The Nectec Gesture Generation System entry to the GENEA Challenge 2020

Ausdang Thangthai, Kwanchiva Thangthai, Arnon Namsanit, Sumonmas Thatphithakkul,
Sittipong Saychum
Speech and Text understanding Research Team
National Electronics and Computer Technology Center (NECTEC), Thailand
ausdang.tha,kwanchiva.tha,arnon.nam,sumonmas.tha,sittipong.say@nectec.or.th

ABSTRACT

This paper describes the gesture generation system developed by Nectec team for GENEA (generation and evaluation of non-verbal behaviour for embodied agents) challenge 2020. To develop the proposed system, this challenge provides the common training dataset of speech (audio and its transcription) and 3D full-body motion capture in the biovision hierarchical (BVH) format, namely the Trinity speech gesture dataset.

Our proposed system consists of pre-processing data and gesture modeling. In terms of data pre-processing, the cleaning data and preparation process of the input (audio and text) and output (gesture motion) features were proposed. For modeling gestures, an encoder-encoder bidirectional LSTM architecture was used to build a gesture model from both acoustic speech and textual information.

To evaluate the proposed system, the held-out dataset of audio and its transcription from the challenge were used to generate the gesture motion, specifically in the body's upper part. Then, the submitted results from all teams and a baseline system were evaluated using a crowdsourced system. The subjective evaluation results show a fair rating result in terms of both appropriateness and human-likeness.

KEYWORDS

gesture motion, encoder-decoder LSTM

1 INTRODUCTION

We have all heard of these two phrases; 'a picture is worth a thousand words' and 'action speaks louder than words'. With human communication, verbal and non-verbal are two channels to convey the message in our daily lives. If we compare the benefits of a picture and action with human communication, non-verbal communication must help us quickly understand the message we want to communicate. Many studies have been reported that around 70-93% of communication is non-verbal such as body movements, head movements, hand gestures, facial expression and vocal tone [8, 9]. For example, lectures can use nodding heads signal to indicate students' understanding. In co-speech gestures, we always use hand gestures linking to the words we speak, such as showing the number while speaking that number. In terms of the virtual world, Virtual humans or virtual robots are becoming popular. Hence, it is important to teach them acting like humans, for example, making virtual humans talking with gestures.

In terms of approach, gesture synthesis can be broadly divided into text-driven and audio-driven. Both approaches aim to synthesise gesture speech motions (representing by a sequence of parameters). The main difference is a given input. A sequence of text is an input for a text-driven approach, while a speech input is an input for a audio-driven approach. With the text-driven approach, the given text has not been used directly as raw text input. It has to be represented with different kinds of features. In the case of limited-domain systems, a voice command to control robot in the form of word, for example,Ogata et al. [10] used a one-hot-vector to represent word index. Then, the combination of these vectors and parameter bias were used to model the relationship between sentences and the robot's motion using recurent neural networks (RNNs). However, a simple one-hot-vector does not suit in general domain systems. One of the reasons is that it is impossible to collect all English words. Yoon et al. [17] applied the pretrained word embedding GloVe for representing each word with embedding vectors [11]. Additionally, the embedding layer with 300 output dimensions was defined as the first hidden layer of their encoder-decoder GRU network architecture. Similarly, Kucherenko et al. [6] used pre-trained word embeddings BERT instead of GloVe. Then, they also used duration and speech features to predict the gesture motion using a simple feed-forward network architecture. With the audio-driven approach, the given speech has not been used directly as raw speech input. It has to be represented with various kinds of features. Mel-frequency Cepstrum Coefficients (MFCC), one of the popular speech feature representations, was employed by Kucherenko et al. [5]. Moreover, mel-frequency power spectrograms were used by Alexanderson et al. [1]. It is interesting to note that they found that there is no difference between MFCC and spectrogram features in terms of objective and subjective tests. Additionally, pitch and intensity were used by Chiu and Marsella [2], called prosodic features.

In this paper, we used a similar idea of Kucherenko et al. [6] that they used the combination of textual and audio information to drive their gesture generation system. The major difference between this work and Kucherenko et al. [6] is that we applied word embedding outside the networks and our models learned by an encoder-decoder bidirectional LSTM neural network architecture. Moreover, we aim to include more information, including sound form and linguistic features.

The rest of this paper is organised as follows. Section 2 explains the full pipeline of the Nectec system. Section 3 presents the results of comparative subjective tests using a crowdsourced system and discussions. Finally, the conclusion is given in Section 4.

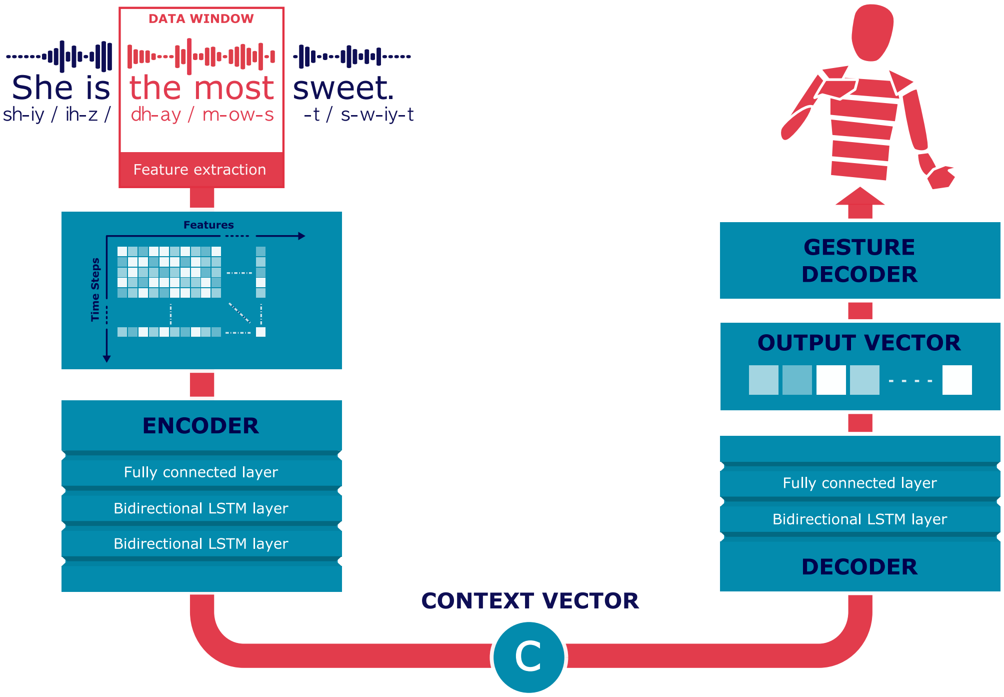

Figure 1: Our text and speech to gesture generation system based on an encoder-decoder LSTM architecture.

2 NECTEC GESTURE GENERATION SYSTEM

We propose the gesture generation system, which is based on both text and speech features. This work extends from an audio-driven baseline system[5]. When it comes to text features, the sound form and linguistic features are extracted from given raw text input. When it comes to speech features, we used the same features as proposed in Kucherenko et al. [5]. More details can be found in this Section.

2.1 Data Preparation

Our initial idea, we know that sound form is directly related to human speech production such as how to pronounce each word and the duration of each sound. Hence, we believe that this information could benefit the training process and lead to better performance. With the GENEA challenge 2020, the Trinity speech gesture dataset [3] is used as the standard dataset and provides to the participants for training the gesture models. However, they do not offer phonetic transcription. They give only 3D full-body motion capture in BVH format, audio in wave format, and word transcription (or writing form) in JSON format. Hence, this subsection describes the process of getting phoneme labels from the Trinity dataset, including sentence segmentation and automatic phoneme alignment.

2.1.1 Sentence Segmentation. The total length of the Trinity dataset is approximately 4 hours and consists of 23 recordings. Noted that the average length of each recording is about 600 seconds. We randomly found the missing words in the word transcriptions that originated from repetitions, revisions, and filler words. Additionally, these inaccurate transcription will lead to misalign because the audio file and transcript text file must be match. To avoid misalignment of the whole recording, this subsection aims to separate the long motion, audio, and a sequence of words from word transcription into short sentence segments.

We found that splitting a long text in each utterance into phrases can be done using an 'alternatives' tag in JSON files (word transcription). However, the average length in each phrase is still longer than 30 seconds. Hence, we design to use 'full stop' to separate each long phrase into short sentences. Finally, there are 2,039 sentences and the average sound length is about 5.79 seconds.

2.1.2 Forced Alignment. The forced alignment aims to automatically find time-stamp of an orthographic transcription in the speech segment. The orthographic transcription can be based on phones, diphones, triphones, syllables, and words. In this paper, we focus on the beginning and ending at word and phone levels using a pronunciation dictionary. There are many aligners toolkit for example Penn forced aligner[1], Prosodylab-aligner[2], EasyAlign[3], and Montreal forced aligner[4]. The Montreal forced aligner is used to perform forced alignment in this paper, which is based on the Kaldi ASR toolkit[12]. Kaldi is a state-of-the-art toolkit for speech recognition and uses for many speech-related tasks such as automatic speech recognition (ASR), speaker verification, and forced alignment.

In data preparation, the Montreal forced aligner requires an audio file in wave file format, pronunciation dictionary, and the

[1] https://web.sas.upenn.edu/phonetics-lab/facilities/
[2] http://prosodylab.org/tools/aligner/
[3] http://latlcui.unige.ch/phonetique/easyalign.php
[4] https://montreal-forced-aligner.readthedocs.io/

corresponding transcript. In our case, 2,039 audio files and its word transcription are available from the previous subsection. However, the dictionary is not available. Hence, we perform the conversion of text to phoneme using flite[5] (small and fast run-time version of festival [14]). After that, unique words and a sequence of phonemes are used to build a pronunciation dictionary. Then, all required data are used to align with a 'mfa_train_and_align' command. Finally, the output consists of word and phone transcriptions, as shown in Figure 2. One of the Montreal forced aligner benefits is the aligner will automatically insert a short pause between words. In this case, we do not require to insert short pauses into transcriptions like the traditional HTK toolkit. We also found that there are 114 audio files that the aligner cannot align some phonemes into speech input. The incorrect text and phoneme transcription are the main reasons for failure, which we ignored all of these failure audio files in this paper.

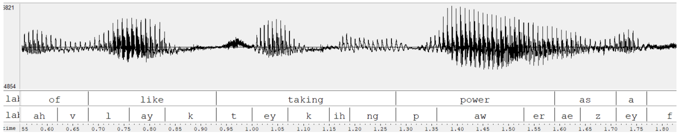

Figure 2: An example output of words and phonemes alignment.

2.2 Input Feature Extraction

This section describes how input features in each frame are represented, which we extract from acoustic speech and text transcription.

2.2.1 Text Features . This section describes how to extract input features from a given sentence text. It begins with the forced alignment module (described in Section 2.1.2) to get where phonemes start and stop. After that, the linguistic features of phone, syllable, word, phrase and sentence level were obtained, as shown in Table 1.

Table 1: Linguistic Features

.

Level	Features
Frame	Current phoneme
	Position of frames in phoneme
	Acoustic class
Phoneme	Phoneme context
Syllable	Position of phonemes in syllable
Word	Word embedding
	Position of syllables in word
Phrase	Position of syllables in phrase
	Position of words in phrase
Sentence	Position of syllables in sentence
	Position of words in sentence
	Position of phrases in sentence

With the frame-level features, **current phoneme** in each frame is encoded to 41-D binary features using a one-hot representation. The value of a one-hot vector is all zero except the phoneme index set to one. **Position of frames in phonemes** is encoded to 3-D binary features based on the index of these three categories (begin, middle or end). 57 questions related to the English language's sound are used to encode to 57-D binary features of **Acoustic class**. The questions are taken from an example of context-dependent label format for HMM-based speech synthesis in the HTS toolkit, for example, 'Is the current phoneme consonant?' or 'Is the current phoneme unvoiced fricative?'.

With the phoneme level features, the gesture motions depend on the behind and ahead articulator, called coarticulatory effects. Hence, **phoneme context** are used to give two preceding and following phonemes as we used 'quinphone context'. Then, this feature is represented by 4 x 41 dimensional binary features using a one-hot representation.

With the word-level features, we intend to use off-the-shelf pre-trained models in this work. The pre-trained model in Spacy[6] is used to convert word to word vectors called as **word embedding** features. There are several Spacy's pretrained models available for English language based on language, genre and size of the word vectors. To simplicity, this paper prefers 'en_core_web_sm' pretrained model, a small-sized English model trained on written web text such as blogs, news and comments. A 96-D numerical features vector represents this feature.

With the rest features, Thangthai et al. [15] reported that the sub level position of phoneme, syllable, word and phrase affects the mouth movements. We believe that these position features (begin, middle, end) could affect the gesture movement as well. Hence, **position of phonemes in syllable**, **position of syllables in word, phrase, sentence**, **position of words in phrase, sentence** and **position of phrase in sentence** are included in this paper.

2.2.2 Speech Features. There are connections between gesture and speech, such as gesture stroke and pitch, vowel onset, stressed syllable, as shown in a review by Wagner et al. [16]. Therefore, we extract 30 dimensions of acoustic features: 26 Mel-frequency cepstral coefficients (MFCCs) and 4 acoustic prosodic features, which are fundamental frequency (f0), energy, and its first derivatives. The acoustic features are obtained from a speech segment at a 25-millisecond window length with 10-millisecond overlapping. Thus the acoustic features are vectorised at 100 frames per second (fps). Then we downsample the acoustic features by averaging over 5 frames interval to match the target gesture motion frame-rate at 20 fps.

2.3 Output Feature Extraction

We extract 40-dimensional features from 3D human motion data using a representation learning method. The 3D human motion data are presented in the biovision hierarchical (BVH) format. To extract output features, we follow the steps proposed by Kucherenko et al. [5] to learn the representation. First, we select 15 joints from the 3D motion data: Spine, Spine1, Spine2, Spine3, Neck, Neck1,

[5] http://cmuflite.org/

[6] https://spacy.io/usage/spacy-101

Head, RightShoulder, RightArm, RightForeArm, RightHand, LeftShoulder, LeftArm, LeftForeArm, and LeftHand. Next, we apply the exponential map [4] to represent the selected joints. Then, the Denoising Autoencoder Networks (DAEs) were trained to encode the exponential map representation. The DAE network consists of an input layer, an output layer, and 3 fully-connected layers where the size of the hidden unit per layers is 325, 40, 325, respectively. The size of input and output vectors of 3D motion features are 45 dimensions (15 joints: 45 3D-coordinates). The motion features are then encoded-decoded, and the bottle-neck layer, which has 40 dimensions, is used as a representation of 3D motion data. To simplify, this 40-D representation will be used as the target output of the proposed system instead of the 45 3D-coordinates.

To trained the representation learning model, the DAEs model was pre-trained on 50 epochs with a learning rate of 0.001 and the mini-batch size of 128. The model was then fine-tuned using adam optimization method to minimize the MSE errors on 20 epochs and the learning rate 0.0001, with a dropout rate at 0.1. To corrupt the input data, we add the Gaussian noise to each feature dimension. Noted that, the original data in each dimension is added by noise factor (0.01) times the standard deviation of that feature dimension.

To generate the 3D motion from speech, the 40-D vectors predicted from the system described in Section 2.4 were decoded by the 'gesture decoder' part of the DAEs resulting in 45 3D-coordinates. However, the motion generated from the model may have discontinuity between frames simply called jerk motions. To reduce the effect of jerky motion outputs, most research papers suggest to smooth the results. Hence, this paper performed the Savitzky–Golay filter in intension to avoid discontinuity results [13].

2.4 Encoder-Decoder Bidirectional LSTM-RNN Architecture

Our network structure is based on encoder-decoder using bidirectional long short term memory (B-LSTM) architecture, as shown in Figure 1. With the input part, the t frame of audio and text features is segmented into fixed of time steps length. This paper is set the number of time steps to 31, which is spanned over 15 frames before and after the current frame. For simplicity, figure 1 assumes that the durations in each phoneme are one frame per phoneme, and the number of time steps is 5. Additionally, the current frame is at the phone /m/.

The encoder-decoder consists of 3 main layers, including encoder, context vector and decoder. The encoder will summarise the input features into a context vector, *c*. Then, the decoder will transform the context vector, *c*, to output features. A fully connected layer is first defined with the encoder layer, which has 512 units with a rectified linear unit (ReLU) activation function. Batch normalisation and 50% dropout are also applied to avoid overfitting and speed up learning between the fully connected and B-LSTM layers. After that, two B-LSTMs are defined, which has 128 units per layer. After that, a 128-D context vector is used to decode output features using 128 units of B-LSTM and 40 units of a fully connected layer. A linear activation function is employed at the output layer. One of the most common loss function, mean squared error (MSE) is used to calculate the difference between the predictions and the ground truth. The Adam optimization algorithm is used to optimise the models with 0.9 and 0.999 for beta1 and beta2, respectively. The size of the mini-batch is set to 128, and a learning rate is set to 0.0003. The maximum number of epochs is set to 300. The model will save at the end of every epoch if the mean squared error of validation is lower down. Moreover, the model will stop training if the mean squared error of validation does not improve more than 15 epochs.

3 EVALUATION RESULTS

The MUSHRA test (MUltiple Stimuli with Hidden Reference and Anchor) for video was used as an evaluation interface to perform the subjective tests in the GENEA challenge 2020 [7]. This test asked participants to watch videos as many times as they required. Then, participants had to rate 0 to 100 from bad to excellent in specific questions based on two studies, including 'human-likeness' and 'appropriateness'. 9 systems were evaluated by 125 participants who passed all attention checks (to avoid scam participants). Where;

N: Natural motion,
M: Mismatched motion,
BA: audio-driven baseline system[5],
BT:Text-driven baseline system[17],
SE: Our system (Nectec System),
S...: Other systems.

3.1 Human-likeness Evaluation

This test aims to measure the quality of the generated gesture motion with a specific question; 'How human-like does the gesture motion appear?'. Participants had to focus only on the generated motion as the videos have no sound.

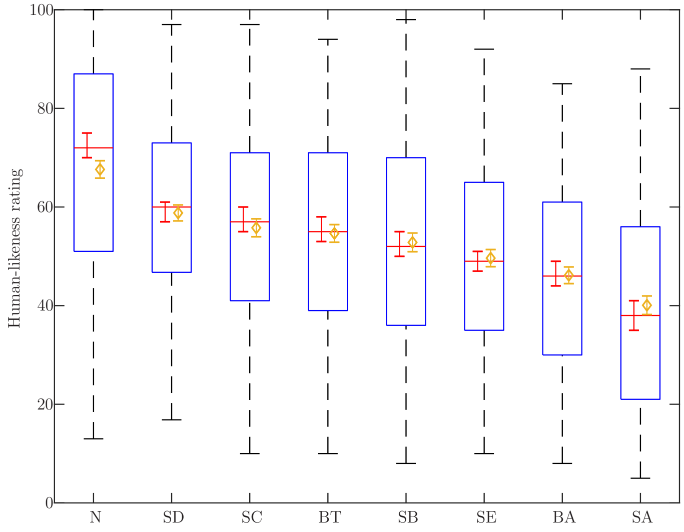

Figure 3: The ratings distribution in the human-likeness evaluation.

3.2 Appropriateness Evaluation

This test investigates the relationship between generated gesture motion and acoustic speech with a specific question; 'How appropriate are the gestures for the speech?'. Participants had to focus on both the generated motion and sound, ignoring motion quality.

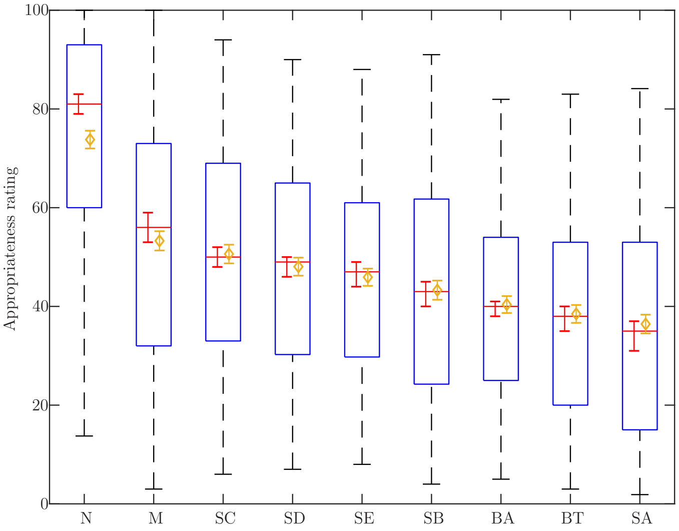

Figure 4: The ratings distribution in the appropriate-likeness evaluation.

3.3 Results Discussion

Figure 3 and 4 shows that the medians and means ratings of human-likeness and appropriateness are identified by the red bars and yellows diamonds. It was not surprising that natural gestures motion got the highest rating in both experiments. The results show that participants easily recognised the difference between real and synthesised motions, as the rating of synthesised gesture motions is far away from that of the natural motions. It can be seen that all submitted models are all at a similar level, but they represent with different distributions. Another interesting result in Figure 4, the high rating from 'M' over the other systems, showed that none of the generated systems is well enough. A possible explanation the duration of the testing videos might be too short to spot the mismatch motion that visualised from 'M' system.

We observed that our model, SE, had not good at head movements based on video results. We found that the avatar's head movements can interpret that they do not talk with the viewers. This might be the primary reason why our system has a lower rating of human-likeness. However, in terms of appropriateness rating, we have got a good rating result comparing 'SB, BA, BT and SA' systems. This result confirmed that our proposed text features using phonetic and linguistic features are useful and related to the audio.

4 CONCLUSION

In this paper, we describe our encoder-decoder LSTM architecture for generating gesture motions using both acoustic and textual information entry to the GENEA challenge 2020. Our framework used only a provided database (the Trinity speech gesture dataset) and off-the-shelf Spacy's pre-trained word embeddings (en_core_web_sm). We used the same features with speech features as proposed in audio-driven baseline system, 'BA'. Interestingly, the appropriatenees rating of our system is higher than both the 'BA' and 'BT' baseline system (p>0.01). In terms of text features, we proposed phonetic and linguistic features. The subjective evaluation results show a fair rating result in terms of both appropriateness and human-likeness.

REFERENCES

[1] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. 2020. Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows. *Computer Graphics Forum* 39, 2 (2020), 487–496. https://doi.org/10.1111/cgf.13946 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13946

[2] Chung-Cheng Chiu and Stacy Marsella. 2011. How to Train Your Avatar: A Data Driven Approach to Gesture Generation. In *Intelligent Virtual Agents*, Hannes Högni Vilhjálmsson, Stefan Kopp, Stacy Marsella, and Kristinn R. Thórisson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 127–140.

[3] Ylva Ferstl and Rachel McDonnell. 2018. IVA: 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] F Sebastian Grassia. 1998. Practical parameterization of rotations using the exponential map. *Journal of graphics tools* 3, 3 (1998), 29–48.

[5] 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* (Paris, France) *(IVA '19)*. Association for Computing Machinery, New York, NY, USA, 97–104. https://doi.org/10.1145/3308532.3329472

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

[7] 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/

[8] David Lapakko. 2007. Communication is 93% Nonverbal: An Urban Legend Proliferates. *Communication and Theater Association of Minnesota Journal* (2007), 7–19.

[9] A. Mehrabian. 1972. *Nonverbal Communication*. Aldine-Atherton. https://books.google.co.th/books?id=Vc5-AAAAMAAJ

[10] T. Ogata, M. Murase, J. Tani, K. Komatani, and H. G. Okuno. 2007. Two-way translation of compound sentences and arm motions by recurrent neural networks. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1858–1863.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[12] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* (Hilton Waikoloa Village, Big Island, Hawaii, US). IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[13] Abraham. Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639. https://doi.org/10.1021/ac60214a047 arXiv:https://doi.org/10.1021/ac60214a047

[14] Paul A Taylor, Alan Black, and Richard Caley. 1998. The Architecture of the Festival Speech Synthesis System. In *The Third ESCA Workshop in Speech Synthesis*. Jenolan Caves, Australia, 147–151.

[15] Ausdang Thangthai, Ben Milner, and Sarah Taylor. 2019. Synthesising visual speech using dynamic visemes and deep learning architectures. *Computer Speech Language* 55 (2019), 101 – 119. https://doi.org/10.1016/j.csl.2018.11.003

[16] Petra Wagner, Zofia Malisz, and Stefan Kopp. 2014. Gesture and speech in interaction: An overview. *Speech Communication* 57 (2014), 209 – 232. https://doi.org/10.1016/j.specom.2013.09.008

[17] 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