Understanding and Improving the Exemplar-based Generation for Open-domain Conversation

Seungju Han[†], **Beomsu Kim**[†], **Seokjun Seo**[†], **Enkhbayar Erdenee**[†], **Buru Chang**[*]
Hyperconnect
{seungju.han,beomsu.kim,seokjun.seo,enkhbayar.erdenee,buru.chang}@hpcnt.com

Abstract

Exemplar-based generative models for open-domain conversation produce responses based on the exemplars provided by the retriever, taking advantage of generative models and retrieval models. However, due to the one-to-many problem of the open-domain conversation, they often ignore the retrieved exemplars while generating responses or produce responses over-fitted to the retrieved exemplars. To address these advantages, we introduce a training method selecting exemplars that are semantically relevant to the gold response but lexically distanced from the gold response. In the training phase, our training method first uses the gold response instead of dialogue context as a query to select exemplars that are semantically relevant to the gold response. And then, it eliminates the exemplars that lexically resemble the gold responses to alleviate the dependency of the generative models on that exemplars. The remaining exemplars could be irrelevant to the given context since they are searched depending on the gold response. Thus, our training method further utilizes the relevance scores between the given context and the exemplars to penalize the irrelevant exemplars. Extensive experiments demonstrate that our proposed training method alleviates the drawbacks of the existing exemplar-based generative models and significantly improves the performance in terms of appropriateness and informativeness.

1 Introduction

Exemplar-based generative models (Wu et al., 2019; Weston et al., 2018; Cai et al., 2019b; Gupta et al., 2021) for open-domain conversation combine a retrieval model (Humeau et al., 2019; Mazare et al., 2018; Kim et al., 2021) and a generative model (Adiwardana et al., 2020; Roller et al., 2021; Zhang et al., 2020; Brown et al., 2020) into a single framework to generate responses in two steps: (1) the retriever searches an exemplar using the given context as a query, and (2) the generator produces a response based on the given context and the retrieved exemplar. Exemplar-based generative models produce more specific responses than vanilla generative models while being more fluent than retrieval models.

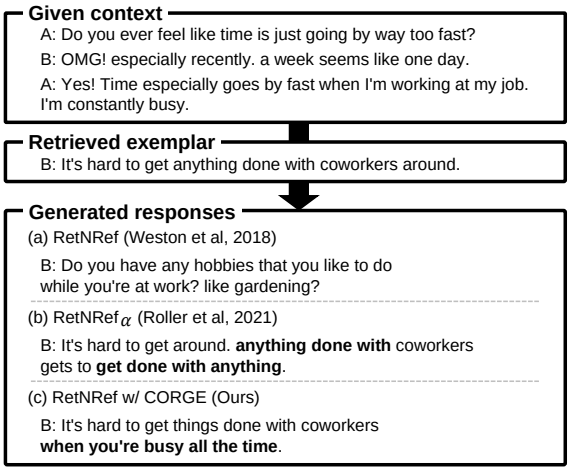

Figure 1: Responses generated by the three exemplar-based generative models. *RetNRef* ignores the exemplar during response generation, *RetNRef*$_\alpha$ generates the response highly over-fitted to the exemplar, and *RetNRef* trained with our training method (CORGE) well utilizes the exemplar to produce a more fluent response than that of the others.

Despite their success, exemplar-based generative models have two major shortcomings. Primitive exemplar-based generative models (Weston et al., 2018; Cai et al., 2019a) tend to ***entirely ignore the exemplars*** and produce responses similar to those of vanilla generative models. This is due to the *one-to-many problem* (Li et al., 2016) where there are many possible responses for each dialogue context. During the training phase, the retrieved exemplar is not helpful for generating the gold response when the exemplar retrieved for the given context is significantly different from the gold response.

[†] Equal contribution
[*] Corresponding author

This leads exemplar-based generative models to ignore the exemplar while generating responses, as shown in Figure 1(a). To address this issue, recent exemplar-based generative models utilize the gold response (Roller et al., 2021) or the slightly perturbed gold response (Cai et al., 2019b) as an exemplar in the training phase. However, these training methods cause the generator to ***rely heavily on the retrieved exemplar***, i.e. the generator resorts to copying the provided tokens, as shown in Figure 1(b). These two disadvantages of existing exemplar-based generative models can adversely affect the quality of the generated response.

Therefore, we introduce ***CORGE*** (COnnecting Retriever and GEnerator), a simple training method of exemplar-based generative models considering the one-to-many problem of the open-domain conversation. As inspired by Wu et al. (2019), CORGE first utilizes the gold response instead of dialogue context as the query for the retriever to select exemplars that are similar to the gold response. The retrieved exemplars ensure that exemplar-based generative models utilize their semantics while generating the gold response at the training phase. Since the exemplars are retrieved by the gold response, some of them are lexically identical or too similar to the gold response. These exemplars lead exemplar-based generative models to be trained to depend on the exemplar heavily. Thus, CORGE then eliminates the exemplars based on the distance between the exemplars and the gold response to alleviate the dependency of the generative models on the exemplars. Here, we employ Jaccard similarity to measure the distance (Guu et al., 2018; Cai et al., 2019a; Wu et al., 2019). However, as the selected exemplars solely depend on the gold response, some of them may be irrelevant to the given context, which results in exemplar-based generative models still ignoring the retrieved exemplar. To solve this, CORGE utilizes the relevance scores between the context and the exemplar to weight the relevant exemplars and penalizes irrelevant exemplars to the given context. Extensive experiments show that CORGE is generally applicable to the existing exemplar-based generative models and improves the quality of generated responses regarding appropriateness and informativeness.

Our main contributions: (1) We analyze the shortcomings of existing exemplar-based generative models derived from the nature of the open-domain conversation, the one-to-many problem. (2) We introduce a training method (CORGE) to improve the quality of generated responses by selecting useful exemplars and weighting the exemplars by relevance scores assessed by the retriever. (3) Through the human evaluation, we demonstrate that CORGE significantly improves the performance of exemplar-based generative models in terms of appropriateness and informativeness.

2 Related Work

2.1 Exemplar-based Generation

While generative models have shown remarkable performance on the open-domain conversation, it is well-known that generative models tend to yield uninformative and bland responses (Li et al., 2016; Liu et al., 2016; Serban et al., 2017; Li et al., 2020; Holtzman et al., 2019; Welleck et al., 2019). Exemplar-based generative models are introduced to overcome the aforementioned problem generative models suffer. Wu et al. (2019) introduce an exemplar-based generative model for open-domain conversation, which retrieves a context-exemplar pair conditioned by the input context and encodes the lexical difference between the input context and the retrieved context to the edit vector. The response is produced by feeding the exemplar and the edit vector to the generator. Weston et al. (2018); Roller et al. (2021) also retrieve the exemplar using the given context as a query and concatenate the exemplar with the context, then feed the concatenated exemplar into the generator to produce the final response for the open-domain conversation. Cai et al. (2019a,b) propose a method that removes the irrelevant information from the exemplar, then uses the masked exemplar to inform the generator to produce the response. Gupta et al. (2021) condition the generator with the retrieved exemplars and the extracted semantic frames of the exemplars, which improves the coherence of generated responses. We do not consider this model as a baseline because their model requires an additional semantic frame extractor, and it can be mutually complemented with our proposed training method.

2.2 Knowledge-grounded Generation

Knowledge-grounded generation models that utilize retrieved results (e.g., relevant documents from Wikipedia) to generate informative responses have been proposed to perform knowledge-intensive NLP tasks (e.g., open-domain question answering). The knowledge-grounded generation has a

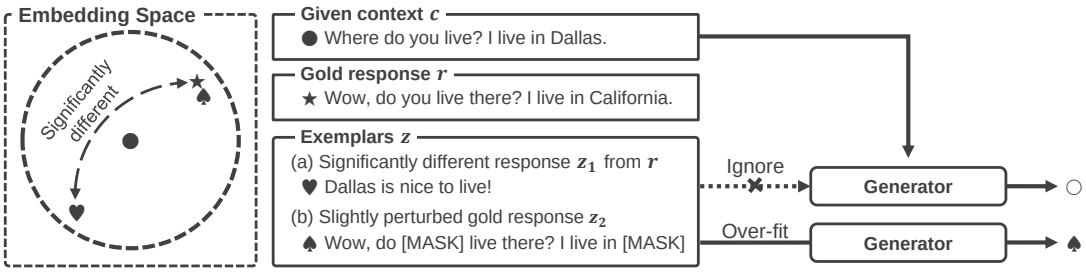

Figure 2: Illustration of the drawbacks of existing exemplar-based generative models. The black dotted line indicates the boundary of the relevant exemplars to the given context.

similar form with the exemplar-based generation. However, the main difference is that knowledge-grounded generative models extract the knowledge from external resources to generate the informative response. Guu et al. (2020) show the effectiveness of pre-training a knowledge retriever with the large-scale language model for open-domain question answering, and Lewis et al. (2020) demonstrate that knowledge-grounded generative models produce more informative and diverse sentences than vanilla generative models on a wide range of knowledge-intensive NLP tasks. Fan et al. (2021) similarly propose a knowledge-grounded generative model for response generation, but they do not focus on the open-domain conversation. In *Method Section*, we demonstrate the difference between our approach and knowledge-grounded generative models, and we show that existing knowledge-grounded generative models are not directly applicable to the open-domain conversation in *Experiments Section*.

3 Preliminaries

3.1 Exemplar-based Generation

Let $D = \{(c_i, r_i) \mid 1 \leq i \leq n\}$ denote the dialogue dataset, which consists of n pairs of context c and response r. Exemplar-based generative models are composed of two components: a retriever $\mathcal{R}$ and a generator $\mathcal{G}$. For a given context c_i, the retriever finds the top-scoring exemplar based on the relevance score $S_{\mathcal{R}}(z, c_i)$ of the exemplar $z \in R$, where R is a pre-defined response set. The generator computes the probability of the response for the context c_i while utilizing the exemplar z as $P_{\mathcal{G}}(r|c_i, z)$.

3.2 Drawbacks of Existing Exemplar-based Generative models

As mentioned in Roller et al. (2021), the primitive exemplar-based generative model (Weston et al., 2018) tends to ignore the retrieved exemplar during response generation due to the one-to-many problem in open-domain conversation (Li et al., 2016). Since its retriever searches an exemplar based on a given context, the retrieved exemplar is often significantly different from a gold response of the generator, although both of the retrieved exemplar and gold response are relevant to the given context, which is shown in Figure 2(a). As the retrieved exemplar is not helpful for generating the gold response, the generator is trained to ignore the retrieved exemplar and to produce a response using only the given context.

To induce the generator to utilize retrieved exemplars more actively, Roller et al. (2021) make use of the gold response, and Cai et al. (2019b) use perturbed gold response as an exemplar rather than using retrieved exemplars during the model training. However, since the exemplar z_i and the gold response r_i are too similar (as shown in Figure 2(b)), the exemplar-based generative model learns to rely overly on the exemplar. Eventually, the generator produces a highly over-fitted response to the exemplar by directly copying the tokens of the exemplar.

4 Method

We hypothesize that selecting semantically relevant but lexically distanced exemplars from the gold response could solve the drawbacks above. To validate this hypothesis, we introduce a training method of exemplar-based generative models, called CORGE. Our proposed training method is illustrated in Figure 3, and the illustrative examples about the exemplars selected by CORGE are described in Table 1.

4.1 Selecting Exemplars Semantically Relevant but Lexically Distanced to the Gold Response

We describe how CORGE selects semantically relevant but lexically distanced exemplars to the gold

response. Conventionally, the retriever selects the exemplars z based on the relevance score $S_{\mathcal{R}}(z, c_i)$ for the given context c_i. However, this searching process could return a significantly different exemplar z from the gold response r_i, and it induces the generator $\mathcal{G}$ to ignore the retrieved exemplar during response generation. Therefore, we select exemplars based on the gold response r_i to ensure that the generator $\mathcal{G}$ utilizes the exemplars inspired by Wu et al.. We select top-k scoring exemplars based on the score $S_{\mathcal{R}'}(z, r_i)$, which we call *k-Nearest Exemplars (kNE)*.[1] These kNE are more semantically related to the gold response r_i than the exemplar obtained by using $S_{\mathcal{R}}(z, c_i)$.

However, some of the selected kNE are lexically identical or too close to the gold response r unintentionally since the retriever searches the exemplars based on the gold response. We observe that using these exemplars also causes the over-fitting problem of generated responses; therefore, the generator excessively copies tokens from the exemplars. From this, we are motivated to filter out the exemplars which are lexically too close to the gold response and preserve the exemplars properly distanced to the gold response to mitigate the over-fitting problem. Here, we employ *Jaccard similarity* to measure the lexical similarity (Guu et al., 2018; Cai et al., 2019a; Wu et al., 2019) between the exemplar and the gold response. Exemplars are filtered out when their Jaccard distance with the gold response r is larger than 0.6, and we replace them with the randomly chosen responses from the pre-defined response set R. The threshold of filtering is empirically chosen as 0.6. The set of the final exemplars z obtained through these steps is referred to as $Z_i = \{z_{i,1}, z_{i,2}, \cdots, z_{i,k}\}$.

4.2 Weighting the Selected Exemplars based on the Relevance Score

As we select the exemplar totally based on the gold response, some of kNE could be relevant to the gold response r_i but irrelevant to the given context c_i. Therefore, we condition the generator with the relevance score of kNE to reward the relevant exemplars and penalize irrelevant exemplars. Using the retriever $\mathcal{R}$, we calculate the relevance score $S_{\mathcal{R}}(z_{i,j}, c_i)$ per each selected exemplar $z_{i,j}$, then apply the softmax function to the relevance score to

[1]Note that $S_{\mathcal{R}}(z, c)$ and $S_{\mathcal{R}'}(z, r_i)$ use the same retriever, but they are computed differently. Please refer to how we calculate the score $S_{\mathcal{R}'}(z, r_i)$ and $S_{\mathcal{R}}(z, c)$ in the Supplementary Materials.

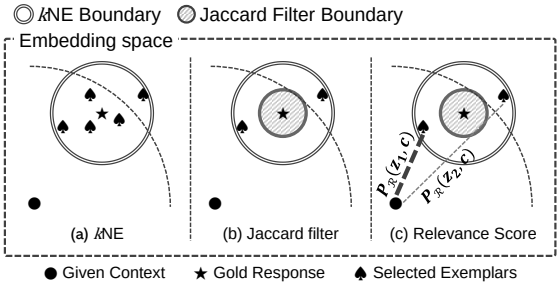

Figure 3: The procedure of our proposed training method, CORGE. (a): Selecting kNE of the gold response r based on $S_{\mathcal{R}'}(z, r)$. (b): Filtering out the exemplars which are too close to the gold response r. (c): Weighting the exemplars z depending on their normalized relevance scores $P_{\mathcal{R}}(z, c)$.

obtain the normalized relevance score $P_{\mathcal{R}}(z_{i,j}, c_i)$. Then we replace the traditional likelihood with the weighted likelihood using the normalized score. Our final training objective is to minimize the loss function $L = \sum_{i=1}^{n} L(r_i, c_i)$ where:

$$L(r_i, c_i) = -\log \sum_{z \in Z_i} P_{\mathcal{R}}(z, c_i) P_{\mathcal{G}}(r_i | c_i, z) \quad (1)$$

The gradient of the generator $\mathcal{G}$ is calculated as follows:

$$\nabla_{\mathcal{G}} L(r_i, c_i) = -\alpha \cdot \sum_{z \in Z_i} P_{\mathcal{R}}(z, c_i) \nabla_{\mathcal{G}} (P_{\mathcal{G}}(r_i | c_i, z)), \quad (2)$$

where $\alpha^{-1} = \sum_{z \in Z_i} P_{\mathcal{R}}(z, c_i) P_{\mathcal{G}}(r_i | c_i, z)$. This equation demonstrates that the gradient of the generator $\mathcal{G}$ is scaled by the normalized relevance score $P_{\mathcal{R}}(z, c_i)$, which indicates that the generator is less updated when the retrieved exemplar z is not relevant to the given context c_i. This procedure helps the model ignore the irrelevant exemplars. Thus, the generator learns to fetch tokens from the exemplar more easily, which is relevant to the gold response.

Difference between CORGE and Knowledge-grounded generative models The way of leveraging the relevance scores is already employed by knowledge-grounded generative models (Lewis et al., 2020; Sachan et al., 2021) in open-domain question answering. However, there is a significant difference between our CORGE and knowledge-grounded generative models. CORGE uses the relevance score $P_{\mathcal{R}}(z, c_i)$ to penalize the irrelevant exemplars z to the given context c_i since the exemplars are retrieved by $S_{\mathcal{R}'}(z, r_i)$. Knowledge-grounded generative models use it as the latent variable to jointly train the retriever $\mathcal{R}$ and generator $\mathcal{G}$. Especially, knowledge-grounded generative models also tend to ignore the retrieved exemplars due

Input Context		
What kind of animals you take care of?		
Gold Response		
I work with a variety of animals. I sometimes work with lions and monkeys.		
Context Retrieval	**Sim**	$P_{\mathcal{R}}(z, c)$
I raise two dogs.	0.1	0.9
***k*NE**	**Sim**	$P_{\mathcal{R}}(z, c)$
I work with a variety of animals.	**0.9**	0.2
He works with various people.	0.3	**0.0**
I work with lots of different animals.	0.5	0.3
I do some work with animals they're amazing creatures.	0.3	0.3

Table 1: Samples of the exemplars selected by CORGE. **Context Retrieval** indicates the exemplar retrieved by using the context as a query, and ***k*NE** shows the exemplars selected by using the gold response as a query. **Sim** measures the lexical similarity between the gold response and the exemplar and $P_{\mathcal{R}}(z, c)$ indicates the normalized relevance score calculated by retriever.

to the one-to-many nature in open-domain conversation when the retriever and generator are jointly trained. On the other hand, we do not perform the joint learning of the retriever and the generator, but freeze the retriever while training the generator.

5 Experiments

5.1 Dataset

We utilize the following four datasets used in Roller et al. (2021), which are Blended Skill Talk (BST) (Smith et al., 2020), ConvAI2 (Zhang et al., 2018), Empathetic Dialogues (ED) (Rashkin et al., 2019), and Wizard of Wikipedia (WoW) (Dinan et al., 2018). To simplify the notation, we denote the concatenated version of these four datasets as ***BST+***. We split BST+ into train, validation, and test sets following Smith et al. (2020).

5.2 Baselines

Retrieval and Generative Models *Bi-encoder 256M* (Mazare et al., 2018) and *Blender 90M* (Roller et al., 2021) are considered as a baseline retrieval model and a baseline generative model. Further, they are also employed as a retriever and a generator of the following exemplar-based generative baselines, respectively.

Exemplar-based Generative Models Since our proposed training method is for training exemplar-based generation models, we first consider recent exemplar-based generation models, *RetNRef* (Weston et al., 2018), *RetNRef*$_\alpha$ (Roller et al., 2021), and *MatToGen* (Cai et al., 2019b), as baselines. *RetNRef* concatenates the retrieved exemplar with the given context as the input of the generator to produce the response. *RetNRef*$_\alpha$ is the dialogue retrieval version of *RetNRef*, which adopts α-blending to escape from simply ignoring the retrieved exemplars ($\alpha = 0.5$). *MatToGen* extracts the meaningful tokens from the exemplar to provide them to the generator.

To verify the effectiveness of our training method, we apply CORGE to *RetNRef* and *MatToGen* instead of their training method. They are denoted as *RetNRef+CORGE* and *MatToGen+CORGE*, respectively.

Knowledge-grounded Generative Models Although *RAG* (Lewis et al., 2020) and *KIF* (Fan et al., 2021) are proposed to perform knowledge-grounded generation tasks, we employ *RAG* and *KIF* as baselines since they have a similar form with exemplar-based generative models. Our experiments demonstrate that these knowledge-grounded generative models cannot be directly applied to the open-domain conversation.

5.3 Evaluation Metrics

To verify the effectiveness of our training method CORGE, we conduct a pair-wise comparison through the human evaluation following Weston et al. (2018). We use two criteria: **Appropriateness** and **Informativeness**. Appropriateness measures how the generated response is fluent, logical, and appropriate to the given context. Informativeness measures how the generated response has meaningful information relevant to the given context. We use Amazon Mechanical Turk to collect the annotations, and more details are described in the Supplementary Material.

We also employ the automatic evaluation metrics, **Perplexity** (PPL), **Dist-**n, and **BLEU** (Papineni et al., 2002), to analyze the generated responses of each model. PPL measures how well the model predicts a response based on the given input context, and lower PPL indicates that the model predicts the response better. To analyze how much the exemplar-based generative model leverages the retrieved exemplar, we introduce two variants of PPL by utilizing conditional probability when exemplars are given: (1) PPL$_{gold}$ uses the

Model Names (A vs. B)	Appropriateness (%)				Informativeness (%)			
	Win Rate	A win	Tie	B win	Win Rate	A win	Tie	B win
RetNRef$_\alpha$ vs. Bi-encoder 256M	44.9	32.0	28.7	**39.3**	47.5	31.3	34.0	**34.7**
RetNRef$_\alpha$ vs. Blender 90M	50.2	**37.3**	25.7	37.0	53.3	**40.3**	24.3	35.4
RetNRef + CORGE vs. Bi-encoder 256M	52.6	**34.0**	35.3	30.7	51.9	**35.7**	31.3	33.0
RetNRef + CORGE vs. Blender 90M	57.7*	**33.7***	41.7*	24.6*	54.6	**30.0**	45.0	25.0
RetNRef + CORGE vs. RetNRef$_\alpha$	53.2	**30.3**	43.0	26.7	51.6	**27.7**	46.3	26.0
RetNRef + CORGE vs. RetNRef	54.4	**41.0**	24.7	34.3	53.4	**37.0**	30.7	32.3
RetNRef + CORGE vs. KIF	57.5*	**37.0***	35.7*	27.3*	50.0	30.0	40.0	30.0
RetNRef + CORGE vs. RAG	53.5	**37.7**	29.7	32.6	52.1	**29.7**	43.0	27.3
MatToGen vs. Bi-encoder 256M	47.1	33.3	29.3	**37.4**	50.9	**36.7**	28.0	35.3
MatToGen vs. Blender 90M	48.1	34.0	29.3	**36.7**	46.3	31.6	31.7	**36.7**
MatToGen + CORGE vs. Bi-encoder 256M	54.2	**43.0**	20.7	36.3	54.4	**41.3**	24.0	34.7
MatToGen + CORGE vs. Blender 90M	58.0*	**35.0***	39.7*	25.3*	58.1*	**36.0***	38.0*	26.0*
MatToGen + CORGE vs. MatToGen	52.6	**33.3**	36.7	30.0	53.3	**32.7**	38.7	28.6
MatToGen + CORGE vs. KIF	57.1*	**44.0***	23.0*	33.0*	52.5	**39.0**	25.7	35.3
MatToGen + CORGE vs. RAG	51.6	**38.3**	25.7	36.0	55.6	**41.3**	25.7	33.0

Table 2: Pair-wise human evaluation results show that our proposed training method improves the performance against the existing exemplar-based generation approaches in terms of appropriateness and informativeness. The win rate is calculated by excluding the tie. * indicates statistical significance (two-tailed binomial test, p < 0.05).

conditional probability $P_{\mathcal{G}}(r|c, r)$, which assumes the situation when the gold response is given as an exemplar, and (2) PPL$_{ret}$ uses the conditional probability $P_{\mathcal{G}}(r|c, z)$ where z is the retrieved exemplar by using $S_{\mathcal{R}'}(z, r)$. Lower PPL$_{gold}$ denotes that the exemplar-based generative model predicts the gold response well when the gold response is given as an exemplar. Lower PPL$_{ret}$ indicates that the exemplar-based generative model well leverages the provided exemplar to predict the gold response. Dist-n (Li et al., 2016) is the ratio of distinct n-grams to a total number of n-grams for all the generated responses, which measures the degree of the diversity of the generated responses. BLEU$_{(z,r)}$ is adopted to measure the degree of the token overlap between the provided exemplar and the generated response pair (z, r). A higher BLEU$_{(z,r)}$ score indicates that the generator copies more from the provided exemplar while generating the response.

5.4 Implementation Details

We provide the details of our implementation in the Supplementary Material. We will the source codes of CORGE for the reproducibility of the conducted experiments.

6 Experimental Results

6.1 Pair-wise Comparison Results

Table 2 shows the pair-wise comparison results through the human evaluation. When *RetNRef* and *MatToGen* adopt our proposed CORGE as their training method, they outperform all baselines except for a case of *RetNRef+CORGE* vs. *KIF* on the informativeness. In detail, *RetNRef+CORGE* and *MatToGen+CORGE* show better performance than *RetNRef*$_\alpha$ and *MatToGen*, respectively, in both metrics. Especially, *MatToGen+CORGE* outperforms *Bi-encoder 256M* and exceeds *Blender 90M*, while *MatToGen* performs worse than *Bi-encoder 256M* and *Blender 90M*. Furthermore, CORGE enlarges the win rate of *RetNRef*$_\alpha$ for *Blender 90M*. These evaluation results demonstrate that CORGE leads the existing exemplar-based generative models to produce more fluent and informative responses.

6.2 Investigating the Exemplar-based Generative Models with Automatic Metrics

Through the automatic evaluation, we verify that existing exemplar-based generative models ignore the provided exemplar or generate responses overfitted to the provided exemplar. As shown in Table 3, *RetNRef+CORGE* and *MatToGen+CORGE* show lower PPL$_{ret}$ than *Blender 90M*, which means that the exemplar-based generative models trained with CORGE make a better prediction of the gold response than *Blender 90M* by utilizing the provided exemplar. *RetNRef+CORGE* has a smaller degree of PPL$_{gold}$ and PPL$_{ret}$ than those of *RetNRef*, which infers *RetNRef+CORGE* leverages the provided exemplar better than *RetNRef*. *RetNRef*$_\alpha$ has lower PPL$_{gold}$ than *RetNRef+CORGE*, however, *RetNRef*$_\alpha$ has higher

Models	PPL_{gold}	PPL_{ret}	Dist-2	Dist-3	$\text{BLEU}_{(z,r)}$-2	$\text{BLEU}_{(z,r)}$-3
Blender 90M	13.79	13.79	0.236	0.372	-	-
Bi-encoder 256M	-	-	0.681	0.881	-	-
RetNRef	8.518	13.37	0.256	0.386	0.030	0.009
RetNRef_α	3.061	16.99	0.530	0.778	0.319	0.201
RetNRef + CORGE	4.863	11.53	0.349	0.520	0.102	0.048
MatToGen	5.291	17.71	0.362	0.567	0.169	0.095
MatToGen + CORGE	5.651	13.45	0.313	0.474	0.069	0.028
RAG	11.84	14.91	0.257	0.390	0.015	0.003
KIF	12.11	15.18	0.238	0.363	0.002	0.000

Table 3: Automatic evaluation results. Since *Blender 90M* can not utilize the exemplar, we report PPL calculated from $P_{\mathcal{G}}(r|c)$ in the place of PPL_{gold} and PPL_{ret}.

PPL_{ret} than *RetNRef+CORGE*. This result demonstrates that $\textit{RetNRef}_\alpha$ does not make good use of the retrieved exemplar except when the gold response is given as the retrieved exemplar. From this observation, we claim that $\textit{RetNRef}_\alpha$ generates a response highly over-fitted to the selected exemplar, which is caused by utilizing the gold response as an exemplar in the training phase. The same goes for *MatToGen*, where applying CORGE mitigates the over-fitting issue.

Higher Dist-n of *RetNRef+CORGE* and *MatToGen+CORGE* compared to *Blender 90M* shows that our exemplar-based generative models produce more diverse responses than the vanilla generative model. Moreover, *RetNRef+CORGE* has higher Dist-n than *RetNRef*, which shows that utilizing the exemplars helps the generator diversify the responses. Although $\textit{RetNRef}_\alpha$ is the only one that achieves comparable Dist-n to that of the vanilla retrieval model, *Bi-encoder 256M*, it is derived from an over-fitting to the exemplar considering the gap between PPL_{gold} and PPL_{ret}, resulting in the degradation of appropriateness and informativeness in human evaluation.

Average $\text{BLEU}_{(z,r)}$ scores implicitly measure the overlap between the retrieved exemplar and the generated response; thus, a higher degree of $\text{BLEU}_{(z,r)}$ indicates that the generator depends more on the retrieved exemplar. *RetNRef* shows a negligible $\text{BLEU}_{(z,r)}$ score, which reaffirms that the model is almost not utilizing the retrieved exemplar. $\textit{RetNRef}_\alpha$ and *MatToGen* have higher $\text{BLEU}_{(z,r)}$ scores compared to *RetNRef+CORGE* and *MatToGen+CORGE*, respectively, which verifies that the former depends more on the retrieved exemplar than the latter.

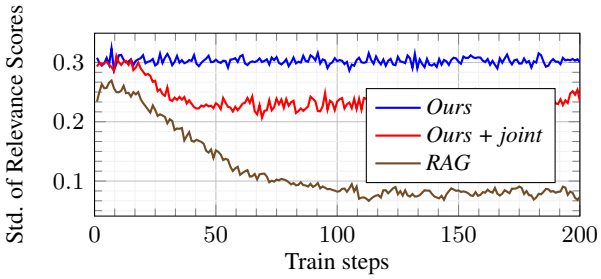

Figure 4: The standard deviation of the normalized retriever score gets smaller when we jointly train the retriever for exemplar-based generative models. *Ours* stands for *RetNRef+CORGE*, and *joint* indicates jointly training the retriever with the generator.

6.3 Incapability of Knowledge-grounded Generative Models in Open-domain Conversation

The automatic evaluation results in Table 3 confirm that knowledge-grounded generative models are ignoring the exemplar. PPL_{gold}, PPL_{ret}, and Dist-n of *RAG* and *KIF* have a similar degree to those of *Blender 90M*, which implies that the exemplars are not providing useful information while generating the response. The average $\text{BLEU}_{(z,r)}$ score also has a poor degree, indicating almost no overlap between the retrieved exemplars and the generated responses. We explain that these results are originated from the difference between the open-domain conversation and knowledge-grounded generation tasks. While training knowledge-grounded generative models, they use $P_{\mathcal{R}}(z,c)$ to fetch the external knowledge. However, the generator also ignores the retrieved exemplar due to the one-to-many nature of the open-domain conversation.

In addition, we observe that jointly training the retriever with the generator causes the retriever stuck in the local minima. As shown in Figure 4, the standard deviation of normalized relevance scores $P_{\mathcal{R}}(z,c)$ computed by the retriever

Input Context	A: No way i do it a lot, at least when i am feeling well enough. B: Awesome. That's the only thing i'll never do. I do like the outdoors though. A: I'm a thrill seaker. Do you like extreme sports?
Bi-encoder 256M Blender 90M	I love extreme sports. Its the only real way I can get the adrenaline boost I need. No, I don 't think I' ve ever done that. I' d love to though.
Exemplar RetNRef	I love extreme sports. Its the only real way I can get the adrenaline boost I need. I don' t really have a lot of time for sports. I spend most of my free time working.
Exemplar RetNRef$_\alpha$	I love extreme sports. Its the only real way I can get the adrenaline boost I need. I love extreme sports I need the adrenaline boost. Its the only way I can get the real adrenaline.
Exemplar RetNRef + CORGE	I love extreme sports. Its the only real way I can get the adrenaline boost I need. I love extreme sports as well. It' s a great way to get out and enjoy life.
Exemplar KIF	nicde. I don' t really have a lot of time for sports. I spend most of my time working.
Exemplar RAG	Do you like sports? I don' t really have a lot of time for sports. I spend most of my time working.

Table 4: Examples for model responses and provided exemplars corresponding to given input contexts.

Models	**PPL$_{\text{gold}}$**	**PPL$_{\text{ret}}$**	**Dist-2**	**Dist-3**	**BLEU$_{(z,r)}$-2**	**BLEU$_{(z,r)}$-3**
RetNRef + CORGE	4.863	11.53	0.349	0.520	0.102	0.048
RetNRef + CORGE − RS	6.482	11.75	0.316	0.478	0.074	0.031
RetNRef + CORGE − kNE	8.657	13.82	0.250	0.380	0.034	0.010
RetNRef + CORGE − JF	1.698	32.91	0.537	0.785	0.332	0.207

Table 5: Results of the ablation study. −RS, −kNE, and −JF denote that relevance score (RS), kNE, and Jaccard filter (JF) are removed from CORGE, respectively.

almost gets near zero when the retriever of RAG is jointly trained. A smaller standard deviation means the relevance scores are getting flattened. Although knowledge-grounded generative models empirically have shown that jointly training the retriever and generator improves the performance in knowledge-intensive NLP tasks (Lewis et al., 2020), in open-domain conversation, the retrieved exemplars are ignored. Thus, the retriever learns to produce an uninformative relevance score. As a result, the retriever collapses, which means the retriever may return inappropriate exemplars to the generator (also shown in the example of KIF and RAG in Table 4). Intriguingly, jointly training the retriever with CORGE also causes the retriever scores to be flattened, as shown in Figure 4, and we empirically observe the minor collapse of the retriever as we experienced in RAG as well. Thus, CORGE does not jointly train the retriever.

6.4 Ablation Study

To verify the effectiveness of each component in CORGE, we conduct the ablation study. In Table 5, PPL$_{ret}$ from *RetNRef+CORGE* is lower than any other ablation counterparts, which confirms each component contributes to predicting the responses. *RetNRef+CORGE*−RS and *RetNRef+CORGE*−kNE have a higher degree of PPL$_{ret}$ and PPL$_{gold}$, which indicates *RS* and *kNE* help the generator to utilize the exemplar while generating the response. *RetNRef+CORGE*−JF provides a strong signal of over-fitting, where it has extremely low PPL$_{gold}$ but exceptionally high PPL$_{ret}$. Dist-n shows our model produces the most diverse responses among the models except *RetNRef+CORGE*−JF, where *RetNRef+CORGE*−JF excessively copies the tokens from the retrieved exemplar. The average BLEU$_{(z,r)}$ scores also show the same trend, where reaffirms the effect of the components of CORGE.

7 Conclusion

In this paper, we introduce a generally applicable training method for exemplar-based generative models to alleviate their disadvantages derived from the one-to-many problem. Our training method selects exemplars that are semantically relevant but lexically distanced from the gold response and weights those exemplars with the relevance score measured by the retriever. Through the extensive analysis, including pair-wise human evaluation, we verify that our method improves the performance of existing exemplar-based generative models in terms of appropriateness and informativeness.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019a. Skeleton-to-response: Dialogue generation guided by retrieval memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1219–1228.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019b. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1866–1875.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting transformers with knn-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Prakhar Gupta, Jeffrey P Bigham, Yulia Tsvetkov, and Amy Pavel. 2021. Controlling dialogue generation with semantic exemplars. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3018–3029.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Beomsu Kim, Seokjun Seo, Seungju Han, Enkhbayar Erdenee, and Buru Chang. 2021. Distilling the knowledge of large-scale generative models into retrieval models for efficient open-domain conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3357–3373.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *arXiv preprint arXiv:1812.07617*.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for*

Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics.

Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779.

A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. 2017. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

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.

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, et al. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*.

Iulian Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92.

Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

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 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

A Implementation Details

A.1 How the Retriever Calculates the Scores

Our retriever follows the architecture of Bi-encoder (Mazare et al., 2018), and the score $S_{\mathcal{R}}(z, c)$ and $S_{\mathcal{R}'}(z, r)$ are calculated as follows:

$$\begin{aligned} S_{\mathcal{R}}(z, c) &= d(z) \cdot q(c), \\ S_{\mathcal{R}'}(z, r) &= d(z) \cdot d(r), \\ d(z) &= \mathbf{BERT}_r(z), \\ d(r) &= \mathbf{BERT}_r(r), \\ q(c) &= \mathbf{BERT}_c(c), \end{aligned} \tag{3}$$

where $d(z)$ and $d(r)$ are encoded vectors produced by response encoder $\mathbf{BERT}_r$ and $q(c)$ is an encoded vector produced by context encoder $\mathbf{BERT}_c$. The notation $\mathcal{R}'$ indicates that it only uses the response encoder instead of using the context encoder together. CORGE is not limited to use Bi-encoder as a retriever and can be applied to other types of a retriever (e.g. Poly-encoder (Humeau et al., 2019)).

A.2 Model Details

As we mentioned in Section 5.2, we employ Bi-encoder 256M and Blender 90M as a retriever and a generator of each exemplar-based generative model, respectively. For MatToGen, additional MLP layers are added to the retriever, as follows the details in Cai et al. (2019b). When training the models, weights of the retriever and the generator are initialized with the pre-trained Bi-encoder 256M and Blender 90M, respectively, For Blender 90M, we use the model released by ParlAI (Miller et al., 2017), which is fine-tuned on the BST+ dataset. For Bi-encoder 256M, we fine-tune the model released by ParlAI on the BST+ dataset, and we follow the hyperparameter settings of Humeau et al. (2019), which are implemented in the ParlAI library. The pre-defined response set is constructed from the BST+ training set, which contains about 400K responses. We use NVIDIA DGX Station A100 for training the models.

A.3 Hyperparameters

When training exemplar-based generative models with CORGE, five (k=5) exemplars are utilized for each training instance. The exemplar-based generators are trained with a batch size of 32 and an initial learning rate of 7e-6, and the learning rate is decayed in half when the training loss meets the plateau. The model is trained until there is no progress in the validation PPL.

A.4 Generation Strategy

When we generate samples using generative model, exemplar-based generative models, and knowledge-grounded generative models, we adopt a beam decoding strategy which is widely used in generative models (Graves, 2012). Following (Roller et al., 2021), we choose a minimum beam length and a beam size as 20 BPE tokens and 10, respectively, and use tri-gram beam blocking on context and response blocks. During the inference phase, both exemplar-based generative models and knowledge-grounded generative models use the top-1 scoring candidate as an exemplar chosen from utilizing the relevance score $S_{\mathcal{R}}(z, c)$.

B Evaluation Details

We prepare dialogue cases that have three-turn input contexts and the gold response from the BST and evaluate them by human pair-wise comparison and automatic evaluation. There are 980 test cases, and we randomly choose 100 test cases for the human evaluation.

B.1 Pair-wise Human Evaluation

As we described in Section 5.3, we use Amazon Mechanical Turk to collect the annotations. Each test case is rated by three annotators to improve the robustness of the evaluation result. We set a maximum number of annotations per worker in order to reduce the potential bias. To control the quality of the annotations, we only allowed annotators who satisfy the following requirements to evaluate our results: (1) HITs approval rate greater than 95%, (2) Location is one of Australia, Canada, New Zealand, United Kingdom, and the United States, (3) Lifetime number of HITs approved greater than 1000, following Li et al. (2018). Figure 5 shows the instructions and the interface for the human evaluation. To mitigate the bias from the annotator, we randomly shuffle the order of the model and the corresponding response.

B.2 Automatic Evaluation

For automatic metrics, we calculate the metric for each case and take the average of those values. When calculating BLEU, we use `sentence_bleu` function in `nltk` python package (Loper and Bird, 2002).

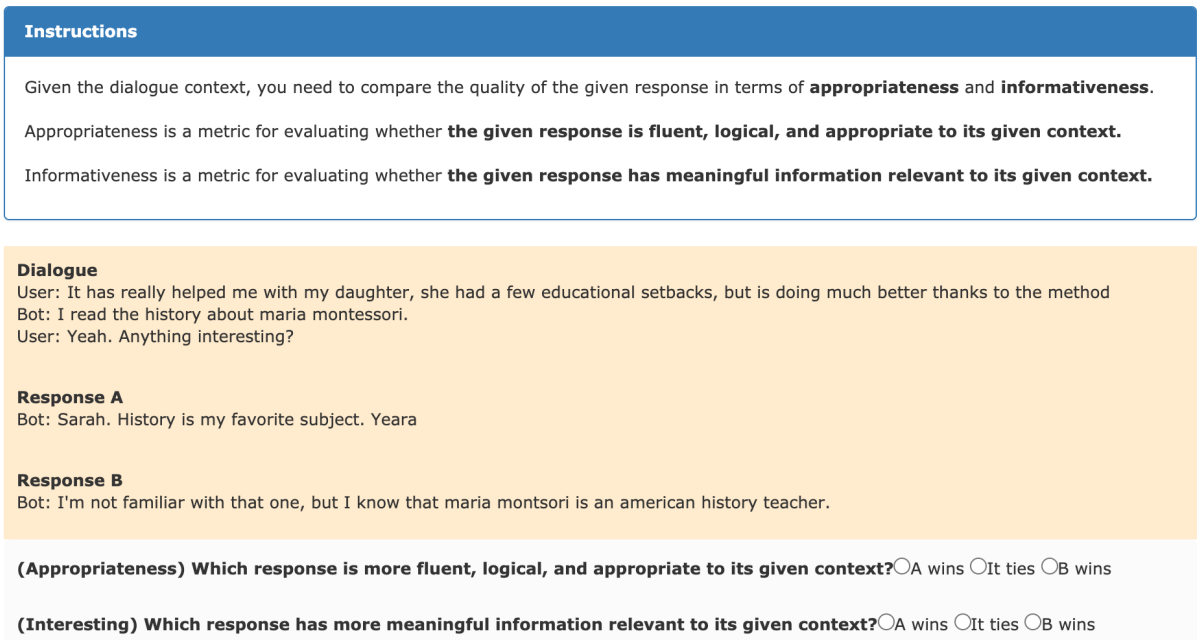

Figure 5: The interface of pairwise human evaluation for appropriateness and informativeness.

C Measuring Inference Time

We measure how much time spend when the model generates the responses. When generating the response, *Blender 90M* takes 0.481 seconds, and *RetNRef+CORGE* takes 0.523 seconds per instance. There is only an 8.7% amount of inference time gap between *Blender 90M* and *RetNRef+CORGE*. This tells us that exemplar-based generation can significantly improve the quality of responses regarding appropriateness, informativeness, and diversity without increasing the amount of time to generate answers. We test our model on NVIDIA DGX Station A100 with PyTorch 1.7.1, CUDA 11.0, CuDNN 8.0, and here we adopt the generation strategy we describe above. When we measure the inference time, we only use a single GPU (NVIDIA A100 GPU, 40GB Memory), and the inference time is measured as the average inference time of 100 response generations.

D Additional Results

We provide additional samples for the retrieved exemplar and the model response from the baselines and our models in Table 6.

Input Context	A: I crash around 230 to 300. I don't have much of a social life. B: Neither do I, you are not alone A: I am alone but I'm not lonely. I have a dog who's great company.
Bi-encoder 256M	Any friends? My dog is one of mine.
Blender 90M	That 's good to hear, I 've a dog too, he' s my best friend.
Exemplar	Any friends? My dog is one of mine.
RetNRef	I am sorry to hear that. Do you have any hobbies that you like to do?
RetNRef$_\alpha$	My dog is one of my friends? Is yours one of any of his friends? lol.
RetNRef + CORGE	What kind of dog is it? I' ve got a chihuahua and she' s my best friend.
Exemplar	I love dogs too. My best friend is my dog.
MatToGen	I love my best friend, he' s the best dog I've ever had. What is your best friend?
MatToGen + CORGE	I love dogs too, my best friend is my dog. Do you have any hobbies?
Exemplar	nicde.
KIF	I am sorry to hear that. Do you have any hobbies that you like to do?
Exemplar	I'm usually alone because I don't like my family.
RAG	I am sorry to hear that. Do you have any hobbies that you like to do?
Input Context	A: Sounds like a horror movie, my favorite kind B: Haha, yeah he melts my heart. A: Did you do anything fun for Halloween?
Bi-encoder 256M	I enjoy Halloween, it's fun to dress up. Do you like it?
Blender 90M	I went to a halloween party with my friends. It was a lot of fun. What about you?
Exemplar	I enjoy Halloween, it's fun to dress up. Do you like it?
RetNRef	No, I don 't think I could do it. I' m not a big fan of halloween.
RetNRef$_\alpha$	I like it, it' s fun to dress up. Do you like halloween? I enjoy dressing up.
RetNRef + CORGE	I like to dress up as a clown, but I don' t think I' ll be doing that this year.
Exemplar	No we went to a haunted house and I got chased often.
MatToGen	No we never got to go to the haunted house we went to once and got chased by a ghost.
MatToGen + CORGE	No, I don 't like to go to haunted houses. I' m scared of them.
Exemplar	nicde.
KIF	Yes, I went to a haunted house and watched all the horror movies. It was so scary!
Exemplar	Are you doing anything cool for halloween
RAG	No, I don' t think I' m going to. I' ve never been a fan of halloween.
Input Context	A: Oh God no! I can't stay tied down to one woman. Why would you want to do that? B: I know right? Most people consider marriage to be involving 2 people but in certain parts of the world that varies between cultures and religions, so it doesn't seem so bad A: Marriage is very good.
Bi-encoder 256M	I agree, but in some cultures, marriage is recommended and unmarried folks are frowned upon!
Blender 90M	I agree! I' m not sure if i could do it though. I don' t think I could.
Exemplar	I agree, but in some cultures, marriage is recommended and unmarried folks are frowned upon!
RetNRef	I agree. I think it' s important for people to have a healthy relationship with each other.
RetNRef$_\alpha$	I agree, but some folks are recommended to marry in some cultures, and some are frown upon!
RetNRef + CORGE	I agree! In some cultures marriage is recommended, but in some countries it' s not recommended.
Exemplar	I agree. I think it' s important for people to have a healthy relationship with each other.
MatToGen	I agree, some cultures are more strict than others, but i agree that some religions are more lenient.
MatToGen + CORGE	I agree! In some cultures marriage is recommended by some religions, but I don' t believe in it!
Exemplar	nicde.
KIF	I agree! I think it' s important to remember that marriage is a legal, social, and financial union.
Exemplar	That is good and great
RAG	I agree! I think it' s important for people to have a healthy relationship with each other.

Table 6: Additional examples for model responses and provided exemplars corresponding to given input contexts.