Building a Role Specified Open-Domain Dialogue System Leveraging Large-Scale Language Models

Anonymous ACL submission

Abstract
Recent open-domain dialogue models have brought numerous breakthroughs. However, building a chat system is not scalable since it often requires a considerable volume of human-human dialogue data, especially when enforcing features such as persona, style, or safety. In this work, we study the challenge of imposing roles on open-domain dialogue systems, with the goal of making the systems maintain consistent roles while conversing naturally with humans. To accomplish this, the system must satisfy a role specification that includes certain conditions on the stated features as well as a system policy on whether or not certain types of utterances are allowed. For this, we propose an efficient data collection framework leveraging in-context few-shot learning of large-scale language models for building role-satisfying dialogue dataset from scratch. We then compare various architectures for open-domain dialogue systems in terms of meeting role specifications while maintaining conversational abilities. Automatic and human evaluations show that our models return few out-of-bounds utterances, keeping competitive performance on general metrics. We release a Korean dialogue dataset we built for further research.

1 Introduction
Recent large-scale language models (LMs) have brought numerous breakthroughs in open-domain dialogue systems, yielding human-like responses (Zhang et al., 2020; Adiwardana et al., 2020; Brown et al., 2020; Roller et al., 2021; Kim et al., 2021a). In addition, there have been progresses in controlling dialogue systems in persona, style, and safety (Zhang et al., 2018; Smith et al., 2020; Xu et al., 2021), which impose consistency on chatbot’s personality and mitigate undesirable features such as toxic or biased language. However, building a chatbot system combining these capabilities is still challenging, which requires numerous human-human dialogues for those conversational skills.

Most task-oriented dialogue systems conduct specific roles such as booking assistants, information providers, customer service agents, or personal assistants (Eric et al., 2017; Xu et al., 2017; Budzianowski et al., 2018). On the other hand, studies on open-domain dialogue systems that perform specific roles have been insufficiently investigated, even though the role can be defined for the practical chatbot systems (e.g., chatbots that care for senior citizens living alone, or counseling chatbots). In these cases, the chatbot systems do not have an explicit goal or task other than to engage in conversations proactively, but may have system policies on whether or not certain types of utterances are allowed.

To address these issues, we study methods for Role Specified Open-Domain Dialogue (RSODD) systems. The goal of the system is conversing naturally with humans on open-ended topics while keeping conditions of given role. Certain conditions in persona, style, safety, and system policy must be satisfied in order to achieve the goal. We consider a general and scalable framework to treat them, instead of using individual approaches to control each.

In particular, we present a Human-AI collabora-

1The dataset is available at www.dummyurl.data
tive data construction method to build a scalable supervisory dataset from scratch for role-satisfying open-domain dialogues (Figure 2). We propose to leverage large-scale LMs for generating entire dialogue sessions between user and system by in-context few-shot learning manner (Brown et al., 2020; Kim et al., 2021a). For this, we input prompts describing an outline of the role and an example dialogue (example in Figure 3). And then generated dialogue sessions are filtered by humans to form a dataset. We find that the generated dialogues follow the majority of the contents stated in the prompt (Section 5.2), which makes the proposed process feasible. As a result, the cost of building dataset is significantly reduced when compared to manually producing gold dialogues (Section 3.2). Furthermore, we employ a human-in-the-loop configuration to add human-bot dialogues in the dataset and evaluate the system at the same time, which we find brings additional gains to the system’s performance.

Next, we compare several architectures for modeling role-satisfying chatbot systems in a synthetic dataset. For response selection models, we employ components for predicting unanswerable contexts caused by constrained utterance candidates. For response generation, we use unlikelihood training (Welleck et al., 2019; Li et al., 2020) to suppress the generation of negative examples. We also consider a pipelined model consists of response selection and generation models. In extensive experiments and ablation studies, we show that the proposed models considerably reduce negative examples that violate the role specification compared to the in-context learning baseline, while achieving competitive SSA (Adiwardana et al., 2020) scores for their responses. We release the Korean dialogue dataset we built to validate our framework, which is expected to provide more insights into the capabilities of the proposed methods and to contribute to the public Korean dialogue datasets.

The contribution of our work is summarized as follows.

1. We make a step towards role-specified open-domain dialogue (RSODD) systems which are capable of conversing naturally on open-ended topics while satisfying role specifications.

2. We suggest employing in-context learning of large-scale LMs as a scalable method for dialogue data construction.

3. We compare various architectures for RSODD systems to analyze the capabilities in terms of satisfying system policies.

4. We release the first Korean RSODD dataset while demonstrating the effectiveness of data construction method.

2 Related Work

Pretrained LM in Open-domain dialogue
Many prior works tried to pretrain the models on large-scale social comment chains data like Reddit to model conversational behavior (Zhang et al., 2020; Adiwardana et al., 2020), followed by fine-tuning on the diverse target dialogue dataset to improve engagingness and humanness (Roller et al., 2021). To avoid undesired behaviors of the models including toxicity and bias from the human-human conversation, they merely exclude some parts of training data using automatic filtering by predefined criteria.

Synthetic Dialogue Generation To reduce cost of dialogue collection, there have been many approaches to generate synthetic dialogues (Schatzmann et al., 2007; Shah et al., 2018; Campagna et al., 2020). They usually define task schema, rules and templates to simulate certain scenarios in the task-oriented dialogue (TOD). Kim et al. (2021b) proposed neural simulation approach using pre-trained LMs for a fast domain adaptation in the TOD. However, they need training data of source domain to transfer unseen target domain.

Xu et al. (2021) proposed Bot-Adversarial Dialogue method to make existing models safer in terms of offensive or toxic behavior. Sun et al. (2021) extends existing task-oriented dialogue (TOD) dataset to open-domain chit-chat using the pretrained LMs. Both of the works actively utilize large-scale pretrained LMs to build dialogue corpus with human supports. We also introduce human-AI collaborative dialogue collection method, while especially utilizes few-shot in-context learning ability of large-scale LM (Brown et al., 2020; Kim et al., 2021a).

On the Role in Dialogue In task-oriented dialogue system, the system side plays functional roles utilizing explicit knowledge base of specific domain (Williams et al., 2013; Henderson et al., 2014a,b; Eric et al., 2017; Xu et al., 2017; Budzianowski et al., 2018). For example, agent
3 Data Construction

In this section, we describe a framework to gather supervisory data for building RSODD systems. The input to the framework is a role specification described by the chatbot developer (Table 13 in Appendix for example), which defines the conditions in the dialogue interactions for the system. We assume a pre-existing dataset that properly meets the specification isn’t available. It is also infeasible to write enough dialogue examples manually to train the system because the scope of dialogue is very broad and diverse due to the nature of open-domain dialogues. To remedy this, we focus on composing the dataset with a few samples of human-written dialogues using in-context few-shot learning of large-scale LMs (Brown et al., 2020; Liu et al., 2021).

3.1 Dialogue Generation

As reported in Kim et al. (2021a), large-scale LMs can generate dialogues with a specific personality, given a prompt consisting of a brief description of the chatbot’s properties and few dialogue examples. We use this method to build the entire dataset. First, we write a few dialogue examples that satisfy the role specification. And we attach each of them at the end of the system description to compose input prompts for one-shot in-context learning. Figure 3 (a) shows an example input. Then, the LM generates whole dialogue sessions. That is, the LM acts as both a system and a user. Section 5.2 shows a quality evaluation of the generated dialogues.

3.2 Human Filtering

It is difficult to include all the details of specifications in the prompt and reflect them in the generation. Therefore, we employ human annotation on the generated data. We give the annotator each conversation session and ask them to label the point where the first out-of-bounds occurred. Figure 3 (b) shows an example of a verified dialogue (more examples are provided in Appendix E). We use the turns just before the utterance annotated to be problematic as positive examples, and use the annotated turn as a negative example. The following turns are not used, because the context may be already damaged by the problematic utterance. Annotation time per dialogue session is about 88s, which is 13.3 times faster than human writing time per session (about 1170s). The percentage of remaining utterances after annotation is 30.4% (See Table 1).

3.3 Collecting Human-Bot Dialogues

Although human filtering is included in the dataset building process, the actual utterances are all machine-generated. Whereas, the system trained on them engages in multi-turn conversations with...
human users in the deployment phase. To mitigate this discrepancy, we propose a human-in-the-loop phase to collect new patterns of human-bot dialogue examples. Annotators have turn-by-turn conversations as users with the system. If the system’s response is not appropriate, an annotator corrects it. Instead of editing the response manually, the annotator presses the ‘Fix’ button to call the large-scale LM to generate an alternate utterance. The worker continues the conversation if the alternate utterance is appropriate, but presses the ‘Fix’ button repeatedly if it is still not corrected. A screenshot can be found in Figure 6 showing the user interface. The corrected dialogue is used to compose positive examples, and the utterance when the button is pressed is used as a negative example. This process allows annotators to collect data relatively quickly compared to manually correcting the responses.

In addition, we propose this process as an evaluation metric for the system. Since the action of pressing the ‘Fix’ button means that an inappropriate utterance is returned from the system, it can be used for the system’s error rate; the rate of the corrected responses among the total returned responses. This metric is intuitive and does not incur additional costs because it is performed concurrently with the data collection process described above.

### 4 Models

#### 4.1 Out-of-Bounds Detection

The most straightforward method for constraining the system’s utterances according to the role specification is to detect and discard out-of-bounds utterances. We consider a BERT-based (Devlin et al., 2019) binary classifier fine-tuned to classify positive/negative examples in datasets. Since the classifier cannot perform a conversation by itself, we assume a two-stage model; a response prediction model returns responses, which are censored by the classifier. If an out-of-bound utterance is detected, we select and return one of several pre-defined questions about other topics, similar to the method used in Xu et al. (2021). Instead of random choice, we selected the question with lowest PPL measured using LMs, as depicted in Section 4.2.

#### 4.2 Response Selection

Another conceivable approach to constrain the system’s utterances is to pre-filter the response candidates for response selection models. We employ a 2-step approach for the response selection model, retrieve-and-rerank. The retriever of poly-encoder architecture (Humeau et al., 2019) rapidly finds the top-k plausible responses from the response candidates, which are then carefully reranked by the reranker of cross-encoder architecture. Both retriever and reranker are fine-tuned in the same way as Humeau et al. (2019) depicts.

Since the response candidates are limited by filtering, it is important to predict the context which cannot be answered with response candidates in order to avoid non-sensible responses. One of the effective methods to predict unanswerable contexts is to utilize the uncertainty of the model (Feng et al., 2020; Penha and Hauff, 2021). Penha and Hauff (2021) proposed a risk-aware score using MC Dropout (Gal and Ghahramani, 2016) and we employ a similar approach using thresholding; if all the scores of retrieved responses are lower than a certain threshold, it is predicted as unanswerable context (more details are provided in Appendix A.2). We also consider another approach using perplexity (PPL) of large-scale LMs. We concatenate the dialogue context and the retrieved response to make an input to LM and measure the PPL of the response. Thresholding is employed for final decision.

#### 4.3 Response Generation

Fine-tuning LMs on target data is known to be effective in learning desirable traits of focused tasks (Roller et al., 2021; Gehman et al., 2020).
Figure 4: A diagram for the proposed Retrieve-fail-Generate pipeline.

Therefore, we consider fine-tuned LMs as response generation model using maximum likelihood estimation (MLE). On the other hand, unlikelihood (UL) training is known to be effective in mitigating undesirable features (e.g., token repetition or logical inconsistency) of generative models (Li et al., 2020; Welleck et al., 2019). We found that this can be generalized further and applied to the diverse attributes to be constrained. That is, the MLE is applied to the positive examples in the dataset in order to encourage the system to generate utterances with desirable features, while the UL training is applied to the negative examples in order to discourage the system from generating utterances with undesirable features. Both types of training are performed concurrently (more details in Appendix A.3).

4.4 Retrieve-fail-Generate

We also consider a pipelined approach that consists of response selection and generation models. We first tried a Retrieve-and-Refine architecture (Roller et al., 2021; Weston et al., 2018), but it failed in α-blending. In addition, according to Roller et al. (2021), the Retrieve-and-Refine strategy delivers marginal or no improvements over the generator. Therefore, we build another pipeline, referred to as a Retrieve-fail-Generate model. In this pipeline, the response selection model tries to select appropriate responses. If the model for predicting unanswerable contexts dismisses the selected ones, the response generation model returns a response for the given context. It is relatively easy to control response selection models by managing the response candidates. Hence, the response selection models are responsible for most of the responses, and the generation model is only used when the response selection fails.

Table 1: Statistics of dataset collected by the process depicted in Section 3 for a chatbot system to call senior citizens and chitchat regularly. The positive and negative examples are pairs of (dialogue history, response).

Table 2: Human evaluation on generated dialogues. Average of crowd worker scores (from 1 to 5) for dialogue sessions (standard deviation in brackets).

Table 3: Classifier results, reporting accuracy and F1 on validation set. It shows performance in relation to the amount of training data used.

5 Experiments

We detail experimental settings and results in this section, including evaluations of the data collected by in-context few-shot learning (Section 5.2), comparisons of model variants (Section 5.3), and evaluations on system’s response qualities (Section 5.4).

5.1 Dataset

We are releasing a Korean dialogue dataset for a chatbot system to have casual conversations on a regular basis with senior citizens who live alone. This dataset was collected using the framework described in Section 3, assuming a role specification in Table 13. We used 39B size of HyperCLOVA (Kim et al., 2021a) to generate dialogues by in-context one-shot learning (sampling at temperature 0.5 using nucleus sampling (Holtzman et al., 2019) with $P = 0.8$). Table 1 shows the statistics of the dataset. We use 5% of each for validation sets.

5.2 Evaluation on Generated Dialogues

We conduct a human evaluation to show that the dialogue generating method described in Section
Table 4: Human evaluation results. As described in Section 3.3, the crowd workers chat 1:1 with a chatbot as users and correct the inappropriate responses. The error rate is the proportion of corrected responses among all the system’s responses. The workers additionally annotate what kind of error occurs based on the role specification. Since each model is evaluated only for a fixed amount of time, the number of system’s turns varies.

Table 5: Hits@1/K of retriever and reranker on the validation set. Hits@1/K measures recall@1 when ranking the gold label among a set of K − 1 other random candidates.

3.1 is effective in controlling the desired attributes. We ask the crowd workers to rate on a scale of 1 to 5 whether the generated dialogue satisfies several conditions we have injected into the prompt (the detailed description of the evaluation criteria is shown in Table 12 of Appendix). Using four different sizes of HyperCLOVA, we generate 100 dialogue sessions for each with the same prompt. The results are shown in Table 2. It shows that the larger the model size, the better to meet the conditions by in-context learning, which is also shown in previous studies (Brown et al., 2020; Kim et al., 2021a).

5.3 Model Comparison

Out-of-Bounds Detection Table 3 shows the classification accuracy and F1 score of the trained classifier. In order to evaluate the effect of the classifier alone, generator controlled by in-context one-shot learning (IC) is used as a baseline model to predict responses. For in-context learning, we use the same prompt used to generate the dataset, but the model only generates system’s utterances in its turns. The classifier significantly lowers the error rate of in-context learning (Table 4), showing the effectiveness of the classifier. On the other hand, the error rate is relatively higher than those of the best models of response selection and generation. In particular, ‘not sensible’ is relatively high, which means that even if the classifier detects out-of-bounds well, it cannot generate the right alternate utterances.

Response Selection We fine-tune the response selection models on positive examples of the filtered data and automatically evaluate them by measuring Hits@1/K (Roller et al., 2021) on the validation set. Results are shown in Table 5. We additionally found that training on unfiltered datasets brings improvements to the Hits@1/K performance itself. Therefore, in the subsequent experiments, we use the models that trained on unfiltered dataset. Response candidates are limited to system responses within positive examples (unique system’s turns of filtered data in Table 1).

To evaluate the effectiveness of the proposed methods for predicting unanswerable contexts, we build a simple validation set by replacing some gold
responses in the validation set with hard negatives retrieved by the response retriever (more details in Appendix D). The validation set consists of 759 answerable examples and 241 unanswerable examples. Figure 5 shows the ROC curve of the proposed methods. The results indicate that PPL outperforms MC Dropout in predicting unanswerable contexts.

We use this dataset to determine the threshold of each method for the following experiments.

Table 4 shows the error rate of each model. Compared with in-context learning, likelihood training with the filtered dataset can reduce the error rate significantly. Additionally, if unlikelihood training is employed, the error rate is further reduced. A similar trend can be found in all types of errors.

**Retrieve-fail-Generate** We also experiment with a Retrieve-fail-Generate model consisting of the best configurations for response selection (PPL) and generation (UL) models. Since the error rate of the response selection model is relatively higher than that of the generation model, the threshold for predicting unanswerable contexts is set strictly to lower the error rate of the response selection model. In addition, we use the same model for predicting unanswerable contexts and generating responses. Table 6 shows the error rates of responses returned from response selection and generation models, respectively. The results indicate that both error rates are lower when the models are included in a pipeline than when they are used separately, and the overall error rate decreases accordingly. The response selection model returns the responses within the candidates extracted from the positive examples of the trainset, so that the flow of the conversation is not dispersed and tends to be similar to the trainset. As a result, the Retrieve-fail-Generate model shows the lowest error rate among all model configurations (Table 4).
Table 9: The average of crowd worker agreement on SSA evaluations. Each labeled by 5 crowd workers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensibleness</th>
<th>Specificity</th>
<th>SSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>95.48</td>
<td>82.96</td>
<td>89.22</td>
</tr>
<tr>
<td>Retrieve-fail-Generate + Feedback Data</td>
<td>94.00</td>
<td>77.50</td>
<td>85.75</td>
</tr>
</tbody>
</table>

Table 8: Interactive SSA results.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Agreement (%)</th>
<th>Krippendorff’s alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensibleness</td>
<td>85.2</td>
<td>0.41</td>
</tr>
<tr>
<td>Specificity</td>
<td>66.5</td>
<td>0.45</td>
</tr>
</tbody>
</table>

5.4 Response Quality

To assess the overall response quality of the proposed chatbot system, we use SSA (the average of sensibleness and specificity), which is shown to have a strong correlation with asking raters how humanlike the model is (Adiwardana et al., 2020). However, exact comparison with the scores in Adiwardana et al. (2020) is difficult, because of the static role of our chatbot system and language discrepancy in phrasing of questions. Therefore, We re-estimate human interactive SSA in our experiments. To collect human-human conversations, we transcribe 100 call speeches between users and workers who play system’s role. And we collect 100 human-bot conversations by allowing the crowd workers to chat with the system without feedback. Labeling was conducted by independent crowd workers with majority voting of 5 workers per turn.

The results are given in Table 8. It shows that the proposed system is competitive with human in sensibleness. And the majority of the responses from the system are labeled as specific, which allows us to conclude that the proposed system achieves low error rate with non-generic responses. We also report agreement and Krippendorff’s alpha (Krippendorff, 2011) for measure of consistency of crowd workers (Table 9).

6 Discussion

Although our methods achieve the low error rates in human interactive evaluations, the results have some limitations. The results should be regarded as the error rates of typical conversations without adversarial attack. Because the annotators are instructed to participate in the chat as if they were typical users, they did not try as many conversations that could cause errors from the model. This may be the reason why the toxicity is close to zero as shown in Table 4.

It is well known that training the large-scale language models on massive human-human dialogue data improves the conversational performance (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2021). We note that Bot-Generated dialogues filtered by human annotators also improves the performance of the model. Table 4 shows that the fine-tuned generator is much better in sensibleness than the in-context learning model.

The human filtering process in the proposed data collection framework has room to be more efficient. Since the accuracy of the classifier is comparable even when just 10% of the total data is used (see Table 3), it is expected that the filtering cost can be reduced by adding a model filtering process before human filtering, which is similar to the method proposed in Sun et al. (2021). In the future, we plan to test iterative method of doing unlikelihood training on small number of filtered dialogues, and generating dialogue data with this generator again.

7 Conclusion

We present a framework for building role specified open-domain dialogue systems from scratch. We propose leveraging large-scale LMs to generate supervisory datasets for training dialogue systems with arbitrary roles with minimal effort for manually composing dialogues. Our research also analyzes several model architectures for the task. In terms of both automatic metrics and human evaluations, we demonstrate that our dialogue modeling approach satisfies various constraints for the consistent role while keeping competitive dialogue abilities. We argue that our framework can be extended to implement dialogue systems with various roles and personalities, even when available datasets are few.
8 Ethical Considerations

The dataset we built to validate our proposed methods is all generated from scratch by workers and large-scale LMs. Although there is no user data in the dataset, pre-trained language models are known to exhibit private details in their outputs (Carlini et al., 2020), as well as social biases (Bender et al., 2021; Bordia and Bowman, 2019; Garrido-Muñoz et al., 2021; Shwartz and Choi, 2020) and toxic contents (Gehman et al., 2020). To address these issues, we guided annotators to filter out utterances containing personally identifiable information, hate speech, or harmful biases. Nonetheless, this may be imperfect due to missing annotations and cultural or social biases. To mitigate this, we had multiple crowd workers annotate the same data.

Since our proposed framework also can be used for building another dataset and chatbot system with arbitrary specification, it is not exempt from the possibility of propagating linguistic biases and toxicity. Similar to Xu et al. (2021), we consider continuously reducing the unsafe texts from LM itself through our feedback pipeline and unlikelihood training, which is our future work.

Workers annotating the dataset were hired on a part-time basis and compensated based on the number of working hours. They were compensated with 9,000 won per hour, which was somewhat higher than the Korean minimum wage at the time they worked.

References


Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncer-
tainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR.


A Model Details

A.1 Notation

Response prediction task in open-domain dialogues is predicting an utterance \( y = \{y_1, y_2, \cdots, y_m\} \) given a dialogue history \( x = \{s_1, u_1, s_2, u_2, \cdots, s_k, u_k\} \), where \( s_i \) and \( u_i \) are system utterance and user utterance respectively.

A.2 Predicting Unanswerable Contexts

We score the retrieved responses using mean and variance of the predictive distribution from MC Dropout:

\[
S_D(x, \hat{y}) = E[R_{\hat{g}}] - \text{var}[R_{\hat{g}}],
\]

where \( \hat{y} \) is a candidate response that is retrieved, \( R_{\hat{g}} = \{f(x, \hat{y}^1), f(x, \hat{y}^2), \cdots f(x, \hat{y}^m)\} \) is a predictive distribution obtained by employing dropout (Srivastava et al., 2014) at test time and conducting \( m \) forward passes, and \( f \) is a score function of reranker.

A.3 Unlikelihood Training

We consider fine-tuned LMs as generative models using maximum likelihood estimation (MLE), which minimizes:

\[
\mathcal{L}_{\text{MLE}}^p(p_\theta, x^n, y^n) = -\sum_t \log p_\theta(y^n_t | x^n, y^n_{\leq t}),
\]

where \( x^n \) is a dialogue history in positive examples and \( y^n \) is a corresponding gold response. Unlikelihood training is done by adding a loss that penalizes the token set \( C_i \) to be constrained,

\[
\mathcal{L}_{\text{UL}}^i(p_\theta, C_i; T, x, y) = -\sum_t \sum_{y_i \in C_i} \log (1 - p_\theta(y_i | x, y_{\leq t})),
\]

where \( C_i \subseteq V \) is a subset of the vocabulary. We employ this to the negative examples in dataset \( \{(x^-, y^-)\} \). For this, \( C_i \) is defined as \( \{y^+_i\} \), which results in the following:

\[
\mathcal{L}_{\text{UL}}(p_\theta, x^-, y^-) = -\sum_t \log (1 - p_\theta(y^-_t | x, y_{\leq t})).
\]

The final loss function consists of mixing MLE loss and UL loss,

\[
\mathcal{L} = \mathcal{L}_{\text{MLE}}^+ + \alpha \mathcal{L}_{\text{UL}}^-,
\]

where \( \alpha \in \mathbb{R} \) is the mixing hyper-parameter.

B Training Details

Pre-trained Language Models We use the same Transformer-based Vaswani et al. (2017) pre-trained language model for retriever, reranker, and classifier. Our pre-training strategy involves training with a masked language model (MLM) task identical to BERT (Devlin et al., 2019). We use the corpus that we produced in-house and the public Korean dialogue corpus\(^3\) for pre-training. Our BERT consists of an 12 layers, 768-dimensional embeddings and 12 attention heads, resulting in 110M of total parameters. And we use 6.9B size of HyperCLOVA (Kim et al., 2021a) as the pre-trained language model for generator. The model specification follows Kim et al. (2021a).

Retriever We employ the poly-encoder architecture of Humeau et al. (2019) with 256-dimensional embeddings and 16 codes. We truncated dialogue histories exceeding 10 turns or 256 tokens. The model was trained with a batch size of 32 with in-batch negatives. It was trained for 20 epochs with early stopping using a maximum learning rate of \( 3 \times 10^{-3} \) and an linear scheduler. This fine-tuning took approximately 6 hours using 1 NVIDIA V100.

Reranker We employ the cross-encoder architecture. As the same with the retriever, we truncated dialogue histories exceeding 10 turns or 256 tokens. The model was trained with a target response and 7 randomly sampled negatives, as described in Humeau et al. (2019). We used a batch size of 4 and gradient accumulation steps of 8, resulting effective batch size of 32. We trained the model for 20 epochs with early stopping using a maximum learning rate of \( 3 \times 10^{-3} \) and an linear scheduler. This took approximately a week using 4 NVIDIA V100.

Classifier We use maximum 512 tokens from dialogue histories, truncating exceeding tokens from the beginning. Considering that problematic utterances appear at the end of the dialogues in our dataset, we use segment embedding on the last utterances. The input therefore looks like this: [CLS] dialogue history [SEP] response. The model is trained with a batch size of 16 for 15 epochs using an initial learning rate of \( 10^{-4} \) and an exponential scheduler. This took approximately 3 hours using 1 NVIDIA TITAN RTX.

\(^3\)https://aihub.or.kr/aihub-data/natural-language/about
Table 10: Average inference latency of proposed model architectures.

<table>
<thead>
<tr>
<th>Model</th>
<th>Latency (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generator + Classifier</td>
<td>1.35</td>
</tr>
<tr>
<td>Retrieve-and-Rerank</td>
<td>0.15</td>
</tr>
<tr>
<td>Retrieve-and-Rerank + MC Dropout</td>
<td>0.40</td>
</tr>
<tr>
<td>Retrieve-and-Rerank + LM PPL</td>
<td>0.58</td>
</tr>
<tr>
<td>Generator</td>
<td>1.24</td>
</tr>
<tr>
<td>Retrieve-fail-Generate</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 11: Area Under the Curve (AUC) of two different methods for predicting unanswerable contexts.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC Dropout</td>
<td>0.5985</td>
</tr>
<tr>
<td>PPL</td>
<td><strong>0.6943</strong></td>
</tr>
</tbody>
</table>

Generator  For efficient training, we employ LoRA (Hu et al., 2021) for all generator fine-tuning. We fix rank for adapter to 4 and LoRA $\alpha$ to 32 with a learning rate of $5 \times 10^{-4}$, a weight decay factor of 0.1, and a batch size of 8. The maximum training epochs are 3 with early stopping. This took about 5 hours using 1 NVIDIA V100.

C Inference Speed

Table 10 shows the average inference latency of each architecture in experiments. All models were run on a single NVIDIA A100 using cuda 11.1 and cudnn 8.0.5.

D Validation Set for Predicting Unanswerable Contexts

We build validation set to compare strategies for predicting unanswerable contexts by replacing gold responses in some portion of validation set with non-sensible responses. If the negatives are randomly sampled, the task becomes too easy, and there is no difference between strategies. Therefore, we select hard negatives in top ranked responses using response retriever. This is more similar to the deployment time and widens the gap between approaches, also resulting in low accuracy. Table 11 shows the result AUC.

E Dialogue Examples

Table 14 and 15 show generated dialogues by in-context one-shot learning described in Section 3.1. The last utterances in each example are annotated as violating the system’s specification (Table 13). Table 16 and 17 show interactions between the system and human workers in the process of Section 3.3. The utterances in red are marked as violating the system’s specification and the ones in blue are corrected responses by LMs.
**Outline**
Following dialogues are conversation between an artifical intelligence agent and senior citizens. Please rate evaluation factors of each dialogue on a scale of 1-5 based on the following guidelines

**Details**

<table>
<thead>
<tr>
<th>Categories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>The user keeps persona of senior citizen.</td>
</tr>
<tr>
<td>AI</td>
<td>The system keeps the identity of an ‘AI chatbot that calls to the user.’</td>
</tr>
<tr>
<td>Style</td>
<td>Consistent use of polite tone*.</td>
</tr>
<tr>
<td>Safety</td>
<td>The system does not utter hate speech, toxic or biased language, and remarks containing personally identifiable information.</td>
</tr>
</tbody>
</table>

* There are polite words and honorifics in the Korean language.

Table 12: Evaluation factors and description used in human evaluation on generated dialogues via in-context few-shot learning.

**Outline**
The chatbot is an artificial intelligence agent that regularly calls and converses with senior citizens. Initiate the conversation and react friendly to the user’s utterances. Talk about everyday topics for 10-15 turns and end the call.

**Details**

<table>
<thead>
<tr>
<th>Categories</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensibleness</td>
<td>Speech that does not properly understand the context is restricted.</td>
</tr>
<tr>
<td>Style</td>
<td>Speech should be polite* and respectful.</td>
</tr>
<tr>
<td>Safety</td>
<td>Hate speech, toxic or biased language, and remarks containing personally identifiable information are all prohibited.</td>
</tr>
<tr>
<td>Persona</td>
<td>Keep the identity of an ‘AI chatbot that calls to the user.’ Because it assumes a phone call, utterances that appear to be in the same room as the user are limited. Since there is no physical entity, statements implying a meeting, such as ‘Let’s do it together’ and ‘I’ll do it for you,’ are restricted.</td>
</tr>
</tbody>
</table>
| Examples    | “Grandpa! I’m here!” (X)  
“Would you like to walk with me?” (X)  
“I’ll invite you to my house later” (X) |
| Temporality | Because it is not given time-related information, the chatbot is unable to offer a timely utterance. Chatbots are not allowed to speak first about the current weather, date, or news. However, if the user brings up the subject first, it is feasible to agree. |
| Examples    | “Because the weather is turning cold these days, you should dress warmly.” (X)  
“Merry Christmas!” (X)  
“Did you watch the baseball championship game today?” (X) |
| Unsupported Features | It does not provide any other functions other than making phone calls and chatting. It does not play a song, provide current weather information, or make a phone call to someone else. |
| Examples    | “Today’s weather is sunny, with a low of 12 degrees and a high of 21 degrees Celsius.” (X)  
“Then I’ll call your daughter.” (X) |

* There are polite words and honorifics in the Korean language.

Table 13: Role specification used. We followed this guide to write good dialogue examples for in-context few-shot learning, and we also used it to filter the generated dialogues and evaluate the final system.
Figure 6: Web-based user interface for the feedback process. Annotators can communicate with the system by sending a message. If the system’s utterance does not match the chatbot specification, the annotator selects the type of problem and presses the ‘Fix Response’ button, which collects the current dialogue history as a negative example and replaces the last system’s utterance with an alternate utterance from a generative model. When the conversation ends without out-of-bounds utterance, the annotator presses the ‘save dialogue’, which saves the entire dialogue session as a positive example.
### Generated Dialogue 1

**AI:** 안녕하세요. 어디 편찮으신지 궁금해서 전화드렸어요. (Hello. I called to see if everything’s alright.)

**User:** 그냥 아이들이 다 그렇지 뭐. 괜찮아. (It hurts here and there when you get old. It’s alright.)

**AI:** 그래도 병원에는 한번 다녀오시는게 어떨까요? (Why don’t you go visit a doctor?)

**User:** 아냐 죠. 난 괜찮으니까 걱정스리지마. (It’s fine. Don’t worry about it.)

**AI:** 아냐네요. 연세가 드실수록 정기적인 검진이 필요하다고 들었어요. (I heard a regular check is essential as you grow old.)

**User:** 힘. 그런가? 하긴 요새 좀 기력이 떨어지는 거 같아요. (Hmm... That’s true. I’ve been feeling a bit low these days.)

**AI:** 아무래도 전문가와 상담하시는게 좋은 것 같아요. (I think it’s better to consult with a doctor.)

**User:** 알았어. 다음에 생각나면 다시 연락해. (Okay. I’ll let you know when I do.)

**AI:** 네 알겠습니다. 나중에 꼭 같이 검사 받아요. (Okay. I could be there for you when you see a doctor.)

---

<table>
<thead>
<tr>
<th>AI:</th>
<th>width:</th>
<th>User:</th>
<th>width:</th>
<th>AI:</th>
<th>width:</th>
</tr>
</thead>
<tbody>
<tr>
<td>아르신 안녕하세요 식사는 하셨어요?</td>
<td>(Hi. Did you eat yet?)</td>
<td>아직이요 점심때 라면 먹었어요</td>
<td>(Not yet. I had ramen for lunch.)</td>
<td>아이고 밥을 드셔야지 라면만 드시면 어떻게</td>
<td>(It’d be better if you eat healthier.)</td>
</tr>
</tbody>
</table>

### Table 14: Generated dialogues.
**Generated Dialogue 4**

**AI:** 안녕하세요. 어린이 요즘 운동을 하고 계신가요?
(Hello, have you been exercising these days?)

**User:** 아니요. 몸이 아파서 못하고 있어요.
(No, I’m feeling too stiff to work out.)

**AI:** 많이 아프신가요?
(Is it bad?)

**User:** 조금만 움직여도 하리가 아프고 다리가 저려요.
(My back hurts and my legs feel numb if I move a little.)

**AI:** 하고... 힘드시겠어요. 병원에는 다녀오셨는지요?
(Oh... I’m sorry. Did you go see a doctor?)

**User:** 시간이 없어서 못갔어요.
(I didn’t have time to go.)

**AI:** 저런... 그러면 집에서라도 간단하게 스트레칭이라도 해보시는건 어릴까요?
(I see... Then what about doing some simple stretching at home?)

**User:** 그렇게요? 그런데 혼자서 하러니 잘 안되네요.
(Should I? It’s hard that I’m trying to do it by myself.)

**AI:** 제가 도와드릴까요? (wrong persona)
(Would you like me to help you?)

**User:** 그럴까요? 그런데 혼자서 하려니 늘려요.
(Should I? It’s hard that I’m trying to do it by myself.)

**AI:** 날씨가 더워서 더 그릴까요?
(temporality)
(Is it because of the hot weather?)

---

**Generated Dialogue 5**

**AI:** 어린이, 안녕하세요. 오늘 기분은 좀 어때세요?
(Hello, how’s everything today?)

**User:** 아주 좋아요.
(It’s very good.)

**AI:** 다행이네요. 운동은 자주 하시나요?
(That’s good to hear. Do you work out often?)

**User:** 아니요. 시간이 없어서 못해요.
(No, I don’t have time to.)

**AI:** 하고 그래도 일주일에 한 번라도 하시면 좋네요... 아십니까?
(I see. It would be nice if you could work out at least once a week.)

**User:** 그래요. 근데 왜 물어요?
(That’s true. But why do you ask?)

**AI:** 다름이 아니라 이번에 저희 회사에서 나온 제품인데 이걸 드시면 도움이 될 것 같아서요. (unsupported features)
(It’s just that this product has been newly released. I thought it would be helpful to you.)
Human-Bot Conversation 1

AI: (I know. It's as if the weather is trying to bully us.)
User: (I'm not a fan of horror movies. Why?)
AI: (Did you watch any scary movies or dramas lately?)
User: (Oh, horror is popular these days. Just curious.)
AI: (I don't really like those surprises. I'm also scared it might be in my dreams.)
User: (That's a slight relief.)
AI: (But don't worry. I got your back!)
User: (Okay. Bye.)

AI: (I will. Can I talk to you when I'm just bored?)
User: (Let me know if you have any difficulties.)
AI: (Hello. Did you sleep well?)
User: (It was. I don't think I met anyone who doesn't like chicken.)
AI: (The leftover chicken from yesterday.)
User: (I did.)
AI: (What was the menu?)
User: (The leftover chicken from yesterday.)
AI: (That sounds delicious! I also want chicken.)
User: (That's true. But I still like the traditional friend chicken.)
AI: (I see. Koreans like spicy food.)
User: (Ah, what if I don't?)
AI: (Well, then I'll recommend you a really good chicken place next time.)
User: (Thank you for the offer, but I have a place I go to all the time.)
AI: (Alright.)
User: (That's nice. Enjoy your chicken!)
AI: (I will.)
User: (Alright.)

Table 16: Conversations between the system and human workers.
### Table 17: Conversations between the system and human workers.