PoKE: Prior Knowledge Enhanced Emotional Support Conversation with Latent Variable

Anonymous Author(s)

ABSTRACT

Emotional support conversation (ESC) task can utilize various *support strategies* to help people relieve emotional distress and overcome the problem they face, which has attracted much attention in these years. The emotional support is a critical communication skill that should be trained into dialogue systems. Most existing studies predict the *support strategy* according to current context to guide response. However, most state-of-the-art works rely heavily on external commonsense knowledge to infer the mental state of the user in every dialogue round. Although effective, they may suffer from significant human effort, knowledge update and domain change in a long run. Therefore, in this article, we focus on exploring the task itself without using any external knowledge. We find all existing works ignore two significant characteristics of ESC. (a) Abundant *prior knowledge* exists in historical conversations, such as the responses to similar cases and the general order of support strategies, which has a great reference value for current conversation. (b) There is a *one-to-many mapping relationship* between context and support strategy, i.e.multiple strategies are reasonable for a single context. It lays a better foundation for the diversity of generations. Taking into account these two key factors, we propose **P**rior **K**nowledge **E**nhanced emotional support model with latent variable, **PoKE**. The proposed model fully taps the potential of prior knowledge in terms of exemplars and strategy sequence instead of external knowledge, and then utilizes a latent variable to model the one-to-many relationship of strategy. Furthermore, we introduce a memory schema to incorporate the encoded knowledge into decoder. Experiment results on benchmark dataset show that our PoKE outperforms existing baselines on both automatic evaluation and human evaluation. Compared with the model using external knowledge, PoKE still can make a slight improvement in some metrics. Further experiments prove that abundant prior knowledge is conducive to high-quality emotional support, and a well-learned latent variable is critical to the diversity of generations.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; • **Information systems → Sentiment analysis**.

KDD '23, , Long Beach, California, United States

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

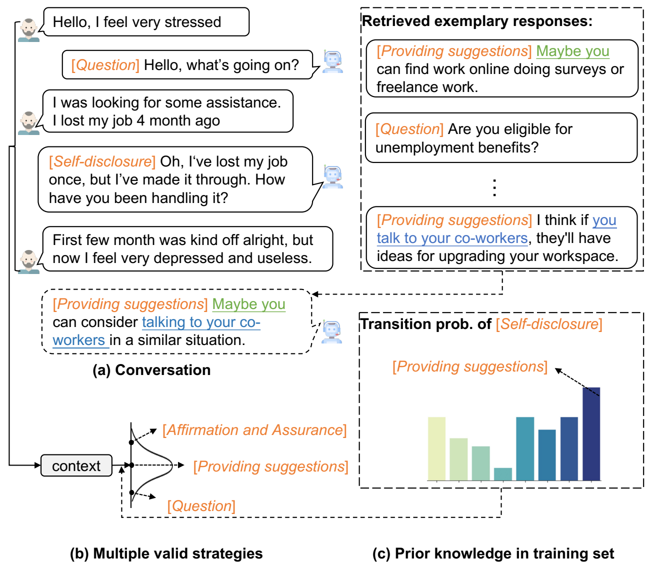


Figure 1: (a) An example to illustrate the ESC task. (b) The one-to-many mapping relationship that there exist multiple valid strategies for a single context. (c) Retrieved exemplary responses give supporter more clues to focus on seeker's problem and express strategy more accurately. Meanwhile, transition probability of strategy provides a good bias to take a correct strategy. Orange text denotes the strategy taken by supporter.

KEYWORDS

dialogue system, emotional support conversation, prior knowledge, latent variable

ACM Reference Format:
Anonymous Author(s). 2018. PoKE: Prior Knowledge Enhanced Emotional Support Conversation with Latent Variable. In *Proceedings of Proceedings of the ACM SIGKDD Conference on Knowlegde Discovery and Data Mining (KDD '23).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Emotional support conversation (ESC) [21] is an emerging and challenging task that devotes to coping effectively with help-seeker's emotional distress and helping them overcome the challenges they face. In general, a well-designed ESC system is crucial for many applications, e.g. customer service chats, mental health support, etc. [21]. Compared to the well-researched emotional and empathetic conversation [19, 24, 32], ESC focuses on reducing users' emotional stress using various *emotional support strategies*, such as *Question, Providing Suggestions*, etc.

Recently, several works have been proposed to explore the ESC task. BlenderBot-Joint [21] generates a strategy token as a prompt to guide the desired response. MISC [37] uses an off-the-shelf generative commonsense model, called COMET [3], to infer the user's mental status, where the COMET can be seen as an external commonsense knowledge base. Then, MISC encodes them additionally and fuses multiple strategies into one response to generate skillfully. GLHG [27] also utilizes COMET to generate the local intention of seeker in each dialogue round, but considers the hierarchical relationship between the seeker's global situation (summarizing the condition of the seeker) and the local intention. Although effective, the commonsense knowledge in COMET need to be carefully integrated into these models to realize their best potential, and the external knowledge base requires a great deal of effort to develop. Further, their model may not be applicable when knowledge base is updated or application domain is changed. Therefore, in this article, we emphasize on exploring the existing knowledge in the dataset and the characteristics of ESC task under the setting of no external knowledge.

Due to the characteristics of ESC, all existing works still suffer two key issues. First, all of them are limited to the scope of the current conversation, but ignore the abundant prior knowledge in global historical conversations. Moreover, they fail to model the one-to-many mapping relationship of strategy, i.e. not only one but multiple strategies could be valid for a single context. These issues lead to the challenge of generating high-quality and diverse responses. We next explain these two issues separately.

Generally, when we attempt to solve help-seeker's problems, we are adept in drawing on related prior knowledge as reference, e.g. psychologists would consult many prior classical cases relevant to current case [25]. In ESC, instead of external knowledge, there also exists much prior knowledge to rely on, such as the (1) exemplary responses to similar cases and (2) the general order of support strategies. This prior knowledge has a great reference value to help explore seeker's problem and decide the target support strategy. An explanatory example in Figure 1 illustrates how prior knowledge guides and benefits emotional support conversation. (1) The retrieved context-related responses from historical conversations, called *exemplars*, can serve as prior knowledge of response. On the one hand, some exemplars, e.g. "I think if you talk to ...", guide supporter to give more emphasis on the key problem "losing job", and thus benefit supporter to focus on and explore seeker's problem. On the other hand, some exemplars, e.g. "Maybe you can find ...", provide a hint to accurately express the target strategy *Providing suggestions* in the sentence pattern starting with "Maybe you". (2) In addition to prior knowledge of response, the transition probability of strategy calculated in training set can act as prior knowledge to help decide the current strategy. This is because the support strategies in ESC follow the procedure of three stages (*Exploration*, *Comforting* and *Action*) [11]. Figure 1(c) shows a transition probability of strategy *Self-disclosure*. It illustrates that after sharing the similar difficulties they faced, supporters tend to use *Providing suggestions* to give advice based on their experience.

Additionally, it is well known that dialogue systems have a one-to-many problem of generation, i.e. given a single context there exists multiple valid responses [44]. In ESC, the supporter is required to take reasonable strategies, so there is also a one-to-many problem of support strategy. As shown in Figure 1 (b), after the seeker states his problem, the supporter can also employ other valid strategies except for the frequently used strategy *Providing suggestions*. Taking the strategy *Question* to take a deeper look at user's problem or *Affirmation and Reassurance* to comfort the user is also a decent choice. Moreover, adopting various strategies is beneficial to diverse responses. In a nutshell, incorporating prior knowledge and modeling the one-to-many mapping relationship of strategy are critical to provide emotional support in ESC task.

To take into account these two significant characteristics of ESC, we propose a novel model called **Prior Knowledge Enhanced** emotional support conversation with latent variable model (**PoKE**). The proposed model could not only fully tap the potential of prior knowledge in terms of exemplars and strategy sequence, but also model the one-to-many mapping relationship of strategy. First, we construct prior knowledge of exemplars and strategy sequence before training. Then we use a fine-tuned dense passage retrieval (DPR) [12] to retrieve a set of responses semantically related to the input context, and build a first-order Markov transition matrix of strategy sequence from training set. To model the one-to-many mapping relationship of strategy, we introduce conditional variational autoencoder (CVAE) [35] to predict diverse probability distribution of strategy conditioned on current conversation and prior knowledge of strategy sequence. Furthermore, we assign exemplars with different attentions according to the distribution of strategy to emphasize those more relevant exemplars. Lastly, we apply the technique of memory schema to effectively incorporate encoded prior knowledge and latent variable into decoder for generation.

The key contributions are summarized as follows: (1) We explore the emotional support conversation task under the setting of no external knowledge base and propose a novel model, **PoKE**. PoKE can promote emotional support conversation by effectively modeling the prior knowledge in terms of exemplars and strategy sequence, and the one-to-many mapping relationship of strategy. (2) We utilize strategy distribution to denoise the exemplars and apply a memory schema to effectively incorporate encoded information into decoder. (3) Experiments on benchmark dataset (i.e., ESConv) of ESC task demonstrate that our method is superior to existing baselines on both automatic evaluation and human evaluation. Compared with the model using external knowledge, PoKE still can make a slight improvement in some metrics. (4) Importantly, we reveal that abundant prior knowledge is conducive to high-quality emotional support, and a well-learned latent variable is critical to the diversity of generations.

2 RELATED WORK

In this section, we first detail some existing proposed methods for the emotional support conversation. Then, because we utilize retrieved exemplars to guide generation and take a latent variable to solve the one-to-many issue of strategy, we will elaborate retrieve-based generation and one-to-many issue in dialogue system.

2.1 Emotional Support Conversation

Before the task ESC is proposed, there are two relevant well researched dialogue systems, i.e. emotional chatting [36, 40, 45] and empathetic responding [19, 20, 24, 30]. Emotional chatting needs to respond in appropriate emotion or the given emotion, such as happy or angry [45]. Empathetic responding needs to understand and feel what user is experiencing, and respond with empathy [30]. Compared with them, the emerging task of ESC aims at reducing help-seeker's emotional stress and help them explore and overcome the problem the face. The first work on ESC task, called BlenderBot-Joint, adopts a chitchat bot BlenderBot [31] as backbone and takes emotional support into account in conversation [21]. Specifically, they encode the context history and predict a strategy token. Then, they concatenate the predicted strategy token to the head of generation to guide the desired response. Meanwhile, they construct an Emotional Support Conversation dataset (ESConv) annotated with support strategies for the ESC task. Based on ESConv, MISC [37] uses an off-the-shelf commonsense model COMET [3] to infer an instant mental state of seeker and encodes them additionally. When predicting strategy, they take the probability of each predicted strategy as weight to get a weighted average representation of strategy, and utilize it for guiding a skillful generation. GLHG [27] considers the hierarchical relationship between the seeker's global situation (summarizing the condition of the seeker) and the local intention (inferred by COMET in each dialogue round) in conversation, and uses a graph neural network to encode their relationship for guiding generation. Note that both MISC and GLHG are constrained by the external knowledge in COMET, which may not be applicable to some specific domain. The external knowledge base like COMET also requires significant human effort to develop. Meanwhile, all of them are limited to the scope of current conversation but ignore abundant prior knowledge existing in the dataset. In contrast, we focus on exploring the existing knowledge and the characteristics of the ESC task without using any external knowledge.

2.2 Retrieve-based Generation

There are lots of works for retrieve-based generation. We will detail some classical studies since our main aim is not to compare with them. Some generative models, like GPT2 [28], perform well on many tasks such as machine translation and question answer [9, 14, 42]. However, recent some works have pointed out that in dialogue system, the generation model just relied on the input context suffers from some issues, such as dull generation (e.g. "I don't know") and hallucination [6, 16, 33]. To prompt model to generate more engaging response, RetNRef [41] proposes a simple but effective retrieve-and-refine strategy. RetNRef appends the retrieved context-relevant responses to context to guide the generation. Similar to this approach, Cai et al. [5] retrieves both literally-similar and topic-related exemplars to guide dialogue generation. Majumder et al. [23] employs dense passage retrieval and introduce three communication mechanisms of empathy to facilitate the generation towards empathy. For the ESC task, the abundant prior knowledge in historical conversations has great reference value for reducing seeker's emotional stress. Besides, the responses with the same strategy are similar in sentence pattern. Thus, we introduce exemplars into generation model and denoise exemplars according to the strategy distribution to emphasize those more relevant exemplars.

2.3 One-to-Many Problem

It is well known that dialogue systems have a one-to-many mapping problem that given a single context, there exist multiple valid responses [6]. To model this one-to-many feature and improve the diversity of generations, many works introduce latent variable to model a probability distribution over the potential responses [7, 8, 43, 44]. DialogVED [6] combines continuous latent variable into the encoder-decoder pre-training framework to generate more relevant and diverse responses. Except for continuous representation of latent variables, some works utilize discrete categorical variables to promote the interpretability of generation [1, 2]. For ESC, there also exist several reasonable support strategies and the corresponding responses at a certain stage. Therefore, it is required to additionally consider the one-to-many mapping relationship of strategies. In our work, we introduce a continuous latent variable to model the distribution over strategy. Furthermore, we employ this strategy distribution to denoise the exemplars at the sequence-level to focus on strategy-relevant exemplars.

3 POKE

Problem Definition. The dialogue context in ESC is an alternating set of utterances from seeker and supporter. Given a sequence of N context utterances $c = (u_1, u_2, \cdots, u_N)$, where each utterance consists of some words, $u_i = (w_1^i, w_2^i, \cdots, w_M^i)$. In the setting of ESC, each utterance of supporter is labeled with a support strategy. There are total 8 support strategies, i.e. *Question*, *Reflection of feelings*, *Information*, *Restatement or Paraphrasing*, *Others*, *Self-disclosure*, *Affirmation and Reassurance*, and *Providing Suggestions* (for more detail please refer to original paper [21]). We use m to denote the total number of strategies in the following parts. Except for the strategy, there is a brief situation s ahead of conversation summarizing the condition of seeker. In this paper, we denote the previous one support strategy taken by supporter as y', and the last utterance of seeker (called post) as p. Then, our model aims at using multiple input information and prior knowledge to generate an emotional support response r by reasonable support strategies.
PoKE Overview. Our devised model uses BlenderBot-small [31] as the backbone. The overview of our method is shown in Figure 2, which consists of four main parts: (a) **prior knowledge module** to retrieve context-related exemplary responses and build a Markov transition matrix of strategy sequence from training set, (b) **unified encoder** to encode multiple input source and exemplars by adding source tokens, (c) **latent variable module** to model the probability distribution of strategy and denoise the exemplars and (d) **knowledge-memory decoder** to effectively incorporate encoded prior knowledge and latent variable into decoder for generation.

3.1 Prior Knowledge Module

Humans tend to use prior knowledge to bias decisions [10], and there is abundant prior knowledge in historical conversation for ESC task. Due to the characteristics of ESC, we consider the prior knowledge of context-related exemplars and the general selection order of support strategies in our work.
Exemplary Responses. We use Dense Passage Retrieval (DPR) [12] as our retriever, which is a dense embedding retrieval model pre-trained on Wikipedia dump. For a target dialogue context with

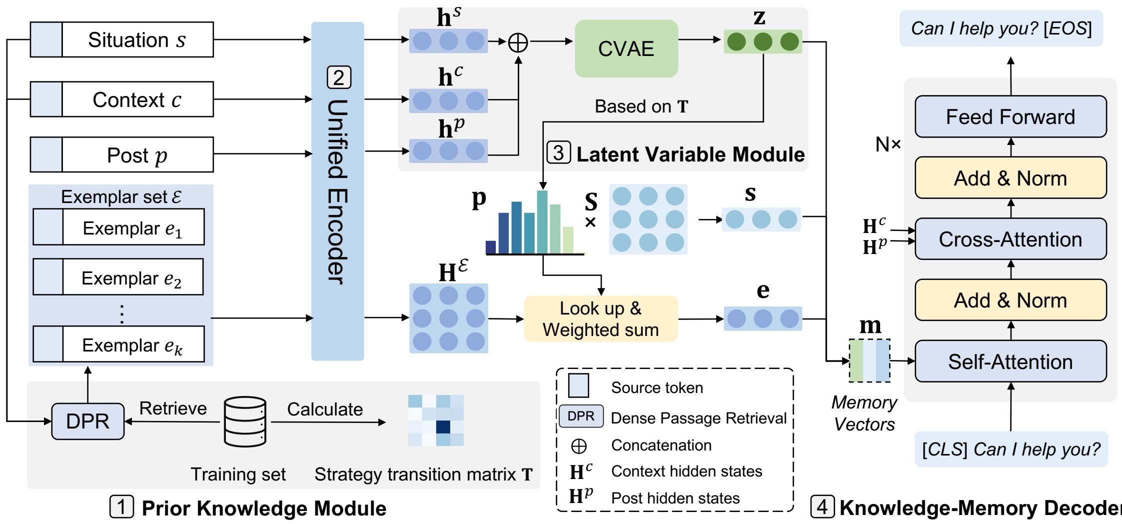


Figure 2: The model architecture of PoKE, which consists of four parts: (1) Prior Knowledge Module to retrieve context-related exemplary and construct a Markov transition matrix of strategy, (2) Unified Encoder to encode multiple input sources and exemplars, (3) Latent Variable Module to sample latent variable for modeling the distribution of strategy and denoising the exemplars (using a Look up & Weighted sum module) and (4) Knowledge-Memory Decoder to incorporate encoded prior knowledge and latent variable into the decoder.

the situation, DPR retrieves a set of possible supporter's responses from training set as *exemplars*. These exemplars have analogous context and situation to the current conversation.

Given the target context c_q with situation s_q, we concatenate them as the *query* input $q = [c_q, s_q]$. For each candidate response r_p, we get its situation s_p and do the same concatenation operation to get the *candidate* input $p = [r_p, s_p]$. Then, DPR calculates the similarity between the query and candidate input using the dot product of their embeddings:

$$\mathrm{sim}(q, p) = E_Q(q)^T E_P(p), \tag{1}$$

where $E_Q(\cdot)$ and $E_P(\cdot)$ are the encoders of query and candidate input respectively. In the end, we select top k candidate responses with the highest similarity as exemplar set $\mathcal{E} = \{e_1, e_2, \cdots, e_k\}$, where e_i denote an exemplar response. Meanwhile, we can get the corresponding strategy set $\mathcal{Y} = \{y_1, y_2, \cdots, y_k\}$, where y_i denotes the strategy label of e_i. As for inference, we use the candidate encoder $E_P(\cdot)$ to pre-compute embeddings of all responses in training set, thus to save the retrieval time of inference. To adapt DPR to the ESC task, we fine-tune DPR on the dataset of ESC.

First-Order Markov Model of Strategy Transition Before making a response, supporter need to think about reasonable strategies at different conversation stage. As pointed in [21], supporters generally follow the procedure of three stages (*Exploration*, *Comforting* and *Action*) [11] to determine the current strategy. Thus, the general strategy order calculated in the training set can serve as prior knowledge to help decide the current strategy. In our work, to urge the model to focus on the previous strategy that has been chosen, we make a simple but effective assumption that the strategy sequence follows Markov chain. Then we calculate a first-order Markov transition matrix $\mathbf{T} \in \mathbb{R}^{(m+1)\times m}$ of strategy from training set, which also considers the case of no previous strategy. Experiment in Section 4.7 demonstrates this is a simple but practical prior knowledge of strategy transition. The calculated strategy transition matrix is shown in Appendix C.1, which is used in Section 3.3 to help model strategy distribution.

3.2 Unified Encoder

We use a multi-layer Transformer-based encoder of BlenderBot [31] to encode multiple information source, including dialogue context, seeker's post, situation and the retrieved exemplars. Note that there are multiple source sequences to consider, so building a parameter-isolating encoder for each source will increase parameters and make training time-consuming. To solve this issue, we design a unified encoder, which is parameter-sharing but prepends a unique *source token* to each input. *Source token* can act as prompt to distinguish different input. There are four *Source token*s including *[CTX]*, *[POST]*, *[ST]* and *[EXEM]* representing context, post, situation and exemplar, respectively.

Firstly, we reconstruct the dialogue context by concatenating them with a special token $[SEP]$ and prepending the *source token* $[CTX]$, i.e. $c = [[CTX], w_1^1, w_2^1, \cdots, [SEP], w_1^2, w_2^2, \cdots, w_M^N]$. Then, we feed this sequence into the encoder to get its contextualized hidden states:

$$\mathbf{H}^c = \mathrm{Enc}(c), \tag{2}$$

where $\mathrm{Enc}(\cdot)$ denotes the encoder, and $\mathbf{H}^c \in \mathbb{R}^{l \times d_h}$ is the hidden states of context sequence with l tokens and hidden size of d_h. To obtain a single sentence-level representation of context, we take the first one hidden state of sequence, i.e. the output hidden state of *source token*, as the context representation:

$$\mathbf{h}^c = \mathbf{H}_0^c. \tag{3}$$

Similarly, for the given situation s, seeker's post p, and each exemplar sequence e_i in exemplars set $\mathcal{E} = \{e_i\}_{i=1}^{k}$, we prepend them with the corresponding *source token* in the same way, and use the encoder to obtain their sequence representations:

$$\begin{aligned} \mathbf{H}^s &= \mathrm{Enc}(s),\ \mathbf{h}^s = \mathbf{H}_0^s; \\ \mathbf{H}^p &= \mathrm{Enc}(p),\ \mathbf{h}^p = \mathbf{H}_0^p; \\ \mathbf{H}^{e_i} &= \mathrm{Enc}(e_i),\ \mathbf{h}^{e_i} = \mathbf{H}_0^{e_i}, \end{aligned} \tag{4}$$

and we use $\mathbf{H}^{\mathcal{E}} = [\mathbf{h}^{e_1}, ..., \mathbf{h}^{e_k}]$ to express the representation of the entire exemplars set $\mathcal{E}$. These representations of multiple source are used to model the latent variable in Section 3.3 and fed into the decoder for generation in Section 3.4.

3.3 Latent Variable Module

In this section, we introduce the workflow of modeling latent variable and how to build strategy distribution to obtain representations of mixed strategy and denoised exemplars.

Latent Variable. To address the one-to-many mapping issues of responses and support strategy at the same time, we utilize the Conditional Variational Autoencoder (CVAE) [35] to model the latent variable. The basic idea of CVAE is to encode the response r along with input conditions to a probability distribution instead of a point. Then, CVAE employs a decoder to reconstruct the response r by using latent variable z sampled from the distribution. We jointly use dialogue context c, situation s, and seeker's post p as the input conditions for estimating the latent variable $\mathbf{z} \in \mathbb{R}^{d_z}$. For brevity, we use a symbol $x = \{c, s, p\}$ to denote the input conditions.

CVAE is trained by maximizing a variational lower bound $\mathcal{L}_{ELBO}$, consisting of two terms: negative likelihood loss of decoder and K-L regularization:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \mathcal{L}_{nll} + \mathcal{L}_{kl} \\ &= \mathbf{E}_{q_\phi(\mathbf{z}|x,r)}\left[\log p_\theta(r|\mathbf{z}, x)\right] \\ &\quad - KL(q_\phi(\mathbf{z}|r, x) \| p_\theta(\mathbf{z}|x)) \end{aligned} \tag{5}$$

where $q_\phi(\mathbf{z}|r, x)$ and $p_\theta(\mathbf{z}|x)$ are called recognition network and prior network respectively (with parameters ϕ and θ), and $p_\theta(r|\mathbf{z}, x)$ is the decoder for generation, which will be illustrated in Section 3.4. Then we can sample latent variable $\mathbf{z}$ from the well-learned Gaussian distribution (for more detail please see Appendix D).

In order to regularize the latent space and model the one-to-many mapping relationship of strategy, we design an extra optimizing objective of strategy, $\mathcal{L}_y$. A strategy prediction network $p_\theta(y|\mathbf{z})$ is used to recover the strategy label y by latent variable $\mathbf{z}$:

$$\mathcal{L}_y = \mathbf{E}_{q_\phi(\mathbf{z}|x,r)}[p_\theta(y|\mathbf{z})], \tag{6}$$

$$p_\theta(y|\mathbf{z}) = \mathbf{p}_y, \tag{7}$$

where $\mathbf{p}$ is denoted as the distribution of strategy. We calculate $\mathbf{p}$ by a fully connected layer and based on the transition matrix $\mathbf{T}$ obtained in Section 3.1:

$$\mathbf{p} = \mathrm{softmax}(\mathbf{W}_y \mathbf{z} + \mathbf{b}_y + \mathbf{T}_{y'}), \tag{8}$$

where $\mathbf{W}_y \in \mathbb{R}^{m \times d_z}$ and $\mathbf{b}_y \in \mathbb{R}^m$ are the learnable parameters, y' is the previous strategy taken by the supporter, which is provided in dataset, and $\mathbf{T}_{y'} \in \mathbb{R}^m$ is the transition probability of y'.

Representation of Mixed Strategy. To model the complexity of strategy expressed in one utterance, and consider multiple valid support strategies, we adopt a method of mixed strategy representation inspired by [19, 37]. First, we create a strategy codebook $\mathrm{S} \in \mathbb{R}^{m \times d_h}$ storing the representation of each strategy. Then, we utilize the strategy distribution $\mathbf{p}$ to get a weighted combination of S, which blends multiple strategy in one representation $\mathbf{s} \in \mathbb{R}^{d_h}$:

$$\mathbf{s} = \mathbf{p} \cdot \mathrm{S}. \tag{9}$$

Representation of Denoised Exemplars. In general, the retrieved exemplars contain irrelevant support strategies. To denoise the exemplars in terms of strategy, we first look up the strategy probability from strategy distribution as the weight for each exemplar. Then, we combine all exemplar representations $\mathbf{H}^{\mathcal{E}}$ at sequence level to obtain a single representation $\mathbf{e} \in \mathbb{R}^{d_h}$ of denoised exemplars:

$$\begin{aligned} \mathbf{e} &= \sum_{i=1}^{k} \frac{\mathbf{p}_{y_i}}{\sum_{j=1}^{k} \mathbf{p}_{y_j}} \cdot \mathbf{H}_i^{\mathcal{E}} \\ &= \sum_{i=1}^{k} \frac{\mathbf{p}_{y_i}}{\sum_{j=1}^{k} \mathbf{p}_{y_j}} \cdot \mathbf{h}^{e_i}, \end{aligned} \tag{10}$$

where $y_i \in [0, m)$ is the strategy label of exemplar e_i, $\mathbf{p}_{y_i}$ denotes the probability of y_i, and the denominator is normalization.

The representations of the latent variable, mixed strategy, and denoised exemplars will be incorporated into the decoder to guide generation, which is illustrated in the following section.

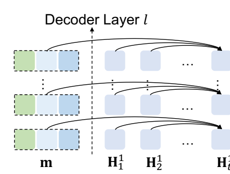


Figure 3: Illustration of memory schema applied in self-attention module in decoder: H_t^l attends both $\mathrm{H}_{<t}^l$ and *memory* vectors m at each layer.

3.4 Knowledge-Memory Decoder

After getting the above-mentioned representations of the latent variable $\mathbf{z}$, mixed strategy $\mathbf{s}$, and denoised exemplars $\mathbf{e}$, the consequent problem is how to effectively incorporate them into decoder[1] for generation. Inspired by [6, 15], we apply a memory schema to inject these encoded knowledge. The memory schema regards the representations of the encoded knowledge as additional *memory vectors* $\mathbf{m}$ for each self-attention layer to attend, as illustrated in Figure 3. We first project the vector of latent variable $\mathbf{z}$ into the d_h-dimensional space:

$$\mathbf{z}_h = \mathbf{W}_z \mathbf{z}, \tag{11}$$

where $\mathbf{W}_z \in \mathbb{R}^{d_h \times d_z}$ is the projection matrix. Thus, we can obtain the memory vectors $\mathbf{m} = [\mathbf{z}_h, \mathbf{s}, \mathbf{e}] \in \mathbb{R}^{3 \times d_h}$ by stacking $\mathbf{z}_h, \mathbf{s}, \mathbf{e}$. Then, we modify the computation of key vector K and value vector

[1]We apply the decoder in BlenderBot [31] to model the distribution $p_\theta(r|\mathbf{z}, x)$ and optimize the negative likelihood loss $\mathcal{L}_{nll}$ in Eq. (5).

V in each self-attention layer by incorporating the memory vectors. Concretely, memory vectors $\mathbf{m}$ are prepended to the hidden states $\mathbf{H}^l$, denoted as $[\mathbf{m}, \mathbf{H}^l]$, to calculate the key vector K and value vector V in each self-attention layer:

$$K = [\mathbf{m}, \mathbf{H}^l]\mathbf{W}^K$$
$$V = [\mathbf{m}, \mathbf{H}^l]\mathbf{W}^V \tag{12}$$

where $\mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_h \times d_h}$ are parameter matrices of key and value, respectively. The memory schema is equivalent to adding some virtual tokens to the response sequence at each layer and enables the decoder to attend all knowledge directly. Besides, we perform multi-head attention over the encoded context $\mathbf{H}^c$ and post $\mathbf{H}^p$ for each layer's cross attention inspired by [37]. In this way, the knowledge is injected into the decoder to guide the generation at each step.

3.5 Training Objective

The final learning objective is defined as the combination of CVAE loss in Eq. (5) and strategy prediction loss in Eq. (6)

$$\mathcal{L}(\varphi) = \mathcal{L}_{ELBO} + \lambda \mathcal{L}_y, \tag{13}$$

where φ denotes the parameters of PoKE, and λ controls the degree of regularizing latent space by strategy. However, directly training this objective may suffer two optimizing challenges, i.e. *KL-vanishing* and *strategy-unstablity*. To alleviate them, we adopt two annealing methods including *KL-annealing* and *Strategy-annealing*.

KL-vanishing. This problem lies in that the decoder overly attends the encoded information of context, and thus ignore the latent variable $\mathbf{z}$, leading to the failure of encoding informative $\mathbf{z}$ [4]. We adopt a *KL annealing* [44] method to solve this issue, i.e. gradually increasing the weight of KL loss in Eq. (5) from 0 to 1 during train.

Strategy-unstablity. At the early stage of training, using latent variable tends to predict unstable and incorrect strategy distribution. Then, this error is propagated to the representation of denoised exemplars and the decoder [19]. To stabilize the training stage, we take a measure of *strategy-annealing*. That is, we use the true distribution of target strategy instead of the predicted distribution by a certain probability α_t and anneal it over time:

$$\alpha_t = \beta + (1 - \beta)e^{-\frac{t}{T}} \tag{14}$$

where β is annealing rate, t is the current iteration step, and T is the annealing steps.

4 EXPERIMENTS

4.1 Dataset

We use the emotional support conversation dataset **ESConv** [21] to evaluate our method. ESConv contains a total of 1,053 dialogues and 31,410 utterances. Each conversation contains a seeker's situation and a dialog context, and each utterance of supporter is annotated by a support strategy that is taken by the supporter. There are 8 different support strategies roughly uniformly distributed across the dataset. Due to the long turns in ESC, we cut each conversation into several pieces with 10 utterances and the last utterance is supporter's response. For training and validation, we split the ESConv into the sets of training/validation/test with the proportions of 7:1.5:1.5. The statistics of original ESConv is shown in Table 8 and the split ESConv is shown in Table 2.

4.2 Evaluation Protocol

Following existing methods, we adopt automatic and human evaluation to evaluate our model and compare with strong baselines.

Automatic Evaluation. We employ perplexity (PPL), BLEU-1 (B-1), BLEU-2 (B-2), BLEU-3 (B-3), BLEU-4 (B-4) [26], ROUGE-L (R-L) [18], Distinct-1 (D-1), Distinct-2 (D-2) [17] automatic metrics to evaluate model performance. PPL is defined as e raised to the power of cross-entropy and is kept as a reference. B-1/2/3/4 and ROUGE-L measure the number of matching n-grams between the model-generated response and the human-produced reference, which reflect the quality generation. D-1/2 is calculated by the number of distinct 1/2-grams divided by the total number of generated words, which indicates the generation diversity.

Human Evaluation. We randomly sample 64 dialogues from the test set and generate responses using our model and one baseline. Then, 3 annotators with relevant backgrounds are prompted to choose the better response based on indicators in [21]: (1) Fluency: which one are more fluent? (2) Identification: which one is more helpful in identifying the seeker's problems? (3) Comforting: which one is more skillful in comforting the seeker? (4) Suggestion: which one provides more helpful suggestions? (5) Overall: generally, which emotional support do you prefer?

4.3 Compared Methods

Since our main purpose is to explore the ESC task under the setting of no external knowledge, we place emphasis on those baselines that do not require any external knowledge. We compare our model with the following baselines, also including a model using external knowledge:

(1) **Transformer** [38]. We use a standard Transformer model, which is trained from scratch by a negative likelihood objective.
(2) **Multi-TRS** [29]. Multi-TRS is a multitask Transformer trained with an additional learning objective of predicting the target emotion.
(3) **MoEL** [19]. MoEL models the distribution of emotion and assigns it to multiple Transformer decoders to softly combine their output.
(4) **BlenderBot-Joint** [21]. BlenderBot-Joint is built on a pre-trained dialogue model, BlenderBot [31]. It generates a strategy token and attaches it to the head of response to guide the desired response.
(5) **MISC** [37]. MISC is also built on BlenderBot but requires external knowledge. It injects external knowledge by inferring the user's fine-grained emotional status using COMET [3]. When generating, they first predict a probability distribution of strategy and use it to obtain a weighted average representation of strategy for guiding generation.

Note that Multi-TRS and MoEL require the emotion label of seeker for training, so we use the conversation-level emotion label provided in ESConv dataset to train them. For a fair comparison, we apply the same hyperparameters for all baselines. The detail of implementation is illustrated in Appendix B

Table 1: Result of automatic evaluation on baseline models and PoKE. * denotes the model requiring external knowledge. The best performance under the setting of no external knowledge is highlighted in bold. Considering the model using external knowledge, the best score is underlined. ↓ indicates that the lower the value, the better the performance.

Model	PPL ↓	B-1 ↑	B-2 ↑	B-3 ↑	B-4 ↑	R-L ↑	D-1 ↑	D-2 ↑
w/o external knowledge								
Transformer	53.85	15.07	4.67	1.78	0.84	13.26	1.49	12.97
MultiTRS	53.08	15.06	4.67	1.74	0.77	13.45	1.56	13.65
MoEL	53.61	17.98	5.96	2.27	1.02	14.08	1.12	11.25
BlenderBot-Joint	**15.71**	16.99	6.18	2.95	1.66	15.13	3.27	20.87
with external knowledge								
MISC*	16.62	17.71	6.44	3.00	1.62	15.57	3.65	22.25
PoKE	15.84	**18.41**	**6.79**	**3.24**	**1.78**	**15.84**	**3.73**	**22.03**

Table 2: Statistics of processed split ESConv.

Category	Train	Valid	Test
# Dialogues	12,235	2,616	2,794
Avg. length of turns	8.57	8.58	8.65
Avg. length of utterances	18.34	18.31	17.04
Avg. length of contexts	157.35	157.18	147.52

Table 3: Human evaluation results.

Comparisons	Indicators	Win	Lose	Tie
PoKE vs. MoEL	Flu.	**61.0**	8.2	29.2
	Ide.	**64.6**	13.3	20.5
	Com.	**68.7**	15.3	14.3
	Sug.	**65.6**	14.8	17.9
	Ove.	**70.2**	14.8	13.3
PoKE vs. MISC*	Flu.	**30.2**	23.4	46.3
	Ide.	**37.5**	29.6	32.8
	Com.	**43.2**	33.3	22.9
	Sug.	**36.4**	30.2	33.3
	Ove.	**45.8**	34.8	19.2

4.4 Experiment Results

Automatic Evaluation. The automatic evaluation results compared with baseline models are shown in Table 1. The results show that PoKE significantly outperforms baselines on the majority of metrics. This indicates PoKE can generate high-quality and more diverse responses, which proves the superiority of PoKE.

Specifically, the Transformer-based models, i.e. Transformer, Multi-TRS, and MoEL, do not perform well on ESConv. This is because these models are initialized with random parameters and trained on ESConv from scratch. Besides, their training objectives are irrelevant to the support strategy and the characteristics of emotional support, so they are hard to handle the challenging ESC task. As for the BlenderBot-based model, i.e. BlenderBot-Joint and MISC, they gain an improvement by a large margin compared to the previous baselines. It is due to the pre-trained dialogue model BlenderBot, which is trained on a large conversation dataset containing multiple conversation skills [34]. For MISC, its D-1 and D-2 are comparatively higher, indicating that it tends to generate more diverse responses. This is because MISC incorporates varied information about seeker's mental state from external knowledge in COMET and merges mixed strategies into one response. However, due to the issue of the local scope of conversation and the one-to-many relationship of strategy, there is still room for improvement.

Compared to those baselines without external knowledge, our proposed model PoKE improves significantly on the majority of metrics. This demonstrates that by effectively exploiting global prior knowledge from historical conversations, PoKE can get more clues to focus on seeker's problem and generate more relevant responses. For the MISC that uses additional external knowledge, PoKE still can obtain a slight improvement in some metrics except diversity. However, PoKE almost achieves the same diversity performance with MISC. This benefits from using latent variable to model the one-to-many mapping relationship between context and support strategy, and latent variable makes it easier to sample infrequent strategies. Moreover, the technique of mixed strategy facilitates expressing diverse strategies in one response. As for PPL, both MISC and PoKE perform worse than BlenderBot-Joint. A recent work proves that PPL is not so reliable for evaluating text quality [39], and due to the insignificant difference of PPL (PoKE only drops by 0.13), we do not further refine the model.

Human Evaluation. The best Transformer-based model MoEL and BlenderBot-based model MISC are used to do a further human evaluation, which is shown in Table 3. The result displays that our proposed PoKE is superior to MoEL and MISC on all indicators, which is nearly consistent with the automatic evaluation results. Significantly, our PoKE outperforms MoEL by a large margin. This is partly due to the pre-trained backbone model Blenderbot, which contains abundant knowledge about communication skills. Compared with MISC, our PoKE that does not rely on external knowledge also achieves a decent performance, especially on aspects of Comforting and Identification. This indicates that the retrieved context-related exemplars contains a lot of information relevant to seeker's problem, which gives model more clues to identify the current problem and comfort seeker.

Overall speaking, under the setting of no external knowledge, our proposed PoKE is superior to baselines on both automatic

Table 4: Analysis of denoised exemplars

Model	PPL ↓	B-2 ↑	B-4 ↑	R-L ↑	D-1 ↑	D-2 ↑
PoKE	**15.84**	**6.79**	**1.78**	**15.84**	**3.73**	**22.03**
w/o denoising	15.81	6.76	1.69	15.60	3.58	21.23

Table 5: The results of PoKE with different CVAE structure.

Model	PPL ↓	B-2 ↑	B-4 ↑	R-L ↑	D-1 ↑	D-2 ↑
Normal CVAE	**15.84**	**6.79**	**1.78**	**15.84**	**3.73**	**22.03**
Variant CVAE	16.07	6.76	1.78	15.61	3.28	20.58

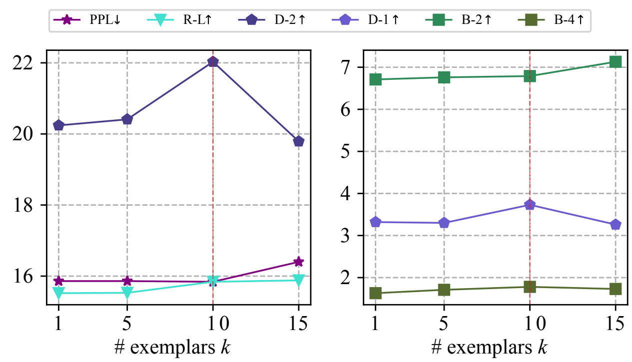


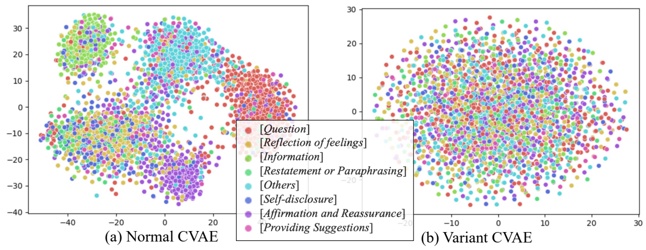
Figure 4: Analysis of the number of exemplars k

evaluation and human evaluation, which proves the superiority and effectiveness of PoKE. Besides, abundant prior knowledge and latent variable help provide better and diverse emotional support in dialogue system.

4.5 Effect of Exemplars

In this section, we explore the effect of exemplars in terms of denoising and quantity. To verify the denoised exemplars in Eq. (10), we implement another variant of PoKE without denoising exemplars, i.e. the representation of exemplars is calculated by averaging, i.e. $\mathbf{e} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{h}^{e_i}$. The result is displayed in Table 4. All metrics drop when not denoising the exemplars. This demonstrates that the strategies of some retrieved exemplars are irrelevant to the current context, and need to be used selectively.

Figure 4 shows that as the number of exemplars increases, the overall performance tends to improve first and then decrease. This is because when exemplars are insufficient, PoKE lacks adequate reference information. When exemplars are too many, there is a lot of redundant and noisy information to distract the generation. Although PoKE (k = 15) can utilize plentiful information to improve quality (higher B-2 and R-L), it pays the price of decreased fluency and diversity (very low PPL and D-1/2). In the end, we decide to retrieve 10 exemplars for each sample considering both the overall effect and training efficiency.



(a) Normal CVAE (b) Variant CVAE

Figure 5: t-SNE visualization of the posterior z for test responses with 8 strategies. (a) Normal CVAE: strategy only acts as output to regularize the latent space. (b) Variant CVAE: strategy is only as input condition of latent variable.

Table 6: The results of ablation study on PoKE variants.

Model	PPL ↓	B-2 ↑	B-4 ↑	R-L ↑	D-1 ↑	D-2 ↑
PoKE w/o **e**	**15.74**	6.59	1.63	15.54	3.53	21.45
PoKE w/o **T**	15.84	6.66	1.70	15.80	3.67	21.65
PoKE w/o **z**	16.13	6.57	1.64	15.42	3.36	20.59
PoKE	15.84	**6.79**	**1.78**	**15.84**	**3.73**	**22.03**

4.6 Effect of CVAE Structure

In this section, we adjust the structure of CVAE to explore the reasonable manner of utilizing strategy. For normal CVAE, namely the PoKE, strategy is only used as the output to regularize the latent space (Eq. (6)). Here, we consider a variant CVAE that strategy is merely as input condition to model latent variable, i.e. the recognition network becomes $q_\phi(\mathbf{z}|x, r, y)$ and $\mathcal{L}_y$ is ignored. We conduct quantitative and visualization experiments to compare these two structures of CVAE.

Table 5 shows that the overall performance of variant CVAE drops a lot, especially in diversity. Meanwhile, the visualization in Figure 5(b) exhibits that the latent space is independent of strategy, so strategy information is vanished from the latent variable. This demonstrates that only taking strategy as input is inadequate to model an informative latent space. In contrast, PoKE has a better diversity (Table 5) and can learn a meaningful latent space highly correlated with the support strategy (Figure 5(a)). This demonstrates that PoKE effectively regularizes the latent space and incorporates the informative latent variable into decoder to generate diverse responses.

4.7 Ablation Study

To understand the importance of prior knowledge and latent variable for providing better emotional support, we conduct an ablation study to investigate the effect of the key components in PoKE. We design several variants of PoKE by removing some specific parts:

PoKE w/o e. Remove the prior knowledge of exemplars, i.e. the the denoised exemplars vector **e** is excluded from *memory* vectors.

PoKE w/o T. Remove the prior knowledge of strategy sequence, i.e. the first-order Markov transition matrix **T** of strategy is ignored when modeling the distribution of strategy.

Table 7: An example of ESConv and the responses generated from PoKE and other SOTA models. The retrieved context-related exemplars are also displayed. Orange text denotes the strategy label of exemplar. Relevant parts have the same color.

Situation	My partner broke up with me, but he did not tell me the reason why.
Context	*Seeker*: I feel like I am in depression. *Supporter*: *[Question]* I'm sorry to hear that. What do you think may have led up to these feelings of being depressed? *Seeker*: My partner broke up with me impromptu. *Supporter*: *[Self-disclosure]* I've been through a few bad breakups myself in the past, so I know how you must be feeling right now. *Seeker*: How did you deal with them, and specifically what was your coping mechanism?
Exemplars	*[Affirmation and Reassurance]* **I can understand your situation**. You think that its normal for you, but other has different outlook. ... *[Providing suggestions]* I tried to focus on other areas of my life and to **critically reflect on my relationship**. Largely, I threw myself into my work and tried to better that aspect of my life, and **hung out with people** that I knew cared about me.
BlenderBot-Joint	I have been able to cope with my ex by talking to him about my feelings. I have found that talking to people helps me to cope.
MISC*	Have you considered seeing a therapist?
PoKE	**I can definitely relate to that**. I have been in that situation, and it really helped to **get outside and have a good conversation** about why I feel that way. I think that it is important to remember to **think about the bright side of your relationship**, too.
Gold Response	Ha, well at first I cried and was mad and just moped and hated everything, but eventually I got tired of no getting anything out of my days. I slowly realized that I had put myself in slow motion and then slowly started to do things I used to do to maybe come out of it.

PoKE w/o z. The CVAE module is removed, and we directly use input conditions instead of latent variable to predict the strategy. In addition, the latent variable **z** is removed from *memory* vectors.

Table 6 shows the results of ablation studies. We can find that almost all variants perform worse than the PoKE, which verifies each component in PoKE. The results of PoKE w/o **e** and w/o **T** show that both generation quality and diversity get worse after removing prior knowledge. This suggests that explicitly using prior knowledge in historical conversations benefits more relevant responses, and plenty of various exemplars help generate responses with higher diversity. However, compared to PoKE, the PPL of PoKE w/o **e** improves slightly. We speculate that exemplars contain some token-level noise, thus impairing fluency. We leave the research of denoising exemplars at the token-level as future work. Regarding the PoKE w/o **e**, D-1 and D-2 drop by a large margin. This result is as expected because the latent variable models the one-to-many mapping relationship of strategy. By sampling latent variable, randomness is introduced to strategy distribution and enables infrequent strategies to be considered.

5 CASE STUDY

Table 7 shows an example of ESConv and the responses generated from PoKE and other SOTA models. From the seeker's situation, we can know the seeker has emotional stress of breaking up with his partner, and he is asking for suggestions. BlenderBot-Joint directly provide a suggestion, but it is not suitable or commonly used. MISC uses the COMET to infer the commonsense that seeing a therapist may help overcome the problem and utilizes it for guiding generation, but it does not combine its own experience. The gold reference shares his solutions of getting rid of emotional stress. Compared with them, PoKE makes a better response thanks to latent variable and prior knowledge. PoKE expresses a mixed strategy smoothly, i.e. affirming the seeker before sharing advice. Additionally, PoKE utilizes abundant reference information about strategy expression and suggestions from exemplars explicitly or implicitly. For instance, (1) "I can definitely..." expresses the strategy of *Affirmation and Reassurance* by explicitly referring to the sentence pattern of the first exemplar, and (2) "get outside ..." as well as "think about ..." implicitly incorporate the suggestions of the last exemplar into the response. Besides, we visualize the correlation between the prior knowledge of strategy and the predicted strategy distribution in Figure 7, which is detailed in Appendix C.2.

6 CONCLUSION

In this paper, we explore the emotional support conversation under the setting of no external knowledge and propose PoKE, a prior knowledge enhanced model with latent variable to provide emotional support in conversation. The proposed PoKE could utilize the prior knowledge in terms of exemplars and strategy sequence, and models the one-to-many mapping relationship of strategy. Then, PoKE utilizes strategy distribution to denoise the exemplars and applies a memory schema to incorporate encoded information into decoder. The experiments on automatic and human evaluation demonstrate the superiority and diversity of PoKE without external knowledge. Moreover, the analytical experiments prove that PoKE can effectively utilize prior knowledge to generate better emotional support and learn an informative latent variable to respond with high diversity. In future work, we will further refine our model to outperform the methods using external knowledge and explore the manner of efficiently incorporating external knowledge.

REFERENCES

[1] Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 85–96.

[2] Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2021. PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2513–2525.

[3] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the Association for Computational Linguistics*. 4762–4779.

[4] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of the Conference on Computational Natural Language Learning*. 10–21.

[5] Hengyi Cai, Hongshen Chen, Yonghao Song, Xiaofang Zhao, and Dawei Yin. 2020. Exemplar Guided Neural Dialogue Generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 3601–3607.

[6] Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. 2022. DialogVED: A Pre-trained Latent Variable Encoder-Decoder Model for Dialog Response Generation. In *Proceedings of the Association for Computational Linguistics*. 4852–4864.

[7] Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. *arXiv preprint arXiv:1908.11527* (2019).

[8] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352* (2018).

[9] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. *Advances in Neural Information Processing Systems* 31 (2018).

[10] Kathleen A Hansen, Sarah F Hillenbrand, and Leslie G Ungerleider. 2012. Effects of prior knowledge on decisions made under perceptual vs. categorical uncertainty. *Frontiers in neuroscience* 6 (2012), 163.

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

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[13] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

[14] Mike Lewis and Angela Fan. 2018. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

[15] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 4678–4699.

[16] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 110–119.

[17] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[18] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[19] 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 and the International Joint Conference on Natural Language Processing*. 121–132.

[20] Zhaojiang Lin, Peng Xu, Genta Indra Winata, Farhad Bin Siddique, Zihan Liu, Jamin Shin, and Pascale Fung. 2020. CAiRE: An End-to-End Empathetic Chatbot. In *Proceedings of the Conference on Artificial Intelligence*. 13622–13623.

[21] 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 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*. 3469–3483.

[22] Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *CoRR* abs/1711.05101 (2017).

[23] Navonil Majumder, Deepanway Ghosal, Devamanyu Hazarika, Alexander F. Gelbukh, Rada Mihalcea, and Soujanya Poria. 2022. Exemplars-Guided Empathetic Response Generation Controlled by the Elements of Human Communication. *IEEE Access* 10 (2022), 77176–77190.

[24] Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander F. Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking Emotions for Empathetic Response Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 8968–8979.

[25] Harald A Mieg. 2001. *The social psychology of expertise: Case studies in research, professional domains, and expert roles.* Psychology Press.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Association for Computational Linguistics*. 311–318.

[27] 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 Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. 4324–4330.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[29] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. I Know the Feeling: Learning to Converse with Empathy. *CoRR* abs/1811.00207 (2018).

[30] 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 Conference of the Association for Computational Linguistics*. 5370–5381.

[31] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, et al. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. 300–325.

[32] Sahand Sabour, Chujie Zheng, and Minlie Huang. 2022. CEM: Commonsense-Aware Empathetic Response Generation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. 11229–11237.

[33] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *Findings of the Association for Computational Linguistics: EMNLP*. 3784–3803.

[34] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can You Put it All Together: Evaluating Conversational Agents' Ability to Blend Skills. In *Proceedings of the Association for Computational Linguistics*. 2021–2030.

[35] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).

[36] Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating Responses with a Specific Emotion in Dialog. In *Proceedings of the Association for Computational Linguistics*. 3685–3695.

[37] 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 60th Annual Meeting of the Association for Computational Linguistics*. 308–319.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[39] Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. 2022. Perplexity from PLM Is Unreliable for Evaluating Text Quality. https://doi.org/10.48550/ARXIV.2210.05892

[40] Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. 2019. Emotion-Aware Chat Machine: Automatic Emotional Response Generation for Human-like Emotional Interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1401–1410.

[41] Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP*. 87–92.

[42] Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 9378–9385.

[43] Min Zeng, Yisen Wang, and Yuan Luo. 2019. Dirichlet Latent Variable Hierarchical Recurrent Encoder-Decoder in Dialogue Generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 1267–1272.

[44] 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*. 654–664.

[45] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *AAAI*.

A ESCONV DATASET

The detailed statistics of the original ESConv are shown in Table 8. The long average length of turns (29.8) indicates that the ESC task needs more turns to provide an effective emotional support for seeker.

Table 8: Statistics of ESConv.

Category	Total	Support	Seeker
# Dialogues	1,053	-	-
# Utterances	31,410	14,855	16,555
Avg. length of turns	29.8	14.1	15.7
Avg. length of utterances	17.8	20.02	15.7
Avg. length of situations	22.85	-	-

B IMPLEMENTATION DETAILS

Similar to BlenderBot-Joint [21] and MISC [37], we use BlenderBot Small [31] as our model's backbone. The default size of hidden state d_h in BlenderBot Small is 512, and the dimension of latent variable d_z is set as 64 by parameter search. According to the result in Section 4.5, we retrieve $k = 10$ exemplars for each context. The coefficient λ in Eq. (13) is set to 1.0. For stable optimization, the total KL annealing steps with 10000, strategy annealing rate β with 1×10^{-3} and steps T with 1000 achieves the best performance. The batch size of training and validation is set to 20 and 50 respectively. We use optimizer AdamW [22] to optimize our model. We train the model for 8 epochs and select the best models based on the perplexity of the validation data. For decoding, we employ Top-k and Top-p sampling methods in previous work [21], and set $k = 30$, $p = 0.9$, temperature $\tau = 0.9$ and repetition penalty to 1.03. For a fair comparison, all methods are implemented using the same hyperparameters and on the Tesla V100 GPU.

C PRIOR KNOWLEDGE OF STRATEGY

C.1 Markov Transition Matrix of Strategy

The first-order Markov transition matrix $\mathbf{T} \in \mathbb{R}^{(m+1)\times m}$ of strategy calculated in the training set is shown in Figure 6. The transition matrix $\mathbf{T}$ containing prior knowledge of strategy selection is simple but practical in ESC task, which is demonstrated in Section 4.7. From this matrix, we can find useful prior knowledge about general patterns of strategy selection. For instance, supporters tend to take *Question* as a conversation starter to acquire more seeker's information. After sharing the similar difficulties they faced, supporters tend to use *Providing suggestions* to give advice based on their experience, and so on.

C.2 Applied in Case Study

For the case in Table 7, we visualize the correlation between the prior knowledge of strategy and the predicted strategy distribution in Figure 7. In that case, the previous strategy taken by the supporter is *Self-disclosure*. According to the first-order Markov transition matrix $\mathbf{T}$ in Figure 6, we can obtain the transition probability of the strategy *Self-disclosure*. Besides, we use Eq. (8) to predict the strategy

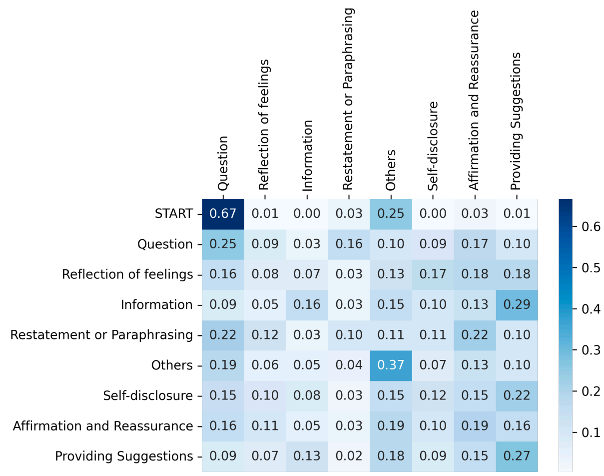


Figure 6: First-order Markov transition matrix T of strategy calculated in training set. START means the current conversation turn is the first round, and there is no previous strategy.

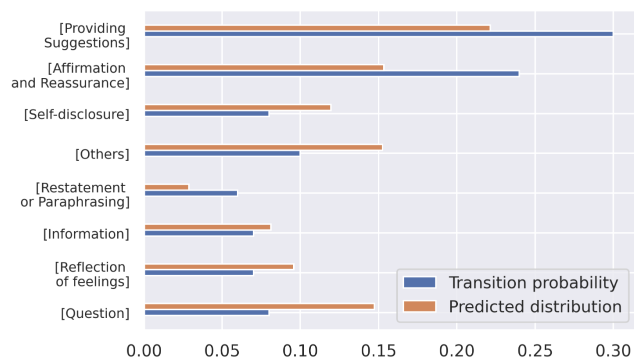


Figure 7: The visualization of transition probability of the previous strategy *Self-disclosure* taken by the supporter and the predicted distribution in case study.

distribution via latent variable and transition probability. Figure 7 shows that the two distributions have a similar pattern, such as the maximum probability of *Providing Suggestions* and the most unlikely strategy *Restatement or Paraphrasing*. This indicates that the simple transition matrix of strategy can provide practical prior knowledge for current strategy decisions. Moreover, according to the predicted strategy distribution, PoKE can further adjust strategy distribution based on the current context (e.g. higher probability of *Question* and *Self-disclosure*).

D CONDITIONAL VARIABLE AUTOENCODER

Mathematically, our goal is to maximize the conditional likelihood of response r for the given conditions x:

$$p(r|x) = \int p(r|\mathbf{z}, x)p(\mathbf{z}|x)d\mathbf{z}, \tag{15}$$

where $p(\mathbf{z}|x)$ involves an intractable marginalization over the latent variable $\mathbf{z}$. To solve that probelm and model the latent variable, CVAE uses a prior network $p_\theta(\mathbf{z}|x)$ to approximate $p(\mathbf{z}|x)$, and a recognition network $q_\phi(\mathbf{z}|x,r)$ to approximate true posterior $p(\mathbf{z}|x,r)$. In general, the latent variables from prior network and recognition network are assumed to fit multivariate Gaussian distribution with a diagonal covariance matrix, i.e. $p_\theta(\mathbf{z}|x) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$ and $q_\phi(\mathbf{z}|r,x) \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\sigma}'^2\mathbf{I})$. Then, CVAE can be trained by maximizing a variational lower bound, consisting of two terms: negative likelihood loss of decoder and K-L regularization:

$$\begin{aligned}\mathcal{L}_{ELBO}(\theta,\phi;r,x) &= \mathcal{L}_{nll} + \mathcal{L}_{kl} \\ &= \mathbf{E}_{q_\phi(\mathbf{z}|x,r)}\left[\log p_\theta(r|\mathbf{z},x)\right] \\ &\quad - KL\left(q_\phi(\mathbf{z}|r,x) \| p_\theta(\mathbf{z}|x)\right) \\ &\leq \log p(r|x),\end{aligned} \tag{16}$$

where $p_\theta(r|\mathbf{z},x)$ is the decoder network for generation, which is illustrated in Section 3.4.

In CVAE, both the prior network and recognition network apply the structure of multilayer perceptron, and then we can calculate the mean $\boldsymbol{\mu} \in \mathbb{R}^{d_z}$ and variance $\boldsymbol{\sigma} \in \mathbb{R}^{d_z}$ in multivariate Gaussian distribution by:

$$\begin{bmatrix} \boldsymbol{\mu} \\ \log\left(\boldsymbol{\sigma}^2\right) \end{bmatrix} = \mathrm{MLP}_p(x) = \mathbf{W}_p[\mathbf{c};\mathbf{s};\mathbf{p}] + \mathbf{b}_p, \tag{17}$$

$$\begin{bmatrix} \boldsymbol{\mu}' \\ \log\left(\boldsymbol{\sigma}'^2\right) \end{bmatrix} = \mathrm{MLP}_q(x,r) = \mathbf{W}_q[\mathbf{c};\mathbf{s};\mathbf{p};\mathbf{r}] + \mathbf{b}_q, \tag{18}$$

where $\mathbf{W}_p \in \mathbb{R}^{2d_z \times 3d_h}$, $\mathbf{b}_p \in \mathbb{R}^{2d_z}$, $\mathbf{W}_q \in \mathbb{R}^{2d_z \times 4d_h}$, $\mathbf{b}_q \in \mathbb{R}^{2d_z}$, and $\mathbf{r}$ is the representation of response reference obtained in the similar way to Eq. (2) and Eq. (3). Then we use the reparameterization trick [13] to sample latent variable $\mathbf{z}$. During training, we sample latent variables from the recognition network and prior network to optimize the CVAE by Eq. (5). While during inference, there is no response reference, so we only sample latent variable from the prior network and pass it to the decoder for generation. For more mathematical details, please refer to [13].