MOJITALK: Generating Emotional Responses at Scale (Supplementary Materials)

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

0.1 Bag-of-Word Loss

In the idea of BoW loss, x can be decomposed into x_o of word order and x_{bow} of words without order. By assuming that x_o and x_{bow} are conditionally independent, $p(x|z,c) = p(x_o|z,c)p(x_{bow}|z,c)$. Given z and c, $p(x_{bow}|z,c)$ is the product over probability of every token in the text:

$$\begin{aligned} p(x_{bow}|z,c) &= \prod_{t=1}^{|x|} p(x_t|z,c) \\ &= \prod_{t=1}^{|x|} \text{softmax}(f(x_t,z,c)) \end{aligned} \tag{1}$$

Function f first maps z, c to space $\mathbb{R}^V$, where V is the vocabulary size, and then chose the element corresponding to token x_t as its logit.

Now the modified objective is written by:

$$\begin{aligned} \mathcal{L}'(\theta_D,\theta_P,\theta_R;x,c) &= \mathcal{L}(\theta_D,\theta_P,\theta_R;x,c) \\ &+\mathbb{E}_{q_R(z|x,c)}(\log p(x_{bow}|z,c)) \end{aligned} \tag{2}$$

Finally, CVAE is trained by minimizing $\mathcal{L}'$.

0.2 Emoji Classifier

The emoji classifier is a skip connected model of Bidirectional GRU-RNN layers and has the same structure as the classifier in (Felbo et al., 2017). This separate neural network uses the same set of hyper-parameters (embedding size, hidden state size, etc.) as in the generation models described below. 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 the uniform distribution [-4e-3, 4e-3].

The classifier gives the probability of all 64 emoji labels. For 32.1% responses in the test set, the probability of the emoji label ranks highest of all emoji labels. In 57.8% of cases, the probability of emoji label is among the five highest. We refer to the two figures as *top-1* and *top-5 accuracy*. Figure 1 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.

0.3 Training Process of the Reinforced CVAE

Algorithm 1 outlines the training process of the Reinforced CVAE. The first step of pretraining is described in the next section. For every training batch, we first compute the variational objective $\mathcal{L}'$ and obtain the generated text. Then we compute the policy gradient $\mathcal{J}'$ from the word probability in the previously generated text and the rewards determined by the emoji classifier. Finally, we conduct gradient descent on the CVAE components using the hybrid objective $\mathcal{L}''$ that is comprised of $\mathcal{L}'$ and $\mathcal{J}'$.

0.4 Experiment Setting

Hyper-parameters For the hyper-parameters of the base model and CVAE 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 as the emoji classifier.

For Reinforced CVAE[1], λ in hybrid objective

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

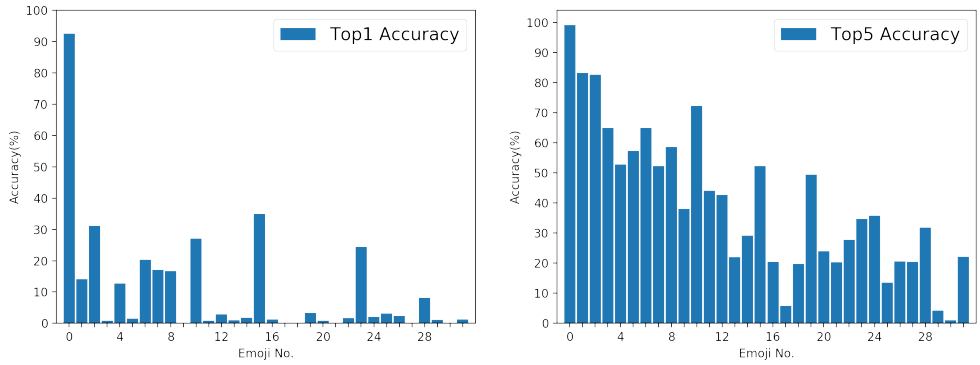

Figure 1: Top-1 and top-5 accuracy of emoji classifier by each emoji label on test set.

input : Total training step N, Training batches, λ

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, α 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 the Reinforced CVAE.

(Eq.6 of the paper) is set to 1, and α in Eq.5 of the paper is empirically given by:

$$\alpha_{x',e} = \begin{cases} 0, & R \textit{ ranks 1 in all labels} \\ 0.5, & R \textit{ ranks 2 to 5 in all labels} \\ 1, & \textit{otherwise} \end{cases} \tag{3}$$

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

Training Setting We use fully converged base SEQ2SEQ model to initialize its counterparts in CVAE models. When training the Reinforced CVAE 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.

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

References

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *CONLL*.

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. *EMNLP*.

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.