MojiTalk: Generating Emotional Responses at Scale

Xianda Zhou
Dept. of Computer Science and Technology
Tsinghua University
Beijing, 100084 China
zhou-xd13@mails.tsinghua.edu.cn

William Yang Wang
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
william@cs.ucsb.edu

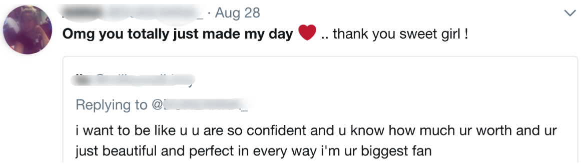
Abstract

Generating emotional language is a key step towards building empathetic natural language processing agents. However, a major challenge for this line of research is the lack of large-scale labeled training data, and previous studies are limited to only small sets of human annotated sentiment labels. Additionally, explicitly controlling the emotion and sentiment of generated text is also difficult. In this paper, we take a more radical approach: we exploit the idea of leveraging Twitter data that are naturally labeled with emojis.

More specifically, we collect a large corpus of Twitter conversations that include emojis in the response, and assume the emojis convey the underlying emotions of the sentence. We then introduce a reinforced conditional variational encoder approach to train a deep generative model on these conversations, which allows us to use emojis to control the emotion of the generated text. Experimentally, we show in our quantitative and qualitative analyses that the proposed models can successfully generate high-quality abstractive conversation responses in accordance with designated emotions.

1 Introduction

A critical research problem for artificial intelligence is to design intelligent agents that can perceive and generate human emotions. In the past decade, there has been significant progress in sentiment analysis (Pang et al., 2002, 2008; Liu, 2012) and natural language understanding—e.g., classifying the sentiment of online reviews. To build empathetic conversational agents, machines must also have the ability of learning to generate emotional sentences.



[redacted] · Aug 28
Omg you totally just made my day ❤️ .. thank you sweet girl !

[redacted]
Replying to @[redacted]
i want to be like u u are so confident and u know how much ur worth and ur just beautiful and perfect in every way i'm ur biggest fan

Figure 1: An example Twitter conversation with emoji in the response (top). We collected a large amount of these conversations, and trained a reinforced conditional variational autoencoder model to automatically generate abstractive emotional responses given any emoji.

One of the major challenges is the lack of large-scale, manually labeled emotional text datasets. Due to the cost and complexity of manual annotation, prior research studies primarily focus on small-sized labeled datasets (Pang et al., 2002; Maas et al., 2011; Socher et al., 2013), which are not ideal for training deep learning models with large amount of parameters.

There do exist a handful of large-scale, emotional corpora in the area of emotion analysis (Go et al., 2016) and a recent dialog dataset with sentiment labels (Li et al., 2017b). However, all of them are condemned to a traditional, small set of human-defined labels, for example, 'happiness,' 'sadness,' 'anger,' etc. or simply binary 'positive' and 'negative.' Such coarse-grained classification makes it difficult to capture the nuances of human emotion.

To circumvent the flaws of human annotation, we propose the use of naturally occurring emoji-rich Twitter data, and extract Twitter conversations with emojis in the response. Our assumption is that the emoji chosen by the user in the response, can be seen as a natural label for the emotion of the response. Using a large collection of Twitter conversations, we then train a conditional generative model to automatically generate the emotional responses. Figure 1 shows an example. We use an attention based sequence-to-sequence model (Sutskever et al., 2014) as a neural baseline to generate abstractive responses.

To generate emotion responses in dialogues, another technical challenge is to control the target emotion labels, as well as to generate the sentences in an abstractive fashion. In contrast to existing work (Huang et al., 2017) that uses information retrieval to generate emotional responses, the research question we are pursuing in this paper, is to design novel techniques that can generate abstractive responses of any given arbitrary emotions, without having human annotators to label a huge amount of training data.

To control the target emotion of the response, we assemble several encoder-decoder generation models, including an standard attention-based Seq2seq model as the base model, and a more sophisticated CVAE model (Kingma and Welling, 2013; Sohn et al., 2015) as VAE is recently found convenient in dialogue generation (Zhao et al., 2017).

We train an emoji text classifier (Felbo et al., 2017) to evaluate the performance of emotion accuracy. To explicitly improve the performance, we then experiment with several extensions to the CVAE model, including a hybrid objective with policy gradient. Additionally, we also conduct a human evaluation to assess the quality of the generated emotional text.

Results suggest that our method is capable of generating state-of-the-art emotional text at scale. Our main contributions are three-hold:

- We provide a publicly available, large-scale dataset of Twitter conversation-pairs naturally labeled with emojis.
- We are the first to use naturally labeled emojis for conducting large-scale emotional response generation for dialogue.
- We apply several state-of-the-art generative models to train an emotional response generation system, and analysis confirms that our models deliver strong performance.

In the next section, we outline related work on sentiment analysis and emoji on Twitter data, as well as neural generative models. Then, we will introduce our new emotional research dataset and formalize the task. Next, we will describe the neural models we applied for the task. Finally, we will show automatic evaluation and human evaluation results, and some generated examples. Experiment details can be found in supplementary materials.

2 Related Work

In natural language processing, sentiment analysis (Pang et al., 2002) is an area that involves designing algorithms for understanding and generating emotional text. Our work is aligned to some recent studies on using emoji-rich Twitter data for sentiment classification. Eisner et al. (2016) proposes a method for training emoji embedding EMOJI2VEC, and combined with WORD2VEC (Mikolov et al., 2013), they apply the embeddings for sentiment classification. DEEPMOJI (Felbo et al., 2017) is closely related to our study: It makes use of a large, naturally labeled Twitter emoji dataset, and train an attentive bi-directional long-short term memory network (Hochreiter and Schmidhuber, 1997) model for sentiment analysis. Instead of building a sentiment classifier, our work focuses on generating emotional responses, given the context and the target emoji.

Our work is also in line with recent progress of the application of Variational Autoencoder (VAE) (Kingma and Welling, 2013) in dialogue generation. advances of deep generative models. VAE (Kingma and Welling, 2013) encodes data in a probability distribution, and then samples from the distribution to generate examples. However, the original frameworks do not support the possibility of generating text conditioning on a certain label. Recently, conditional VAE (CVAE) (Sohn et al., 2015; Larsen et al., 2015) was proposed to incorporate conditioning option in the generative process. Recent research in dialogue generation shows that language generated by VAE models enjoy significant greater diversity than traditional Seq2seq models (Zhao et al., 2017), which is a preferable property toward building a true-to-life dialogue agents.

In dialogue research, our work aligns with recent advances of sequence-to-sequence models (Sutskever et al., 2014) using long-short term memory networks (Hochreiter and Schmidhuber, 1997). We use this model as a baseline, but its vanilla version cannot control the target emotion of the generated text. Li et al. (2016) use a reinforcement learning algorithm to improve the vanilla sequence-to-sequence model for non-task-oriented dialog systems, but their reinforced and

0	😂	175,282	9,218	16	😎	9,033	472	32	😌	5,289	269	48	👊	2,619	152
1	😭	36,513	1,966	17	🙏	8,970	485	33	😋	4,873	241	49	😴	2,422	110
2	😊	28,965	1,482	18	😏	8,830	468	34	😐	4,788	238	50	😣	2,224	108
3	😉	23,796	1,222	19	😒	7,914	471	35	💯	4,479	259	51	😓	2,185	108
4	👍	18,844	988	20	😢	7,899	442	36	💙	4,414	209	52	🙊	1,622	76
5	😘	16,093	841	21	😔	7,885	408	37	😞	4,323	208	53	♥	1,474	60
6	😩	16,091	918	22	💀	7,729	415	38	👏	4,082	205	54	😷	1,326	77
7	😁	14,765	798	23	💖	6,771	330	39	😑	3,975	230	55	😖	1,193	65
8	😍	14,318	728	24	😳	6,608	331	40	💪	3,855	211	56	😠	1,048	43
9	😅	13,380	741	25	😄	6,425	344	41	😫	3,771	202	57	🙅	663	35
10	💕	13,169	718	26	🙌	6,287	338	42	😡	3,659	182	58	✋	605	22
11	👀	13,084	657	27	🙈	6,241	317	43	😪	3,656	207	59	💔	392	31
12	❤	12,472	675	28	💜	6,036	338	44	✌	3,074	162	60	💟	240	10
13	☺	10,397	530	29	😬	5,743	288	45	✨	2,933	139	61	🔫	230	13
14	👌	9,615	489	30	😕	5,590	259	46	💁	2,925	163	62	🎶	144	10
15	😜	9,078	468	31	😤	5,322	302	47	😈	2,811	158	63	🎧	120	10

Figure 2: This is a table showing all 64 emoji labels, and number of conversations labeled by each emoji. Items are in the format of 'emoji No. / image of emoji / conversation number in train set / conversation number in test set'.

its follow-up adversarial models (Li et al., 2017a) also do not model emotions or conditional labels. Zhao et al. (2017) recently introduced conditional VAE for dialog modeling, but they did no model emotions in the conversations, and no reinforcement learning was considered in this model. Hierarchical Recurrent Encoder-Decoder (HRED) (Sordoni et al., 2015) is very similar to the work of (Li et al., 2016) and its latent variable extension (Serban et al., 2017) further improves the performance. Both models cannot explicitly condition on turn-based labels.

3 Dataset

Social media contains large amount of conversations, and people use emojis extensively in their posts. However, not all emojis are used to express emotion and frequency of emojis are unevenly distributed. Inspired by DeepMoji (Felbo et al., 2017), we use 64 common emojis as labels (see Figure 2), and collect a large corpus of Twitter conversations containing those emojis.

3.1 Rules for Data Collection

We crawled conversation pairs on Twitter from 12th to 14th of August, 2017. Responses must include at least one of the 64 emoji labels. Emojis with only tone difference are considered the same emoji. For both original tweets and responses, only English tweets without multimedia contents (such as URL, image or video) are allowed, since we assume that those contents are as important as text itself for machine to understand the conversation.

3.2 Data preprocessing

During data preprocessing, all mentions and hashtags are removed, and punctuations and emojis are separated if they are adjacent to words. Words with digits are all treated as the same special symbol.

In some cases, users use emojis and symbols in a cluster to express emotion extensively. To normalize the data, words with more than two repeated letters, symbol strings of more than one repeated punctuations symbols or emojis are shortened, for example, '!!!!' is shortened to '!', and 'yessss' to 'yess'. Note that we do not reduce words all the way to linguistically simplest form ('yes' in the example), since length of repeated letters represents the intensity of emotion. By distinguishing 'yess' from 'yes', the emotion intensity is partially preserved in our dataset.

If a Tweet contains less than three alphabetical words, the conversation is not included in the dataset. Then all symbols, emojis and words are tokenized. Finally, we build a vocabulary of size 20K according to token frequency. Any tokens outside the vocabulary are replaced by a special token.

3.3 Emoji Labeling

Then we label responses with emojis. If there are multiple types of emoji in a response, we use the emoji with most occurrences inside the response. Among those emojis with same occurrences, we choose the least frequent one across the whole corpus, on the hypothesis that less frequent tokens better represent what the user wants to express. The last occurrence of emoji label is taken out from the response.

We randomly split the corpus into 629,559 / 32,600 conversation pairs for train/test set[1]. Distribution of responses across different emoji labels is presented in Figure 2.

4 Generative Models

In this work, our goal is to generate emotional responses to the original Tweet. The emotion is explicitly linked to an emoji label.

4.1 Base: Sequence-to-Sequence Models

Traditional studies use deep recurrent architecture and encoder-decoder models to generate conversation responses, mapping original texts to target

[1] We will release the dataset.

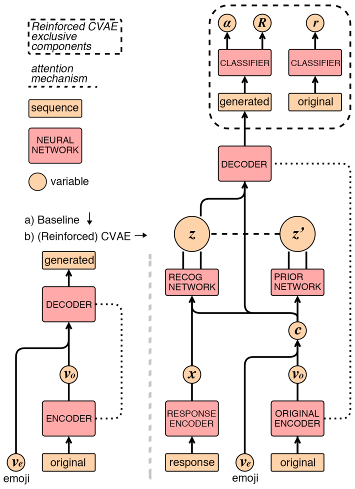


Figure 3: From bottom to top is a forward pass of data during training. During training of Reinforced CVAE, emoji label is also fed to classifier, which is not shown in the figure.

responses. Here we use a sequence-to-sequence (SEQ2SEQ) model (Sutskever et al., 2014) with scaled Luong attention mechanism (Luong et al., 2015) as our baseline model (See Figure 3).

We use randomly initialized embedding vectors to represent each word. To specifically model the emotion, we compute the embedding vector of the emoji label the same way as word embeddings. The emoji embedding is further reduced to smaller size vector v_e through a dense layer. We pass the embeddings of original Tweets through a bidirectional RNN encoder of GRU cells (Schuster and Paliwal, 1997; Chung et al., 2014). The encoder outputs a vector v_o that represents the original tweet. Then v_o and v_e are fed to a 1-layer RNN decoder of GRU cells. Response is then generated from the decoder.

4.2 Conditional Variational Autoencoder (CVAE)

Having similar encoder-decoder structures, SEQ2SEQ model can be easily extended to a Conditional Variational Autoencoder (CVAE) (Sohn et al., 2015). Figure 3 illustrates the model: response encoder, recognition network, and prior network are added on top of the SEQ2SEQ model. Response encoder has the same structure to original Tweet encoder, but it has separate parameters. We use embeddings to represent Twitter responses and pass them through response encoder.

Mathematically, CVAE is trained by maximizing a variational lower bound on the conditional likelihood of x given c, according to:

$$p(x|c) = \int p(x|z,c)p(z|c)dz \tag{1}$$

z, c and x are random variables. z is the latent variable. In our case, the condition $c = [v_o; v_e]$, target x represents the response. Decoder is used to approximate $p(x|z,c)$, denoted as $p_D(x|z,c)$. Prior network is introduced to approximate $p(z|c)$, denoted as $p_P(z|c)$. Recognition network $q_R(z|x,c)$ is introduced to approximate true posterior $p(z|x,c)$ and will be absent during generation phase. By assuming that the latent variable has a multivariate Gaussian distribution with a diagonal covariance matrix, the lower bound to $\log p(x|c)$ can then be written by:

$$\begin{aligned}-\mathcal{L}(\theta_D, \theta_P, \theta_R; x, c) &= \mathrm{KL}(q_R(z|x,c)||p_P(z|c)) \\ &\quad -\mathbb{E}_{q_R(z|x,c)}(\log p_D(x|z,c))\end{aligned} \tag{2}$$

θ_D, θ_P, θ_R are parameters of those networks.

Note that decoder still has an attention mechanism connected to the original Tweet encoder, which makes our model deviate from previous works of CVAE on text data. Based on attention memory as well as c and z, a response is finally generated from decoder.

When dealing with text data, VAE models tends to deteriorate to plain SEQ2SEQ model. Some previous methods effectively alleviate this problem, which are also important to keep a balance between the two items of the loss. We use two techniques of KL annealing (Bowman et al., 2015) and bow loss (Zhao et al., 2017) in our model.

4.3 Reinforced CVAE

Reinforced CVAE is the CVAE model above combined with policy gradient method. First, we train an emoji classifier on our dataset separately and fix its parameters thereafter. The classifier is a skip connected model of Bidirectional GRU-RNN layers (Felbo et al., 2017).

During policy training, we first get generated response x' by forward pass x and c through CVAE, then feed generation x' to classifier and get the probability of the emoji label as reward R. Let θ be parameters of our network, REINFORCE algorithm (Williams, 1992) is used to maximize the expected reward of generated responses:

$$\mathcal{J}(\theta) = \mathbb{E}_{p(x|c)}(R_\theta(x,c)) \tag{3}$$

The gradient of Equation 3 is approximated using the likelihood ratio trick (Glynn, 1990; Williams, 1992):

$$\nabla \mathcal{J}(\theta) = (R - r)\nabla \sum_{t}^{|x|} \log p(x_t|c, x_{1:t-1}) \quad (4)$$

r is the baseline value to keep estimate unbiased and reduce its variance. In our case, we directly pass x through emoji classifier and compute the probability of the emoji label as r. The model then encourages response generation that has $R > r$.

As REINFORCE objective is unrelated to response generation, it may make the generation model quickly deteriorate to some generic responses. To prevent the training from running wild, we propose two straightforward techniques to constrain policy training:

1. Adjust rewards according to the rank of emoji label probability. The rationale is that when rank of emoji label probability is high enough, it has already succeeded in emotion modeling, thus no need to adjust parameters toward higher probability on this response. Modified policy gradient is written as:

   $$\nabla \mathcal{J}'(\theta) = \alpha(R - r)\nabla \sum_{t}^{|x|} \log p(x_t|c, x_{1:t-1}) \quad (5)$$

   where $\alpha \in [0, 1]$ is a variant coefficient. The higher R ranks in all types of emoji label, the closer α is to 0.

2. Train Reinforced CVAE by a hybrid objective of REINFORCE and variational lower bound objective, learning to generate responses toward a better emotion accuracy:

   $$\mathbf{min}_\theta \mathcal{L}'' = \mathcal{L}' - \lambda \mathcal{J}' \quad (6)$$

   where λ is a balancing coefficient.

Algorithm 1 outlines the training process of Reinforced CVAE.

5 Experimental Results and Analyses

To generally evaluate the performance of our models, generation perplexity and top-1/top-5 emoji accuracy on test set as metrics. Perplexity indicates how much difficulty the model is having when generating responses. We also use top-5 emoji accuracy, since meaning of different emojis may overlap with only a subtle difference. Machine may learn that similarity and give multiple possible labels as answer.

```
input : Total training step $N$, Training
        batches, $\lambda$
Pretrain CVAE by minimizing Eq.2;
$i = 0$;
while $i < N$ do
    Get next batch B and target responses T in B;
    procedure Forward pass B through CVAE
        get generation G;
        get probability $P$ of all words in G;
        get variational lower bound objective $\mathcal{L}'$;
    Compute $R$, $\alpha$ by emoji classifier using G;
    Compute $r$ by emoji classifier using T;
    $\mathcal{J}' = \alpha(R - r)\sum \log P$;
    $\mathcal{L}'' = \mathcal{L}' - \lambda\mathcal{J}'$;
    Conduct gradient descent on CVAE using $\mathcal{L}''$;
    $i++$;
end
```

Algorithm 1: Training of Reinforced CVAE.

Model	Perplexity	Emoji Accuracy	
		Top1	Top5
Baseline	130.7	34.2%	57.7%
CVAE	37.1	41.2%	75.6%
Reinforced CVAE	38.1	42.2%	77.3%

Table 1: Generation perplexity and emoji accuracy.

As is shown in Table 1, CVAE significantly reduces the perplexity and increases the emoji accuracy over baseline model. The Reinforced CVAE also adds to the emoji accuracy at the cost of a slight increase in perplexity. These results confirm that proposed methods are effective toward the generation of emotional responses.

When converged, the second item of variable lower bound objective, namely KL loss, is 26.8/25.4 for CVAE/Reinforced CVAE respectively. The models achieve a balance between items of loss, confirming that they've successfully learned a meaningful latent variable.

In following parts of this section, we are going to take a closer look to the generation quality as well as our models' capability of expressing emotions.

Model	Unigram	Bi-	Tri-
Baseline	0.0061	0.0199	0.0362
CVAE	0.0191	0.131	0.365
Reinforced CVAE	0.0160	0.118	0.337
Target responses	0.0353	0.370	0.757

Table 2: Type-token ratios for model generation. Scores of tokenized human-generated target responses are given for reference.

5.1 Generation Diversity

Generation of SEQ2SEQ model is monotonous as several generic responses occur repeatedly across the whole generation. SEQ2SEQ model also learns to generate "i'm not" or "i'm not sure if" for the beginning of many responses, while CVAE models generate responses of much more language diversity. To showcase this disparity, we report the diversity score computed by counting the number of distinct unigrams/bigrams/trigrams and scaling the count by the total number of those n-grams.

As shown in Table 2, results show proposed models beat baseline by a large margin. Diversity scores of Reinforced CVAE are reasonably compromised, since it's generating more emotional responses.

5.2 Controllability of Emotions

There are potentially multiple types of emotion in reaction to an utterance. Our work makes it possible to generate a response of an arbitrary emotion by conditioning the generation on a specific type of emoji. We conducted experiments by replacing user-generated label with all other emojis in the 64 emoji labels. Note that multiple responses may be responding to the same tweet, so in this experiment, we eliminate duplicate original tweets in the dataset. There are 30,299 unique original tweets in the test set.

Figure 4 shows top-5 accuracy of each type of the first 32 emoji labels when we generating responses on the test set conditioned on the same emoji.

Results show that proposed models increase the accuracy over every type of emoji label. Notify that Reinforced CVAE model sees a bigger increase on the less common emojis, confirming the effect of the emoji specified policy training. This is a general evaluation showing the capability of proposed model. Accuracy may be low for some emojis, as they are uncommon across the data set, or generally not suitable in reaction to some original tweets.

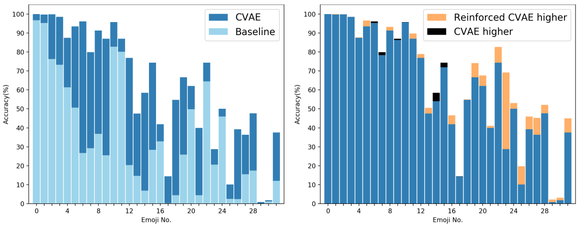


Figure 4: Top5 emoji accuracy of the first 32 emoji labels. **Left**: CVAE v. baseline. **Right**: Reinforced CVAE v. CVAE. If Reinforced CVAE scores higher, the margin is shown in orange. If lower, it's shown in black.

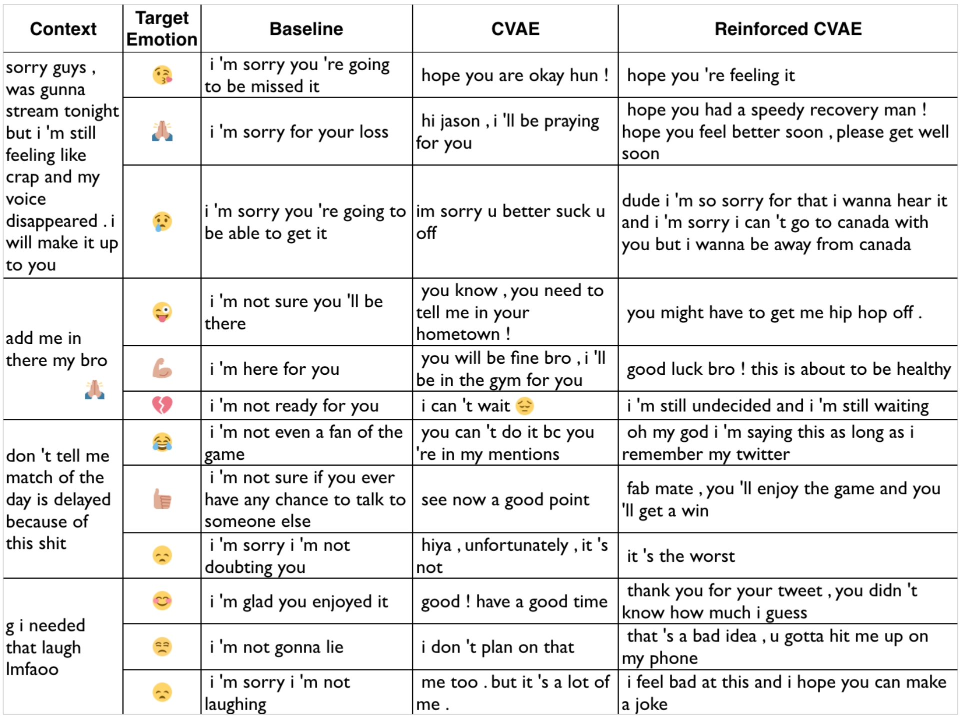
Setting	Model v. Baseline	Win	Lose	Tie
reply	CVAE	42.4%	43.0%	14.6%
reply	Reinforced CVAE	40.6%	39.6%	19.8%
emoji	CVAE	48.4%	26.2%	25.4%
emoji	Reinforced CVAE	50.0%	19.6%	30.4%

Table 3: Results of human evaluation. Tests are conducted pairwise between proposed models and the baseline model.

5.3 Human Evaluation

We employ crowdsourced judges to evaluate a random sample of 100 items, each being assigned to 5 judges on Amazon Mechanical Turk. We present judges original tweets and generated responses. In the first setting of human evaluation, judges are asked to decide which of the two generated response better reply the original tweet. In the second setting, the emoji label is presented, and judges are asked to pick the one they decide better fits the emoji. (The two settings of evaluation are conducted separately, so that it will not affect judges' verdicts.) Order of two generated responses under one item is permuted. Ties are permitted for answers. We batch five items as one assignment and insert a item with two identical outputs as sanity check. Anyone who failed to choose 'tie' for that item is rejected from our test.

We then conducted a Turing test. Each item we present judges an original tweet, its reply by human, and its response generated from Reinforced CVAE model. We ask judges to decide which of the two given responses is written by human. Other parts of setting are similar to above mentioned tests. It turned out 18% of the test subjects mistakenly chose machine-generated responses as human written, and 27% stated that they were



Context	Target Emotion	Baseline	CVAE	Reinforced CVAE
sorry guys , was gunna stream tonight but i 'm still feeling like crap and my voice disappeared . i will make it up to you	😘	i 'm sorry you 're going to be missed it	hope you are okay hun !	hope you 're feeling it
	🙏🏼	i 'm sorry for your loss	hi jason , i 'll be praying for you	hope you had a speedy recovery man ! hope you feel better soon , please get well soon
	😢	i 'm sorry you 're going to be able to get it	im sorry u better suck u off	dude i 'm so sorry for that i wanna hear it and i 'm sorry i can 't go to canada with you but i wanna be away from canada
add me in there my bro 🙏🏼	😜	i 'm not sure you 'll be there	you know , you need to tell me in your hometown !	you might have to get me hip hop off .
	💪🏼	i 'm here for you	you will be fine bro , i 'll be in the gym for you	good luck bro ! this is about to be healthy
	💔	i 'm not ready for you	i can 't wait 😔	i 'm still undecided and i 'm still waiting
don 't tell me match of the day is delayed because of this shit	😂	i 'm not even a fan of the game	you can 't do it bc you 're in my mentions	oh my god i 'm saying this as long as i remember my twitter
	👍🏼	i 'm not sure if you ever have any chance to talk to someone else	see now a good point	fab mate , you 'll enjoy the game and you 'll get a win
	😞	i 'm sorry i 'm not doubting you	hiya , unfortunately , it 's not	it 's the worst
g i needed that laugh lmfaoo	😊	i 'm glad you enjoyed it	good ! have a good time	thank you for your tweet , you didn 't know how much i guess
	😒	i 'm not gonna lie	i don 't plan on that	that 's a bad idea , u gotta hit me up on my phone
	😞	i 'm sorry i 'm not laughing	me too . but it 's a lot of me .	i feel bad at this and i hope you can make a joke

Figure 5: Some examples from our generated emotional responses. Context is the original Tweet, and target emotion is the emotion that we would like to generate. The three columns on the right are generated emotional responses.

not able to distinguish between the two responses. This indicates a preliminary success toward generating human-like language.

When it comes to inter-rater agreement, it is ideal if all five judges choose the same answer, and in worst scenario, only two judges choose the same answer at most. The ratio for agreement by 5:4:3:2 is 0.317:0.33:0.31:0.053, showing that our test has a reliable inter-rater agreement.

5.4 Case Study

Finally, we sampled some generated responses from all three models, and list them in Figure 5. Given an original Tweet, we would like to generate responses for three different target emotions. Generally, we can see that generated emotional responses from proposed models are better than from baseline both on emotion expression and general quality, while generation from SEQ2SEQ model is monotonous and tedious. Furthermore, Reinforced CVAE gains on emotion expression over CVAE.

Interestingly enough generation from SEQ2SEQ seems to be mostly grammatically correct. With all the diversity of language on Twitter, SEQ2SEQ only choose to generate from most frequent expressions, forming a predictable pattern for its generation. On the contrary, generation from CVAE model is diverse, which is in line with previous quantitative analysis. However, the generated responses are sometimes too diversified and implausible to reply the original tweet. The problem is rooted in the nature of CVAE and partially aggravated by our training setting that gives CVAE too much freedom.

Sometimes, Reinforced CVAE tends to generate lengthy response by stacking up sentences. It learns to break the length limit of sequence generation during hybrid training, since the variational lower bound objective competing with REINFORCE objective. The situation would be more serious is λ in Equation 6 is set higher.

6 Conclusion and Future Work

In this paper, we investigate the possibility of using naturally annotated emoji-rich Twitter data for emotional response generation. More specifically, we collected more than half a million Twitter conversations with emoji in the response, and assumed that the emoji chosen by the user expresses the emotion of the Tweet. We applied several state-of-the-art neural models to learn a generation system that is capable of giving response with arbitrary emotion. We performed automatic and human evaluations to understand the quality of generated responses. We trained a large scale emoji classifier, and ran the classifier on the generated responses to evaluate the emotion accuracy of the generated response.

We also performed an Amazon Mechanical Turk experiment, by which we compared our models with a baseline sequence-to-sequence model on metrics of relevance and emotion. Experimentally, it is shown that our model is capable of generating high-quality emotional responses, without the need of laborious human annotations.

We believe our work marks a step toward building serviceable dialogue agents. We are also looking forward to transferring the idea of naturally-labeled emojis to more specific domain of text and multi-turn dialog generation. Due to the nature of social media text, some emotions, such as fear and disgust, are underrepresented in the dataset, and the distribution of emojis is unbalanced to some extent. Future work should include accumulating more data and balance the ratio of different emojis, as well as advancing toward more sophisticated generation methods.

References

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *SocialNLP at EMNLP* .

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524* .

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*. pages 249–256.

Peter W Glynn. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33(10):75–84.

Alec Go, Richa Bhayani, and Lei Huang. 2016. Sentiment140. *Site Functionality, 2013c. URL http://help. sentiment140. com/site-functionality. Abruf am* 20.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Chieh-Yang Huang, Tristan Labetoulle, Ting-Hao Kenneth Huang, Yi-Pei Chen, Hung-Chen Chen, Vallari Srivastava, and Lun-Wei Ku. 2017. Moodswipe: A soft keyboard that suggests messages based on user-specified emotions. *EMNLP Demo* .

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300* .

Jiwei Li, Will Monroe, Alan Ritter, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* .

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *EMNLP* .

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957* .

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 142–150.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pages 79–86.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*. pages 3295–3301.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*. pages 3483–3491.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714* .

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960* .

A Supplementary Materials

A.1 Emoji Classifier

For the emoji classifier used in the Reinforced CVAE method, we train it on our train set by mapping response Tweets to their emoji label, with a dropout rate of 0.2 and an Adam optimizer of a 1e-3 learning rate with gradient clipped to 5. RNN layers and word embeddings in the classifier have a dimension of 128. All weights of dense layers are initialized by glorot uniform initializer (Glorot and Bengio, 2010) and word embeddings are initialized by sampling from uniform distribution [-4e-3, 4e-3].

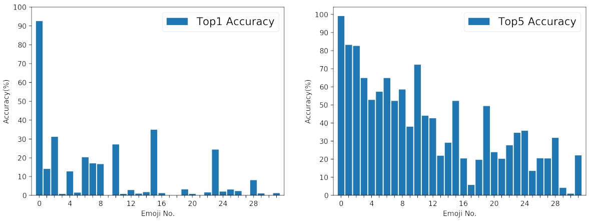


Figure 6: Top-1 and top-5 accuracy of emoji classifier by each emoji label on test set.

The classifier gives probability of all 64 emoji labels. For 32.1% responses in test set, probability of the emoji label ranks highest of all emoji labels. In 57.8% of cases, probability of emoji label is among the five highest. We refer to the two figures as *top-1* and *top-5 accuracy*. Figure 6 shows the top-1 and top-5 accuracy of the 32 most frequent emoji labels. Accuracy for less common emojis may be low, since they are underrepresented in the dataset.

A.2 Hyperparameters

For the hyper-parameters of baseline model and proposed models, we use word embeddings of 128 dimensions and RNN layers of 128 hidden units for all encoders and decoders. The size of emojis' embeddings is contracted to 12 through a dense layer of $tanh$ non-linearity. We set the size of latent variables to 268. MLPs in recognition/prior network are 3 layered with $tanh$ non-linearity. All other training settings are the same with emoji classifier's.

For Reinforced CVAE[2], λ in hybrid objective (Equation 6) is set 1, and α in Equation 5 is empirically given by:

$$\alpha_{x',e} = \begin{cases} 0, & \textit{R ranks 1 in all labels} \\ 0.5, & \textit{R ranks 2 to 5 in all labels} \\ 1, & \textit{otherwise} \end{cases} \tag{7}$$

where reward R is the probability of emoji label e computed by the classifier, and x' is the generated response.

Pretraining is vital to the success of CVAE models, since it is essentially hard for them to learn a latent variable space from total randomness. We use fully converged baseline SEQ2SEQ model to initialize its counterparts in CVAE models. When

[2] We will release the source code for MOJITALK and pretrained models on Github.com.

trained with emoji classifier, instead of using hybrid loss function from the beginning, we introduce the policy loss only after 2 epochs of training.

For our final models, we use bow loss along with KL annealing to 0.5 at the end of the 6th epoch. Note that KL weight does not anneal to 1 at last, meaning that our models do not strictly follow the objective of CVAE (Equation 2). However, lower KL weight gives the model more freedom to generate text. We can view this technique as early stopping (Bowman et al., 2015), finding a better result before model converges on the original objective.

To exploit the randomness of latent variable, during generation, we sample the result of CVAE models 5 times and choose the generated response with highest probability of designated emoji label as the final generation.