A Face-to-Face Neural Conversation Model

Hang Chu[1,2]   Daiqing Li[1]   Sanja Fidler[1,2]
[1]University of Toronto   [2]Vector Institute
{chuhang1122, daiqing, fidler}@cs.toronto.edu

Abstract

Neural networks have recently become good at engaging in dialog. However, current approaches are based solely on verbal text, lacking the richness of a real face-to-face conversation. We propose a neural conversation model that aims to read and generate facial gestures alongside with text. This allows our model to adapt its response based on the "mood" of the conversation. In particular, we introduce an RNN encoder-decoder that exploits the movement of facial muscles, as well as the verbal conversation. The decoder consists of two layers, where the lower layer aims at generating the verbal response and coarse facial expressions, while the second layer fills in the subtle gestures, making the generated output more smooth and natural. We train our neural network by having it "watch" 250 movies. We showcase our joint face-text model in generating more natural conversations through automatic metrics and a human study. We demonstrate an example application with a face-to-face chatting avatar.

1. Introduction

We make conversation everyday. We talk to our family, friends, colleagues, and sometimes we also chat with robots. Several online services employ robot agents to direct customers to the service they are looking for. Question-answering systems like Apple Siri and Amazon Alexa have also become a popular accessory. However, while most of these automatic systems feature a human voice, they are far from acting like human beings. They lack in expressivity, and are typically emotionless.

Language alone can often be ambiguous with respect to the person's mood, unless indicative sentiment words are being used. In real life, people make gestures and read other people's gestures when they communicate. Whether someone is smiling, crying, shouting, or frowning when saying "thank you" can indicate various feelings from gratitude to irony. People also form their response depending on such

Figure 1: Facial gestures convey sentiment information. Words have different meanings with different facial gestures. Saying *"Thank you"* with different gestures could either express gratitude, or irony. Therefore, a different response should be triggered.

context, not only in what they say but also in how they say it. We aim at developing a more natural conversation model that jointly models text and gestures, in order to act and converse in a more natural way.

Recently, neural networks have been shown to be good conversationalists [33, 15]. These typically make use of an RNN encoder which represents the history of the verbal conversation and an RNN decoder that generates a response. [16] built on top of this idea with the aim to personalize the model by adapting the conversation to a particular user. However, all these approaches are based solely on text, lacking the richness of a real face-to-face conversation.

In this paper, we introduce a neural conversation model that reads and generates both a verbal response (text) and facial gestures. We exploit movies as a rich resource of such information. Movies show a variety of social situations with diverse emotions, reactions, and topics of conversation, making them well suited for our task. Movies are also multi-modal, allowing us to exploit both visual as well as dialogue information. However, the data itself is also extremely challenging due to many characters that appear on-screen at any given time, as well as large variance in pose, scale, and recording style.

Our model adopts the encoder-decoder architecture and adds gesture information in both the encoder as well as the decoder. We exploit the FACS representation [8] of ges-

---

demo/data: http://www.cs.toronto.edu/face2face

Figure 2: Example conversations from our MovieChat dataset. Each row shows two examples, left shows query face and text, right shows target face and text. Our dataset has various conversation scenarios, such as simple conversations shown in the first and second rows on the left, as well as more challenging cases shown on the right.

tures, which allows us to effectively encode and synthesize facial gestures. Our decoder is composed of two levels, one generating the verbal response as well as coarse gesture information, and another level that fills in the details, making the generated expressions more natural. We train our model using reinforcement learning that exploits a trained discriminator to provide the reward. We show that our model generates more appropriate responses compared to multiple strong baselines, on a large-scale movie dataset. We further showcase NeuralHank, an expression-enabled 3D chatting avatar driven by our proposed model.

The rest of the paper is organized as follows. Sec. 2 reviews the related work. In Sec. 3 we introduce our dataset to facilitate face-to-face conversation modeling. In Sec. 4 we describe our approach. Sec. 5 provides extensive evaluation and introduces our chit-chatting avatar.

2. Related Work

Dialogue systems have been explored since the 60', with systems like ELIZA [34] and PARRY [5] already capable of engaging in relatively complex conversations. These approaches have mainly been based on hand-coded rules, thus were not able to adapt to users and topics, and usually seemed unnatural. In [20], the authors formulated the problem as statistical machine translation, where the goal was to "translate" the query posts in blogs into a response. This problem setting is typically harder than traditional translation from one language to another, since the space of possible responses is more diverse.

Conversation modeling has recently been gaining interest due to the powerful language models learned by neural networks [33, 15, 16]. [33] was the first to propose a neural conversation model, which exploited the encoder-decoder architecture. An LSTM encoder was used to represent the query sentence while the decoder LSTM generated a response, one word at a time. The model was trained on a large corpus of movie subtitles, by using each sentence as a query and the following sentence as a target during training. Qualitative results showed that meaningful responses were formed for a variety of queries. In parallel, the Skip-Thought model [12, 37] adopted a similar architecture, and was demonstrated to be effective in a variety of NLP tasks

as well as image-based story-telling.

Since neural conversation models typically produce short and more generic sentences, the authors in [15] proposed an improved objective function that encouraged diversity in the generator. In [22], the authors exploited a hierarchical encoder-decoder, where one GRU layer was used to model the history of the conversation at the sentence level, and the second level GRU was responsible for modeling each sentence at the word level. This model was extended in [24] by adding latent variables aiming to capture different topics of conversation, allowing the model to achieve a higher diversity in its response.

An interesting extension was proposed in [16] which aimed at personalizing conversations. The model learned a separate embedding for each person conversationalist, jointly with dialogue. The purpose of the embedding was to bias the decoder when generating the response. This allowed for a more natural human-like chit-chat, where the model was able to adapt to the person it was speaking to.

Most of these works are based solely on language. However, humans often use body gestures as an additional means to convey information in a conversation. An interesting approach was proposed in [14, 13] which aimed at synthesizing body language animations conditioned on speech using a HMM. This approach required motion capture data recorded during several conversation sessions.

Face capture has been a long-studied problem in computer vision, with many sophisticated methods such as [2, 25, 10]. The FirstImpression dataset [3] was collected to facilitate the need of data in gesture recognition. Face synthesis has been widely studied in both vision and graphics communities. [28] proposed a reconstruction algorithm that captures a person's physical appearance and persona behavior. [31] transfers facial gesture from a source video to a target video to achieve realistic reenactment. [29] further transformed audio speech signal into a talking avatar using an RNN-based model.

In our approach, we aim to both encode and generate facial gestures jointly with language, by exploiting a large corpora of movies. Movies feature diverse conversations and interactions, and allow us to use both visual as well as dialogue information.

(a) (b) (c)

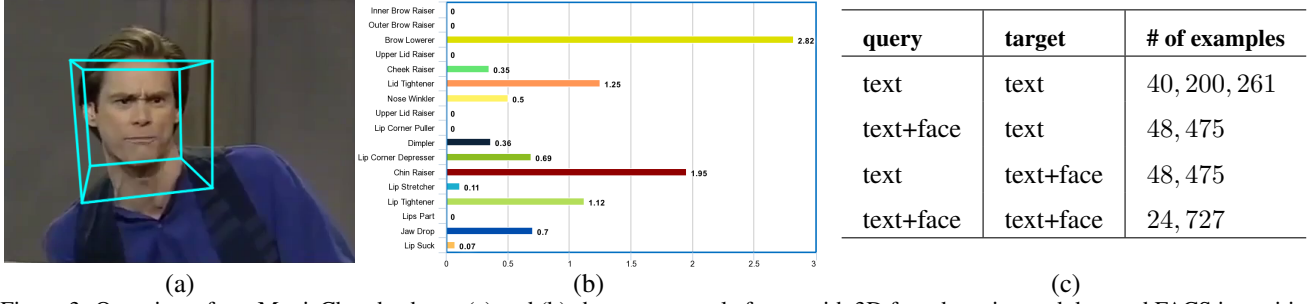| query	target	# of examples
text	text	$40,200,261$
text+face	text	$48,475$
text	text+face	$48,475$
text+face	text+face	$24,727$

Figure 3: Overview of our MovieChat database. (a) and (b) show an example frame with 3D face detection and detected FACS intensities. We obtain detections using the off-the-shelf OpenFace [2] package. (c) shows the scale of our MovieChat database. Our database is by far the largest language-face conversation video dataset.
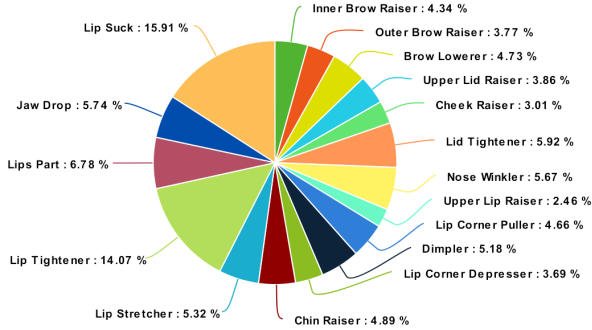

Figure 4: List of gestures recorded in the MovieChat dataset, and percentage of frames where each gesture is dominant.

3. The MovieChat Dataset

Datasets of considerable size are key to successfully training neural networks. In our work, we seek a dataset containing people engaging in diverse conversations, that contains both video as well as transcribed dialogues.

Towards this goal, we build the MovieChat dataset. We take advantage of the large movie collection of MovieQA [30], which contains clips from 250 movies, covering more than half of each movie in duration. To track 3D faces and detect facial gestures, we use the off-the-shelf OpenFace [2] package. Tracking and detection runs in real-time while maintaining good accuracy. This makes processing of such a large volume of video data possible.

However, even the best automatic face detector occasionally fails. Certain recording styles, such as the shaky and free-cam clips, make our processing more challenging. To address these problems and improve the quality of our dataset, we further divide all movies into short, single sentence clips by exploiting the time stamps stored in their subtitles file. We only keep clips where a single face is detected across all of its frames, and discard the rest of the clips. This is to avoid ambiguous dialog-face association when multiple characters appear in a single shot. Additionally, we remove fast-cut clips where the speaker's face is not fully visible throughout the clip. Finally, we also remove clips in which tracks are extremely shaky, which often suggests tracking failure. We observe significant quality improve-
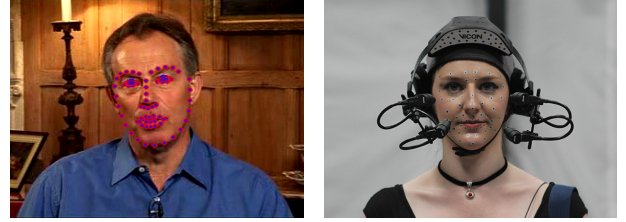

Figure 5: Facial Landmark (FL) systems. Left shows an example of "in the wild" landmarks [6, 36, 2], which fails to capture subtle gesture information. Right shows invasive motion capture landmarks [1].

ment after these filtering steps, with only rare failure cases.

We build our final dataset with the remaining clips. We record image frames, time stamps, 3D face poses, facial gestures, and transcribed dialogues. Fig. 3 shows an example, and provides statistics summarizing our dataset. Fig. 4 shows the recorded gestures and their statistics in our MovieChat data.

4. Face-to-Face Neural Conversation Model

We first explain our facial gestures representation using Facial Action Coding System (FACS) [8]. We then describe our proposed model.

4.1. FACS Gesture Representation

Various approaches are available for representing gesture numerically, e.g. Six Universal Expressions (SUE) [4], Facial Landmarks (FL) [6, 36], and FACS [8].

SUE [4] categorizes gesture into six emotions: anger, disgust, fear, happiness, sadness and surprise. It is effective in encoding high-level emotion, but it is overly abstract to describe detailed gestures. Each emotion involves a combination of up to 6 muscle movements, making it difficult for face synthesis and animation.

FL [6, 36] represents gesture using landmark points. Typically, 68 points are used to track corner-edge keypoint positions of the face. Compared to SUE, FL carries more details. However, FL has two disadvantages. First, FL does not contain complete gesture information. The cheek
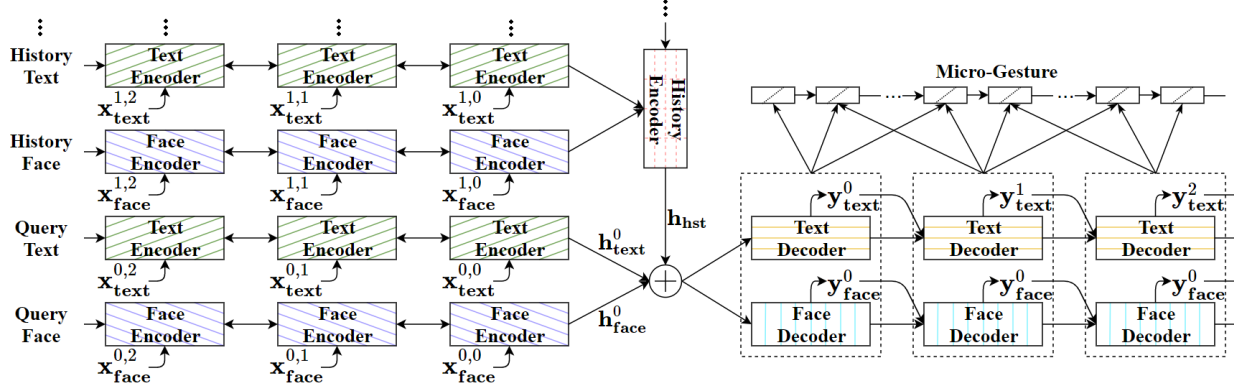
Figure 6: Our face-to-face conversation model. Our model consists of 6 RNNs shown in different colors. First, the text-face sequences of query and conversation history are encoded by text and face encoders (only one history sentence is depicted). History sentence encodings are further encoded by the history encoder. Next, encodings are added to form the context vector, which the text and face decoders are conditioned on. Finally, we generate frame-level, micro-gesture animation controls based on the word decodings.

and forehead regions, which are texture-less but contain many muscles, are usually missing. Second, FL is anatomically redundant. 5 landmarks are used to outline one brow, while its underlying motion is lower dimensional that involves 2 muscle intensity values. Therefore, FL is less desirable for our task. It should be noted there are variations of FL that places landmarks across all muscles evenly. They are widely used in motion capture, e.g. Cara [1] in Figure 5. This FL system requires visible marks on the character face, thus making large scale data collection difficult.

We adopt FACS [8] in this paper. Particularly, we use 18 action unit each controls a face muscle, as well as 3 dimensions to represent the 3D head pose. Compared to the SUE and FL, FACS not only captures subtle detail gestures, but also produces highly interpretable gesture representation which makes animation simple and straight-forward. We detect FACS from images using the off-the-shelf Open-Face software [2].

4.2. Face-to-Face Conversation Model

Following previous work on conversation modeling [24, 27], we adopt the RNN encoder-decoder architecture, but adapt it to our face-to-face conversation task. Our proposed model consists of 6 RNN modules that capture and generate information across different modalities and resolutions. Fig. 6 provides an overview of the model. Overall, our model is an encoder-decoder framework that is trained with RL and GAN.

Notation. Our algorithm takes a series of paired text and gesture sequences as input. These represent the query sequence as well as a recent conversation history of N sequences. Here, let $\mathbf{x}^0$ denote the current query sequence, while $\mathbf{x}^n$ indexes the n-th sequence of the current history. We will use subscripts **text** and **face** to denote the data from the two modalities.

Sentence encoders. We synchronize text and gestures at the word level. Let $\mathbf{x}_{\mathbf{text}}^{n,\ell}$ represent a one-hot encoding of the ℓ-th word in the n-th sentence. To keep the representation consistent and to simplify the multi-dimensional gesture data, we set $\mathbf{x}_{\mathbf{face}}^{n,\ell}$ as a similar one-hot encoding of the closest gesture template. Gesture templates are obtained via k-means ($k = 200$) clustering of all gestures in the training set. We define our sentence-level encoders as bidirectional RNNs, i.e.

$$\mathbf{h}_{\mathbf{text}}^n = \text{BiLSTM}(\{\mathbf{x}_{\mathbf{text}}^{n,\ell}\}_\ell)$$
$$\mathbf{h}_{\mathbf{face}}^n = \text{BiLSTM}(\{\mathbf{x}_{\mathbf{face}}^{n,\ell}\}_\ell) \quad (1)$$

where the BiLSTM computes the forward and backward sentence encodings $\vec{h}^n$ and $\overleftarrow{h}^n$, respectively, concatenates them, and applies a linear layer on top.

History encoder. To model the context of the conversation, we take a history of N sequences (excluding query), and add another bidirectional RNN over the encoded text and gesture sequences:

$$\mathbf{h}_{\mathbf{hst}} = \text{BiLSTM}(\{\mathbf{h}_{\mathbf{text}}^n \oplus \mathbf{h}_{\mathbf{face}}^n\}_n) \quad (2)$$

where $\oplus$ denotes vector concatenation.

Sentence decoders. We use two decoders that generate the target text and gesture sequences. The output follows the same one-hot encoding as query and history sentences. We condition the target sentence generation on the joint text-face-history context vector, which is obtained by the summation of the query text encoding, query face encoding, and history encoding. Concretely,

$$\mathbf{h}_{\mathbf{enc}} = \mathbf{h}_{\mathbf{text}}^0 + \mathbf{h}_{\mathbf{face}}^0 + \mathbf{h}_{\mathbf{hst}} \quad (3)$$

where $\mathbf{h}_{\mathbf{enc}}$ is the final encoding that we condition our generation decoders on. We use two independent single-

directional RNN decoders, i.e.

$$\mathbf{h}_{\mathbf{dec}}^{\ell} = \text{LSTM}(\mathbf{h}_{\mathbf{dec}}^{\ell-1} \mid \mathbf{h}_{\mathbf{enc}}, \mathbf{y}^{\ell-1})$$
$$\mathbf{y}^{\ell} = \underset{\mathbf{y}}{\arg\max}\, p(\mathbf{y} \mid \mathbf{h}_{\mathbf{dec}}^{\ell}), \qquad (4)$$

where $\mathbf{y}^{\ell}$ is either ℓ-th output word or gesture, and p computes a softmax over a linear layer on top of the hidden state.

Micro-gesture generator. The output of our gesture decoder is a gesture template (cluster). We observe that although templates are sufficient for representing semantics, they are insufficient for synthesizing vivid, high framerate animations. We use the micro-gesture module to fill in this resolution gap, effectively interpolating between the consecutive discrete gestures, and adding relevant variations to the final gestures. We define this module as a frame-level RNN. For the t-th frame, we synthesize its micro-gesture based on the two most adjacent words, i.e.

$$\mathbf{h}_{\mathbf{micro}}^{t} = \text{LSTM}(\mathbf{h}_{\mathbf{micro}}^{t-1} \mid \mathbf{y}_{\mathbf{text}}^{t-\delta} \oplus \mathbf{y}_{\mathbf{face}}^{t-\delta}, \mathbf{y}_{\mathbf{text}}^{t+\delta} \oplus \mathbf{y}_{\mathbf{face}}^{t+\delta})$$
$$(5)$$

where δ denotes the interpolation interval ($t - \delta$ indexes the previous word, and $t + \delta$ the next one). We obtain the frame-level gesture by linearly regressing $\mathbf{h}_{\mathbf{micro}}^{t}$ to each individual gesture dimension. These gesture values directly control the muscle intensities that drive a 3D avatar.

Policy gradient optimization. As typical, we train the model with the cross-entropy loss. However, the default cross-entropy training suffers from exposure bias, as the model is only exposed to ground truth samples during training. For our decoder networks, we alleviate this problem by optimizing directly for the desired metrics using policy gradient optimization. In this setting, the *policy* takes the form of a decoder RNN, and an *action* is a sentence sampled from the policy denoted by $\tilde{\mathbf{y}}$ (for either **text** or **face**). Our goal is to expose the model to more samples of $\tilde{\mathbf{y}}$, and discover a policy to achieve higher *reward* under the metric of choice evaluated at the end of the sequence (e.g. $F1$-score). This is denoted by $\mathbf{R}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}_{\mathbf{gt}})$, where $\hat{\mathbf{y}}_{\mathbf{gt}}$ is the ground truth sequence. The policy gradient (using a single sample) is computed as follows ($\mathbf{h}$ short for $\mathbf{h}_{\mathbf{enc}}$):

$$\nabla J_{\mathbf{pg}}(\theta) = [\mathbf{R}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}_{\mathbf{gt}}) - \mathbf{b}]\nabla_{\theta} \log p_{\theta}(\tilde{\mathbf{y}}|\mathbf{h}) \qquad (6)$$

where $J_{\mathbf{pg}}$ is the objective function, and $\mathbf{b}$ is the baseline to help reduce the variance of the gradients. We follow [19], and compute the baseline by greedy decoding conditioned on the same $\mathbf{h}$. Computing the baseline in this way mimics the inference strategy at test time, thus obtaining positive gradients whenever the sampled sequence scores higher than the current greedy sequence.

For computing the reward, we will exploit standard metrics such as $F1$-score, as well as learned reward functions as explained next.

Reward via an adversarial discriminator. Conversation models suffers from dull responses [17], while diverse dialogues are preferred in practical scenarios. We address this problem by using a sequence GAN [35], following the idea of [7] for captioning. The *generator* is our decoder network, while the *discriminator* is another network that distinguishes whether the resulting sequence is machine-generated (fake) or real. We can formulate this as a minmax problem, i.e.

$$\min_{\theta} \max_{\eta} J_{\mathbf{gan}}(p_{\theta}, \mathbf{D}_{\eta}) \qquad (7)$$

where $\mathbf{D}$ is the discriminator producing probability value $[0, 1]$, η being its parameters. Specifically, the GAN objective is defined as

$$J_{\mathbf{gan}} = \mathbb{E}_{\hat{\mathbf{y}}_{\mathbf{gt}}}[\log \mathbf{D}_{\eta}(\mathbf{h}, \hat{\mathbf{y}}_{\mathbf{gt}})] + \mathbb{E}_{\tilde{\mathbf{y}} \sim p_{\theta}}[\log(1 - \mathbf{D}_{\eta}(\mathbf{h}, \tilde{\mathbf{y}}))]$$
$$(8)$$

The discriminator is conditioned on $\mathbf{h}$, trying to both learn what good sequences are and their consistency with the query sequences. When training the discriminator we follow [7], to also add mis-matched query-sequence pairs in the discriminator training step to improve the generation's semantic relevance. We use $\mathbf{D}$ to compute the reward for policy-gradient optimization.

Implementation details. We substantiate all LSTMs with a 1024-d LSTM cell [9] on top of a 512-d embedding layer, followed by a linear layer with hyperbolic tangent non-linearity to compute the final encoding. Our GAN discriminator is implemented as a 3-layer, 512-d MLP that takes sentence encoding and context vectors as inputs.

Nested hierarchical neural networks are difficult to train from scratch in an end-to-end fashion. We observe the same for our model. To train our model successfully, we first pre-train our text and face encoders on single sequence corpora. Then we freeze the encoder modules and use them to generate setence-level encodings, which is used to pre-train our history model. Similarly, we pre-train decoders to make them familiar with the context. After all modules are pre-trained, we jointly finetune the entire network.

To train our decoder, we adopt the MIXER [18] strategy. We initialize the policy network via MLE. Then we gradually anneal MLE steps and blend in RL steps temporally. We keep this process until all time steps are replaced by RL. To train our discriminator, we mix the same ratio of sampled sequences, ground truth sequences, and mis-matched ground truth. In PG training, we observe that balanced positive and negative rewards are also helpful for the training process. In our case, we randomly discard samples until average reward is equal to the baseline reward. We use clipped gradient descent in our pre-training steps, and Adam [11] in all other training steps.

We pre-train our micro-gesture module on the FirstImpression dataset [3], which contains close-up talking videos that allows high precision tracking of micro-gesture. To

	perp.	beam=1			beam=3			beam=5		
		pre. %	rec. %	F1	pre. %	rec. %	F1	pre. %	rec. %	F1
Text [12, 27]	32.53	23.18	15.58	17.12	25.00	17.13	18.62	24.70	16.91	18.34
Text+RandFace	32.65	22.92	15.99	17.27	24.74	17.32	18.57	24.71	17.82	18.84
Text+Face	30.17	24.25	17.52	18.69	24.78	18.60	19.40	24.34	18.74	19.37
History-RNN [23]	31.15	23.99	19.46	19.59	23.79	20.11	19.67	23.37	20.50	19.68
History-FC	30.39	24.49	19.61	19.88	24.38	20.50	20.14	23.70	20.45	19.91
Ours-MLE	30.08	25.16	19.72	20.17	24.50	20.32	20.11	23.75	20.47	19.89
Ours-F1	31.91	25.16	20.24	20.42	24.48	20.26	20.02	24.06	20.33	19.96
Ours-GAN	31.60	25.23	20.19	20.44	24.56	20.31	20.08	24.11	20.38	19.97

Table 1: The mind-reading text results on text. Second column lists word *perplexity* (lower the better). Third to last columns list unigram *precision*, *recall*, and $F1$-*score* (higher the better) across different beam search size. For each column, we mark the **best** and **second best** results in red and blue color. We underscore the **overall best** result across all methods and all beam sizes.

	perp.	beam=1			beam=3			beam=5		
		pre. %	rec. %	F1	pre. %	rec. %	F1	pre. %	rec. %	F1
Face [27]	18.98	26.48	9.83	12.96	22.41	8.18	10.82	20.74	7.55	10.02
Face+RandText	18.94	26.63	10.01	13.15	22.54	8.15	10.82	20.20	7.43	9.80
Face+Text	17.20	29.46	10.89	14.41	25.84	9.46	12.57	24.82	9.14	12.15
History-RNN [23]	20.30	20.84	7.33	9.81	20.84	7.33	9.81	20.84	7.33	9.81
History-FC	20.26	20.86	7.35	9.83	20.81	7.33	9.80	20.84	7.33	9.81
Ours-MLE	17.18	35.81	13.74	18.07	30.44	11.43	15.10	28.25	10.58	13.49
Ours-F1	17.20	36.17	13.92	18.28	30.42	11.43	15.09	28.30	10.63	14.06
Ours-GAN	17.19	36.06	13.85	18.20	30.43	11.38	15.05	28.12	10.52	13.92

Table 2: The mind-reading test results on gesture. Legend same as Table 1.

synchronize words and gestures at the frame level, we perform speech recognition with Bluemix, and force the alignment with existing transcripts with the Smith-Waterman algorithm [26]. We reduce the jittering effect of our final generation using an online Savitzky-Golay filter [21].

5. Experiments

We evaluate our model through automatic metrics with a "mind-reading" test, and through a human study with a NeuralHank chatting avatar controlled by our model. We randomly split MovieChat into 4:1:1 train-val-test, and keep the split in all experiments.

5.1. The Mind-Reading Test

In the first experiment, we evaluate how well the model's generation matches with the ground truth target text and gesture sequences. This reflects the model's ability to produce appropriate, human-like responses. We note that this is an extremely challenging task, particularly for producing fair evaluation. Due to the multi-modal nature of chit-chat conversations, there exists many plausible responses to the same query, and the ground truth only represents one mode among many. Therefore, we refer to this evaluation as the mind-reading test.

We evaluate the model at both word and sentence level. At the word level, we evaluate the *perplexity*, i.e. the likelihood of generating the correct next target word, given the source and correct previous words in the target sequence. This measures coherence of the textual and facial language models. At the sentence level, we evaluate *precision*, *recall*, and $F1$-*score* between the words in generated sentences and ground truth.

We compare our approach to five baselines: **1.)** Text(Face): The classic Seq2Seq [12, 27] method that uses single modality only (either text or face) and only the query sequence (no history); **2.)** Text+Face(Face+Text): Two encoders for both text and face query senquences without history; **3.)** Text+RandFace(Face+RandText): Same model as previous but trained with randomized face(text) query sentences; **4.)** History-RNN: Modeling conversation history as well as query text(face) using a hierarchical RNN, which is similar to to [23]; **5.)** History-FC: Same as previous but directly connects history sentences to the decoder with fully connected layers. This exploits the potential in conversation history, at the cost of inflexibility to history length N and significantly heavier models. For our model, we compare Ours-MLE, Ours-F1 ($F1$-score as reward), and Ours-GAN. We use beam search with varying sizes for all methods.

From Table 1, 2, it can be seen that our method achieves the best performance. Due to non-overlapping conversation scenes between data splits, the improvement of our methods is meaningful and generalizable. Therefore, our experiments quantitatively prove the common intuition that seeing the face makes understanding the conversation easier and better, backing up the main argument of this paper. Our base model can be further improved using reinforcement and adversarial training. GANs do not achieve better automatic metric score than directly setting metric reward for PG. However, GANs are able to generate more diverse and interesting responses, as we will later show in Sec. 5.2. This finding is in accordance with image captioning [7].
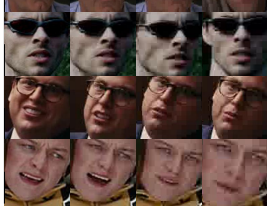
source text	source face sequence	true target text	text only [12, 27]	text+face
we went to the hickory stick, we had a drink, two drinks.		and then? and then i went home alone.	we drank a bottle of champagne.	and then i went to bed.
she doesn't know where he is.		i don't know where he is.	i'm sorry.	i don't know where she is.
and he sleeps only one hour a night.		he's a great man.	he sleeps in the same bed.	he's a good man.
a night that marked the opening of a new chapter.		in world history.	for the future.	in the history of the world.
i hope you're not a hothead like sonny.		he's a good kid.	he's got a lot of something.	he's a good kid.
i guess they was worried they wouldn't find a vein in my arm.		what's that number?	what's that?	i don't think so.
oh, he's so cute.		oh, my god.	oh, my god	he's so cute.
can you hear me? i'm still here. scott. stop.		i'm here. scott. stop.	i'm sorry.	what the f*** are you doing here?
so i don't really remember, yeah.		yeah, right. stupid.	yeah, yeah, yeah.	well, you know what? i'm sorry.
i can't feel my legs.		i can't feel my legs.	and i can't breathe.	it's too much.

Figure 7: Success and failure cases of using face along with text. Top five rows show successful examples where adding facial gesture information produces sentences closer to the ground truth. Bottom five rows show failure modes, including face detection failure in the sixth row, and detecting another face that does not belong to the speaker in the seventh row.

The role of gestures. In Table 1, Text+Face outperforms only Text, indicating that gesture information helps text understanding. Text+RandFace does not achieve significant improvement despite a slightly better $F1$-score. This verifies the improvement of Text+Face is indeed due to the effectiveness of gesture data, instead of the additional encoder stream. This justifies our argument that gesture information is useful for text understanding. Our method outperforms both History-RNN and History-FC, showing the compatibility between gesture and history information.

The role of text. Similarly, from Table 2 it can be seen that in understanding and generating face gestures, text information is helpful. This confirms the mutual benefit between both text and gesture information.

The role of history. In both text and face, History-FC outperforms History-RNN. This indicates that there is still room for further improvement for better history encoders. However, History-RNN remains the preferable option, for its smaller model size and flexibility to varying history length, which is important in practical scenarios. Interestingly, history method outperforms Text+Face in text mind reading, indicating that multiple sentences of text history is more helpful than a query face sequence. It is the opposite in gesture mind reading, indicating that when guessing facial gestures, seeing the source face and react accordingly can be more helpful than knowing a series of text-only history sentences. This can be also partially due to the nature of movie data, where both source and target can be conveyed by the same character.

5.2. The NeuralHank Chatbot

Here, we test how our model's performance in the eyes of real human users. We create a virtual chatbot named NeuralHank, that is controlled by our model. This experiment aims to demonstrate the more realistic potential of our model and provides a pilot study towards new applications, e.g. AI assistants and gaming/HCI.

In NeuralHank, we ensemble a series of off-the-shelf packages to convert our model's generation into a real talking avatar. We use Microsoft Speech API to render text as audio, while also keeping record of viseme time tags. We then render FACS gestures using Maya's Facial Animation Toolset, with its default character Hank. For Hank's lip motion, we simply use the viseme event record with tangent interpolation over time. In training, we continue for a few more epochs after the early stopping point until training loss is below a certain threshold. We found this makes the model's generation more particular, which is helpful for building a lively avatar.

We compare three methods: **1.)** *noMicro-noGAN* that uses beam search without GANs, and only word-level face decoder without micro-gesture RNN; **2.)** *Micro-noGAN* that uses the micro-gesture RNN, and sampling without GANs; **3.)** *Ours* as our full model.

Figure 8: NeuralHank examples. `Q` is the query text. `A1`, `A2`, and `A3` are generated by *noMicro-noGAN*, Micro-noGAN, and *Ours*, respectively. We also show one animation frame generated by our method. First two rows show that our GAN-based model generates more diverse and interesting responses. Last row shows failure cases where our method generates confusing responses. Please refer to our project page for animation videos, more examples, and chatting with Hank live through a webcam.

	text %	face %	overall %	
noMicro-noGAN	48.8	39.0	46.2	
Micro-noGAN	**51.2**	**61.0**	**53.8**	
noMicro-noGAN	44.8	35.3	42.4	*pairwise*
Ours	**55.2**	**64.7**	**57.6**	
Micro-noGAN	46.1	48.8	46.7	
Ours	**53.9**	**51.2**	**53.3**	
noMicro-noGAN	31.5	25.0	29.8	
Micro-noGAN	32.5	36.8	33.5	*accumu.*
Ours	**36.0**	**38.2**	**36.6**	

Table 3: AMT user study on interestingness and naturalness. The evaluation is conducted in form of pairwise comparison. We further accumulate number of votes for different methods.

We conduct a human study via Amazon Mechanical Turk, by asking participants to rate different methods' responses on the same query. We ask participants to choose the more interesting and natural response, in terms of text, gestures, and overall. For query subjectivity, we randomly choose 65 query sentences from our held-out test set and run all methods using them as inputs. Most participants are not well-trained experts. It is important to make our study easy to follow. To achieve this, we only display a pair of different methods' generations in random order, instead of showing all three together. We also intentionally set query text as the most important information, and set query gesture and history as zero. This makes our task easy to understand, while not affecting the fairness of comparison because the methods only differ on the decoder side.

We request 10 Turkers for each sample. This results in

5850 answers from 37 unique participants. We further use exam questions to filter out the noisy participant responses. The questions are verified samples where one answer is obviously better, e.g. a spot on, grammar error free, and fun sentence, versus a simple and boring yes/no answer.

It can be seen from Table 3 that micro-gesture significantly improves gesture quality. Our full model with adversarial training achieves the best user rating from all three perspectives. Compared to no-GAN methods that tend to produce universally correct but less interesting responses, GAN methods produces generally more diverse and interesting responses. However, GAN methods also suffer from occasional confusing or offensive responses.

Fig. 8 shows generated samples. We only show one key generated facial gesture per example. Our project page contains videos which better reflect the quality of generations.

6. Conclusion

We proposed a face-to-face neural conversation model, an encoder-decoder neural architecture trained with RL and GAN. Our approach used both textual and facial information to generate more appropriate responses for the conversation. We trained our model by exploiting rich video data in form of movies. We evaluated our model through a mind-reading test as well as a virtual chatting avatar. In the future, we aim to learn body controllers as well, model the personalities of the conversation participants, as well as capture more high-level semantics of the situation [32].

References

[1] https://www.vicon.com/products/camera-systems/cara-1. 3, 4

[2] T. Baltrusaitis, P. Robinson, and L.-P. Morency. Openface: an open source facial behavior analysis toolkit. In *WACV*, pages 1–10, 2016. 2, 3, 4

[3] J.-I. Biel and D. Gatica-Perez. The youtube lens: Crowd-sourced personality impressions and audiovisual analysis of vlogs. *IEEE Trans. on Multimedia*, 15(1):41–55, 2013. 2, 5

[4] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *IJCV*, 25(1):23–48, 1997. 3

[5] K. M. Colby. Modeling a paranoid mind. *Behavioral and Brain Sciences*, 4:515–534, 1981. 2

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001. 3

[7] B. Dai, D. Lin, R. Urtasun, and S. Fidler. Towards diverse and natural image descriptions via a conditional gan. In *ICCV*, 2017. 5, 6

[8] P. Ekman, W. V. Freisen, and S. Ancoli. Facial signs of emotional experience. *Journal of personality and social psychology*, 39(6):1125, 1980. 1, 3, 4

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[10] L. Hu, S. Saito, L. Wei, K. Nagano, J. Seo, J. Fursund, I. Sadeghi, C. Sun, Y.-C. Chen, and H. Li. Avatar digitization from a single image for real-time rendering. In *SIGGRAPH Asia*, 2017. 2

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *NIPS*, 2015. 2, 6, 7

[13] S. Levine, P. Krahenbuhl, S. Thrun, and V. Koltun. Gesture controllers. *ACM Trans. on Graphics*, 29(4), 2010. 2

[14] S. Levine, C. Theobalt, and V. Koltun. Real-time prosody-driven synthesis of body language. *ACM Trans. on Graphics*, 28(5), 2009. 2

[15] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *arXiv:1510.03055*, 2015. 1, 2

[16] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan. A persona-based neural conversation model. In *arXiv:1603.06155*, 2016. 1, 2

[17] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv:1606.01541*, 2016. 5

[18] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv:1511.06732*, 2015. 5

[19] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *CVPR*, 2017. 5

[20] A. Ritter, C. Cherry, and W. B. Dolan. Data-driven response generation in social media. In *EMNLP*, 2011. 2

[21] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 6

[22] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *arXiv:1507.04808*, 2015. 2

[23] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv:1507.04808*, 2015. 6

[24] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *arXiv:1605.06069*, 2016. 2, 4

[25] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 2

[26] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. 6

[27] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 4, 6, 7

[28] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. What makes tom hanks look like tom hanks. In *ICCV*, 2015. 2

[29] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. In *SIGGRAPH*, 2017. 2

[30] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 3

[31] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niesner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 2

[32] P. Vicol, M. Tapaswi, L. Castrejon, and S. Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *CVPR*, 2018. 8

[33] O. Vinyals and Q. Le. A neural conversational model. In *arXiv:1506.05869*, 2015. 1, 2

[34] J. Weizenbaum. Eliza, a computer program for the study of natural language communication between man and machine. *ACM*, 9(1):36–45, 1966. 2

[35] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017. 5

[36] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 3

[37] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *ICCV*, 2015. 2