Automatic Dialogue Generation with Expressed Emotions

Chenyang Huang, Osmar R. Zaïane, Amine Trabelsi, Nouha Dziri
Department of Computing Science, University of Alberta
{chuang8,zaiane,atrabels,dziri}@ualberta.ca

Abstract

Despite myriad efforts in the literature designing neural dialogue generation systems in recent years, very few consider putting restrictions on the response itself. They learn from collections of past responses and generate one based on a given utterance without considering, speech act, desired style or emotion to be expressed. In this research, we address the problem of forcing the dialogue generation to express emotion. We present three models that either concatenate the desired emotion with the source input during the learning, or push the emotion in the decoder. The results, evaluated with an emotion tagger, are encouraging with all three models, but present better outcome and promise with our model that adds the emotion vector in the decoder.

1 Introduction

Automatic dialogue generation (Ritter et al., 2011) aims at generating human-like responses given a human-to-human dialogue history. Most conversational agents are specialized for a specific domain such as travel booking (Xu and Rudnicky, 2000) and are typically finite state-based or template-based. Open domain dialogue systems have seen a growing interest in recent years thanks to neural dialogue generation systems, based on deep learning models. These systems do not encode dialog structure and are entirely data-driven. They learn to predict the maximum-likelihood estimation (MLE) based on a large training corpus. The machine learning-based system basically learns to predict the words and the sentence to respond based on the previous utterances. However, while such a system can generate grammatically correct and human-like answers, the responses are often generic and non-committal instead of being specific and emotionally intelligent. For instance, we can not dictate a particular emotion to express. In this paper, we consider a model in which the wished emotion to be expressed is injected to direct the response generation. For example, if the user says: "I just missed my deadline." If we want the system to respond with sadness, it could be "I am sorry to hear that.", but we can also force the response to express anger: "You should never do it again!"

There are some challenges to tackle this task.

- The current neural dialogue models are not satisfactory in general.
- There is a lack of dialogue corpora that are labeled with emotions.
- The evaluation is hard because emotion is subjective and sometimes ambiguous.

The idea is to use an emotion mining from text classifier (Yadollahi et al., 2017) to predict the emotion or emotions expressed in the source utterance, then decide based on the detected emotions, which emotion e is expressed in the response. The response is evaluated using the same emotion classifier and is declared successful if e is predicted from the response. The emotion tagger we use is based on the work in (Yadollahi et al., 2017) but uses a deep learning model and trains on 9 emotions: *anger, disgust, fear, guilt, joy, love, sadness, surprise*, and *thankfulness*. These are based on the six basic emotions from Ekman's model (Ekman, 1992), to which we added guilt, love and thankfulness in the context of an open ended conversational agent that we aim to be emotionally intelligent for companionship to elderly users.

In this paper, we proposed three approaches to make our model of our conversational agent generate responses expressing specific emotions. The first two approaches add the emotion as a token with the input during the learning either before the

utterance sentence or after, and the third approach injects the desired emotion directly in the decoder.

2 Related Work

Vinyals and Le (2015) adopted the Sequence-to-sequence (Seq2Seq) model used in machine translation (Sutskever et al., 2014) in the task of automatic response generation. Seq2Seq learns to generate a sequence of words from another sequence of words as input. Since then, many works based on this framework have been conducted to improve the response quality from different points of view. Reinforcement learning has also been adopted to force the model to have longer discussions (Li et al., 2016b). Serban et al. (2017) proposed a hierarchical encoder to generate a response from more utterances. Moreover, there are also attempts to avoid generating dull, short responses (Li et al., 2017a,b).

3 Embed Emotion into Seq2Seq Models

Seq2Seq is a conditional language model which takes as input message-response pairs (X, Y), where $X = x_1, x_2, \cdots, x_m$ and $Y = y_1, y_2, \cdots, y_n$ are sentences consisting of sequences of words. The goal of the model is to minimize the cross entropy loss $\mathcal{L} = \log p(Y|X)$. Despite the variants of Seq2Seq models, they usually consist of two major components: encoder and decoder. The encoder embeds a source message into a vector which is then fed into the decoder. The decoder generates $\hat{Y} = \hat{y_1}, \hat{y_2}, \cdots$ step by step. This procedure can be described as $c = \text{Encoder}(X)$, $Y = \text{Decoder}(c)$. In our case, each (X, Y) pair is assigned with an additional desired response emotion e. Our goal is therefore to minimize $-\log p(Y|X, e)$. We propose two methods to tackle this task based on how to embed e, either concatenating an emotion token to the input message, or injecting the emotion into the decoder.

3.1 Seq2Seq with Attention

The choice of our encoder is LSTM (Hochreiter and Schmidhuber, 1997) and it can be formulated as the following.

$$h_t^{En}, c_t^{En} = \text{LSTM}^{En}(M(x_i), [h_{t-1}^{En}; c_{t-1}^{En}])$$
$$h_0^{En} = c_0^{En} = \mathbf{0} \quad (1)$$

Where h_t^{En} and c_t^{En} are encoder's hidden state and cell state at time t. $M(x)$ is the vector representation of word x (Mikolov et al., 2013). In our experiments, we apply the state-of-the-art *FastText* (Joulin et al., 2016) pre-trained model.

Adapting attention mechanism in sequence generation has shown promising improvement (Bahdanau et al., 2014; Luong et al., 2015). In our case, we use the global attention with general score function (Luong et al., 2015) under the assumption that generated words can be aligned to any of the words in the previous dialogue utterance. We use another LSTM to decode the information, the decoder with attention can be described as:

$$\boldsymbol{h}^{En} = [h_1^{En}, h_2^{En}, \cdots, h_m^{En}] \quad (2)$$
$$\hat{h}_t = \alpha_t \cdot \boldsymbol{h}^{En} \quad (3)$$
$$\alpha_t = \text{Softmax}(h_t^{De} W_a \boldsymbol{h}^{En}) \quad (4)$$
$$h_t^{De}, c_t^{De} = \text{LSTM}^{De}(M(y_i), [\hat{h}_{t-1}; c_{t-1}^{De}]) \quad (5)$$
$$\hat{h}_0 = h_m^{En}, c_0^{De} = c_m^{En} \quad (6)$$

Where h_t^{De} and c_t^{De} are hidden state and cell state. α_t is the attention weights over all hidden states of encoder. W_a is a trainable matrix which is initialized randomly.

3.2 Embedding Emotion

Our first model is inspired by Google's multilingual neural machine translation system (Johnson et al., 2016). Generating different types of emotional responses can be an analogy to translating the same sentence into different languages. The implementation is straight forward; we make each emotion a single token and concatenate it with the input X so that our model has the target of minimizing $\log p(Y|X')$, where $X' = \text{Concat}(e, X)$. This approach reduces the two individual inputs into one so that they can be trained on normal Seq2Seq models. Further more, we consider the concatenation in two ways, before X and after X, as the following.

$$X_1 = \{e, x_1, x_2, \cdots, x_m\} \ (Enc - bef) \quad (7)$$
$$X_2 = \{x_1, x_2, \cdots, x_m, e\} \ (Enc - aft) \quad (8)$$

Both of the methods are embedding the desired emotion into an encoder. We name them *Enc-bef* and *Enc-aft*, respectively. e is the emotion of the generated response and is obtained from Y by an emotion mining classifier. Both models require to change the m in (2) and (6) to $m + 1$.

Li et al. (2016a) proposed a modified Seq2Seq model that allows models to learn the speaking

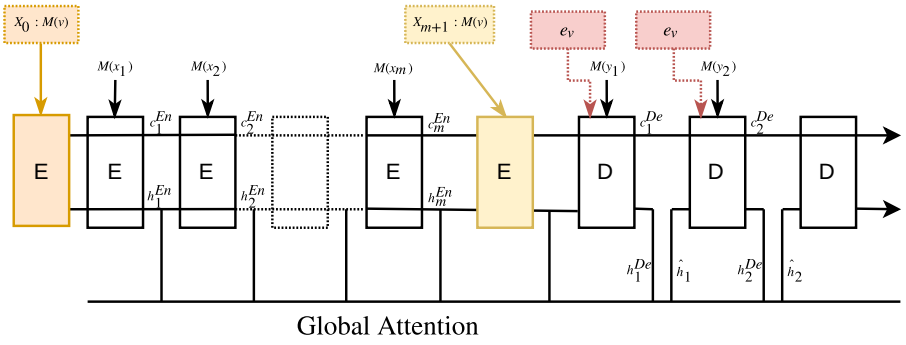

Figure 1: Three models to embed emotion: orange and yellow are the addition emotion tokens to concatenate with the source sentences for model *Enc-bef* and *Enc-aft* respectively. The salmon-colored blocks represent the emotion vectors which need to be feed into decoder of model *Dec* repeatedly.

styles of different people from a movie script corpus. Our third approach adapts their idea but instead of embedding people/speaker into the decoder, we feed the emotion vectors v_e during the decoding. Equation (5) is changed to $h_t^{De}, c_t^{De} = \text{LSTM}^{De}(M(y_i), [\hat{h}_{t-1}; c_{t-1}^{De}; v_e])$. v_e is drawn from a trainable embedding layer. We name this model *Dec*. The models are shown in Figure 1.

4 Dataset

To train the dialogue models, we use the OpenSubtitles dataset (lis, 2016). Precisely, we use the pre-processed data by (Li et al., 2016a) and further removed duplicates. The total amount of utterances is 11.3 million, each utterance has a minimal length of 6 words.

Since there is no existing dialogue data set labeled with emotions, we trained our own emotion classifier to tag the corpus. We use the CBET dataset [1] (Yadollahi et al., 2017; Shahraki and Zaiane, 2017), it contains 9 emotions and 81k instances. Each instance is labeled with up to two emotions. The emotions are *anger, surprise, joy, love, sadness, fear, disgust, guilt*, and *thankfulness*. We train a bidirectional LSTM (Graves et al., 2005) model and achieve an F1-score of 68.4% with precision 49.1% and recall 52.9% on these emotions. To tag the target utterances with higher confidence, we use a threshold to separate those utterances that do not express emotion. 34.01% are thus labeled as Non-emotion. 'Non-emotion' is treated as a special emotion when training the dialogue models, but it is not considered in the evaluation.

[1] http://www.cs.ualberta.ca/~zaiane/data/CBET/CBET.csv

5 Experiments and Evaluation

5.1 Seq2Seq

With the purpose of comparison, the parameters of the three models are set to be the same. The dimensions of LSTM hidden units are set to 600. Adam optimizer (Kingma and Ba, 2014) with learning rate of 0.0001 is used. The size of the vocabulary space is set to 25,000, which is the same as that in (Li et al., 2016a). We also use *FastText* (Joulin et al., 2016) pre-trained word embedding which is shared by the LSTMs in both encoder and decoder and set to trainable. We held out 50k samples from the whole dataset as test set. 95% of the remaining is used to train the dialogue models, and 5% of it is used for evaluation and preventing overfitting.

5.2 Accuracy of Expressed Emotions

In this research, we tackle the problem of training a generative model that can respond while expressing a specific emotion. Unlike the work by (Li et al., 2016a), expensive human evaluation is not needed. Instead, we evaluate the output using an emotion mining classifier to see whether the intended emotion is among the detected ones. For each input utterance, we let the model generate responses for each of the 9 emotions and check, using the emotion classifier, which emotion is indeed expressed in the output. Hence, the emotions' accuracies of the generated responses are estimated by the emotion classifier. Different from the procedure of tagging, where we put a threshold to enforce a higher precision, the most possible emotion is chosen in the evaluation. The results are shown in Table 1.

Emotion	*Enc-bef*	*Enc-aft*	*Dec*
anger	60.34%	62.44%	68.24%
fear	89.34%	86.46%	87.52%
joy	45.76%	41.36%	48.53%
love	56.96%	55.32%	59.13%
sadness	94.16%	93.93%	94.22%
surprise	84.46%	85.11%	87.22%
thankfulness	87.89%	89.51%	91.06%
disgust	78.06%	76.94%	79.01%
guilt	93.25%	92.16%	91.22%
Average	76.69%	75.91%	78.46%

Table 1: Per class accuracy of generated response

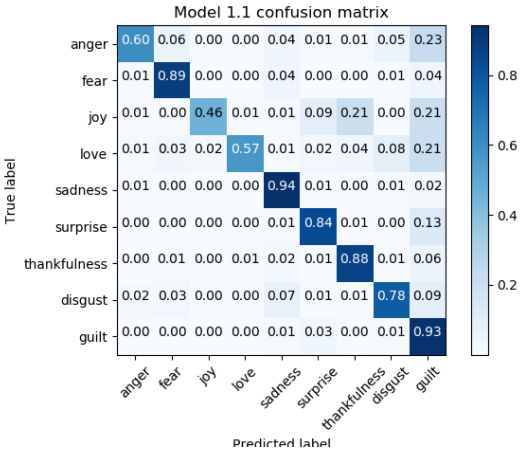

Figure 2: Confusion matrix of model *Enc-bef*

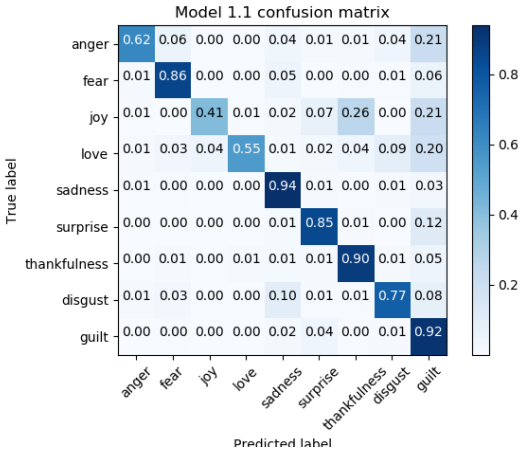

Figure 3: Confusion matrix of model *Enc-aft*

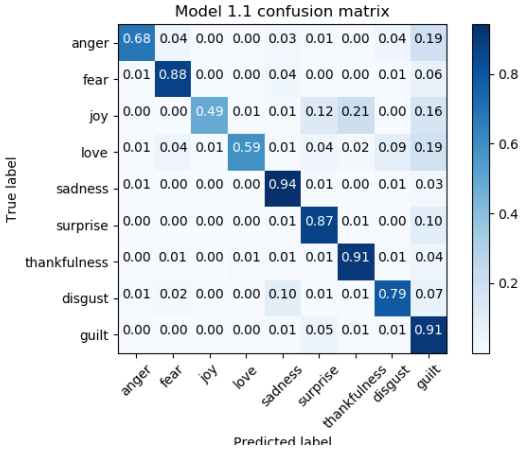

Figure 4: Confusion matrix of model *Dec*

6 Results and Discussion

From Table 1, we can observe that *Dec* has better overall average accuracies than *Enc-bef* and *Enc-aft*. The average accuracies of *Enc-bef* and *Enc-aft* are very close. However, we notice some discrepancies in the individual emotions' accuracies. For instance, *fear* is better captured by *Enc-bef*, while *anger* has a much better accuracy for *Dec*.

To further inspect the results, we also show the normalized confusion matrix of each model respectively, as in Figure 2, 3 and 4. We can notice obvious dark colored diagonals for the three figures. This indicates that all the three proposed models, indeed, have the ability to generate responses with given emotions. From these figures, we find that models tend to generate the responses with *guilt* regardless of the desired emotion. All the three models tend to generate *thankfulness* while they were instructed to express *joy*.

The patterns of confusion matrices of model *Enc-aft*, *Enc-bef* and *Dec* are close to each other. However, Dec model has a slightly better overall performance.

Tables 2 and 3 display examples of generated responses, according to different emotions, given a source utterance extracted from the test set. We can observe that the generated text is: (1) related to the source text; (2) expresses the desired emotions. For instance, when responding to "What is she doing here?", the generated text employs "she" rather than "he". The models are also able to express the emotion of *fear* by generating the word "afraid". When instructed to respond to the previous utterance "I didn't realize you were here", and to express *guilt*, all the models are able to generate "I am sorry". In terms of semantics, while the source is mentioning "here", the *Dec* model is able to answer with "I was just in the garden" which remains coherent with the location context.

Since increasing the diversity is not the target of this work, our models also suffer from this com-

source	what is she doing here ?
target	how do i open this god forsaken window !
anger *Enc-bef*	she 's going to kill herself
anger *Enc-aft*	she 's going to kill you
anger *Dec*	she 's supposed to be in the house
fear *Enc-bef*	i 'm afraid i can 't tell you
fear *Enc-aft*	she 's trying to kill herself
fear *Dec*	i 'm afraid she 's not here
love *Enc-bef*	she wants to see you in the bedroom
love *Enc-aft*	she 's in love with you
love *Dec*	she wants to be with you
disgust *Enc-bef*	she 's not allowed to leave
disgust *Enc-aft*	she 's going to be sick
disgust *Dec*	she 's the one who raped me

Table 2: Examples of generated responses by the three models with emotions *anger*, *fear*, *love* and *disgust*.

source	i didn 't realize you were here
target	maybe i should leave so
you	can continue
joy *Enc-bef*	i 'm here to make a phone call
joy *Enc-aft*	i 'm so happy for you
joy *Dec*	i was just in the garden house
sadness *Enc-bef*	i thought you were gonna be here
sadness *Enc-aft*	she 's trying to kill herself
sadness *Dec*	i thought i 'd be here
guilt *Enc-bef*	i 'm sorry i didn 't
guilt *Enc-aft*	i 'm sorry i didn 't know
you	were here
guilt *Dec*	i 'm sorry i didn 't hear you
surprise *Enc-bef*	i 'm here to find out
surprise *Enc-aft*	i thought you were going to be here
surprise *Dec*	i thought you might be here

Table 3: Examples of generated responses by the three models with emotions *joy*, *sadness*, *guilt* and *surprise*.

mon problem of Seq2Seq models. Similar to generating "I don't know" regardless of source sentences, in Seq2Seq models (Sordoni et al., 2015; Serban et al., 2016; Li et al., 2016b), our model tends to generate "I <unk>l be back in a minute" for emotion *anger*. The diversity of words that are used for each emotion are low, e.g., generations for emotion *fear* often have the word "gun" and the responses of emotion "sadness" often start with "I don't want ". This is clearly a side effect from our training data.

7 Conclusion

Emotional intelligence is the ability to monitor interlocutor's emotions and in turn appropriately express emotions in response. In our case, monitoring emotions in utterances is done using an emotion mining classifier. We assume that given some mapping rules, we can decide to express a specific emotion in the response. For instance if the message expresses sadness, the response could express compassion or surprise depending upon context. The work presented herein focuses solely on generating a response that expresses a given desired emotion, and assumes the emotion to be expressed is given via these mapping rules. However, one could automatically learn the emotion to express given the emotion in the message directly from the data by changing the input message-response pairs (X, Y) into $((X, e_X), (Y, e_Y))$ where e_X is the emotion in the message and e_Y is the emotion in the response. In this paper, we show that it is indeed possible to generate fluent responses that express a desired emotion. We present three models to do so. Despite the differences among the models, they are all trained towards minimizing $-\log p(Y|X, e)$ and all converge. The expression of some emotions (*guilt*, *sadness* and *thankfulness*) even reach accuracies over the 90%.

In our early experiments, we tagged each of the target utterance with the most possible emotion regardless of its confidence, wrongly assuming that all target utterances have a significant emotion. Although, our generative models can still be forced to produce the desired emotions, the quality of the generated sentences in terms of expressed emotions is below what is presented in Table 1 where the utterances without emotions (below a certain threshold) were labeled by "Non-Emotion". This shows the importance of learning to express emotions only from the utterances that indeed strongly convey measurable emotions. The other sentences are still kept to contribute in building the language model. We believe that adding reasoning to the mix can further enhance the emotional intelligence of a conversational agent.

References

2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

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

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion* 6:169–200.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005* pages 753–753.

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

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* .

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 994–1003.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017a. Data distillation for controlling specificity in dialogue generation. *arXiv preprint arXiv:1702.06703* .

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

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547* .

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 583–593.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*. pages 3776–3784.

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.

Ameneh Gholipour Shahraki and Osmar R Zaiane. 2017. Lexical and learning-based emotion mining from text. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*.

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.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* .

Wei Xu and Alexander I Rudnicky. 2000. Task-based dialog management using an agenda. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*. Association for Computational Linguistics, pages 42–47.

Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)* 50(2):25.