Generating Responses Expressing Emotion in an Open-domain Dialogue System

Chenyang Huang[0000−0003−2811−6008] and Osmar R. Zaïane[0000−0002−0060−5988]

Department of Computing Science,
University of Alberta, Edmonton, Canada
{chuang8,zaiane}@ualberta.ca

Abstract. Neural network-based Open-ended conversational agents automatically generate responses based on predictive models learned from a large number of pairs of utterances. The generated responses are typically acceptable as a sentence but are often dull, generic, and certainly devoid of any emotion. In this paper we present neural models that learn to express a given emotion in the generated response. We propose four models and evaluate them against 3 baselines. An encoder-decoder framework-based model with multiple attention layers provides the best overall performance in terms of expressing the required emotion. While it does not outperform other models on all emotions, it presents promising results in most cases.

Keywords: Open-domain dialogue generation · Emotion · Seq2seq · Attention mechanism

1 Introduction

Open-domain conversational systems [3, 21, 2] tackle the problem of generating relevant responses given an utterance as input. Compared to the non-open-domain scenario, also known as task-oriented dialogue generation, where it is possible for agents to rely on knowledge for a narrowed domain and detect intent then use specific templates to generate responses, such as a travel booking system [30].

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.

However, an absolute “automatic" system may find itself in situations where any inattentive response, even if correct and to the topic, may be improper or even negligent or offending. For example, if a person is expressing loneliness, or the death of a friend, the response should better be expressing empathy and support rather than a generic and careless possible response. In this work, we are tackling the problem of how to control the emotions expressed in generated responses. As shown in Fig. 1, the proposes methods take as input a source sentence as well as an emotion to be expressed. It will not only respond with an relative sentence, but also express the given emotion in the response.

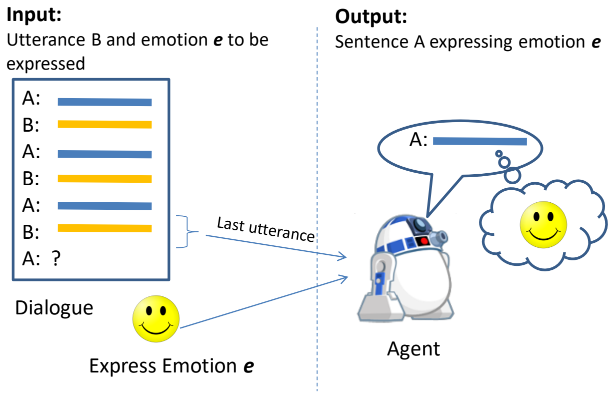


Fig. 1. The task of generating response given emotions

To this end, a sufficiently large dialogue corpus labeled with emotions is required to train such a system which is both open-domain and emotionally intelligent. We follow the pipelines described in [7]:

1. Train an emotion mining from text classifier [31].
2. Label the Opensubtitles dataset [15] with emotions using the classifier.
3. Design and train models using the labeled dialogue corpus.
4. Evaluate the models.

In addition, we proposed 4 models: *Dec-start*, *Dec-att*, *Dec-proj* and *Dec-trans*. Experiment results show that all the proposed models are capable of the task and *Dec-att* outperforms the baseline models in [7] without adding too many parameters to the neural models.

2 Related Work

Emotion category classification serves as one of the basics of this work. The task of emotion mining from text is mainly composed of the following 3 aspects.

- Identify the categories?
- Obtain a large, high quality labeled dataset?
- Train a good classifier for emotions from text?

P. Ekman, one of the earliest emotion theorists, suggested 6 basic emotions in 1972 [4]: *anger, disgust, fear, joy, sadness* and *surprise*. The following work by P. Shaver [25] and W. G. Parrott [20] suggest removing *disgust* and adding *love*. A. G. Shahraki [31, 24] combine the aforementioned emotion models by involving two additional emotions (*guilt* and *thankfulness*) but keeping *disgust*.

Human annotation is one straightforward approach to obtain labeled datasets but it is costly. Only very small amount of manually labeled categorical dataset is available at the time of writing this paper. As an alternative, distant supervision [18] has been investigated in many emotion detection researches [19]and proven to be efficient. Generally, they harvest tweets with emotion-carrying hashtags which are used as a surrogate of emotion labels.

Having tweets labeled with emotions, training a classifier is a task of supervised text classification which has already been a well-studied area[26, 14]. The recent state-of-art models are usually neural network models[33] with pre-trained

With the rise of deep learning, the success of the technology was also demonstrated in automatic response generation. The Sequence-to-sequence (Seq2seq) model which was shown effective in machine translation[27], was adopted in response generation for open domain dialogue systems [29]. Instead of predicting a sequence of words in the target language from a sequence of words in the source language, the idea is to predict a sequence of words as a response of another sequence of words. In a nutshell, Seq2seq models are a class of models that learn to generate a sequence of words given another sequence of words as input. 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 [12]. [23] proposed a hierarchical framework to process context more naturally. Moreover, there are also attempts to avoid generating dull, short responses [11, 13].

The work in [10] is able to generate personalized responses given a specific speaker, which can be considered as one of the first attempts that control the generations of seq2seq models. In terms of controlling emotions, [32] tackles this problem with a sophisticated memory mechanism while [7] uses three concise but efficient models to achieve equally good performance.

3 Seq2seq with Attention

Seq2Seq is a conditional language model which takes as input source-target 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. By maximizing the probability of $P(Y|X)$, these models can generate estimated sentences $\hat{Y}$ given any input X.

Despite the variants of Seq2seq models, they usually consist of two major components: encoder and decoder. Such models can be referred as an encoder-decoder framework. The encoder will embed a source message into a dense vector representation $\boldsymbol{s}$ which is then fed into the decoder. The encoder and decoder are usually randomly initialized and jointly trained afterwards.

The decoder will generate $\hat{Y} = \hat{y_1}, \hat{y_2}, \cdots$ in an autoregressive fashion. This procedure can be described as $\boldsymbol{s} = \text{Encoder}(X)$, $\hat{Y} = \text{Decoder}(Y, \boldsymbol{s})$.

The choice of our encoder is an LSTM [6] and it can be formulated as the following:

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

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 [17]. In our experiments, we apply the state-of-the-art *FastText* [8] pre-trained model.

Adapting attention mechanism in sequence generation has shown promising improvement [1, 16]. In our case, we use the global attention with general score function [16] 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:

$$\boldsymbol{h}^{En} = [h_1^{En}, h_2^{En}, \cdots, h_m^{En}] \tag{2}$$

$$\alpha_t = \text{softmax}(h_t^{De} \text{Tanh}(W_a \boldsymbol{h}^{En})) \tag{3}$$

$$\hat{h}_t = \alpha_t \cdot \boldsymbol{h}^{En} \tag{4}$$

$$h_t^{De}, c_t^{De} = \text{LSTM}^{De}\Big(M(y_i), \big[\hat{h}_{t-1}; c_{t-1}^{De}\big]\Big) \tag{5}$$

$$y_i = \text{argmax}\Big(\text{softmax}\big(Proj(h_i^{De})\big)\Big) \tag{6}$$

$$\hat{h}_0 = h_m^{En},\ c_0^{De} = c_m^{En}$$

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.

The seq2seq with attention mechanism is shown in Fig. 2, where **E** and **D** represent two LSTM models for encoder and decoder respectively. α_t in Equation 3 is known as attention scores. According to equations (3),(4) and (5), the attention score has to be calculated at every decoding step repeatedly. In Fig. 2, the illustration of the attention layer is only for $h_1^{De} \rightarrow h_2^{De}$. $Proj(h_i^{De})$ is a linear layer that projects the hidden state of time step i to the one dimensional space of vocabulary. After normalization, e.g. softmax, the output of $Proj(h_i^{De})$ is the probability of the next token. The most possible one is considered as $\hat{y}_i$. During the training, the loss is calculated by comparing the difference between Y and $\hat{Y}$. The resulting cross entropy loss can be calculated by Equation 7,

$$H(Y, \hat{Y}) = \sum_{i=1}^{n} \Big[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \Big] \tag{7}$$

4 Emotion Injection

As mentioned above in Section 3, general seq2seq models are learning the probability of $P(Y|X)$. While in the task of controlling responses by an instructed emotion, each (X, Y) pair is assigned with an additional desired response emotion e. The goal is therefore to estimate the target sequence Y given the joint probability of $X \cap e$, which can be written as $P(Y|X \cap e)$,

4.1 Baseline models

We propose three models (*Enc-bef*, *Enc-aft* and *Dec*) in [7]. *Enc-bef* and *Enc-aft* are models that inject an emotion e in the encoder by putting special tokens before or after the input sequence X. The *Dec* model, on the other hand, puts e at each decoding step, which is similar to the method in [10]. *Dec* changes (5) to the following:

$$h_t^{De}, c_t^{De} = \mathrm{LSTM}^{De}\Big(M(y_i), [\hat{h}_{t-1}; c_{t-1}^{De}; v_e]\Big) \tag{8}$$

In Equation (8) v_e is randomly initialized and trainable vector for each of the emotions. To be more precise, we will refer to the *Dec* model as *Dec-rep* in the follow text.

4.2 Proposed models

The Encoder-Decoder framework provides us with a flexible foundation which makes joining additional modules into it straightforward and intuitive. In this work, inspired by [7] and [32], four models are proposed.

Dec-start In [7] *Enc-bef* and *Enc-aft* models have been shown to be successful and effective. By creating a special token T_e for every emotion, these two methods are essentially modifying X to $\left[T_e; X\right]$ or $\left[X; T_e\right]$ in both training and evaluating. As shown in Fig. 2, to start decoding, a special token <*s*> is fed into the decoder. h_1^{De} is obtained by calculating $LSTM^{De}(M(\text{<s>}), [h_m^{En}, c_m^{En}])$.

In this Dec-start model, we simply substitute the start token <*s*> with an emotion token T_e as shown in Equation (9).

$$h_1^{De} = LSTM^{De}\Big(M(T_e), \big[h_m^{En}, c_m^{En}\big]\Big) \tag{9}$$

Dec-trans As an alternative, we multiply the h_t^{De} with another matrix to transform the hidden state of time t with respect to the emotion to be expressed. Denote the transforming matrix as $Trans_e$, Equation 6 is changed as following:

$$y_i = \mathrm{argmax}\Bigg(\mathrm{softmax}\Big(Trans_e\big(Proj(h_i^{De})\big)\Big)\Bigg) \tag{10}$$

Dec-proj [32] also proposed an external memory which maps the hidden state h_t^{De} into a slightly different vocabulary space for each of the emotions. By taking a step forward, we propose a *Dec-proj* model which will make h_t^{De} to totally independent vocabulary spaces. This is done by making unique projection layer $Proj_e$ for each of the emotion. Equation 6 is thus changed to the following:

$$y_i = \mathrm{argmax}\Big(\mathrm{softmax}\big(Proj_e(h_i^{De})\big)\Big) \tag{11}$$

Dec-att The attention mechanism has been proven to be very powerful in many sequence to sequence tasks. [28] even outperformed many tasks by using an attention only encoder-decoder model. [16] proposed three methods to calculate the attention score. The one we chose in Equation 3 is referred to as *general* score in their paper. It is a parameterized method compared to *dot* score. Since the general attention has individual parameters, making different attention layer for different emotion is possible as well. The *Dec-att* model changes Equation 3 to the following:

$$\alpha_t = \text{softmax}(h_t^{De} \text{Tanh}(W_e \boldsymbol{h}^{En})) \tag{12}$$

Compared to the original equation, the universe matrix for calculating attention score W_a is replace with W_e for each of the given emotion e. Fig. 2 shows where the emotion is injected into standard seq2seq with attention model.

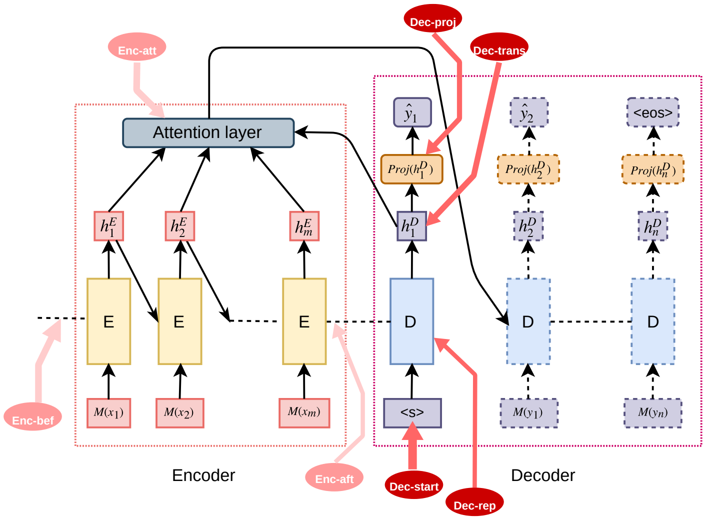


Fig. 2. Comparison among the four proposed models and three baseline models

5 Dataset

As mentioned in Section1, there is no dataset that contains pairs of dialogue exchanges and corresponding emotions. But there are categorical datasets for emotion classification in text. For example, *Cleaned Balanced Emotional Tweets (CBET)* dataset [31, 24], *Twitter Emotion Corpus (TEC)* [19] and the *International Survey on Emotion Antecedents and Reactions (ISEAR)* [22]. There are some other dimensional datasets but they can not directly fit into this task. Table 1 shows more details about the aforementioned datasets.

Table 1. Details of three categorical datasets for emotion classification

Dataset	# of categories	# instances	Emotions
CBET	9	81,163	anger, surprise, joy, love, sadness, fear, disgust, guilt , thankfulness
TEC	6	21,051	anger, disgust, fear, joy, sadness, surprise
ISEAR	7	7,666	joy, fear, anger, sadness, disgust, shame, guilt

Considering the size of the datasets and also for consistency with the work in [7], we choose the CBET dataset to train a text classifier and use that to tag the corpus which is used to train a dialogue model.

By applying a bidirectional LSTM [5] model with self-attention [14] We achieve a slightly better results than that in [7], which has a F1-score of 54.33% with precision of 66.20% and recall of 51.29%.

The OpenSubtitles dataset [15] is one of the largest and most popular dataset to train open domain dialogue systems. Following the work in [7], we use the pre-processed data by [10] and further remove duplicate lines. The total number of utterances is 11.3 million, each utterance has at least 6 words.

In addition, a threshold is applied to approximate a *Non-emotion* category. This means, in evaluation, if the most possible emotion of an instance still has a very low 'confidence', this instance would be considered as not containing any of those emotions. In our experiment, by setting the threshold to 0.35, approximately 35% of the sentences are below the threshold. *Non-emotion* is treated as a special emotion when training the dialogue models, but it is not considered in the evaluation.

6 Experiments

6.1 Parameters setup

For the purpose of comparison, the parameters of the proposed models are set to as close to the baseline models as possible. The dimension for both encoder LSTM and decoder LSTM is 600. The dropout ratio is 0.75. The choice of optimizer is Adam [9] with learning rate set to 0.0001. The number of the vocabulary is set to 25,000. *FastText* [8] pre-trained word embedding model is used and set to trainable. The size of held-out test set is 50k samples. The training and development split ratio is 0.95 to 0.05. The padding length is 30.

6.2 Evaluation metric

The main goal of this research is to check the ability of generating responses while given a specific emotion. That being said, the quality and relevance of generated responses are not the focus of this research. Hence, our interest lies in checking if the generated sentences contains the instructed emotions or not. The size of the test set is 50k and 7 models are evaluated: 3 baselines and our 4 proposed models. Further more, every instance in the test set is assigned to 9 emotions for the model. Thus, a total of 3,150k ($50k \times 7 \times 9$) responses are generated for evaluation.

Fortunately, unlike the work by [10], expensive human evaluation is not needed. Instead, we evaluate the output using the emotion mining classifier again. Since every source sentence we generate 9 responses (i.e. one for each emotion), for each emotion category, we check the proportion of the responses where the corresponding emotions are indeed expressed. Such proportion is considered as the *estimated accuracy*. Therefore, for each model, we can obtain 9 *estimated accuracy* scores for the 9 emotions.

7 Results and Discussion

7.1 Result analysis

The *estimated accuracy* scores of the 7 models are shown in Table 2. Moreover, we draw the confusion matrices of the 4 proposed models to show the misclassification errors (Fig. 2). For the confusion matrices of the 3 baseline models, please refer to [7].

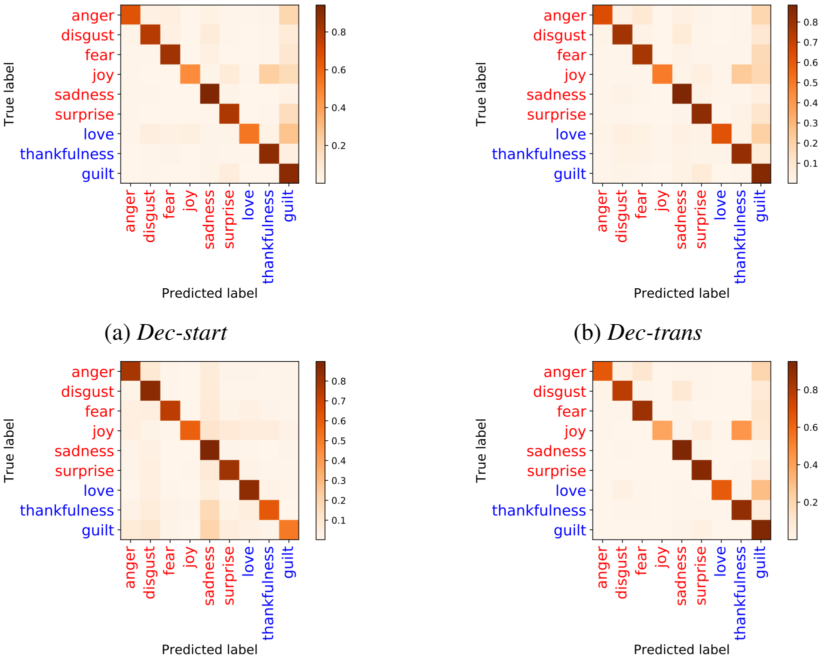
Table 2. Per class accuracy of generated response

Emotion	*Enc-bef*	*Enc-aft*	*Dec-rep*	***Dec-start***	***Dec-trans***	***Dec-proj***	***Dec-att***
anger	60.18%	62.30%	67.95%	66.81%	64.27%	**78.48**%	65.09%
disgust	77.98%	76.79%	79.02%	78.42%	78.33%	**86.43**%	78.29%
fear	**86.40**%	84.17%	83.52%	84.10%	77.15%	73.70%	86.00%
joy	45.69%	41.15%	48.30%	47.42%	49.69%	**59.12**%	38.71%
sadness	94.19%	93.98%	94.21%	94.18%	88.42%	89.83%	**95.09**%
surprise	84.47%	85.09%	87.21%	80.55%	83.61%	80.56%	**92.54**%
love	56.38%	54.69%	58.32%	54.25%	62.82%	**85.14**%	64.56%
thankfulness	87.69%	89.31%	**90.83**%	89.44%	82.03%	61.80%	89.11%
guilt	93.19%	92.17%	91.20%	90.68%	86.64%	50.92%	**94.40**%
Average	76.24%	75.52%	77.84%	76.21%	74.77%	74.00%	**78.20**%

From the table, we can see that despite the fact that the *Dec-att* model only achieves 38.71% accuracy for the emotion *joy*, it still outperforms the others on most of the emotions. From Fig. 2, one can observe a significant mismatch between *joy* and *thankfulness*. Instead of expressing *joy*, *Dec-att* conveys *thankfulness* which could also be considered reasonable. However, *love* is also confused with *guilt*. Note that the measured accuracy is also subject to the accuracy of the emotion tagger used.

It is also noticeable that the performance of model *Dec-start* is close to that of models *Enc-bef* and *Enc-aft*. This is expected considering the models are simply injecting the information of emotions by only one special token. The highlighted numbers in Table 2 show the best accuracy of each emotion. Model *Dec-rep*, *Dec-proj* and *Dec-att* have at least 2 best scores whereas the others almost have none.

To compare the extended emotion model with Ekman's basic emotions, we highlight the two group of emotions in both Fig. 3 and Table 2. The emotions in red are the six basic emotions, the blue ones are those added by [31].



(a) *Dec-start*

(b) *Dec-trans*

(c) *Dec-proj*

(d) *Dec-att*

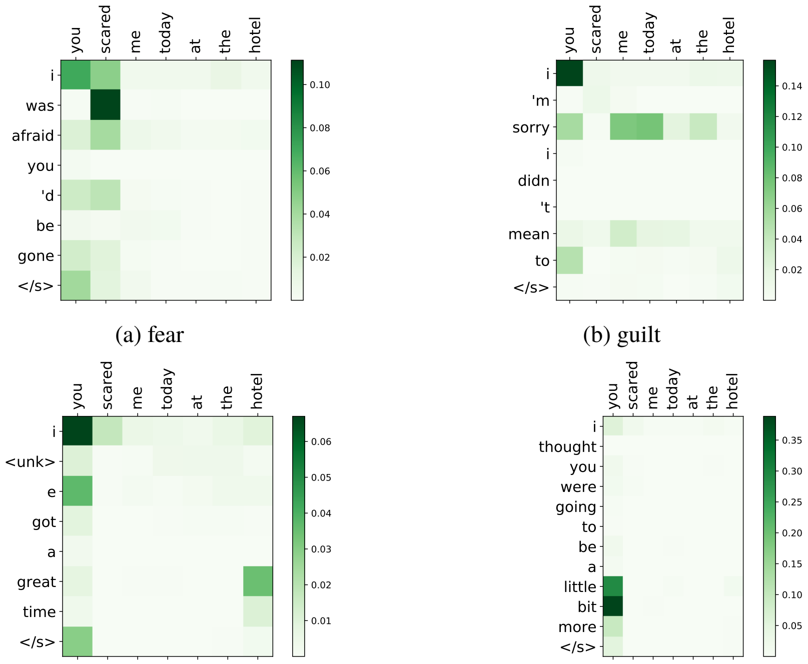
Fig. 3. Confusion matrices of 4 proposed models

7.2 Dec-att model visualization

To show how the *Dec-att* model works, we chose an example utterance and show how the attention scores vary with respect to responses with different emotions. The attention is visualized by heatmaps in Fig. 4. To respond to the utterance *You scared me today at the hotel*, the model focused on *scared* when expressing *fear*. When conveying *guilt*, except for focusing on the pronoun *you*, it focused on *me* and *today* and show a strong preference to using the word *sorry*. When responding with *joy*, it focused on the word *hotel*. Interestingly, to response to the utterance with *sadness*, the model did no pay attention to any words except for the pronoun, but it did try to answer with the phrase *little bit more*.

7.3 Parameter cost

Apart from the performance of the models, another important comparison of deep learning models is their sizes. Considering that all the 7 models are based on the basic seq2seq with attention model. We only need to compare the additional parameters that are needed. Let's denote the size of vocabulary space as $|V|$, the length of source sentences as m, the dimension of the decoder LSTM as D, and the number of emotions as S. The comparison of the models in terms of these parameters is shown in Table 3.



(a) fear

(b) guilt

(c) joy

(d) sadness

Fig. 4. An example of the attention scores of *Dec-att* model

It has to be mentioned that S in our experiments is 10: 9 emotions plus an *non-emotion* category.

Table 3. Comparison of the models in terms of additional space for required parameters

Model	Additional para. in symbols	Additoinal para. in our exp.
Enc-bef	0	0
Enc-aft	0	0
Dec-rep	$D \times S$	6,000
Dec-start	0	0
Dec-trans	$D \times D \times S$	3,600,000
Dec-proj	$\|V\| \times D \times S$	150,000,000
Dec-att	$m \times D \times S$	180,000

From the above table, *Dec-proj* is the least cost efficient model. *Dec-rep* and *Dec-att* are both outperforming models considering their performance.

8 Conclusion and Perspectives

In this work, we propose four models that are able to automatically generate a response while conveying a given emotion. We compare our models with the baseline models in [7] in terms of both performance and efficiency. Our *Dec-att* model outperforms the strongest baseline and we show how it works using attention heatmaps. *Dec-rep* and *Dec-att* turn out to be both effective and efficient.

However, in this work, we did not experiment with any combinations of the models. It is shown in [32] that the combination of external and internal memory outperforms each of the single model. We think the combination of *Dec-rep* and *Dec-att* has a potential to give a better result. One major limitation of this work is that we heavily rely on the accuracy of the emotion mining classifier and assume it is of acceptable accuracy. Moreover, the main effort of this research lies on generating responses accurately and efficiently but without focusing on properties like grammar, relevance and diversity.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 (2014)
2. Bessho, F., Harada, T., Kuniyoshi, Y.: Dialog system using real-time crowdsourcing and twitter large-scale corpus. In: Proc. of the Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 227–231. Association for Computational Linguistics (2012)
3. Bickmore, T.W., Picard, R.W.: Establishing and maintaining long-term human-computer relationships. ACM Transactions on Computer-Human Interaction **12**(2), 293–327 (2005)
4. Ekman, P., Friesen, W.V., Ellsworth, P.: Emotion in the Human Face: Guide-lines for Research and an Integration of Findings. Pergamon (1972)
5. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional lstm networks for improved phoneme classification and recognition. Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005 pp. 753–753 (2005)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
7. Huang, C., Zaiane, O., Trabelsi, A., Dziri, N.: Automatic dialogue generation with expressed emotions. In: Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics. vol. 2, pp. 49–54 (2018)
8. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv:1612.03651 (2016)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
10. Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., Dolan, B.: A persona-based neural conversation model. In: Proc. of the Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 994–1003 (2016)
11. Li, J., Monroe, W., Jurafsky, D.: Data distillation for controlling specificity in dialogue generation. arXiv:1702.06703 (2017)
12. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D.: Deep reinforcement learning for dialogue generation. arXiv:1606.01541 (2016)
13. Li, J., Monroe, W., Shi, T., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. arXiv:1701.06547 (2017)
14. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv:1703.03130 (2017)

15. Lison, P., Tiedemann, J.: Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles (2016)
16. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1412–1421 (2015)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
18. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. pp. 1003–1011. Association for Computational Linguistics (2009)
19. Mohammad, S.M.: # emotional tweets. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. pp. 246–255. Association for Computational Linguistics (2012)
20. Parrott, W.G.: Emotions in social psychology: Essential readings. Psychology Press (2001)
21. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the conference on empirical methods in natural language processing. pp. 583–593. Association for Computational Linguistics (2011)
22. Scherer, K.R., Wallbott, H.G.: Evidence for universality and cultural variation of differential emotion response patterning. Journal of personality and social psychology **66**(2), 310 (1994)
23. Serban, I.V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A.C., Bengio, Y.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: AAAI. pp. 3295–3301 (2017)
24. Shahraki, A.G., Zaiane, O.R.: Lexical and learning-based emotion mining from text. In: Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing (2017)
25. Shaver, P., Schwartz, J., Kirson, D., O'connor, C.: Emotion knowledge: Further exploration of a prototype approach. Journal of personality and social psychology **52**(6), 1061 (1987)
26. Silva, J., Coheur, L., Mendes, A.C., Wichert, A.: From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review **35**(2), 137–154 (2011)
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
29. Vinyals, O., Le, Q.: A neural conversational model. arXiv:1506.05869 (2015)
30. Xu, W., Rudnicky, A.I.: Task-based dialog management using an agenda. In: Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3. pp. 42–47. Association for Computational Linguistics (2000)
31. Yadollahi, A., Shahraki, A.G., Zaiane, O.R.: Current state of text sentiment analysis from opinion to emotion mining. ACM Computing Surveys (CSUR) **50**(2), 25 (2017)
32. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: Emotional conversation generation with internal and external memory. In: Proc. of the AAAI Conference on Artificial Intelligence (2018)
33. Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., Xu, B.: Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. arXiv:1611.06639 (2016)