et al.[19] used *TedTalk dataset*. They used vector representation of text input as the input to the network. Their results also indicated the large portion of beat gestures with a limited number of other gestures. Seeing data-driven approaches fail to generate diverse gestures, we decided to exploit pre-defined gesture animations (for clarity sake, we call this *Gesture dataset*). We formulated our approach to generate (learn) rules from a large dataset of real human gestures. Instead of learning gestures from the dataset directly (as other data-driven approaches do), we made our approach to use the diverse gesture animations in *Gesture dataset*. We also designed our automatic

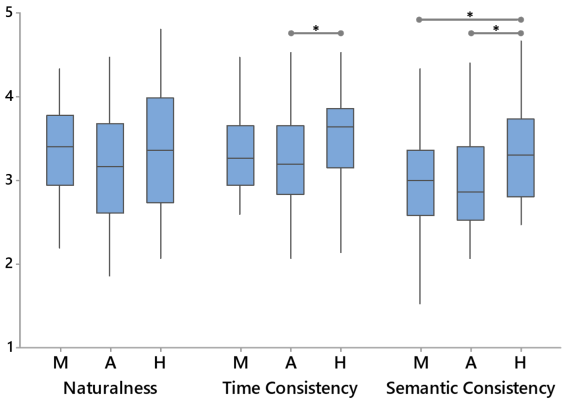

Figure 5: Results of subjective measures. Whiskers in the box plots are extended to represent the data points less than 1.5 interquartile range (M: *Manual*, A: *Auto*, H: *Hybrid*, *: $p < .05$).

rule generation not to be biased to a subset of gesture animations in *Gesture dataset*. In the automatic rule generation procedure, we tuned our similarity check threshold to be 0.92. This threshold was obtained empirically. We use the intuition that if the distribution in the resulting rule map is spread across all gestures without bias toward one or two gestures then it's a good map. Along with the threshold, we also tested different types of selection criteria. We set three different criteria to select a gesture animation that matches the human gesture performed in *TedTalk dataset*. The first criterion was random selection above the cut-off threshold. The second criterion was the maximum similarity above the cut-off threshold. The third criterion was the maximum similarity without any cut-off threshold. The cut-off threshold was determined empirically by analyzing a small sample from *TedTalk dataset* with different similarities. We found that co-speech gestures generated from the first criterion with 0.92 as a threshold for similarity check were closer to the ground truth. Sample from the *TedTalk dataset* served as ground truth. The distributions of gesture animations used are shown in Figure 6. The distributions of the second and third criterion look similar (the threshold for those were set to 0.92). Because both use maximum similarity except that the second one has a cut-off threshold. The third criterion generated more rules but the quality was low. The second criterion provided the quality but the distribution was biased. First criterion trade-off and gave a better distribution. This approach, however, has a strong dependency on the diversity of the gesture animations in *Gesture dataset*. If *Gesture dataset* is limited then the resulting co-speech gestures used will be limited. We used 57 gesture animations. If more gestures can be added then the results will improve.

Improving Rule Matching

Semantic mismatches between spoken words and co-gestures are a well known issue of data-driven approaches. One the other hand, human experts created rule-based approaches can generate semantically sounding co-speech gestures, but lack the frequency of gesturing or repeat the same gestures. To preserve the semantic consistency and increase the frequency of gesturing, we focused on the rule matching algorithm in the animation sequence generation procedure as described in Section Animation sequence generation. In order to make semantic-aware matching we used GloVe word embedding matrix. We compared the word embedding method with the Levenshtein distance for the same task. As shown in table 1, we took a short passage from the Geroge Orwell's novel "Animal Farm." The passage is *"Even when you have conquered him do not adopt his vices. No animal must ever live in a house or sleep in a bed or wear clothes or drink alcohol or smoke tobacco or touch money or engage in trade. All the habits of Man are evil"*. The result of the comparison indicates that GloVe preserved the semantics and context in a much better way in 1,6,7,9 then the simple string matching. GloVe performed worst on 4 and others are insignificant.

Synergistic Effects of Hybrid Rules

In Section , we compared co-speech gestures generated by human-made rules (*Manual*) with those produced by our automatically generated rules (*Auto*). In the comparisons, we also included the mixture of those two rules (*Hybrid*), exploiting the benefit of the text-driven approach, i.e., the ease of expanding rules. In the objective evaluations, the *Manual* and *Hy-*

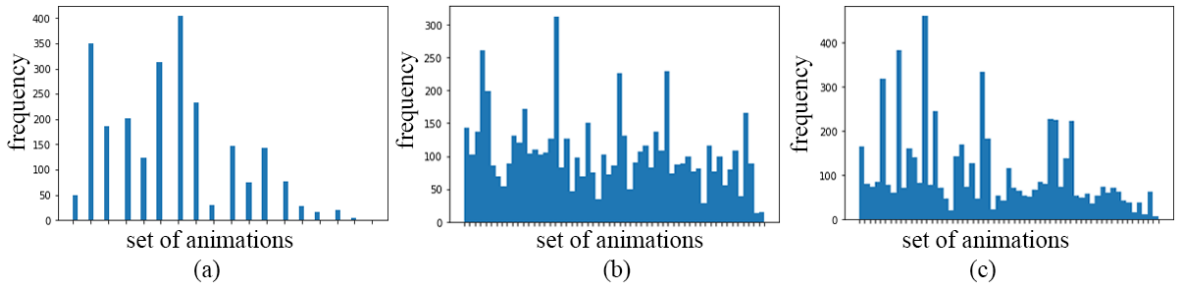

Figure 6: Gesture distribution with different criteria. x-axis is set of animations and y-axis is frequency of each animation. Left to Right: a) Random selection above cut-off threshold. b) Maximum similarity above cut-off threshold. c) Maximum similarity.

1	Input: Even when you have conquered GloVe: you have invaded me in LD: see when you lose your
2	Input: him do not adopt his GloVe: that did not keep him LD: him to do so the
3	Input: vices No animal must ever GloVe: every animal they see including LD: species of animals out there
4	Input: live in a house or GloVe: them and live in a LD: live in a world of
5	Input: sleep in a bed or GloVe: a book in bed my LD: been in a cave for
6	Input: wear clothes or drink alcohol GloVe: or to wear anything or LD: we create our own echo
7	Input: or smoke tobacco or touch GloVe: or smoking or that people LD: go home today or to
8	Input: money or engage in trade GloVe: have to trade part of LD: us to engage in a
9	Input: All the habits of Man GloVe: habits of mind and these LD: or the arms of an

Table 1: GloVe and Levenshtein distance(LD) comparison

brid generated more diverse gestures compared to the *Manual*, with higher frequency of gesturing. However, compared to *Manual*, *Hybrid* seemed to generate more biased use of gestures. Interestingly, this biased but diverse use of gestures produced synergistic effects in the subjective evaluation. The co-speech gestures produced by *Hybrid* were perceived as more timely and semantically correct. That might be because *Hybrid* rules use the best features of both rules. The *Manual* provides a limited but strong association of gestures with text, whereas *Auto* offers a relatively weak but extensive association of gestures. *Hybrid* takes strongly associated (with text) gestures from *Manual* due to high priority. As shown in Figure 3, *Manual* generated a single gesture whereas *Auto* and *Hybrid* generated two gestures each. The gesture of *Manual* is a *negative* animation, which is strongly associated with the text. In *Auto*, the first animation is a *emphasize* animation. The second animation is a *you* animation in *Auto*. *Hybrid* uses the *negative* animation from the *Manual* and a *you* animation from the *Auto* to generate a better animation sequence then both of the other methods.

Conclusion and Future Work

We presented an approach to automate the generation of rule-based co-speech gesture mapping from publicly available video dataset by utilizing pre-defined gesture animations. Our approach synthesis rules without the intervention of human experts. By using pre-trained word embedding, our method preserves the context of input. Our method does not depend on any specific Text-to-Speech system. It ensures the gen-

eral time consistency of the animation sequence by taking into account the length of the generated audio. From evaluation, we conclude that gesture generation can be improved significantly by extending the limited manual mapping with our method.

The limiting factor in our method is the *Gesture dataset* and the selection of upper-body joints excluding hands. Some of the animations in the *Gesture dataset* varies only in hand motion. This may result in miss-classification due to exclusion of hand pose. Due to the large number of rules and use of word embedding, the memory and computation cost increases significantly. Due to the high memory and computation cost, this method is not feasible to run on mobile and wearable devices. For these type of devices, this method can be deployed on cloud [24]

The future work can include full-body pose along with hand pose estimation in the video dataset and sentiment analysis of the text. This can improve the experience more. By increasing the number of gestures, mapping can be text to multiple animations taking into account the sentiment of the text. The other direction can be personalized automatic rule generations. For personalization, the pose and text data will need more personality related tags like gender, age and personality traits.

In conclusion, our method automatically generates a large set of text-to-gesture rules from the pose and text data by using pre-defined gesture animations. The text-to-gesture rules generated by our method can significantly improve the limited manual mapping.

Acknowledgement

This research is supported by the Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program and KIST Flagship Project.

References

[1] M L Knapp and J A Hall. *Nonverbal communication in human interaction*. 1972.

[2] Michael Studdert-Kennedy. Hand and Mind: What Gestures Reveal About Thought, 1994.

[3] Mahoro Anabuki, Hiroyuki Kakuta, Hiroyuki Yamamoto, and Hideyuki Tamura. Welbo: An embodied conversational agent living in mixed reality space. In *Conference on Human Factors in Computing Systems - Proceedings*, 2000.

[4] Jorge Arroyo-Palacios and Richard Marks. Believable Virtual Characters for Mixed Reality. In *Adjunct Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2017*, 2017.

[5] Vanya Avramova, Fangkai Yang, Chengjie Li, Christopher Peters, and Gabriel Skantze. A virtual poster presenter using mixed reality. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[6] Roberta Castano, Giada Dotto, Rossella Suma, Andrea Martina, and Andrea Bottino. Virtual-me: A library for smart autonomous agents in multiple virtual environments. In *Communications in Computer and Information Science*, 2014.

[7] Stefan Kopp, Bernhard Jung, Nadine Leßmann, and Ipke Wachsmuth. Max – A Multimodal Assistant in Virtual Reality Construction. *Künstliche Intelligenz*, 2003.

[8] Octavian M. Machidon, Mihai Duguleana, and Marcello Carrozzino. Virtual humans in cultural heritage ICT applications: A review, 2018.

[9] Deborah Richards. Agent-based museum and tour guides. 2012.

[10] Spyros Vosinakis, Nikos Avradinis, and Panayiotis Koutsabasis. Dissemination of Intangible Cultural Heritage Using a Multi-agent Virtual World. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[11] Anton Bogdanovych, Tomas Trescak, and Simeon Simoff. Formalising believability and building believable virtual agents. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.

[12] Ferdinand De Coninck, Zerrin Yumak, Guntur Sandino, and Remco Veltkamp. Non-verbal behavior generation for virtual characters in group conversations. In *Proceedings - 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2019*, 2019.

[13] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. BEAT: The Behavior Expression Animation Toolkit. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*, 2001.

[14] Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.

[15] Dirk Heylen, Stefan Kopp, Stacy C. Marsella, Catherine Pelachaud, and Hannes Vilhjálmsson. The next step towards a function markup language. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008.

[16] Jina Lee and Stacy Marsella. Modeling speaker behavior: A comparison of two approaches. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[17] Michael Neff, Michael Kipp, Irene Albrecht, and Hans Peter Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics*, 2008.

[18] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. Analyzing Input and Output Representations for Speech-Driven Gesture Generation. In *IVA 2019 - Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, 2019.

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

[20] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning Individual Styles of Conversational Gesture. jun 2019.

[21] Ylva Ferstl, Michael Neff, and Rachel McDonnell. Multi-objective adversarial gesture generation. *Proceedings - MIG 2019: ACM Conference on Motion, Interaction, and Games*, 2019.

[22] Michael Kipp. ANVIL A generic annotation tool for multimodal dialogue. In *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology*, 2001.

[23] Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. Gesture controllers. In *ACM SIGGRAPH 2010 Papers, SIGGRAPH 2010*, 2010.

[24] Ghazanfar Ali, Hong Quan Le, Junho Kim, Seung Won Hwang, and Jae In Hwang. Design of seamless multi-modal interaction framework for intelligent virtual agents in wearable mixed reality environment. In *ACM International Conference Proceeding Series*, 2019.

[25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014.

[26] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.