Sequence-to-Sequence Predictive models: from Prosody to Communicative Gestures

Fajrian Yunus, Chloé Clavel, Catherine I Pelachaud

Sequence-to-Sequence Predictive models: from Prosody to Communicative Gestures

Fajrian Yunus
fajrian.yunus@upmc.fr
ISIR
Sorbonne University
Paris, France

Chloé Clavel
chloe.clavel@telecom-paristech.fr
LTCI
Télécom ParisTech
Palaiseau, France

Catherine Pelachaud
catherine.pelachaud@upmc.fr
CNRS
ISIR, Sorbonne University
Paris, France

ABSTRACT

Communicative gestures and speech prosody are tightly linked. Our aim is to predict when gestures are performed based on prosody. We develop a model based on a seq2seq recurrent neural network with attention mechanism. The model is trained on a corpus of natural dyadic interaction where the speech prosody and the gestures have been annotated. Because the output of the model is a sequence, we use a sequence comparison technique to evaluate the model performance. We find that the model can predict certain gesture classes. In our experiment, we also replace some input features with random values to find which prosody features are pertinent. We find that the F_0 is pertinent. Lastly, we also train the model on one speaker and test it with the other speaker to find whether the model is generalisable. We find that the models which we train on one speaker also works for another speaker of the same conversation.

KEYWORDS

Machine Learning, Neural Network, Gesture

ACM Reference Format:
Fajrian Yunus, Chloé Clavel, and Catherine Pelachaud. 2018. Sequence-to-Sequence Predictive models: from Prosody to Communicative Gestures. In *Proceedings of WACAI 2020: Workshop sur les "Affects, Compagnons Artificiels et Interactions (WACAI 2020).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Human naturally gesticulates while speaking [15]. There are different families of gestures [20] that vary depending on the types of information they convey. Gesture helps the locutor to form what he wants to convey and also helps the listener to comprehend the speech [11]. Therefore, it is desirable for a virtual agent which interacts with humans to show natural-looking gesticulation behaviour. Because of that, researchers have been working on automatic gesture generation in the context of human-computer interaction [6, 7, 17, 23]. So far, most of prior gesture generation works simplify the problem by generating only one type of gesture. For example, the algorithm proposed by Kucherenko et al [17] generates only beat gestures while the algorithm proposed by Bergman et al [4] generates only iconic gestures. An ideal generator should be able to generate various types of gestures. Therefore, it is desirable to know when different types of gesture are performed. Here, we attempt to compute when a virtual-agent should perform a certain type of gesture. We are doing this in a larger context of generating natural-looking gestures within the context of human-computer interaction.

We compute the gesture class based on the speech prosody. For the computation, we use a recurrent neural network with an attention mechanism [2] to learn the relationship. This technique is based on sequence-to-sequence formulation [25]. The input is the speech prosody, broken into time-steps, and the output is the sequence of gesture classes. Our input features are the F_0, F_0 direction score, and intensity. They are known to be related to gestures. Besides that, by limiting the number of features, we mitigate the problem of the curse of dimensionality.

In section 2, we explain the relevant prior works about gesture, gesture generation techniques, and the evaluation techniques in the related work section. In section 3, we explain the dataset we use for our experiments. We explain the raw content, the various annotations provided in the dataset, and how we process the data. In section 4, we explain the model which we use and how it is implemented. The evaluation of the model is presented in section 5. The experiments are described in the section 6. Finally, we discuss our results in Section 7 and draw some conclusions.

2 RELATED WORK

McNeill [20] splits gestures into four classes: metaphorical gesture, deictic gesture, iconic gesture, and beat gesture; metaphorical gesture is to convey an abstract concept, deictic gesture is to point at an object or a location, iconic gesture is to describe a concrete object by its physical properties, and beat gesture marks the rhythm. A gesture, except beat, consists of several phases, namely preparation, pre-stroke-hold, stroke, post-stroke-hold, hold and retraction [20]. The stroke phase is obligatory while the preparation, the hold and the retraction phases are optional. When multiple gestures are performed consecutively, the gesture phases are chained together. For example, if gesture X precedes gesture Y, there might not be any retraction before gesture Y's stroke. Successive gestures coarticulate one from the others. Beat gesture, on the contrary, has no phase.

Embodied Conversational Agents are virtual agents endowed with the capacity to communicate verbally and nonverbally [6]. Several models have been developed to drive the nonverbal behaviors of these agents [4, 6]. The earliest gesture generators are rule-based

WACAI 2020, 3–5 June 2020, Île d'Oléron, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

systems [6, 23]. However, the rules governing the relationship between gesture and speech are too complex to be manually specified. These past years, researchers develop machine-learning based gesture generators. There are two classes of the machine-learning based gesture generators, namely prosody-based model and text-based model. Beat gestures are characterized by their rhythmic pattern. They are often produced with a soft open hand gestures. The hand shape and movements of non-beat gestures are mainly related to semantic.

The prosody-based generators [7, 17] have a similar formulation: they express the problem as a time series prediction problem where the input is the prosody and the output is the gesture motion. They generate beat gestures. There are also text-based generators [1, 4]. They generate iconic, metaphoric, and deictic gestures because those gestures are related to the semantics, which are inferred from the text. The semantics are then used to predict the gestures. Unlike the prosody-based generators, they have more varied problem formulations.

A common simplification which has been used in many existing gesture generation works is that gestures can be inferred from the speech. However, gestures and speech are generated from a common process [20]. In some cases, gestures and speech complement each other instead of conveying the same information [20]. Nevertheless, in our work, we apply that simplification too. Speech, as we assume for this work, consists of two components: the prosody and the text.

3 DATASET

We use the Gest-IS English corpus [24]. In the dataset, there are various dialogues between the same two speakers. Each dialogue is contained in a video. The total length of those videos is around 50 minutes. The corpus has been annotated along different layers: gesture phase, gesture types, chunk boundaries, classification annotations on whether the gesture is communicative or non-communicative, and the speech transcription data. With these data, we have to decide which sub-data is pertinent.

We decide a sub-data to be pertinent if it is part of an utterance. We use the transcription timestamps from the corpus to find the start and end of the speeches. In the data, there are many times when the person does not speak. We use these gaps to demarcate the utterances. However, it is also normal to have short gaps within an utterance. Therefore, we need a threshold to decide whether a gap is short enough to actually be a part of the same utterance. For this, we use the concept of Inter-Pausal Unit (IPU) [18] to define one utterance, which in turn we consider as one sample. By using the IPU concept, we consider two consecutive speeches whose gap is less than 200 milliseconds to be one utterance [22].

After splitting the data into samples where each sample is one utterance, we use OpenSmile [13] to extract the prosodic features with 100 milliseconds time-steps. To avoid the curse of dimensionality problem, we limit ourselves to only F_0, F_0 direction score, and intensity because they are known to be related to gestures [9, 19].

The model takes as input only the prosodic features. It does not consider any semantic feature. Thus, we decide to divide the communicative gestures into beats and all other gestures, that is deictic, metaphoric and iconic gestures. We classify the gestures into four classes, namely "NoGesture", "Beat", "NonBeatNonStroke", "NonBeatStroke". The gesture class depends on what gesture is being performed at that particular time-step. "NoGesture" refers to the time when the person does not gesticulate. "Beat" refers to the time when the person uses beat gesture. "NonBeatNonStroke" refers to the time when the person does a non-stroke phase (e.g. preparation, retraction) of either metaphoric, iconic, or deictic (i.e. non-beat) gestures. "NonBeatStroke" refers to the time when the person does the stroke phase of either metaphoric, iconic, or deictic gestures (i.e. non-beat). Note that beat gestures do not have stroke or non-stroke phases, therefore beat is treated as a separate class.

Each sample is one utterance surrounded by IPUs. The utterances are natural utterances, therefore they have different lengths. Unfortunately, the technical constraint of recurrent neural network requires all samples to have the same length. Therefore, we pad the inputs with 0-vectors and we pad the outputs with "suffix" auxiliary class so that all samples have the same length. In our full dataset, we have 3851 time-steps of "NoGesture"s (6.71%), 946 time-steps of "Beat"s (1.65%), 3303 time-steps of "NonBeatNonStroke"s (5.76%), 2739 time-steps of "NonBeatStroke"s (4.77%), and 46536 time-steps of the auxiliary "suffix"s (81.11%).

4 MODEL

We use recurrent neural network with attention mechanism [2] to perform the prediction; it is based on the our previous work [26]. Our implementation is based on the Keras [1] code of Zafarali [2]. The model takes 3 speech prosody input features: F_0, F_0 direction score and intensity. The prosody input itself is treated as a time series. The output is another time series, namely the gesture class sequence. The schema of the model is presented in the figure 1. The original Zafarali's code is to do date format translation, therefore it uses word embedding. Our input is already in the form of vector of numbers, therefore we remove the word embedding. Unlike the standard sequence-to-sequence model where there are only encoder and decoder, the recurrent neural network with attention mechanism has an attention map between the encoder and the decoder. The map allows any encoder to directly influence any decoder. The map is a set containing the weights of the connections between the encoders and the decoders. The learning uses categorical cross-entropy as the loss function and Adam as the optimiser.

5 EVALUATION TECHNIQUE

The recurrent neural network with attention mechanism is based on the sequence-to-sequence formulation. This formulation is pioneered by Sutskever et al [25]. Sutskever et al use the formulation to create a language translator. To evaluate their model, Sutskever et al use BiLingual Evaluation Understudy (BLEU). BLEU, however, is specific to language translation task. Chorowski et al [8] use phoneme error rate (PER) to evaluate their sequence-to-sequence model. Meanwhile, Bahdanau et al [3] use Character Error Rate (CER) and Word Error Rate (WER) to evaluate their model. All of these prior works are evaluated by using domain specific measurements. Therefore, we devise our way of evaluating the model.

[1] https://keras.io/
[2] https://github.com/datalogue/keras-attention

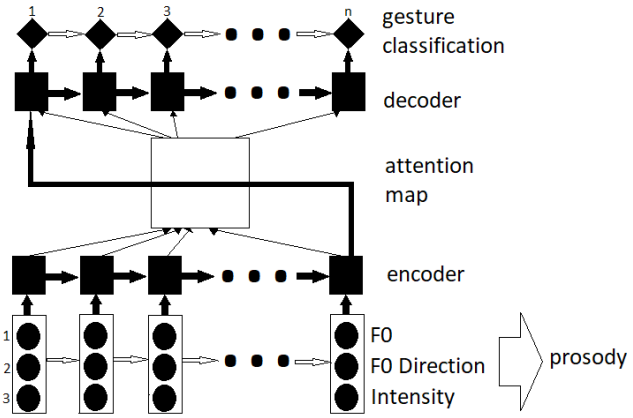

Figure 1: Our Neural Network Model

To evaluate the model, we use a general-purpose sequence similarity measurement technique to compare the similarity between the prediction and the ground truth, and we use the similarity measure as the performance measure. We use the sequence comparison algorithm proposed by Dermouche and Pelachaud [10] as our evaluation technique. It measures the city-block distance between a block in the ground truth and a block in the prediction. If the distance is below a certain threshold, then they are considered as matches. The precise match formula is at formula 1. We define b_{ps} as the start of the prediction block and b_{pe} as the end of the prediction block. Correspondingly, we define b_{ts} as the start of the ground truth block and b_{te} as the end of the ground truth block. We also define T as the distance threshold. We define prediction block and ground truth block as matches when formula 1 is true.

$$\left|b_{ps} - b_{ts}\right| + \left|b_{pe} - b_{te}\right| \leq T \tag{1}$$

We measure the performance based on how many blocks match and how long those blocks are, and then we normalise against the length of the sample and the proportion of that particular class. The length of the blocks (for the same class) matters because the ideal matches should be as long as possible. The normalisation against the proportion of that particular class is done because the class distribution is not balanced. For instance, "beat" comprises less than 9% of the total time-steps (after excluding the "suffix" auxiliary class).

We also introduce the concept of "insertion" and "deletion". A block which exists in the prediction but has no match in the ground truth is considered to be "inserted". This is similar to *false positive*: we predict what actually does not happen. Similarly, a block which exists in the ground truth but has no match in the prediction is considered to be "deleted". This is similar to *false negative*: we fail to predict something which actually happens. The precise definition of alignment, insertion, and deletion score are at formula 2. In the formula, n stands for the number of samples in the dataset, t_c is the number of time-steps of class c in the dataset, p_c is proportion of class c in the dataset, l is sample length (which is the same for all samples), $b.d$ stands for deleted block, d_c is the deletion score of class c, $b.i$ stands for inserted block, $b.p$ stands for predicted block, $b.t$ stands for ground truth block, and a_c is the alignment score of class c. The ideal alignment score is 1 while the ideal deletion and insertion score are 0. It means everything is aligned and there is neither deleted nor inserted block. The insertion score of class c can exceed 1 if we predict class c more frequently than it actually occurs. On the other hand, the deletion score is always between 0 and 1. The deletion score of class c is 1 when we fail to predict any of the block of that class. For the alignment score, if the predictor is accurate but slightly overestimates the length of the block, then the alignment score will be slightly higher than 1. On the other hand, if the predictor is accurate but often slightly underestimates the length of the block, then the alignment score will be slightly lower than 1.

$$\begin{aligned} p_c &= \frac{t_c}{n \times l} \\ d_c &= \frac{\Sigma_{b.d} length(b.d)}{n \times l \times p} \\ i_c &= \frac{\Sigma_{b.i} length(b.i)}{n \times l \times p} \\ a_c &= \frac{\Sigma_{b.p,b.t.aligned} length(b.p) + length(b.t)}{2 \times n \times l \times p} \end{aligned} \tag{2}$$

6 EXPERIMENT

We split our data with the proportion of 64% training data, 16% validation data, and 20% testing data. This is chosen according to the common 80%/20% rule. To optimise the model, we vary the dimensions of the encoder and the decoder. The dimensions of the encoder and decoder are varied from 1 to 3, because our data has three features. A challenge we face is that the loss function used in the training measures the match at the same time step, and the network is optimised according to that loss. This is slightly different from the metric we use in our evaluation. Therefore, we have to train the model many times to get a good result.

We run a series of experiments. In experiment 1, we generate random outputs according to the data distribution and match it against the ground truth output in order to establish a baseline performance. In experiment 2, we perform training and testing with our entire dataset in order to observe how well our model learns. In experiment 3, we do ablation studies. Here, we replace certain features with random values to observe how much the model learns about the structure of the data and how each feature affects the performance of the model. In experiment 4 and 5, we train the model with one speaker only and then we test it the other speaker in order to find out whether the model is generalisable to the other speaker of the same conversation. It should be noted that conversation partners are known to align their speech prosody [14].

In **experiment 1 (random output)**, we generate random outputs according to the probability distribution of the gesture classes, while completely ignoring the prosody input. Specifically, we measure two sets of probabilities, namely the probabilities that a sample is started by a particular class and the probabilities that a class follows another (or the same) class. This is done because our data consist of sequences, where an element affects the element at the next time step. We match this result against the output from out ground truth. We do this 50 times and we measure the mean of their performances. This can be seen as an extremely simple predictor

Table 1: Results

Random output result (result 1)			
	Alignment	Insertion	Deletion
Beat	0.084	1.043	0.945
NonBeatStroke	0.228	0.746	0.78
NonBeatNonStroke	0.232	0.655	0.775
NoGesture	0.464	0.629	0.509
Suffix	0.964	0.015	0.011
Trained and tested with the entire data (result 2)			
	Alignment	Insertion	Deletion
Beat	0.494	3.494	0.519
NonBeatStroke	0.568	0.38	0.459
NonBeatNonStroke	0.234	0.141	0.683
NoGesture	0.559	0.52	0.235
Suffix	0.997	0	0
All input features are randomised (result 3)			
	Alignment	Insertion	Deletion
Beat	0	0	1
NonBeatStroke	0.304	0.52	0.914
NoGesture	0.455	0.556	0.669
Using intensity only (result 4)			
	Alignment	Insertion	Deletion
Beat	0	0	1
NonBeatStroke	0.577	0.629	0.823
NoGesture	0.68	0.937	0.508
Using F_0 and the F_0 direction score only (result 5)			
	Alignment	Insertion	Deletion
Beat	0.375	3.181	0.594
NonBeatStroke	0.787	0.464	0.486
NoGesture	0.568	0.617	0.268
Using F_0 only (result 6)			
	Alignment	Insertion	Deletion
Beat	0.766	3.869	0.594
NonBeatStroke	0.66	0.418	0.488
NoGesture	0.547	0.589	0.287

Table 2: Results (cont)

Using F_0 direction score only (result 7)			
	Alignment	Insertion	Deletion
Beat	0	0	1
NonBeatStroke	0.346	0.555	0.918
NoGesture	0.437	0.573	0.623
Trained with the 1st speaker tested on the 2nd speaker (result 8)			
	Alignment	Insertion	Deletion
Beat	0.553	2.076	0.489
NonBeatStroke	0.604	0.479	0.47
NoGesture	0.54	0.369	0.248
Trained with the 2nd speaker tested on the 1st speaker (result 9)			
	Alignment	Insertion	Deletion
Beat	0.41	3.38	0.413
NonBeatStroke	0.487	0.553	0.601
NoGesture	0.49	0.4	0.107

and thus can be seen as the baseline result. The result is in table 1 result 1.

In **experiment 2 (training and testing with the entire data)**, we train and test the neural network model with the entire data with the 80% and 20% split mentioned earlier. Note that in this data, we have two speakers. We mix and shuffle the data, and then split them into training, validation, and testing data. The result is in table 1 result 2.

In **experiment 3 (ablation study)**, we want to observe how much the model we obtain in experiment 2 learns about the structure of the data and how each feature affects the performance of the model. In order to do that, we use the model and data used in experiment 2, but we replace the some or all input features (intensity, F_0, and F_0 direction score) with random values.

In the 1st sub-experiment, in order to observe how much the model learns the structure of the data, we randomise all input features (table 1, result 3). This way, we force the model to make "educated guesses" about the outputs without being able to know what the inputs are. Unlike in experiment 1 where the random outputs are generated based on two explicitly-set probability distributions, here we use a model whose prediction ability comes only from the training.

In the subsequent sub-experiments, we keep some features while randomising the others in order to find which features are tied to gesture classes. In the 2nd sub-experiment, we keep only the intensity (table 1, result 4). In the 3rd sub-experiment, we flip the condition, so we keep the F_0 and F_0 direction score (table 1, result 5). After that, to isolate the individual effect of the F_0 and the F_0 direction score, we keep the F_0 only (table 1, result 6) and F_0 direction score only (table 2, result 7).

In **experiment 4 (trained with the 1st speaker, tested on the 2nd speaker)**, we train the model with the 1st speaker, and then we test it on the 2nd speaker. In **experiment 5 (trained with the 2nd speaker, tested on the 1st speaker)**, we flip the condition. The results of both experiments are in table 2, result 8 and 9. We do both experiments 4 and 5 to find whether the trained models are generalisable to the other speaker of the same conversation.

7 DISCUSSION AND CONCLUSION

We observe in the performance of the random output (table 1, result 1), not all classes are equally easy to predict. For example, "Beat" with the alignment score of 0.084, is harder to predict than all other classes. The "Suffix" class is easy to predict, but is a mere auxiliary class we add as a workaround of a technical constraint, so this class has no significance.

In the performance of the model which is trained and tested with the entire data (table 1, result 2), we observe that its alignment score outperforms the random output (table 1, result 1), except on the "suffix" and "NonBeatNonStroke" classes. Nevertheless, the model outperforms the random output's alignment scores on the "Beat", "NonBeatStroke", and "NoGesture" classes which suggests that the model can predict those classes based on the intensity, F_0, and F_0 direction score.

In our case, the difficulty of predicting "Beat" is partly explainable by the lack of data. "Beat" occurs rarely in the corpus, therefore the difficulty of predicting it is expected. This leads us to the question on whether we would be able to predict "Beat" better if we have more data. Although "Beat" is known to be related to prosody [20], we are not aware of any prior work which attempts to distinguish "Beat" from other gestures; for example, [7, 17] only use prosody as their inputs, which means they implicitly assume that all gestures are "Beat". Besides that, "Beat" are not necessarily performed by hands. "Beat" can also be performed by head or facial movements [5, 12, 16]; head nods and raised eyebrows often punctuate pitch accents. Our corpus does not have head nor facial movement annotation, which means we lose some information. Indeed, the incompleteness of the corpus and the nature of our errors are problems we want to investigate. It is possible that this absence of head and facial movement annotation contributes to the difficulty of predicting the "Beat".

On the "NonBeatStroke" class, our predictor is able to surpass the random output generator. This class encompasses the stroke of all communicative gestures except beat gestures. Our model is able to predict where a gesture stroke (other than beat) is aligned with the acoustic features we considered. This phase is well-studied in gesture literature as it carries the meaning of the gesture. This phase usually happens at around or slightly before the pitch accent [20]. In our case, we have the F_0, intensity, and the F_0 direction score features as our input. Pitch accents are characterised by these acoustic features, but not solely by them. Using only these three acoustic features may not allow us to fully capture where pitch accents occur.

On the "NotBeatNonStroke" class our model does not do the prediction well. As a recall, this class contains all the gesture phases (e.g., preparation, hold, retraction) except the stroke phase for all gestures but the beats. Indeed, in all our experiments, we never obtain a good alignment on this class. This class is made of different gesture phases that may not correspond to the same prosodic profile.

Our model predicts well the "NoGesture" class. This class predicts when no gesture occurs. We select our samples only when the person is speaking. This class allows us to predict when the person is speaking without gesticulating.

In our 1st ablation study, where we replace the entire speech prosody input with random values and use in on the trained model (table 1, result 3), we observe that it completely fails to predict "Beat" (deletion score of 1) while its performance on "NonBeatStroke" is slightly better than the random output one (alignment score of 0.304 against 0.228). On the other hand, its alignment score on the "NoGesture" class is similar to the random output. This result suggests that "Beat" is tied to speech prosody as reported in the literature [20]. For the "NonBeatStroke" and "NoGesture" the result suggests that the model did learn the distribution of the gesture class. Unlike the random output in the table 1 result 1 where the distribution is given, the model "knows" the distribution of the data only from the learning.

In the sub-experiment where we use the intensity alone (table 1, result 4), we find again that the model completely fails to predict "Beat", which suggests that "Beat" is not tied to intensity alone. However, "NotBeatStroke" and "NoGesture" alignment scores are similar to the performance of the model when it is tested with all input features (table 1, result 2), which suggests that these two features are tied to intensity. Finally, we find that with only F_0 (table 1, result 6), we get alignment scores which are similar to what we get when we use all input features. It suggests that F_0 is tied and is very pertinent to the "Beat", "NonBeatStroke", and "NoGesture" classes. It should be noted, however, that the three features we use, namely F_0, F_0 direction score, and intensity are not necessarily independent to each other, especially in the case of natural human conversation which we deal with. This ablation study is to find which features are more relevant to our task, namely gesture class prediction.

In experiments 4 and 5 where we train the model with one speaker and test it on the other speaker of the same interaction (table 2, result 8 and 9), we find that the models outperform the random output (table 1, result 1), which suggests that some generalisability exists even-though people have different gesturing styles. These results may also be due as participants in a conversation tend to align to each other [21].

Currently, we do the prediction based only on the prosody. Some prior works rely also on prosodic features [7, 17]. Some other prior works do the prediction based only on the text [1, 4]. Based on this state, one interesting and relevant research question is how to combine both modalities. In the future, we intend to go into this direction.

8 DISCLAIMER

As of 30 January 2020 when this work is submitted for WACAI 2020, an almost identical version of this work is also being reviewed for a submission for ICME 2020.

ACKNOWLEDGMENTS

 This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769553. This result dissemination reflects only the authors's views. The European Commission is not responsible for any use that may be made of the information it contains. We also thank Katya Saint-Amand for providing the Gest-IS corpus [24].

REFERENCES

[1] Chaitanya Ahuja and Louis-Philippe Morency. 2019. Language2Pose: Natural Language Grounded Pose Forecasting. In *2019 International Conference on 3D Vision (3DV)*. IEEE, 719–728.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4945–4949.

[4] Kirsten Bergmann and Stefan Kopp. 2009. GNetIc–Using bayesian decision networks for iconic gesture generation. In *International Workshop on Intelligent Virtual Agents*. Springer, 76–89.

[5] D. Bolinger. 1989. *Intonation and its Uses*. Stanford University Press.

[6] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2004. Beat: the behavior expression animation toolkit. In *Life-Like Characters*. Springer, 163–185.

[7] Chung-Cheng Chiu and Stacy Marsella. 2014. Gesture generation with low-dimensional embeddings. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 781–788.

[8] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*. 577–585.
[9] Alice Cravotta, M Grazia Busà, and Pilar Prieto. 2019. Effects of Encouraging the Use of Gestures on Speech. *Journal of Speech, Language, and Hearing Research* 62, 9 (2019), 3204–3219.
[10] Soumia Dermouche and Catherine Pelachaud. 2016. Sequence-based multimodal behavior modeling for social agents. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. ACM, 29–36.
[11] James E Driskell and Paul H Radtke. 2003. The effect of gesture on speech production and comprehension. *Human Factors* 45, 3 (2003), 445–454.
[12] P. Ekman. 1979. About brows: Emotional and conversational signals. In *Human ethology: Claims and limits of a new discipline: contributions to the Colloquium*, M. von Cranach, K. Foppa, W. Lepenies, and D. Ploog (Eds.). Cambridge University Press, Cambridge, England; New-York, 169–248.
[13] Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 1459–1462.
[14] Howard Giles, Anthony Mulac, James J Bradac, and Patricia Johnson. 1987. Speech accommodation theory: The first decade and beyond. *Annals of the International Communication Association* 10, 1 (1987), 13–48.
[15] Jana M Iverson and Susan Goldin-Meadow. 1998. Why people gesture when they speak. *Nature* 396, 6708 (1998), 228.
[16] E. Krahmer and M. Swerts. 2004. More about brows. In *From Brows till Trust: Evaluating Embodied Conversational Agents*, Z. Ruttkay and C. Pelachaud (Eds.). Kluwer.
[17] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. 2019. Analyzing Input and Output Representations for Speech-Driven Gesture Generation. *arXiv preprint arXiv:1903.03369* (2019).
[18] Rivka Levitan and Julia Hirschberg. 2011. Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. In *Twelfth Annual Conference of the International Speech Communication Association*.
[19] Daniel P Loehr. 2012. Temporal, structural, and pragmatic synchrony between intonation and gesture. *Laboratory Phonology* 3, 1 (2012), 71–89.
[20] David McNeill. 1992. *Hand and mind: What gestures reveal about thought*. University of Chicago press.
[21] Laura Menenti, Simon Garrod, and Martin Pickering. 2012. Toward a neural basis of interactive alignment in conversation. *Frontiers in Human Neuroscience* 6 (2012), 185. https://doi.org/10.3389/fnhum.2012.00185
[22] Klim Peshkov, Laurent Prévot, and Roxane Bertrand. 2013. Prosodic phrasing evaluation: measures and tools. *Proceedings of TRASP 2013* (2013).
[23] Brian Ravenet, Chloé Clavel, and Catherine Pelachaud. 2018. Automatic Nonverbal Behavior Generation from Image Schemas. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1667–1674.
[24] Katya Saint-Amand. 2018. Gest-IS: Multi-lingual Corpus of Gesture and Information Structure. *Unpublished Report* (2018).
[25] 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.
[26] Fajrian Yunus, Chloé Clavel, and Catherine Pelachaud. 2019. Gesture Class Prediction by Recurrent Neural Network and Attention Mechanism. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. ACM, 233–235.