A Cue Adaptive Decoder for Controllable Neural Response Generation

Weichao Wang
School of Computer Science and Engineering, Northeastern University
Shenyang, China
xinghuashuying@126.com

Shi Feng
School of Computer Science and Engineering, Northeastern University
Shenyang, China
fengshi@cse.neu.edu.cn

Wei Gao
School of Information Systems, Singapore Management University
Singapore
weigao@smu.edu.sg

Daling Wang
School of Computer Science and Engineering, Northeastern University
Shenyang, China
wangdaling@cse.neu.edu.cn

Yifei Zhang
School of Computer Science and Engineering, Northeastern University
Shenyang, China
zhangyifei@cse.neu.edu.cn

ABSTRACT

In open-domain dialogue systems, dialogue cues such as emotion, persona, and emoji can be incorporated into conversation models for strengthening the semantic relevance of generated responses. Existing neural response generation models either incorporate dialogue cue into decoder's initial state or embed the cue indiscriminately into the state of every generated word, which may cause the gradients of the embedded cue to vanish or disturb the semantic relevance of generated words during back propagation. In this paper, we propose a Cue Adaptive Decoder (CueAD) that aims to dynamically determine the involvement of a cue at each generation step in the decoding. For this purpose, we extend the Gated Recurrent Unit (GRU) network with an adaptive cue representation for facilitating cue incorporation, in which an adaptive gating unit is utilized to decide when to incorporate cue information so that the cue can provide useful clues for enhancing the semantic relevance of the generated words. Experimental results show that CueAD outperforms state-of-the-art baselines with large margins.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**; *Natural language processing*; Natural language generation.

KEYWORDS

dialogue generation, vanishing gradient problem, disturbing gradient problem, cue adaptive decoder

ACM Reference Format:
Weichao Wang, Shi Feng, Wei Gao, Daling Wang, and Yifei Zhang. 2020. A Cue Adaptive Decoder for Controllable Neural Response Generation. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380008

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380008

1 INTRODUCTION

Non-task-oriented dialogue systems aim to develop chatbots capable of conversing with humans naturally in the open domains [15]. Most previous studies focused on the single-turn conversation by utilizing encoder-decoder framework [11, 19, 21]. Some systems also tried to incorporate semantic information and context dependency by considering dialogue history [16–18, 20, 22]. Because of the limited capacity of the models in capturing useful semantic relevance among various utterances, many of such methods may generate dull and uncontrollable responses. Recently, different kinds of dialogue cues, such as persona, emotion, emoji, etc., were considered for controlling and guiding the generation with an attempt to keep them on a par with desirable responses [10, 24–26].

Dialogue cue is referred to as a kind of prior information that can provide certain guidance and interpretability to the generation. Typically each generation is given a specific type of cue as input for the response generator to consider. For example, Zhou et al. [26] provided a Twitter conversation dataset containing 64 fine-grained emojis as emotion cues, and each response is emotionally consistent with a provided emoji. Different strategies have been developed to utilize cues in the decoding process. A cue can be incorporated either directly into decoder's initial state [26] or instead into every time step during generation [10]. These approaches inevitably result in critical gradient-related issues. Incorporating a cue into decoder's initial state can make the introduced cue suffer from vanishing gradient problem (dubbed as vanishing cue gradient, which is incurred similarly as the general vanishing gradient [2]). On the other hand, introducing a cue at each time step of decoder ignores the fact that not all the generated words rely on the cue, and the gradients on those cue-irrelevant words could distort the representation learning of the introduced cue, which we call the problem of disturbing cue gradient.

Semantic relevance obtained from conversation context plays a key role in the response generation process. Cues can provide useful hints that will enhance the semantic relevance for generating controllable and meaningful responses. Table 1 shows an example of single-turn conversation incorporated with a cue of "*sadness*" emotion used to control the generation of a response conveying sadness corresponding to "lonely life". Intuitively, the first candidate response is preferable which appears to be emotional and

● **cue-relevant word** ● **cue-irrelevant word**

Triggering message
I think I will always be alone, living a lonely life.
Response candidates (cue: *sadness* emotion)
A lonely person can't afford to get hurt! ✓ What happened? X

Table 1: An example of conversation incorporated with a cue of "*sadness*" emotion. The first candidate is a more natural and emotional response by containing sadness-relevant words.

compassionate. Meanwhile, not all the generated words rely on the emotion, such as those cue-irrelevant words marked in red color. In general, to learn better response representation, it is essential to endow the decoder with a capacity to decide whether to resort to the cue or not at each time step, which has not been well studied.

In this paper, we work towards controllable conversation response generation task, for which we try to incorporate dialogue cue in a way more reasonable and effective. We focus on learning cue's representation and incorporation in the decoding process for generating each word regardless of specific encoding method used, and thus we assume that the decoder's initial state has been available with the encoding operation. A novel Cue Adaptive Decoder (CueAD) is proposed to embed dialogue cue and decoder neural network in a unified framework. Specifically, the cue representation is modeled by a revised GRU model, which stores adaptive guidance information used to facilitate the incorporation of the cue. We design an adaptive gating unit to determine the degree of the cue's participation so that CueAD could automatically decide when to use the cue information at each decoding step. As a result, the cue vector can also be learned with the words reflecting their essential meaning and incorporated at the right places, which consequently mitigates the impacts of vanishing/disturbing cue gradients.

Our contributions are summarized into three folds:

- We propose an extensible framework to learn and incorporate dialogue cue effectively for controllable response generation. Our method is generic. By configuring different types of cue, it can be easily applied to various neural network-based response generation systems.
- We extend GRU network with a cue representation to store adaptive cue information, and propose an end-to-end CueAD model to control when and to what extent to incorporate the cue, which can provide appropriate semantic clues to better guide and interpret the response generation.
- Our proposed model consistently outperforms the strong baseline methods on two real-world conversation datasets.

2 RELATED WORK

Most existing studies for response generation obtain semantic relevance only from dialogue content without exploring any external prior knowledge. In single-turn dialogue generation, Shang et al. [19] proposed a Neural Response Machine with encoder-decoder framework. Gu et al. [7] designed a COPYNET model for response generation. Zhang et al. [23] developed a controllable method to handle different utterance-response relationships by focusing on specificity. In multi-turn dialogue generation, Serban et al. [17] explored using dialogue history with a hierarchical recurrent encoder-decoder neural network. Xing et al. [22] proposed a hierarchical attention model for generating response by selecting the most important utterance and words from multi-turn utterances. In these models, the response only conditions on the triggering utterance, which may lead to dull or uncontrollable responses due to their limited capacity of capturing semantic relevance.

To enhance semantic relevance, researchers introduced dialogue cues with different types of cues. In single-turn dialogue generation, Li et al. [10] proposed a speaker model to incorporate persona cue for handling the issue of speaker consistency. Zhou and Wang [26] incorporated emotion cue based on Conditional Variational Autoencoders (CVAE) for generating emotional language. Zhou et al. [25] embedded emotion cue into a dialogue model with internal and external memory. Cues are also utilized in multi-turns dialogue generation. Zhao et al. [24] studied a cue-guided CVAE model using dialogue act as a cue. Shen et al. [20] built an SPHRED model to incorporate sentiment cue for controlling responses. In these methods, however, cues failed to work effectively due to vanishing and disturbing gradient problems. To the best of our knowledge, our proposed approach is the first to introduce and apply the adaptive cue gating unit for improving end-to-end response generation.

3 OUR PROPOSED MODEL

3.1 Problem Description

Assume that a dialogue consists of triggering utterance(s) $U = u_1 u_2 \ldots u_M$, and a response utterance $Y = y_1 y_2 \ldots y_N$ which is associated with a cue index $c \in C$ indicating a specific type of cue in a cue set of C, such as emotion, persona, emoji, etc. Note that u_j is the j-th word in triggering utterance(s), and y_i is the i-th word in response utterance. Typically, utterances in a dialogue are tied to some fixed cue type, which provides a general semantic hint for response generator to use. Single-turn conversation contains only one trigger-response pair of utterances while multi-turns conversation contains historical interaction between participants consisting of multiple turns of utterances with more contextual information.

Let the cue c be represented as an embedding h_c, which is to be learned by the response generator to represent the specific attributes of the cue. We focus on the decoding (i.e., response generation) process by estimating the generation probability of current word $y_i \in Y$ at time i, i.e., $p(y_i|y_1, \ldots, y_{i-1}, U, h_c)$, for producing controllable response utterance based on the cue. Note that the cue index is known in advance and cue type is assumed invariable in the entire generation process, but the cue representation h_c needs to be learned. We try to generate an appropriate response not only semantically relevant to U but also favorably consistent to c.

3.2 Incorporate Cue into Encoder-Decoder

In a traditional encoder-decoder model, the conditional probability of generating a word y_i at time i is defined as:

$$p(y_i|y_1, ..., y_{i-1}, U) = \mathrm{softmax}(W \cdot [h_i, c_i] + b) \tag{1}$$

where h_i is the hidden state in the i-th time step, c_i is the context vector to allow the decoder to pay different attention to different

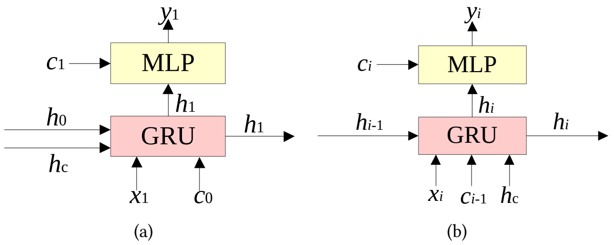


Figure 1: Two methods incorporating dialogue cue.

parts of input at different steps [1], W and b are trainable parameters. After softmax, we get a distribution in a V-dimensional vector for predicting the generated word y_i, and V is the vocabulary size.

Let x_i denote the current input word at time i despite the fact that it may take a different value in training and test. Each x_i is first mapped to its embedding, and the decoder state at time i is represented as a hidden vector h_i which is generally computed by:

$$h_i = f(h_{i-1}, [x_i, c_{i-1}]) \tag{2}$$

where $f(\cdot)$ is the process that can be defined specifically in Eq. 3 below. Note that when training the decoder, x_i takes as input the (i-1)-th target word $\hat{y}_{i-1}$ from the gold utterance, however, when testing the input instead becomes the generated word y_{i-1} from the previous generation step. We use GRU [4] model for word generation process which calculates h_i as:

$$\begin{aligned} r_i &= \sigma(W_r \cdot [h_{i-1}, [x_i, c_{i-1}]]) \\ z_i &= \sigma(W_z \cdot [h_{i-1}, [x_i, c_{i-1}]]) \\ \tilde{h_i} &= \tanh(W_h \cdot [r_i \odot h_{i-1}, [x_i, c_{i-1}]]) \\ h_i &= (1 - z_i) \odot h_{i-1} + z_i \odot \tilde{h_i} \end{aligned} \tag{3}$$

where $\odot$ is element-wise multiplication, r_i is a reset gate, z_i is an update gate, $\sigma(\cdot)$ is a sigmoid function, and W_r, W_z, W_h are parameters. Note that h_0 is initialized with the *encoder state*. For each generated utterance, the cross-entropy loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_i = \sum_{i=1}^{N} - \hat{y}_i \log(y_i) \tag{4}$$

where N is number of steps in decoder, $\hat{y}_i$ is the ground-truth distribution of the i-th target word, and y_i is the predicted distribution the i-th generated word with Eq. 1.

A cue can be designated to guide the generation of words consistent to the desired response. Zhou et al. [26] input emotion cue into decoder's initial state. Li et al. [10] input persona as cue into each time step of the model's decoder process. Suppose the cue embedding h_c is available, these two cue incorporation methods are illustrated in Figure 1.

In Figure 1(a), the concatenation of cue embedding h_c and encoder state h_0 is regarded as decoder's initial state. Given the input word embedding x_i, the h_i is computed as:

$$h_i = \begin{cases} f([h_{i-1}, h_c], [x_i, c_{i-1}]) & i = 1 \\ f(h_{i-1}, [x_i, c_{i-1}]) & i > 1 \end{cases} \tag{5}$$

Vanishing cue gradient problem: As the cue is incorporated into the initial state only, the gradient of cue tends to vanish in the long run similarly as the general vanishing gradient problem of recurrent neural networks [2]. As a result, semantic hints introduced by the cue could not provide useful clues for producing forward tokens, and its effect tends to diminish down the line.

In Figure 1(b), the input of each step is the concatenation of x_i and h_c, and h_i is computed by:

$$h_i = f(h_{i-1}, [x_i, h_c, c_{i-1}]) \tag{6}$$

In this model, the cue participates in the generation of each word and vanishing gradient could be avoided. Nevertheless, the indiscriminate involvement of cue could lead to disturbing gradient, allowing the cue-irrelevant words to mislead the learning process.

Disturbing cue gradient problem: As Figure 1(b) shows, y_i depends on h_i while h_i depends on h_c. During error back-propagation when learning h_c, the gradient w.r.t. h_c is calculated as:

$$\frac{\partial \mathcal{L}}{\partial h_c} = \sum_{i=1}^{N} \frac{\partial \mathcal{L}}{\partial \mathcal{L}_i} \cdot \frac{\partial \mathcal{L}_i}{\partial y_i} \cdot \frac{\partial y_i}{\partial h_i} \cdot \frac{\partial h_i}{\partial h_c} \tag{7}$$

where h_i is defined in Eq. 6, and the gradient of h_c in each step is adjusted by $\frac{\partial h_i}{\partial h_c}$ with coefficient $\frac{\partial \mathcal{L}}{\partial \mathcal{L}_i} \cdot \frac{\partial \mathcal{L}_i}{\partial y_i} \cdot \frac{\partial y_i}{\partial h_i}$. However, it is unreasonable to adjust h_c's gradient in every step. If the target word to be generated is a cue-irrelevant word (e.g. "the" or "of"), which affects h_c's gradient, the model will be misled to learn that the cue-irrelevant word is strongly related to h_c. As a result, cue-irrelevant words would disturb the learning of h_c. If the desired cue is altered, say from "sadness" to "happiness", since both of the cue representations (i.e., h_c) are adjusted by the same set of sentiment-irrelevant words, the learned h_c of "happiness" tends to be similar as that of "sadness" which is not desired.

3.3 Cue Adaptive Decoder (CueAD)

Existing models suffer from vanishing gradient and disturbing gradients of the introduced cue because they lack ability to distinguish whether the word to be generated needs the guide from the cue. Therefore, we introduce a new adaptive cue representation, which is a hidden neural state for facilitating the cue incorporation properly in response generation. With the adaptive cue representation, we endow the decoder with the capability to know when to use the learned cue embedding in the time steps of decoding process.

What is adaptive cue representation? Since the cue-related information is strongly connected with words to be generated in each decoding step that could reflect the characteristic of the cue, in the current time step of decoding, we want to adaptively incorporate cue information so that the model can choose not to use the cue when the target word does not depend on the cue information. We call such properly incorporated cue an *adaptive cue representation*. By extending GRU to GRU+ with an adaptive gating unit, we can extract the *adaptive cue representation*, which is illustrated in Figure 2. Specifically, based on the GRU model (see Eq. 3), we add an adaptive gate β_i using a multi-layer perceptron network and a sigmoid function $\sigma(\cdot)$, and concatenate the decoder semantic h_i, word embedding x_i, and context vector c_i as the input of the MLP network. The adaptive cue representation a_i is defined as:

$$\begin{aligned} \beta_i &= \sigma(MLP_\beta[h_i, x_i, c_i] + b_\beta) \\ a_i &= \beta_i \times h_c \end{aligned} \tag{8}$$

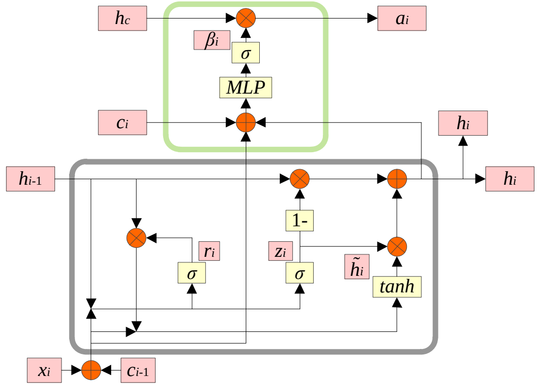


Figure 2: GRU+: Cue adaptive GRU model.

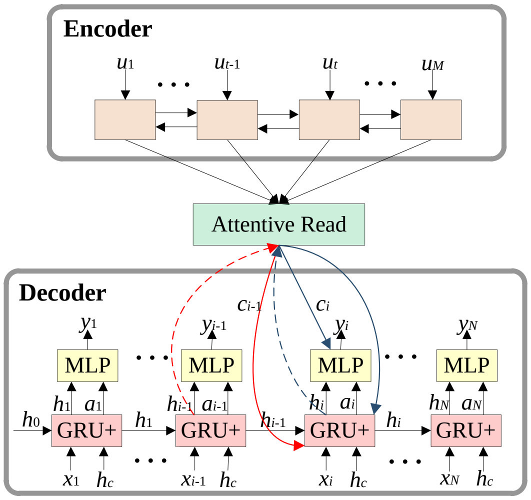


Figure 3: The CueAD model with GRU+.

where the output dimension of MLP_β is 1, β_i is a scalar whose value is in the rage of [0,1] indicating the importance of cue at step i. Intuitively, we use gate β_i to control the extent of incorporation of cue representation h_c in the i-th step, and the obtained adaptive cue representation a_i is used to generate target word.

With the adaptive cue representation, we propose our CueAD model as shown in Figure 3. Since the semantics of dialogue content in our training data is generally compatible with the provided cue, we can learn the cue representation h_c consistently from data. We initialize h_c randomly, which is then continuously updated and embedded with CueAD for capturing the semantic hints of the designated cue to strengthen its semantic relevance for better text generation.

In our model, the probability of a word to be generated is governed by decoder semantic h_i, context vector c_i and adaptive cue representation a_i. The conditional probability of y_i is defined as:

$$p(y_i|y_1, \cdots, y_{i-1}, U, h_c) = \text{softmax}(W \cdot [h_i, c_i, a_i] + b) \qquad (9)$$

In this way, our model can decide not only whether or not but also to what extent to incorporate the cue.

In the training process, the value of β_i is adaptively adjusted in generation steps to obtain appropriate cue representation h_c. In the inference process, the model totally depends on itself where the input is from its previous output because of the non-availability of the actual response, which is known as exposure bias problem [14]. A bad output at a certain step can affect the semantics of h_i and c_i in future generation, and further affect the calculation of adaptive gating value β_i. As a consequence, the cue information cannot be incorporated effectively. During inference, to relieve the impact of exposure bias problem on cue incorporation, and better control and interpret the response generation, we set the value of β_i to 1 in all steps to take full advantage of learned cue representation.

Why can CueAD mitigate vanishing gradient and disturbing gradients of cue? With the adaptive gating unit, CueAD can decide when to incorporate the cue information in training process. Incorporating cue at appropriate time steps in the generation intermittently amplifies the signal. Hence, its vanishing gradient can be mitigated. Furthermore, both h_i and a_i contribute to the generation of y_i (see Figure 3 and Eq. 9), but only a_i depends on h_c. So, the gradient w.r.t. h_c is computed as:

$$\frac{\partial \mathcal{L}}{\partial h_c} = \sum_{i=1}^{N} \frac{\partial \mathcal{L}}{\partial \mathcal{L}_i} \cdot \frac{\partial \mathcal{L}_i}{\partial y_i} \cdot \frac{\partial y_i}{\partial a_i} \cdot \left(\frac{\partial a_i}{\partial h_c} = \beta_i \right) \qquad (10)$$

which indicates that the gradient of h_c depends on β_i, that is, h_c's gradient can be adjusted by the importance of cue for generating y_i. Specifically, if the target word to be generated is cue-irrelevant, β_i would be close 0, thus avoiding adjusting h_c's gradient. If the target word to be generated is cue-relevant, e.g. "lonely", β_i would be close to 1, thus resulting in using the cue. In this way, h_c can be learned only by those words reflecting its essential semantics without being interfered by cue-irrelevant words, thus mitigating the disturbing gradient problem.

4 EVALUATION

4.1 Datasets and Experimental Setup

Single-turn twitter corpus tagged with emoji labels (SE) was released by [26], which has 642,159/10,000/10,000 single-turn conversation pairs for training/validation/test with 64 common emojis naturally labeled. The responses of this corpus are labeled with 64 common emojis. Each dialogue contains 25.92 words on average.

Furthermore, to verify the performance in multi-turns conversation, we construct a **multi-turn emotional dialogue corpus (ME)** by crawling Reddit[1] comments to train our model. The ME dataset[2] is constructed in three steps:

1) **Building an Emotion Classifier:** We collect 72,165 reddit comments tagged with emojis to train the emotion classifier. We consider 5 emotion categories: *Disgust*, *Happy*, *Like*, *Anger* and *Other*, and each category has 14,433 comments. We partition the dataset into training, validation, and test sets with a 8:1:1 ratio. We train a Bi-LSTM emotion classifier with an accuracy of 0.657.

2) **Dialogue Data Filtering:** We remove the dialogues shorter than 4 turns and longer than 7 turns, and remove the utterances with less than 5 words and more than 30 words. After this, there

[1] http://www.reddit.com

[2] https://drive.google.com/drive/folders/1Fu6K32WvY5raLg_H5FouhyYLEQ3luAV3

Corpus	Models	embed-avg	greedy	distinct-1	distinct-2	cue accuracy
SE	Seq2seq	0.581	0.429	0.006	0.017	0.371
	Cue-I (emoji)	0.584	0.432	0.006	0.018	0.550
	Cue-E (emoji)	0.593	0.439	0.006	0.023	0.561
	Cue-IE (emoji)	0.595	0.440	0.006	0.024	0.561
	CueAD (emoji)	**0.619**	**0.462**	**0.008**	**0.027**	**0.573**
ME	Seq2seq	0.672	0.484	0.011	0.045	0.298
	Cue-I (emotion)	0.674	0.485	0.010	0.043	0.531
	Cue-E (emotion)	0.685	0.499	0.012	0.054	0.619
	Cue-IE (emotion)	0.686	0.501	0.013	0.056	0.620
	CueAD (emotion)	**0.694**	**0.509**	**0.015**	**0.065**	**0.681**

Table 2: Effect of CueAD on SE and ME corpus with different cue information.

are 833,178 dialogues left, and the average number of words in context/response is 96.59/24.40 respectively, and the average number of context dialogue turns is 5.27. Then, we randomly sample 15,000 dialogues for validation and another 15,000 dialogues for test.

3) **Annotation Dialogue Dataset with Emotion:** We use the well trained Bi-LSTM emotion classifier to annotate the filtered conversation dataset. This cue annotation strategy using a pre-trained classifier follows the similar methods in [25] and [24].

For both corpora, we use the 300d Glove embeddings [13] pre-trained on Wikipedia as word embeddings, and we use top 40,000 frequent words as the vocabulary. We empirically set the size of cue embedding to 50, and the size of Bi-GRU encoder and decoder states are set to 300. We set the size of mini-batch to 30. All datasets are tokenized using the NLTK tokenizer [3], and all the initial weights are sampled from a uniform distribution [-0.08, 0.08]. We optimize our model using Adam [8] with learning state of 0.0001 on SE and 0.0002 on ME, and gradient clipping is set to 5. The beam search method is adopted and the size is set to 5.

4.2 Baselines

For all baselines, we use sequence-to-sequence framework based on GRU with attention mechanism for conversation generation. Each baseline method has its specific way to incorporate cue information.

Cue applied in the initial state (Cue-I): The cue is only applied in decoder's initial state with a method following Figure 1(a). This way of incorporating cue information is similar to [26].

Cue applied in every step (Cue-E): The cue information is applied in every decoding time step with a method following Figure 1(b). This way of cue incorporation is similar to [10].

Cue applied in the initial state and every step (Cue-IE): The cue is applied in not only decoder's initial state but also every decoding time step. This way of incorporation is similar to [24].

4.3 Evaluation Metrics

Distinct: The Distinct1 (Distinct2) [9] is the ratio between the number of distinct unigrams (bigrams) in generated responses and the total number of generated unigrams (bigrams), which is used to evaluate the diversity of generated responses.

Embedding-based metrics: We use embedding average (*embed-avg*) and greedy matching (*greedy*) [12] to evaluate the topic similarity between generated responses and ground-truth responses.

Cue accuracy: We define cue accuracy as the agreement between the expected cue category and the cue category of the generated response predicted by a pre-trained classifier. For SE corpus, we apply the pre-trained DeepMoji classifier (of which the accuracy is 0.585) [5], and use top-5 accuracy to evaluate whether the 5 most likely categories contain the label category. For ME corpus, we apply the pre-trained Bi-LSTM classifier previously used for dialogue annotation, and use top-1 accuracy for evaluation. The higher cue accuracy reflects better controllability and interpretability.

Human evaluation: We recruit three Master's students who are independent of the project as human judges. To every judge, we show the trigger utterance(s) of a test example with two generated responses in random order, one from CueAD model and the other from Cue-IE model. Each judge is asked to choose a better one based on content appropriateness (i.e., grammatical correctness, content coherence) and cue consistency. The cue consistency refers to the consistency between the gold cue category and the cue category judged by human. The tie is permitted. Each judge individually judges 400 examples for each pair of compared methods.

4.4 Results and Analyses

Table 2 shows the results of comparison on the two corpora. We can observe that with the two embedding-based evaluation metrics, the baseline methods with cue incorporation perform better than their counterparts without it, because cue helps to strengthen semantic relevance. Our method further improves the performance as it relieves vanishing gradient and disturbing gradient effects, which helps to generate responses with high topical similarity to ground-truth responses, rendering better semantic relevance.

Besides, we obtain better cue representation reflecting its essential semantics, which helps to generate more diverse responses. This is confirmed by the best performance of our method based on the two distinct metrics. Furthermore, our method performs best in cue accuracy, which confirms that CueAD model can effectively control the generation following the direction of specific cue label that is consistent to the desired responses, so that one can better interpret the generated responses from the perspective of cue. We also obverse that Cue-E performs better than Cue-I, which indicates the vanishing cue gradient problem worsens performance compared with the disturbing gradients. Further, Cue-IE improves slightly over Cue-E, which might be because the cue participates

Case	Cue	Message	Response
1	💕	love you, sweet girl 😚	CueAD: love you more and miss you Cue-IE (emoji): well, thank you
2	happy	i'm really glad to hear that. they are too important to screw up. we all didn't get that from our book to movie adaptations.=> honestly a book should at most be a tv show, movies can't take the time to show everything that's needed true. and i don't know understand why the writers allow complete deviations from the books.=> i know with hitchiker's guide, douglas adams just really, really hated doing something the same way twice so he made an effort to change things each time yeah=>	CueAD: i think it is desirable to have a miniseries adaptation Cue-IE (emotion): i am not sure if it's real

Table 3: Comparison between CueAD and Cue-IE based on the generated responses in two typical cases. The utterances in multi-turn dialogue are separated by '=>'.

Criteria	CueAD vs. Cue-IE	Win	Loss	Tie	Kap
CA	SE (emoji)	0.428	0.247	0.325	0.466
	ME (emotion)	0.387	0.268	0.345	0.431
CC	SE (emoji)	0.398	0.272	0.330	0.446
	ME (emotion)	0.480	0.232	0.288	0.482

Table 4: Results of human evaluation. The evaluation criteria include content appropriateness (CA) and cue consistency (CC). Fleiss' Kappa (Kap) is applied.

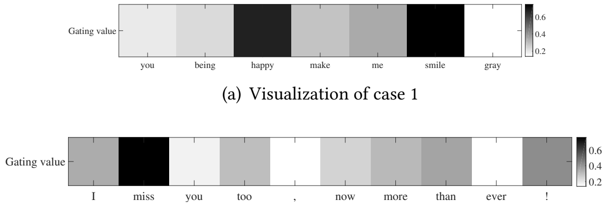
in the generation of each word and the vanishing cue gradient problem caused by Cue-I does not exist in Cue-E. Regarding the cue accuracy, CueAD model makes stronger improvements on the ME corpus than on the SE corpus, because the dialogues in ME contains long context information, which provides more abundant semantics for calculating the value of adaptive gating unit.

Table 4 shows the human evaluation results of different models. For each criterion, after counting numbers of win, loss and tie of each human judge, we average the three rates from all the three judges as the final evaluation result of this criteria. We can see that CueAD model outperforms all baseline methods with mostly moderate agreement among the evaluators according to Fleiss' Kappa [6]. Compared with Cue-IE, CueAD model improves 18.1%, 11.9% on content appropriateness (win-loss), and improves 12.6%, 24.8% on cue consistency on the SE and ME corpora, respectively.

4.5 Case Study

As shown in Table 3, we study two cases randomly chosen from the test sets of the two corpora for comparing CueAD model and the baselines. We can see that the cue information is successfully incorporated into the responses generated by CueAD model, e.g., the "💕" cue in case 1, and the "*happy*" cue in case 2, which confirms that our method can generate appropriate responses being coherent to the triggering utterance(s), and can well control and interpret the response generation. In contrast, the cue information cannot be reflected clearly in baseline methods. Besides, the appropriate cue provides accurate semantic hints so that our model generates longer and more diverse responses.

To further illustrate our CueAD model, we visualize the values of the adaptive gating unit of two another cases with "💕" cue in training part of SE corpus. The triggering utterances of the two cases are "life made me smile a lot today" and "i miss you so much" respectively. The visualization is shown as Figure 4 where darker color means higher gating value. We can see that the well-trained CueAD model can accurately utilize the cue-relevant words, i.e., "miss" and "smile" corresponding to the "💕" cue. This indicates that CueAD learns cue representation in a more appropriate way by reducing the effects of vanishing gradient and disturbing gradients of the cues commonly occurring in traditional methods.



(a) Visualization of case 1

(b) Visualization of case 2

Figure 4: Visualization of value of adaptive gating in the well-trained CueAD model.

5 CONCLUSION

In this paper, we propose a cue adaptive decoder network named CueAD for controllable dialogue response generation. With an adaptive cue gating unit designed to facilitate cue incorporation, CueAD could know when and to what extent to take advantage of the cue in each generation time step, and improves the cue representation that can reflect its essential semantics for enhancing the semantic relevance for response generation. The experiment results show that our model can take advantage of cue information effectively, and clearly outperforms state-of-the-art response generation methods.

ACKNOWLEDGMENTS

The work was supported by the National Key R&D Program of China under grant 2018YFB1004700, National Natural Science Foundation of China (61872074, 61772122), and the CETC Joint fund.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[2] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* 5, 2 (1994), 157–166.

[3] Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*, Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle (Eds.). The Association for Computer Linguistics.

[4] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*. 103–111.

[5] 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. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 1615–1625.

[6] Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement* 33, 3 (1973), 613–619.

[7] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[9] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. *Computer Science* (2016).

[10] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[11] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 2157–2169.

[12] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 2122–2132.

[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://www.aclweb.org/anthology/D14-1162/

[14] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence Level Training with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[15] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. 583–593.

[16] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. 2017. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 3288–3294.

[17] 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 *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. 3776–3784.

[18] 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 *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 3295–3301.

[19] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. 1577–1586.

[20] Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A Conditional Variational Framework for Dialog Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*. 504–509.

[21] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. In *Proceedings of the 32nd International Conference on Machine Learning, ICML Workshop 2015*.

[22] Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical Recurrent Attention Network for Response Generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 5610–5617.

[23] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to Control the Specificity in Neural Response Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 1108–1117.

[24] Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 654–664.

[25] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 730–739.

[26] Xianda Zhou and William Yang Wang. 2018. MojiTalk: Generating Emotional Responses at Scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 1128–1137.