Improving Multi-turn Emotional Support Dialogue Generation with Lookahead Strategy Planning

Yi Cheng[1*], **Wenge Liu**[2*], **Wenjie Li**[1†], **Jiashuo Wang**[1],
Ruihui Zhao[3], **Bang Liu**[4‡], **Xiaodan Liang**[5], **Yefeng Zheng**[3]

[1]Hong Kong Polytechnic University   [2]Baidu Inc., Beijing, China   [3]Tencent Jarvis Lab
[4]RALI & Mila, Université de Montréal   [5]Sun Yat-sen University
{csycheng,cswjli,csjwang}@comp.polyu.edu.hk,
{kzllwg,xdliang328}@gmail.com, bang.liu@umontreal.ca,
{zacharyzhao,yefengzheng}@tencent.com

Abstract

Providing Emotional Support (ES) to soothe people in emotional distress is an essential capability in social interactions. Most existing research on building ES conversation systems only considers single-turn interactions with users, which is over-simplified. In comparison, multi-turn ES conversation systems can provide ES more effectively, but face several new technical challenges, including: i) how to conduct support strategy planning that could lead to the best supporting effects; ii) how to dynamically model the user's state. In this paper, we propose a novel system named MultiESC to address these issues. For strategy planning, drawing inspiration from the A* search algorithm, we propose lookahead heuristics to estimate the future user feedback after using particular strategies, which helps to select strategies that can lead to the best long-term effects. For user state modeling, MultiESC focuses on capturing users' subtle emotional expressions and understanding their emotion causes. Extensive experiments show that MultiESC significantly outperforms competitive baselines in both strategy planning and dialogue generation. Our codes are available at https://github.com/lwgkzl/MultiESC.

1 Introduction

Almost every human has experienced emotional distress, even if not suffering from any mental disorders. Frequently, people deal with the distress by seeking Emotional Support (ES) from social interactions (Langford et al., 1997; Greene, 2003). Nevertheless, ES from family and friends is not always available (Webber and Mascari, 2018). With the potential of providing more people with in-time support, developing Emotional Support Conversation (ESC) systems has attracted much attention. However, since early ES datasets are constructed
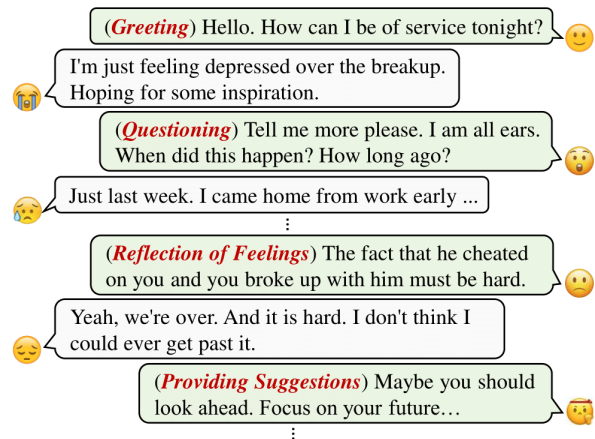

Figure 1: An example of an emotional support conversation between the support-seeker (left) and the supporter (right). The support strategies adopted by the supporter are presented in red italics before the utterances.

by crawling post-response pairs from online forums, they only contain single-turn conversations (Medeiros and Bosse, 2018; Sharma et al., 2020). Thus, most of the existing research on ESC also only considers single-turn interactions with the user (Medeiros and Bosse, 2018; Sharma et al., 2020, 2021), which is over-simplified and has limited support effects. It was not until recently that Liu et al. (2021) released the first large-scale multi-turn ES dataset, ESCONV. They also designed an ESC framework, suggesting the conversation procedures and support strategies for multi-turn ESC.

Compared to the single-turn scenario, developing multi-turn ESC systems faces several new challenges. One significant challenge is *support strategy planning*. As pointed out in the psychological literature, particular procedures and strategies are indispensable for effective emotional support (Greene, 2003; Hill, 2009). As in Fig. 1, the supporter strategically soothes the support-seeker by first caringly inquiring about the situation, then resonating with the seeker's feelings, and finally providing suggestions to evoke positive emotions.

---

*Equal contribution.
†Corresponding author.
‡Canada CIFAR AI Chair.

3014

Notably, strategy planning in ESC should be conducted on a long planning horizon. That is, instead of merely considering the dialogue history or foreseeing the immediate effect after using the strategy, the system should further *look ahead*, to consider how much the adopted strategy would contribute to reducing the user's emotional distress at a long run. Though some strategies may not directly provide comfort, they are still essential for reaching the long-term dialogue goal, such as greetings at the beginning of the conversation and inquiring about the user's experiences.

Another challenge for multi-turn ESC is how to *dynamically model the user's state* during the conversation. Prior works on emotion-related dialogue tasks mainly detect the user's coarse-grained emotion type to enhance dialogue generation (Lin et al., 2019; Majumder et al., 2020; Li et al., 2020a). However, such practice is not completely appropriate for ESC. The reason is that the user's emotion in ESC almost stays the same type, such as being sad, throughout the conversation. Instead, it often changes subtly in terms of emotion intensity. Besides, effective ES requires more than only identifying the user's emotion. A thorough understanding of the user's situation is also essential.

In this paper, we propose a multi-turn ESC system MultiESC to address the above issues. For *strategy planning*, we draw inspiration from the A* search algorithm (Hart et al., 1968; Pearl, 1985) and its recent application in constrained text generation (Lu et al., 2021), which addressed the challenge of planning ahead by incorporating heuristic estimation of future cost. In MultiESC, we develop lookahead heuristics to estimate the expectation of the future user feedback to help select the strategy that can lead to the best long-term effect. Concretely, we implement a strategy sequence generator to produce the probability of the future strategy sequences, and a user feedback predictor to predict the feedback after applying the sequence of strategies. For *user state modeling*, MultiESC captures the user's subtle emotion expressed in the context by incorporating external knowledge from the NRC VAD lexicon (Mohammad, 2018). Moreover, it identifies the user's emotion causes (i.e., the experiences that caused the depressed emotion) to more thoroughly understand the user's situation.

In summary, our contributions are as follows:

- We propose a multi-turn ESC system, MultiESC, which conducts support strategy planning with foresight of the user feedback and dynamically tracks the user's state by capturing the subtle emotional expressions and the emotion causes.
- It is a pioneer work that adopts A*-like lookahead heuristics to achieve dialogue strategy selection on a long planning horizon.
- Experiments show that MultiESC significantly outperforms a set of state-of-the-art models in generation quality and strategy planning, demonstrating the effectiveness of our proposed method.

2 Related Work

Emotional Support Conversation Systems. Since early ES datasets were mainly composed of single-turn conversations (Medeiros and Bosse, 2018; Sharma et al., 2020), most research on developing ESC systems only considered the simplified scenario of single-turn interactions with the user (Sharma et al., 2021; Hosseini and Caragea, 2021). The few works that developed multi-turn ES chatbots rely on predefined templates and hand-crafted rules (Zwaan et al., 2012), which suffer from limited generality. It was not until last year that Liu et al. (2021) released the first multi-turn ESC dataset ESCONV. Following Liu et al. (2021), Peng et al. (2022) and Tu et al. (2022) recently explored data-driven multi-turn ESC systems. Peng et al. (2022) proposed a hierarchical graph network to capture both the global context and the local user intention. They did not consider strategy planning, which is critical in multi-turn ESC. Tu et al. (2022) proposed to enhance context encoding with commonsense knowledge and use the predicted strategy distribution to guide response generation. Nevertheless, their method of strategy prediction, directly implemented with a vanilla Transformer encoder, was relatively preliminary and did not consider any user-feedback-oriented planning as we do.

Empathetic Response Generation. Empathetic Response Generation (ERG) (Rashkin et al., 2019) is a research area closely related to ESC, as being empathetic is a crucial ability for providing emotional support (Greene, 2003; Pérez-Rosas et al., 2017). However, ERG does not has the explicit goal of proactively soothing the user's negative emotion. Instead, it only reactively generates responses that are consistent with the user's emotion (Lin et al., 2019; Majumder et al., 2020; Li et al., 2020a; Zheng et al., 2021; Wang et al., 2021).
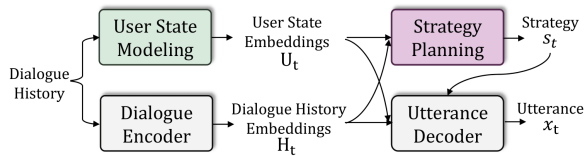
Figure 2: The overall framework of MultiESC. Details about the user state modeling and the strategy planning modules are shown in Fig. 3 and Fig. 4, respectively.

Figure 3: The architecture of the user state modeling module in MultiESC.

3 Preliminaries

ESConv. Our research is conducted on ESCONV. It is a long conversation dataset, with an average of 29.8 utterances in each dialogue. It also includes rich annotations, such as the *strategies* adopted by the supporter and the *user feedback* scores. There are overall eight types of strategies (e.g., *question*, *reflection of feelings* and *self-disclosure*). The user feedback score indicates how much the user's emotional distress is reduced during the conversation. They are marked by the support-seekers on a Likert scale with five levels after every two turns. More data statistics are provided in the appendix.

NRC VAD Lexicon. The NRC VAD lexicon includes the Valence-Arousal-Dominance (VAD) scores of 20,000 English words. The VAD score of a word measures its underlying emotion in three dimensions: valence (pleased-displeased), arousal (excited-calm), and dominance (dominant-submissive). For example, the VAD scores of "loneliness" and "abandon" are (0.15, 0.18, 0.22) and (0.05, 0.52, 0.25), respectively. The VAD model captures a wide range of emotions and allows different emotions to be comparable.

Problem Formulation of ESC. Denote the utterances from the system and the user at the i-th round of the conversation are respectively (x_i, y_i),[1] while the user's state is u_i (i=1, 2, ..., n_R). Suppose the set of all support strategies is $\mathcal{S}$. At the t-th turn, given the dialogue history $\mathcal{H}_t = \{(x_i, y_i)\}_{i=1}^{t-1}$, the system tracks the user states $\mathcal{U}_t = \{u_1, u_2, ..., u_{t-1}\}$ from $\mathcal{H}_t$ and generates the next utterance x_t, using an appropriate support strategy $\hat{s}_t \in \mathcal{S}$.

4 Methodology

As shown in Fig. 2, our proposed system Multi-ESC consists of four modules. The dialogue encoder first converts the dialogue history $\mathcal{H}_t$ into the embeddings $\mathbf{H}_t$. At the same time, the user

state modeling module extracts the user state information, producing the embeddings $\mathbf{U}_t$. Then, given $\mathbf{H}_t$ and $\mathbf{U}_t$, the strategy planning module selects the strategy s_t. Finally, the utterance decoder generates the utterance x_t, adopting the strategy s_t.

4.1 Dialogue Encoder

The dialogue encoder module is implemented with a Transformer encoder (Vaswani et al., 2017). We concatenate the utterances in $\mathcal{H}_t$ and keep the last N tokens of the concatenation as its input sequence. Given the input, it produces the dialogue history embeddings $\mathbf{H}_t \in \mathbb{R}^{N \times d_{\text{emb}}}$.

4.2 User State Modeling

Fig. 3 illustrates the workflow of user state modeling. To identify the user's state at the i-th round of the conversation, we first extract the emotion cause mentioned at this round, denoted as c_i, with an off-the-shelf detector[2] trained on a large-scale emotion cause detection dataset (Poria et al., 2021). For example, in Fig. 3, c_1="*I have not seen my friends for a long time*". Then, we concatenate the dialogue content x_i, y_i and the emotion cause c_i with special separator tokens to form the input of a Transformer encoder. Here, the system's utterance x_i is also considered because it often provides necessary context for understanding the user's state. The input sequence is represented as the positional sum of emotion embeddings, word embeddings, and positional embeddings.

The emotion embeddings are used to fuse the emotion information. They are obtained as follows. We train multiple emotion vectors $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{n_{\text{emo}}}\}$ to represent the underlying emotions of different words. Concretely, we split the VAD space into multiple subspaces by dividing the valence and the arousal dimensions, respectively, into n_V and n_A intervals of equal length. Each

---

[1] We suppose that ESCs are always initiated by the system (or the supporter).

Strategy Score: $\mathbf{F}(s_t) = g(s_t) + \lambda \cdot h(s_t)$

History-based Score: $g(s_t) = -\log \mathrm{P}(s_t | \mathrm{H_t, U_t})$

Lookahead Score: $h(s_t) = \sum_{s_{>t} \in \hat{S}_L} [\mathrm{P}(s_{>t} | s_t, \mathrm{H_t, U_t}) \cdot f(s_t, s_{>t}, \mathrm{U_t})]$

Process of Calculating a Strategy Score during Inference

Probability of the Next Strategy s_t

$\mathrm{P}(s_t | \mathrm{H_t, U_t})$

Probability of the Following Strategy Sequence $s_{>t}$

$\mathrm{P}(s_{>t} | s_t, \mathrm{H_t, U_t})$

Strategy Sequence Generator

Beam Search

k Most Possible $s_{>t}$

$\hat{S}_L = \arg\mathrm{top}k \, \mathrm{P}(s_{>t} | s_t, \mathrm{H_t, U_t})$

User Feedback Predictor

Predicted User Feedback

$f(s_t, s_{>t}^k, \mathrm{U_t})$

$f(s_t, s_{>t}^2, \mathrm{U_t})$
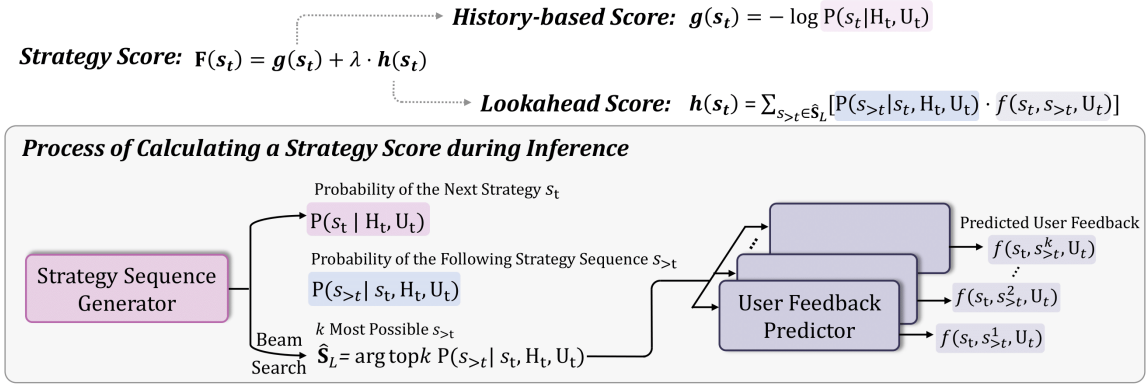
$f(s_t, s_{>t}^1, \mathrm{U_t})$

Figure 4: The process of calculating the strategy score, using a strategy sequence generator and a user feedback predictor. At each turn, our model selects the next strategy that maximizes the score of $F(s_t)$.

emotional subspace is represented as one emotion vector e_j.[3] To construct the emotion embeddings, we retrieve the VAD score of each input token from the NRC VAD lexicon to identify which emotional subspace it belongs, and then we represent it as the corresponding emotion vector. For those tokens without VAD annotation, we use a special emotion vector to represent them.

Finally, the encoded hidden vector $\mathbf{u}_i$ corresponding to the [CLS] token is used to represent the user state at the i-th round. The user state embeddings $\mathbf{U}_t$ is the concatenation of all the user state embeddings before the t-th round, that is, $\mathbf{U}_t=[\mathbf{u}_1; \mathbf{u}_2; ...; \mathbf{u}_{t-1}]$.

4.3 Strategy Planning with Lookahead Heuristics

We develop a strategy score function to evaluate whether to adopt a particular strategy (e.g. *question* or *self-disclosure*) by comprehensively considering the dialogue history and the potential user feedback. Formally, at the t-th round, MultiESC adopts the strategy $\hat{s}_t$ that maximizes the score function:

$$\hat{\mathbf{s}}_t = \arg\max_{s_t \in \mathcal{S}} F(s_t), \qquad (1)$$

where $F(\cdot)$ is the strategy score function.

In the following, we will first introduce the strategy score function and then explain how MultiESC calculates the strategy scores with two components: a strategy sequence generator and a user feedback predictor, as presented in Fig. 4. Finally, we will describe the architectures of the two components.

Strategy Score Function. Our method draws inspiration from the classical search algorithm, A* search (Hart et al., 1968), which conducts lookahead planning in a heuristic way. At each step, it searches the highest-scoring path by selecting an action that maximizes the sum of the score so far and a heuristic estimation of the future score. Similarly, we define our strategy score function as:

$$F(s_t) = g(s_t) + \lambda \cdot h(s_t), \qquad (2)$$

where $g(s_t)$ is a *history-based score*; $h(s_t)$ is a *lookahead score* that heuristically estimates the future user feedback; λ is a hyper-parameter that balances the weights of the two terms.

The history-based score $g(s_t)$ computes the conditional probability distribution of the next strategy purely based on the dialogue history and the previous user states. Formally, it is defined as:

$$g(s_t) = -\log \Pr(s_t | \mathbf{H}_t, \mathbf{U}_t). \qquad (3)$$

Previous research on dialogue strategy prediction generally followed this history-based scheme (Zhou et al., 2019; Joshi et al., 2021; Dutt et al., 2021), though they may vary in their methods of obtaining the representations of $\mathbf{H}_t$ and $\mathbf{U}_t$. However, such practice overlooks the strategy's future effects and how much it could help in achieving the long-term dialogue goal. In our work, we incorporate the lookahead score to alleviate this issue.

The lookahead score $h(s_t)$ heuristically estimates *the mathematical expectation of the future user feedback score*[4] after adopting the strategy s_t at the t-th round. Ideally, to select the strategy that could lead to the best final result, we want to

---

estimate the user feedback score at the end of the conversation, that is:

$$h(s_t) = \mathrm{E}[f(s_t, s_{>t}, \mathbf{U}_t)|s_t, \mathbf{H}_t, \mathbf{U}_t]$$
$$= \sum_{s_{>t} \in \mathcal{S}^\omega} \Pr(s_{>t}|s_t, \mathbf{H}_t, \mathbf{U}_t) \cdot f(s_t, s_{>t}, \mathbf{U}_t).$$
(4)

$\mathrm{E}(\cdot)$ represents the mathematical expectation; $s_{>t}$ is the future strategy sequence to be used after the t-th round till the end of the conversation; $\mathcal{S}^\omega$ is the set of all possible strategy sequences; $f(s_t, s_{>t}, \mathbf{U}_t)$ denotes the user feedback score after successively applying s_t and $s_{>t}$ to comfort a user whose previous states are $\mathbf{U}_t$.

However, Eq. 4 is hard to calculate, because the space of $\mathcal{S}^\omega$ is too large and it is difficult to estimate the user feedback $f(\cdot)$ after too many turns (i.e. if the strategy sequence $s_{>t}$ is too long). Thus, we approximate Eq. 4 as follows. First, we only look ahead for the limited L turns. We estimate the expectation of the user feedback score after L turns instead of at the end of the conversation. Then, to further narrow the space of $\mathcal{S}^\omega$, we only consider the k most possible future strategy sequences. Formally, Eq. 4 is approximated as:

$$h(s_t) = \sum_{s_{>t} \in \hat{\mathcal{S}}_L} \Pr(s_{>t}|s_t, \mathbf{H}_t, \mathbf{U}_t) \cdot f(s_t, s_{>t}, \mathbf{U}_t),$$
$$\hat{\mathcal{S}}_L = \arg\operatorname{topk}_{s_{>t} \in \mathbf{S}_L} \Pr(s_{>t}|s_t, \mathbf{H}_t, \mathbf{U}_t),$$
(5)

where $\mathcal{S}_L$ is the set of the strategy sequences whose lengths are less than L.

Strategy Score Calculation in MultiESC. MultiESC calculates the strategy scores with a Strategy Sequence Generator (SSG) and a User Feedback Predictor (UFP). The function of SSG is to sequentially predict $s_{\geq t}$ based on $\mathbf{H}_t$ and $\mathbf{U}_t$, where $s_{\geq t}$ is the strategy sequence that will be used in the following L rounds ($s_{\geq t}$=$[s_t; s_{>t}]$). At the l-th timestep, it outputs the predicted strategy distribution:

$$\Pr(s_{t+l}|s_{t:t+l}, \mathbf{H}_t, \mathbf{U}_t), \qquad (6)$$

where l=1, 2, ..., L and $s_{t:t+l}$ denotes the already-generated strategy sequence before the l-th timestep. The function of UFP is to estimate the user feedback score $f(s_{\geq t}, \mathbf{U}_t)$.

As shown in Fig. 4, to calculate the strategy score of a particular strategy s_t, we first use SSG to derive the history-based score $g(s_t)$ from its predicted strategy distribution at the first timestep.

Next, we use SSG to find the set of the k most possible future strategy sequences $\hat{\mathbf{S}}_L$ through beam search. For each strategy sequence $s_{>t}$ in $\hat{\mathbf{S}}_L$, we obtain its probability by:

$$\Pr(s_{>t}|s_t, \mathbf{H}_t, \mathbf{U}_t) = \prod_{l=2}^L \Pr(s_{t+l}|s_{t:t+l}, \mathbf{H}_t, \mathbf{U}_t).$$

We then leverage UFP to estimate the user feedback score after successively applying s_t and $s_{>t}$. Combining the predicted probabilities of the strategy sequences and the estimated user feedback scores, we obtain the lookahead score $h(s_t)$ as in Eq. 5. Finally, given $g(s_t)$ and $h(s_t)$, the overall strategy score is obtained as in Eq. 2.

Strategy Sequence Generator. SSG is developed upon the architecture of the Transformer decoder. Its only difference from the original Transformer decoder is that it adopts the multi-source attention mechanism to selectively attend to the dialogue history $\mathbf{H}_t$ and the user state information $\mathbf{U}_t$. Specifically, the strategy sequence $s_{\geq t}$ is first fed to a masked multi-head attention layer, producing the contextualized strategy sequence representations $\mathbf{P}_t$. Then, $\mathbf{P}_t$ interacts with $\mathbf{H}_t$ and $\mathbf{U}_t$ respectively through cross attention layers as:

$$\hat{\mathbf{H}}_t = \mathtt{MH\text{-}ATT}(L(\mathbf{P}_t), L(\mathbf{H}_t), L(\mathbf{H}_t)),$$
$$\hat{\mathbf{U}}_t = \mathtt{MH\text{-}ATT}(L(\mathbf{P}_t), L(\mathbf{U}_t), L(\mathbf{U}_t)),$$

where $\mathtt{MH\text{-}ATT}(\cdot)$ represents the multi-head self-attention mechanism. An information fusion layer is utilized to combine them:

$$\mu = \mathrm{ReLU}(\mathbf{W}_\mu[\hat{\mathbf{H}}_t; \hat{\mathbf{U}}_t] + \mathbf{b}_\mu),$$
$$\hat{\mathbf{P}}_t = \mu \cdot \hat{\mathbf{H}}_t + (1 - \mu) \cdot \hat{\mathbf{U}}_t,$$

where $\mathbf{W}_\mu$ and $\mathbf{b}_\mu$ are trainable parameters. Next, $\hat{\mathbf{P}}_t$ is fed to a connected feed-forward network with residual connections around the sub-layers. We denote the hidden states produced here as $\widetilde{\mathbf{P}}_t$. Finally, the strategy distribution at the l-th timestep is predicted as:

$$\Pr(s_{t+l}|s_{t:t+l}, \mathbf{H}_t, \mathbf{U}_t) = \mathrm{softmax}(\mathbf{W}_s\widetilde{\mathbf{P}}_t + \mathbf{b}_s),$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are trainable parameters. To train the SSG, we use the negative log-likelihood of the ground-truth strategy s_{t+l}^* as its loss function.

User Feedback Predictor. UFP predicts the user feedback score $f(s_{\geq t}, \mathbf{U}_t)$ by first encoding $s_{\geq t}$ with a Transformer encoder, denoted as $\mathrm{TRS_{UFP}}$. Specifically, we leverage a trainable strategy matrix $\mathbf{E}_s \in \mathbb{R}^{|\mathcal{S}| \times d_{\mathrm{emb}}}$ to represent different types

Model	PPL↓	B-1↑	B-2↑	B-3↑	B-4↑	R-L↑	MET↑	CIDEr↑
MoEL (Lin et al., 2019)	264.11	19.04	6.47	2.91	1.51	15.95	7.96	10.95
MIME (Majumder et al., 2020)	69.28	15.24	5.56	2.64	1.50	16.12	6.43	10.66
EmpDG (Li et al., 2020a)	115.34	18.08	6.46	3.02	1.52	15.89	6.93	10.73
DialoGPT-Joint (Liu et al., 2021)	15.71	17.39	5.59	2.03	1.18	16.93	7.55	11.86
BlenderBot-Joint (Liu et al., 2021)	16.79	17.62	6.91	2.81	1.66	17.94	7.54	18.04
MISC (Tu et al., 2022)	16.16	-	7.31	-	2.20	17.91	-	-
GLHG (Peng et al., 2022)	15.67	19.66	7.57	3.74	2.13	16.37	-	-
MultiESC	**15.41**	**21.65**	**9.18**	**4.99**	**3.09**	**20.41**	**8.84**	**29.98**
MultiESC *w/o* emotion	18.43	18.93	7.68	4.05	2.41	20.15	7.89	24.33
MultiESC *w/o* cause	15.68	20.07	8.76	4.64	2.77	19.82	8.60	26.73
MultiESC *w/o* strategy	15.60	19.72	8.24	4.42	2.70	20.35	8.25	27.77
MultiESC *w/o* lookahead	15.71	21.52	9.15	4.81	3.02	20.39	8.43	29.81

Table 1: Automatic evaluation results on the generation quality.

of strategies. Given the strategies in $s_{\geq t}$, we concatenate their corresponding strategy vectors as the input of TRS$_{\text{UFP}}$, so we have

$$\mathbf{B} = \text{TRS}_{\text{UFP}}[\text{Emb}([\texttt{CLS}]; s_{\geq t})],$$

where $\text{Emb}(\cdot)$ represents the operation of the embedding layer that maps the strategies in $s_{\geq t}$ to their corresponding vectors in $\mathbf{E}_s$. Suppose the encoded hidden state corresponding to the [CLS] token is $\mathbf{q}_s$. Next, we pass the user state embeddings through a Long-Short Term Memory (LSTM) network (Cheng et al., 2016):

$$\hat{\mathbf{U}}_t = \text{LSTM}(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{t-1}),$$

We then use $\mathbf{q_s}$ to attend to the hidden states $\hat{\mathbf{U}}_t = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, ..., \hat{\mathbf{u}}_{t-1}]$ through an attention layer:

$$\widetilde{\mathbf{u}}_f = \sum_{i=1}^{t-1} a_i \hat{\mathbf{u}}_i,$$

$$a_i = \frac{\exp(\hat{\mathbf{u}}_i^\top \mathbf{W}_a \mathbf{q}_s)}{\sum_{j=1}^{t-1} \exp(\hat{\mathbf{u}}_j^\top \mathbf{W}_a \mathbf{q}_s)},$$

where $\mathbf{W}_a$ is a trainable matrix. Finally, we obtain the predicted user feedback score by passing $\widetilde{\mathbf{u}}_f$ through a single feedforward layer.

We leverage the ground-truth user feedback scores annotated in the ESCONV dataset as supervision to train the UFP, and use the Mean Squared Error (MSE) as its loss function $\mathcal{L}_f$.

4.4 Utterance Decoder

Given the user state embeddings $\mathbf{U}_t$, the dialogue history embeddings $\mathbf{H}_t$, and the selected strategy $\hat{\mathbf{s}}_t$, the utterance decoder aims to produce the next utterance x_t. Its architecture is the same as that of the strategy sequence generator (§ 4.3), except for the input sequence. To guide dialogue generation with the selected strategy $\hat{\mathbf{s}}_t$, we prepend the strategy embedding of $\hat{\mathbf{s}}_t$ before the embeddings of the utterance sequence as the input of the utterance decoder. The negative likelihood of the ground-truth token in the target utterance is used as the generation loss $\mathcal{L}_g$. More details on the training procedure are provided in the appendix.

5 Experiments

5.1 Experimental Setup

Baselines. Our baselines include three empathetic response generators: **MoEL** (Lin et al., 2019), **MIME** (Majumder et al., 2020), and **EmpDG** (Li et al., 2020a); and four state-of-the-art methods on the ESCONV dataset: **DialoGPT-Joint**, **BlenderBot-Joint** (Liu et al., 2021), **MISC** (Tu et al., 2022), and **GLHG** (Peng et al., 2022). More details about them are described in the appendix.

Implementation Details. We follow the original division of ESCONV for training, validation, and testing. We initialize the parameters of the dialogue encoder and the utterance decoder of MultiESC with the BART-small (Lewis et al., 2020) model from the HuggingFace library (Wolf et al., 2019). There are n_{emo}=65 types of emotion vectors, with n_{V}=n_{A}=8. In the strategy planning module, we set λ=0.7 and L=2. The beam size k is set to be 6 when searching the set of the most possible strategy sequences $\hat{\mathbf{S}}_L$. Since the codes of MISC and GLHG were not released, we directly refer to the results reported in their original papers. For the other baselines, we use their released codes to conduct our experiments. Our model has 145.6M parameters, which is in the same order as the baselines. For reference, BlenderBot-Joint, DialoGPT-Joint, and GLHG have 90M, 117M, and 92M parameters, respectively. More implementation details are provided in the appendix.

MultiESC vs.	MoEL			BlenderBot-Joint			w/o strategy			w/o lookahead		
	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
Fluency	**64.1**‡	18.0	18.0	35.2	**42.9**	21.9	38.3	**41.4**	20.3	**41.4**	37.5	21.1
Empathy	**53.1**‡	34.4	12.5	**44.5**	43.8	11.7	**43.8**†	29.7	26.5	35.9	**39.1**	25.0
Identification	**69.5**‡	22.7	7.9	**48.4**‡	32.8	18.8	**56.3**‡	32.8	10.9	**46.9**‡	28.9	14.2
Suggestion	**71.9**‡	14.8	13.3	**60.9**‡	23.4	15.6	**52.3**†	36.7	10.9	**44.5**†	30.5	25.0
Overall	**65.6**‡	20.3	14.1	**58.6**‡	31.3	10.2	**55.5**‡	30.5	14.0	**46.1**†	32.0	21.9

Table 2: Human interactive evaluation results (%). The columns of "Win/Lose" indicate the proportion of cases where MultiESC wins/loses in the comparison. $\dagger/\ddagger$ denote p-value $< 0.1/0.05$ (statistical significance test).

Model	Accuracy	Weighted-F1	Feedback
DialoGPT-Joint	26.03	23.86	2.87
BlenderBot-Joint	29.92	29.56	3.05
MISC	31.61	-	-
MultiESC	**42.01**	**34.01**	3.85

Table 3: The strategy planning performance of Multi-ESC and the baseline methods.

Model	Accuracy	Weighted-F1	Feedback
MultiESC$_{k=1}$	38.72	30.12	3.59
MultiESC$_{k=2}$	39.53	30.61	3.62
MultiESC$_{k=3}$	41.33	32.83	3.75
MultiESC$_{k=4}$	41.61	33.30	3.67
MultiESC$_{k=5}$	41.78	33.64	**3.93**
MultiESC$_{k=7}$	41.79	33.92	3.88
MultiESC$_{k=8}$	41.79	33.97	3.92
MultiESC	**42.01**	**34.01**	3.85
w/o lookahead	38.76	30.21	3.36

Table 4: The strategy planning performance of different variants of MultiESC. Note that the beam size of MultiESC is set to be 6 (see § 5.1).

5.2 Automatic Evaluation of Generation Quality

To evaluate the generation quality, we adopt the following metrics: perplexity (**PPL**), BLEU-1/2/3/4 (**B-1/2/3/4**) (Papineni et al., 2002), ROUGE-L (**R-L**) (Lin, 2004), METEOR (**MET**) (Lavie and Agarwal, 2007), and **CIDEr** (Vedantam et al., 2015).

Comparison with Baselines. As shown in the upper part of Table 1, MultiESC performs significantly better than the baseline models in all the metrics. It performs exceptionally well in the CIDEr metric, which measures the similarity between TF-IDF weighted n-grams (i.e., the words that frequently appear in many utterances contribute less to the score). This result demonstrates that MultiESC is more capable of including critical information in the responses, catering to particular situations of users. Another finding is that the three empathetic generators (i.e., MoEL, MIME, and EmpDG) achieve significantly worse perplexity and CIDEr scores than the other models. Through analysis, we find that they tend to include content that commonly appears in many samples (e.g., "I'm sorry to hear that", "I can understand that"). This is probably because they only focus on how to generate responses that can display an understanding of the user's emotion, which is insufficient for ESC.

Ablation Study. To analyze the effects of different components on the downstream generation, we compare MultiESC with its following variants: (1) *w/o* **emotion** does not incorporate the emotion embedding layer in the user state modeling module; (2) *w/o* **cause** does not incorporate emotion cause

extraction for user state modeling; (3) *w/o* **strategy** directly generates utterances without first predicting the used strategy; (4) *w/o* **lookahead** conducts strategy planning without the lookahead heuristics to estimate the future user feedback scores.

As shown in the lower part of Table 1, the ablation of any component can cause a drop in the automatic evaluation results, demonstrating the indispensability of each part. In comparison, the ablation of the emotion embedding layer ("*w/o* emotion") leads to the most significant performance drop, as understanding the user's emotional states plays the most central role in ESC. The difference between the automatic evaluation results of the full model and "*w/o* lookahead" is relatively small. It is because they would generate exactly the same responses if they select the same strategy. The effects of lookahead planning heuristics are more evident in the human interactive test, since the adoption of different strategies at one turn would trigger different responses from the user and lead to different dialogue directions in the future rounds.

5.3 Human Interactive Evaluation

We recruit four graduate students with linguistic or psychological background as annotators to chat with the models for human interactive evaluation. We randomly sample 128 dialogues from the test set of ESCONV. The annotators are asked to act as the support seekers in these dialogue samples
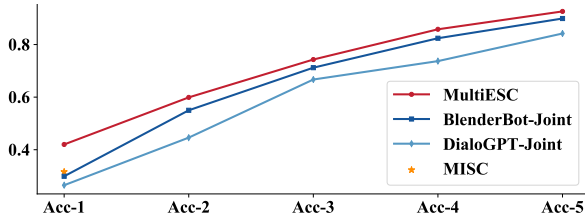
Figure 5: The top-n strategy prediction accuracy of MultiESC and the baseline methods.

by learning their situations and simulating their process of seeking emotional support by chatting with the models. Given MultiESC and a compared model, the annotators are asked to choose which one performs better (or select *tie*) in terms of the following metrics, following Liu et al. (2021): (1) **Fluency**: which model generates more fluent and understandable responses; (2) **Empathy**: which model has more appropriate emotion reactions, such as warmth, compassion and concern; (3) **Identification**: which model explores the user's situation more effectively to identify the problem; (4) **Suggestion**: which model offers more helpful suggestions; (5) **Overall**: which model provides more effective emotional support overall.

As shown in Table 2, we can see that the advantage of MultiESC over MoEL is substantial in all the metrics. It also outperforms BlenderBot-Joint in the overall supporting effects, though relatively inferior in terms of fluency, probably because the backbone of BlenderBot-Joint is extensively pre-trained on large-scale dialogue corpora (Roller et al., 2021). Compared with "*w/o* strategy", MultiESC is able to show more empathy, more clearly inquire about the user's situation, and provide more specific suggestions, demonstrating the importance of explicit strategy planning in ESC. Comparing MultiESC with "*w/o* lookahead", we can see that the incorporation of lookahead heuristics brings significant improvement in the dimensions of *identification* and *suggestion*.

5.4 Analysis of Strategy Planning

We evaluate the strategy planning module individually, using the following metrics: **Accuracy**, the proportion of prediction results that are the same as the ground-truth labels; **Weighted F1**, the weighted average of F1 scores in different classes while considering the class imbalance; **Feedback**, the next user feedback score that would be given after the predicted strategy is adopted, simulated with an user feedback predictor as illustrated in § 4.3.

| **Bot**: What can I help you with? |
| **User**: Lately, my friends started to make jokes about my face being ugly. They even called my face as flat as a pan! What should i do? I see them everyday! |
| **MoEL**: I'm sorry to hear that. Have you been able to find that it would be better to let yourself know how you feel about it. |
| **BlenderBot-Joint**: You should tell them to calm down. |
| **MultiESC**: (*Question*) That's horrible! Have you tried talking to them about it? |
| **MultiESC *w/o* lookahead**: (*Self-disclosure*) I can understand how that would make you feel. I have had to deal with a lot of bullies and I know how hard it can be. |
| **MultiESC *w/o* strategy**: I can understand that. I know why that would be upsetting. |
| **Ground-Truth**: (*Affirmation and Reassurance*) I'm sure you are very handsome! They could just be jealous. |

Table 5: Case study. The upper part is an example of the dialogue history in the test set of ESCONV. The lower part shows the responses from different models.

Comparison with Baselines. We compare MultiESC with the three baselinses capable of strategy planning (i.e., DialoGPT-Joint, BlenderBot-Joint, and MISC). The results are shown in Table 3. We can see that MultiESC performs the best in all the metrics with an absolute improvement of 10.4% and 4.45% in accuracy and weighted F1, respectively. As shown in Fig. 5, MultiESC also surpasses the baselines in all the top-n accuracy.

Analysis of MultiESC Variants. We analyze the following variants of our strategy planning method: (1) **MultiESC**$_{k=?}$: the model with different beam sizes when searching the set of k most possible strategy sequences $\hat{\mathbf{S}}_L$; (2) *w/o* **lookahead**: the model without the lookahead heuristics. As shown in Table 4, the strategy planning performance steadily improves with the increase of the beam search size when $k \leq 6$, as the larger beam size can result in a more precise estimation of the future user feedback. Nevertheless, further increasing k to consider more strategy sequences of low probabilities does not continue improving the performance apparently when $k>6$. Our full model also significantly outperforms "*w/o* lookahead" in all the metrics, especially regarding the feedback score. It demonstrates that our lookahead heuristics can help the model better plan the conversation and provide more effective emotional support.

5.5 Case Study

Table 5 presents a case study of the responses generated by different models. We can see that the utterances from MultiESC and its two variants are

more consistent with the context and more empathetic than those of the two baseline models. Further comparing MultiESC and its two variants, the utterance from the "*w/o* strategy" seems general and less engaging. The responses generated by MutiESC and "*w/o* lookahead" are both of high quality. Nevertheless, with the incorporation of the lookahead heuristic, MultiESC tends to proactively explore the user's situation at the beginning stage of the conversation instead of directly comforting the user, which aligns with the suggested procedure for providing emotional support (Hill, 2009).

6 Conclusion

In this paper, we explored the task of developing multi-turn Emotional Support Conversation (ESC) systems, with focus on how to strategically plan the conversation procedure. To this end, we proposed a novel ESC system, MultiESC, that conducts strategy planning with lookahead heuristics to estimate the long-term effect of the adopted strategy on the user. Moreover, we also proposed some effective mechanisms to dynamically model the user's state in multi-turn ESCs. The empirical results showed that MultiESC achieves significant improvement compared with a set of strong baselines in both generation quality and strategy planning.

Limitations

Though our proposed method exhibits large improvement compared with the existing baselines, we believe that the research on emotional support chatbots still has a long way to go. Compared with the human supporters, the utterances generated by the chatbots are usually general and repetitive, unable to show a personalized, in-depth understanding of the user's experiences or provide very specific and constructive suggestions on how to change the situation. This issue might be alleviated through the incorporation of commonsense knowledge, which will be included in our future research direction. Other issues, such as how to construct more trustworthy and safe emotional support chatbots, also require much further exploration.

Ethical Considerations

It needs to be clarified that the term "emotional support" in our paper mainly refers to peer support, like the one we can seek from family and friends in daily conversation. We do not claim to construct chatbots that can provide professional psycho-counseling or psychological treatment. Still, it needs particular caution when using such systems, and considerable further efforts are required to construct safer ESC systems. For example, crisis-warning mechanisms to detect users who have tendencies of self-harming or suicide are desirable.

Our experimental dataset, ESConv, is a well-established, publicly-available benchmark. It has filtered the sensitive and private information during the dataset construction. The participants in our human evaluation were transparently informed of our research intent and were paid reasonable wages.

References

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Ritam Dutt, Sayan Sinha, Rishabh Joshi, Surya Shekhar Chakraborty, Meredith Riggs, Xinru Yan, Haogang Bao, and Carolyn Rose. 2021. ResPer: Computationally modelling resisting strategies in persuasive conversations. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 78–90.

Jennifer C Greene. 2003. *Handbook of communication and social interaction skills*. Psychology Press.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Clara E Hill. 2009. *Helping skills: Facilitating, exploration, insight, and action*. American Psychological Association.

Mahshid Hosseini and Cornelia Caragea. 2021. It takes two to empathize: One to seek and one to provide. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13018–13026.

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan Black, and Yulia Tsvetkov. 2021. DialoGraph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*.

Catherine Penny Hinson Langford, Juanita Bowsher, Joseph P Maloney, and Patricia P Lillis. 1997. Social support: a conceptual analysis. *Journal of Advanced Nursing*, 25(1):95–100.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Qintong Li, Hongshen Chen, Zhaochun Ren, Pengjie Ren, Zhaopeng Tu, and Zhumin Chen. 2020a. EmpDG: Multi-resolution interactive empathetic dialogue generation. In *Proceedings of the International Conference on Computational Linguistics*, pages 4454–4466.

Qintong Li, Piji Li, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. 2020b. Knowledge bridging for empathetic dialogue generation. *arXiv e-prints*, pages arXiv–2009.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019. MoEL: Mixture of empathetic listeners. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 121–132.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3469–3483.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in Adam. *arXiv preprint arXiv:1711.05101*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2021. Neurologic A*esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*.

Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking emotions for empathetic response generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 807–819.

Lenin Medeiros and Tibor Bosse. 2018. Using crowdsourcing for the development of online emotional support agents. *International Conference on Practical Applications of Agents, Multi-Agent Systems*, pages 196–209.

William R Miller and Stephen Rollnick. 2012. *Motivational interviewing: Helping people change*. Guilford Press.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Judea Pearl. 1985. Heuristics: Intelligent search strategies for computer problem solving. *The Addison-Wesley Series in Artificial Intelligence*.

Wei Peng, Yue Hu, Luxi Xing, Yuqiang Xie, Yajing Sun, and Yunpeng Li. 2022. Control globally, understand locally: A global-to-local hierarchical graph network for emotional support conversation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.

Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, and Lawrence An. 2017. Understanding and predicting empathic behavior in counseling therapy. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13(5):1317–1332.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325. Association for Computational Linguistics.

Ashish Sharma, Inna W Lin, Adam S Miner, David C Atkins, and Tim Althoff. 2021. Towards facilitating empathic conversations in online mental health support: A reinforcement learning approach. In *Proceedings of the World Wide Web Conference*, pages 194–205.

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 5263–5276.

Quan Tu, Yanran Li, Jianwei Cui, Bin Wang, Ji-Rong Wen, and Rui Yan. 2022. MISC: A mixed strategy-aware model integrating COMET for emotional support conversation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 308–319. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575. IEEE Computer Society.

Jiashuo Wang, Wenjie Li, Peiqin Lin, and Feiteng Mu. 2021. Empathetic response generation through graph-based multi-hop reasoning on emotional causality. *Knowledge-Based Systems*, 233:107547.

Jane M Webber and J Mascari. 2018. *Disaster mental health counseling: A guide to preparing and responding*. American Counseling Association Foundation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DialoGPT: Large-scale generative pre-training for conversational response generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

Chujie Zheng, Yong Liu, Wei Chen, Yongcai Leng, and Minlie Huang. 2021. CoMAE: A multi-factor hierarchical framework for empathetic response generation. In *Findings of the Association for Computational Linguistics*, pages 813–824.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched Transformer for emotion detection in textual conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 165–176.

Yiheng Zhou, Yulia Tsvetkov, Alan W Black, and Zhou Yu. 2019. Augmenting non-collaborative dialog systems with explicit semantic and strategic dialog history. In *International Conference on Learning Representations*.

Janneke M Zwaan, Virginia Dignum, and Catholijn M Jonker. 2012. A conversation model enabling intelligent agents to give emotional support. In *Modern Advances in Intelligent Systems and Tools*, pages 47–52. Springer.

Appendix

A. Training Procedure

For training of the strategy sequence generator, we use the negative log-likelihood of the ground-truth strategy s_{t+l}^* as its loss function $\mathcal{L}_s$. For the utterance decoder, the negative likelihood of the ground-truth token in the target utterance is used as the generation loss $\mathcal{L}_g$. The strategy sequence generator and the utterance decoder are trained jointly, with the total loss as $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_g$. For training of the user feedback predictor, we use the Mean Squared Error (MSE) as its loss function $\mathcal{L}_f$. Since the user feedback scores in the ESCONV dataset are mainly between 2-5, we augment the training data by randomly generating 5,000 strategy sequences and regarding them as samples with the score 1. It is trained independently from the strategy predictor and the utterance decoder.

B. Baselines

MoEL (Lin et al., 2019) adopts several decoders focusing on different types of emotional utterances, whose outputs are combined to generate the final utterances.

MIME (Majumder et al., 2020) follows the architecture of MoEL and adds extra mechanisms to combine the results from different decoders.

EmpDG (Li et al., 2020a) learns how to generate responses consistent with the user's emotion via an adversarial learning framework.

DialoGPT-Joint and **BlenderBot-Joint** (Liu et al., 2021) are developed on the backbones of DialoGPT (Zhang et al., 2020) and BlenderBot (Roller et al., 2021), respectively. They prepend a special token, denoting the predicted support strategy, before the generated utterance to generate content conditioned on a predicted strategy.

MISC (Tu et al., 2022) enhances context encoding with commonsense knowledge and uses the predicted strategy distribution to guide the emotional support dialogue generation. It predicts the strategy distribution using a vanilla Transformer encoder.

GLHG (Peng et al., 2022) adopts a graph neural network to model the relationships between the user's emotion causes, intentions and the dialogue history for emotional support dialogue generation.

C. Implementation Details

We initialize the parameters of the dialogue encoder and the utterance decoder of MultiESC with the BART-small (Lewis et al., 2020) model from the HuggingFace library (Wolf et al., 2019). The maximum length of the input sequence for the dialogue encoder is N=512. The dimensions of all the hidden embeddings are d_{emb}=768. There are n_{emo}=65 types of emotion vectors, with $n_V=n_A=8$.

In the strategy planning module, we set λ=0.7, which results in the best performance on the validation set among $\lambda \in \{0.1, 0.2, ..., 1.0\}$. For the number of lookahead rounds L, we experiment with $L \in \{1, 2, ..., 5\}$. We find that the performance on the validation set is the best when L =2 and L =3. The performances when L =2 and L =3 are very close, but considering the computation efficiency, we set L =2 in the following experiments. For the searching beam size k, we experiment with $k \in \{1, 2, ..., 10\}$, and finally set it to be 6, because it strikes the best balance between performance and efficiency. We choose the above hyperparameters by manual tuning, and the selection criterion is the strategy prediction accuracy on the validation set.

AdamW (Loshchilov and Hutter, 2018) is used as optimizer; its initial learning rate is set to be 5×10^{-5} and adaptively decays during training. The batch size is set to be 32. Since the codes of MISC and GLHG were not released, we directly refer to the results reported in their original papers. For the other baselines, we use their released codes to conduct our experiments. Each model is trained up to 10 epochs, and the checkpoints that achieve the best perplexity on the validation set are used for evaluation.

Our model has 145.6M parameters. The hardware we used was one GPU of NVIDIA Tesla V100. The overall training time is about two hours.

D. Dataset

Our experiments are conducted on the ESCONV dataset (Liu et al., 2021)[5]. It is an English dataset. To construct the dataset, they recruited crowd-workers, who had learned the common procedures and strategies for providing emotional support, to converse with volunteers that needed emotion support through an online platform. The crowd-workers were required to annotate the strategy adopted at each turn, and the support-seekers were asked to give feedback every two rounds on a Likert scale with five levels, indicating how much their emotional distress is reduced. The dataset contains 1,300 long dialogues with 38,350 utterances. There

---

[5]https://github.com/thu-coai/Emotional-Support-Conversation

Strategy	Proportion
Question	21.77%
Restatement or Paraphrasing	6.46%
Reflection of Feelings	8.05%
Self-disclosure	9.34%
Affirmation and Reassurance	16.13%
Providing Suggestions or Information	22.02%
Greetings	8.72%
Others (Unlabelled)	7.49%

Table 6: The strategy distribution.

Providing Suggestions	*Information*
I understand that. What if that is the case? You may need to talk to them and let them know how you feel about that.	Is it possible to reframe how you look at your clients' dire financial situations?
Have you thought of contacting a debt relief program? In some cases they can substantially reduce debt to something much more manageable.	It may not be for you. I think you should think about the pros and cons of keeping your position. It might make things clearer for you.

Table 7: Examples of the responses labelled with the strategies *Providing Suggestions* and *Information* in the original ESConv dataset.

is an average of 29.5 utterances per dialogue and an average 16.7 tokens per utterance. We follow the original division of the ESCONV dataset for training, validation, and testing, with the ration of 8:1:1. There are overall 8 types of support strategies. Referring to Liu et al. (2021), their original definitions are as follows:

- *Question*: ask for information related to the problem to help the help-seeker articulate the issues that they face.
- *Restatement or Paraphrasing*: a simple, more concise rephrasing of the support-seeker's statements that could help them see their situation more clearly.
- *Reflection of Feelings*: describe the help-seeker's feelings to show the understanding and empathy.
- *Self-disclosure*: share similar experiences or emotions that the supporter has also experienced to express your empathy.
- *Affirmation and Reassurance*: affirm the support-seeker's ideas, capabilities, and strengths to give reassurance and encouragement.
- *Providing Suggestions*: provide suggestions about how to change the current situation.
- *Information*: provide useful information to the help-seeker, for example with data, facts, opinions, resources, or by answering questions.
- *Others*: other support strategies that do not fall into the above categories.

In our experiments, we make some adaptions to the strategy annotation. On one hand, we find that the definition of *Providing Suggestions* and *Information* are often hard to differentiate. As shown in Table 7, some responses annotated as *Providing Suggestions* can also be regarded as providing useful information, and those labelled as *Information* also offer suggestions. Thus, we merge these two categories into one type of strategy, named as "*Providing Suggestions or Information*".

On the other hand, we find that the existence of the "*Others*" category largely impedes the model

performance. As illustrated in the above definition, some responses in this category are exchange of pleasantries, which is the case for approximately 50% of the responses annotated as *Others*. We argue that the exchange of pleasantries is also an essential strategy, as it can help to establish a friendly connection with the user (Miller and Rollnick, 2012). Thus, we define a new strategy category, named as "*Greetings*" for such kind of responses. We obtain the annotation of them by first using a set of regular expression matches and then manually double-checking the results. For the rest of responses labelled as *Others*, we directly regard them as unlabelled data and do not include them in the training of the strategy planning module, because we find that many of them can actually be classified as the other types of strategies but were mislabelled, and this strategy are not helpful in enhancing the response generation. The strategy distribution after the above adaptions is presented in Table 6.