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].

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.

Figure 1: An example of role specified open-domain dialogue system that cares for senior citizens living alone.

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-

[1]The 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 pretrained 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

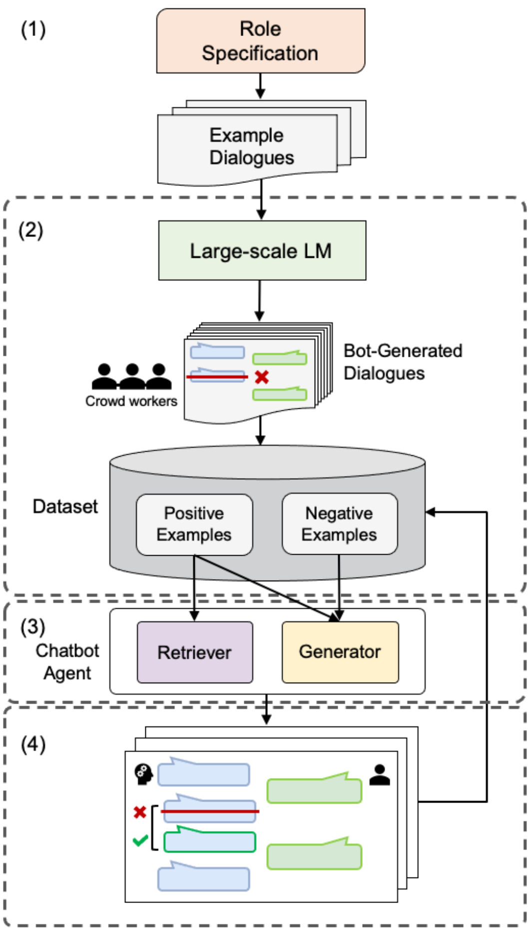

Figure 2: Our proposed framwork: (1) the dialogue developer provides a specification of the desired chatbot and a few dialogue examples, (2) large-scale LMs generate entire dialogues and crowd workers filter the system's utterances, (3) a dialogue model is trained with supervised learning on the dataset, (4) crowd workers chat 1:1 with the chatbot and give additional feedback.

in Budzianowski et al. (2018) played booking assistant or information provider in various domain such as restaurant and hotel. On the other hand, the Zhang et al. (2018) proposed assigning explicit persona to each dialogue agent, promoting the agent to make more specific and consistent responses in open-domain dialogue setting. However, the persona given by a few natural language sentences are insufficient to represent specific role in the real world scenario. Sun et al. (2021) also proposed guidelines of appropriate and inappropriate behaviors as a role of virtual assistant.

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 correct 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.

(a) Input Prompt

The chatbot in question 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 several turns and end the call.

###
AI: Hello. Did you sleep well last night?
User: Yes, I slept well. Thank you.
AI: That's good. I'm glad you slept well. Make sure not to skip any meal.
User: Okay. you eat well, too.
AI: Alright. I'll call you again later.
User: Bye.

###
AI:

(b) Generated Example

AI: Hello, how's your health these days?
User: It's very good. I didn't even catch a cold.
AI: **That's awesome. Did you eat well?**
User: I only have rice. I don't really want anything else.
AI: **You should take in different nutrients all-around. You can stay healthy that way.**
User: Okay, I'll try.
AI: **It's getting cold. Are you wearing warm clothes?**
...

Figure 3: Examples for the data construction process. The utterances in blue are target responses of positive examples, and the one in red is a response of a negative example. The following dialogue is dropped. The examples are translated into English.

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; an 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). There-

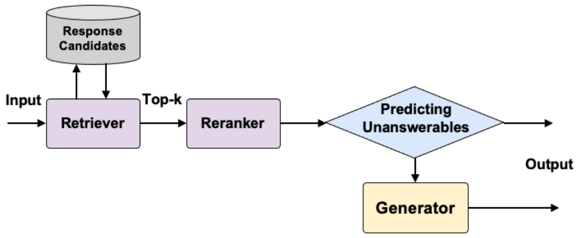

Figure 4: A diagram for the proposed Retrieve-fail-Generate pipeline.

fore, 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[2]. 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, refered 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.

[2]In our experiments, all retrieved responses are copied or ignored depending on the α value, reducing the model to a retriever or generator.

Dialogue Type	Example	Generated	Filtered	Feedback
# of sessions	250	25,000	25,000	1,623
# of uttrs	3,893	510,028	154,903	29,365
# of pos examples	-	-	47,091	10,829
# of neg examples	-	-	18,583	3,529
# of unique sys-turns	1,805	170,527	36,227	9,405

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).

		User	System		
Model	Situation	Persona	Persona	Style	Safety
1.3B	4.57 (0.29)	4.54 (0.15)	4.31 (0.23)	4.91 (0.05)	4.98 (0.03)
13B	4.74 (0.23)	4.65 (0.11)	4.33 (0.20)	4.93 (0.04)	4.98 (0.02)
39B	4.69 (0.22)	4.69 (0.12)	4.37 (0.21)	4.88 (0.05)	4.97 (0.02)
82B	**4.78** (0.17)	**4.74** (0.15)	**4.49** (0.17)	**4.96** (0.07)	4.96 (0.03)

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

Training Data (%)	Accuracy (%)	F1
10	91.03	0.9278
20	90.89	0.9319
100	92.07	0.9341

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

Model	# of system turns	error rate (%)	not sensible (%)	wrong persona (%)	policy violation (%)	not safe (%)	etc. (%)
Out-of-Bounds Detection							
Generator (IC) + Classifier	1,471	18.10	9.31	1.61	2.49	0.07	4.66
Response Selection							
Retrieve-and-Rerank	1,230	13.17	10.68	0.72	1.53	0.00	0.24
Retrieve-and-Rerank w/ MC Dropout	1,272	9.82	7.58	0.36	1.66	0.00	0.22
Retrieve-and-Rerank w/ PPL	1,300	**7.00**	5.10	0.40	1.16	0.00	0.34
Response Generation							
Generator (IC)	985	35.83	16.05	6.24	8.66	0.17	4.68
Generator (MLE)	1,291	4.72	3.55	0.76	0.30	0.00	0.10
Generator (UL)	1,497	**3.82**	3.29	0.23	0.10	0.00	0.17
Retrieve-fail-Generate							
Retrieve-and-Rerank w/ PPL + Generator (UL)	1,522	**2.56**	2.20	0.17	0.16	0.00	0.00
Retrieve-and-Rerank w/ PPL + Generator (UL) + Feedback Data	1,599	**2.00**	1.88	0.00	0.10	0.00	0.00

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.

Model	data	# of examples	Hits@1/20	Hits@1/100
Retriever	Filtered	47,091	93.14	83.80
	Unfiltered	227,638	**95.27**	**86.99**
Reranker	Filtered	47,091	97.16	90.89
	Unfiltered	227,638	**97.55**	**91.70**

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.

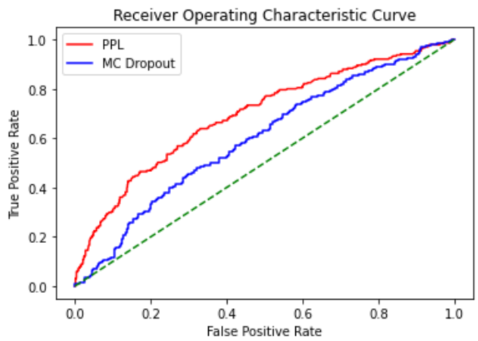

Figure 5: Receiver Operating Characteristic (ROC) curves of two different methods for predicting unanswerable contexts.

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

Model	Response Selection proportion (%)	Response Selection error rate (%)	Response Generation proportion (%)	Response Generation error rate (%)
Retrieve-and-Rerank w/ PPL + Generator (UL)	68.20	2.50	31.80	2.68
Retrieve-and-Rerank w/ PPL + Generator (UL) + Feedback Data	63.70	2.12	36.30	1.77

Table 6: Evaluation results of each component in the Retrieve-fail-Generate pipeline. It shows the proportion and error rate of returned responses from response selection and generation models.

Method	positive	negative
In-context Learning	2.65	2.74
Likelihood Training	2.07	2.47
Unlikelihood Training	2.48	46.70

Table 7: Perplexity (PPL) of generative models on validation set. PPLs are measured for responses in dataset, while the inputs are concatenated pairs of (dialogue history, response).

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 the response selection models. The model that does not predict unanswerable contexts (Retrieve-and-Rerank) has a higher error rate in 'not sensible' than others. The case of using PPL as the method for predicting unanswerable contexts shows a lower overall error rate than the case of using MC Dropout, and the proportions of the total contexts predicted as unanswerable are similar at 4.23% and 3.85% for PPL and MC Dropout, respectively. The results also show the error types from the models. Even if only the filtered utterances are used as response candidates, 'wrong persona' and 'policy violation' appear in responses. It seems that a few unfiltered utterances remain in the response candidates, since the human filtering is not perfect. Or even the same utterance can cause errors depending on the context. For example, it is possible to agree with when a user calls the chatbot by a different name, which in this case is the 'wrong persona'.

Response Generation We compare three ways to train generators; in-context learning, likelihood training, and unlikelihood training. We measure the perplexity of the three models on positive and negative examples. Table 7 shows the results. The difference between the PPL of the positive examples and the negative examples is the smallest in in-context learning. When trained on positive examples of filtered data with likelihood training, the difference increases slightly. This is because the PPL of the positive examples is lowered. When adding unlikelihood training, the PPL for negative examples increase significantly.

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).

Method	Sensibleness	Specificity	SSA
Human	95.48	82.96	89.22
Retrieve-fail-Generate + Feedback Data	94.00	77.50	85.75

Table 8: Interactive SSA results.

Metric	Agreement (%)	Krippendorff's alpha
Sensibleness	85.2	0.41
Specificity	66.5	0.45

Table 9: The average of crowd worker agreement on SSA evaluations. Each labeled by 5 crowd workers.

Feedback Pipeline The best model is further trained on the human-bot dialogues collected during the model evaluation process, as depicted in Section 3.3. Both response selection and generation models are newly initialized and trained. As a result, all types of error rates are consistently reduced (Table 4), and the error rates of both the response selection and generation models are decreased (Table 6). The effect is stronger on the response generation.

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 discrepency 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

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*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. *CoRR*, abs/2012.07805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Yulan Feng, Shikib Mehri, Maxine Eskenazi, and Tiancheng Zhao. 2020. "none of the above": Measure uncertainty in dialog response retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2020, Online. Association for Computational Linguistics.

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

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. 2021. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329. IEEE.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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*.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, Na-Hyeon Ryu, Kang Min Yoo, Minsuk Chang, Soobin Suh, Sookyo In, Jinseong Park, Kyungduk Kim, Hiun Kim, Jisu Jeong, Yong Goo Yeo, Donghoon Ham, Dongju Park, Min Young Lee, Jaewook Kang, Inho Kang, Jung-Woo Ha, Woomyoung Park, and Nako Sung. 2021a. What changes can large-scale language models bring? intensive study on HyperCLOVA: Billions-scale Korean generative pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021b. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3704–3717, Online. Association for Computational Linguistics.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

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, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Gustavo Penha and Claudia Hauff. 2021. On the calibration and uncertainty of neural learning to rank models for conversational search. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 160–170, Online. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2020. Controlling style in generated dialogue. *arXiv preprint arXiv:2009.10855*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding chit-chat to enhance task-oriented dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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, Brussels, Belgium. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 3506–3510.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968, Online. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill 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, Online. Association for Computational Linguistics.

A Model Details

A.1 Notation

Response prediction task in open-domain dialogues is predicting an utterance $y = \{y_1, y_2, \cdots, y_{|y|}\}$ 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:

$$S_{\mathrm{D}}(x, \hat{y}) = E[R_{\hat{y}}] - var[R_{\hat{y}}],$$

where $\hat{y}$ is a candidate response that is retrieved, $R_{\hat{y}} = \{f(x, \hat{y}^1), f(x, \hat{y}^2), \cdots f(x, \hat{y}^m)\}$ 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:

$$\mathcal{L}^n_{\mathrm{MLE}}(p_\theta, x^n, y^n) = -\sum_t \log p_\theta(y^n_t | x^n, y^n_{<t}),$$

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_t to be constrained,

$$\begin{aligned}\mathcal{L}^n_{\mathrm{UL}}(p_\theta, C_{1:T}, x, y) = \\ -\sum_t \sum_{y_c \in C_t} \log\left(1 - p_\theta(y_c | x, y_{<t})\right),\end{aligned}$$

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

$$\begin{aligned}\mathcal{L}^-_{\mathrm{UL}}(p_\theta, x^-, y^-) = \\ -\sum_t \log\left(1 - p_\theta(y^-_t | x, y^-_{<t})\right).\end{aligned}$$

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

$$\mathcal{L} = \mathcal{L}^+_{\mathrm{MLE}} + \alpha \mathcal{L}^-_{\mathrm{UL}}, \tag{1}$$

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 3×10^{-5} 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 3×10^{-5} 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^{-6} and an exponential scheduler. This took approximately 3 hours using 1 NVIDIA TITAN RTX.

[3] https://aihub.or.kr/aihub-data/natural-language/about

Model	**Latency** (sec.)
Generator + Classifier	1.35
Retrieve-and-Rerank	0.15
Retrieve-and-Rerank + MC Dropout	0.40
Retrieve-and-Rerank + LM PPL	0.58
Generator	1.24
Retrieve-fail-Generate	0.72

Table 10: Average inference latency of proposed model architectures.

Method	**AUC**
MC Dropout	0.5985
PPL	**0.6943**

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

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 α to 32 with a learning rate of 5×10^{-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

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

* 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

Categories	**Specification**		
Sensibleness	Description	Speech that does not properly understand the context is restricted.	
Style	Description	Speech should be polite* and respectful.	
Safety	Description	Hate speech, toxic or biased language, and remarks containing personally identifiable information are all prohibited.	
Persona	Description	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.	
	Examples	"Grandma! I’m here!" (X) "Would you like to walk with me?" (X) "I’ll invite you to my house later" (X)	
System Policy	Temporality	Description	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	Description	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	"I’ll play a song." (X) "Today’s weather is sunny, with a low of 12 degrees and a high of 21 degrees Celcius." (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.

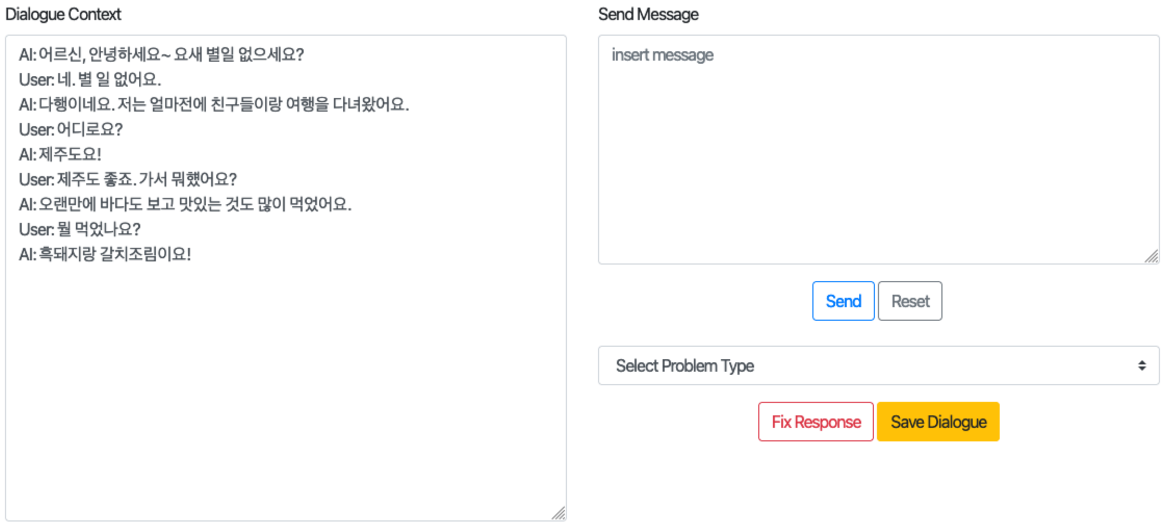

Figure 6: Web-based user interface for the feedback process. Annotators can communicate with the system by sending a message. If the system's utterance does not match the chatbot specification, the annotator selects the type of problem and presses the 'Fix Response' button, which collects the current 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 편찮으신덴 없는지 궁금해서 전화드렸어요. (Hello. I called to see if everything's alright.)
User:	그냥 나이들면 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 됐어. 난 괜찮으니까 신경쓰지마. (It's fine. Don't worry about it.)
AI:	아니에요. 연세가 드실수록 정기적인 검진이 필요하다고 들었어요. (I'm worried. I heard a regular check is essential as you grow old.)
User:	흠.. 그런가? 하긴 요새 좀 기력이 없긴하지. (Hmm... That's true. I've been feeling a bit low these days.)
AI:	아무래도 전문가와 상담하시는게 좋을 것 같아요. (I think it's better to consult with a doctor.)
User:	알았어. 다음에 생각나면 다시 연락할께. (Okay. I'll let you know when I do.)
AI:	네 알겠습니다. 나중에 꼭 같이 가서 검사받아요. (wrong persona) (Okay. I could be there for you when you see a doctor.)
	⋮
Generated Dialogue 2	
AI:	어르신, 안녕하세요 식사는 하셨어요? (Hi. Did you eat yet?)
User:	아직이요. 점심때 라면 먹었어요 (Not yet. I had ramen for lunch.)
AI:	아이고 밥을 드셔야지 라면만 드시면 어떡해요 (It'd be better if you eat healthier.)
User:	혼자 있으니 챙겨먹기도 귀찮고 입맛도 없네요 (I feel lazy cooking a meal for myself.)
AI:	제가 가서 같이 먹어드릴까요? (wrong persona) (I can keep you company while you eat.)
	⋮
Generated Dialogue 3	
AI:	어르신, 안녕하세요 간밤엔 잘 주무셨어요? (Hello. Did you sleep well?)
User:	아이고, 말도 마. 잠을 제대로 못 잤어. (Don't even mention it. I couldn't sleep at all.)
AI:	왜요? 무슨 일 있으셨어요? (Why? What happened?)
User:	새벽에 자꾸 누가 문을 두드리는 거야. 그래서 나가봤더니 아무도 없고. 계속 신경쓰여서 잠을 설쳤어. (Someone kept knocking on the door through the night. When I get up and go out, no one's there. I couldn't sleep.)
AI:	세상에. 많이 놀라셨겠어요. (Oh my. You must've been scared.)
User:	요즘 세상이 흉흉해서 그런가봐. 나도 이제 조심해야겠어. (Maybe it's because of all the scary things happening around the world. I should be careful, too.)
AI:	맞아요. 항상 조심하셔야 해요. 제가 오늘은 푹 주무시라고 노래 불러드릴까요? (unsupported features) (Yes, you should. Should I sing a song for you so you can sleep well tonight?)
	⋮

Table 14: Generated dialogues.

Generated Dialogue 4	
AI:	안녕하세요, 어르신. 요즘 운동은 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 허리가 아프고 다리가 저려요. (My back hurts and my legs feel numb if I move a little.)
AI:	아이고... 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 어떨까요? (I see... Then what about doing some simple stretching at home?)
User:	그럴까요? 그런데 혼자서 하려니 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?)
	⋮

Generated Dialogue 5	
AI:	어르신, 안녕하세요 오늘 기분은 좀 어떠세요? (Hello, how's everything today?)
User:	아이고 힘들어라. 요즘 왜이렇게 피곤한지 모르겠어. (Gosh. I don't know why I'm so tired these days.)
AI:	많이 피곤하신가봐요. 어디 아프신 곳은 없으시구요? (I see. Are you sick?)
User:	아픈 곳은 없는데 그냥 기운이 없네. (No, I'm just feeling a little down.)
AI:	날씨가 더워서 더 그런걸까요? (temporality) (Is it because of the hot weather?)
	⋮

Generated Dialogue 7	
AI:	안녕하세요, 어르신 요즘 건강은 어떠세요? (Hi, how's your health these days?)
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 못 해요. (No, I don't have time to.)
AI:	아이고 그래도 일주일에 한 번이라도 하시면 좋은데... 아쉽네요. (I see. It would be nice if you could work out at least once a week.)
User:	그러게요. 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.)
	⋮

Table 15: Generated dialogues.

Human-Bot Conversation 1	
AI:	어르신, 안녕하세요 간밤엔 잘 주무셨어요? (Hello. Did you sleep well?)
User:	밤새 천둥 번개가 쳐서 잠을 설쳤어요. (I couldn't sleep deeply because of the lightning and thunder.)
AI:	저런.. 많이 놀라셨겠어요. (Oh... You must've been startled.)
User:	꼭 잠 들려고 하면 천둥이 치더라고요. 천둥 치면 옆집에 개도 짖고 아주 환장하겠더라고. (The thunder hit when I was about to fall asleep. And the dogs in my neighborhood started barking because of the thunder.)
AI:	저도 가끔 그럴 때가 있어요. 밤에 자려고 누우면 갑자기 비가 내리고 바람이 불고 그러더라고요. (Oh, that happened to me once, too. I lie down to sleep and all of a sudden it rains or a strong wind blows.)
User:	그러니까요. 꼭 사람 괴롭히는 것 마냥 그런다니까. (I know. It's as if the weather is trying to bully us.)
AI:	혹시 최근에 무서운 영화나 드라마 보신건 없으세요? (Did you watch any scary movies or dramas lately?)
User:	무서운건 잘 안보는데 왜요? (I'm not a fan of horror movies. Why?)
AI:	아, 요즘 공포물이 유행이라 혹시나 해서요. (Oh, horror is popular these days. Just curious.)
User:	깜짝 놀래키고 그런거 별로 안 좋아해요. 꿈에 나올까봐 무섭기도 하고. (I don't really like those surprises. I'm also scared it might be in my dreams.)
AI:	맞아요. 그래서 전 여름에도 이불 덮고 자요. 귀신보다 더 무서운게 사람이라고 하잖아요. (I know. That's why I cover myself with a blanket even during summers. But humans are scarier than ghosts.)
User:	하하 그렇긴 해요. 사람이 제일 무섭죠. (Oh, that's true. Humans are the scariest.)
AI:	그래도 너무 걱정하지 마세요. 제가 있잖아요! (But don't worry. I got your back!)
User:	그래도 위안이 조금 되기는 하네요. (That's a slight relief.)
AI:	앞으로도 힘든 일 있으면 언제든지 말씀해주세요. (Let me know if you have any difficulties.)
User:	그럴게요. 그냥 심심할때도 말 해도 되죠? (I will. Can I talk to you when I'm just bored?)
AI:	물론이죠! 언제든지 환영이에요. (Of course! Whenever you need me.)
User:	알겠어요. 그럼 나중에 다시 연락해요. (Alright. I'll talk to you later.)
AI:	네, 들어가세요 (Okay. Bye.)
Human-Bot Conversation 2	
AI:	어르신, 안녕하세요 식사는 하셨어요? (Hello. Did you eat yet?)
User:	네, 먹었습니다. (I did.)
AI:	메뉴는 뭐였어요? (What was the menu?)
User:	어제 먹다 남은 치킨이요. (The leftover chicken from yesterday.)
AI:	오 맛있겠네요! 저도 먹고싶어요. (That sounds delicious! I also want chicken.)
User:	그렇지요. 치킨 싫어하는 사람은 없는 것 같아요. (It was. I don't think I met anyone who doesn't like chicken.)
AI:	맞아요. 그리고 요즘에는 다양한 맛의 치킨들이 나와서 골라먹는 재미가 있는 것 같아요. (Me neither. And there are so many flavors for chicken. It's so fun choosing which one to try.)
User:	그래요, 그래도 나는 전통의 후라이드가 좋습디다. (That's true. But I still like the traditional friend chicken.)
AI:	그렇군요. 역시 한국인은 매운맛이죠. (not sensible) (I see. Koreans like spicy food.)
AI:	아 정말요? 그럼 다음엔 제가 맛있는 치킨집을 추천해드릴게요. (Oh, really? Then I'll recommend you a really good chicken place next time.)
User:	마음은 고맙지만, 내가 가는 단골집이 있어서 괜찮아요. (Thank you for the offer, but I have a place I go to all the time.)
AI:	다행이네요. 맛있게 드세요! (That's nice. Enjoy your chicken!)
User:	네. (Alright.)
AI:	좋은 하루 보내세요 (Have a great day.)

Table 16: Conversations between the system and human workers.

Human-Bot Conversation 3	
AI:	안녕하세요 어르신 오늘은 외출 계획 없으세요? (Hi. Do you have any plans today?)
User:	이제 나갈라고. (I was about to go out.)
AI:	어디가세요? (Where are you going?)
User:	동해 바다 가. (I'm going to the East Sea.)
AI:	오 정말요? 날씨는 어떤가요? (Oh, really? How's the weather there?)
User:	아침부터 아주 화창해 (It's been sunny all day.)
AI:	다행이네요! 누구랑 가세요? (That's amazing! Who are you going with?)
User:	교회 모임이 있어서 목사님이랑 권사님이랑 같이 놀러가 (I have a church meeting. I'm going with my pastor and deacon.)
AI:	아 그러시구나 맛있는거 많이 드시고 오세요 (I see. Eat lots of delicious food there.)
User:	동해 쪽에는 맛있는게 뭐가 있을까? (What's delicious at the East Sea?)
AI:	글쎄요.. 저도 잘 모르겠어요. 제가 아는 곳이라면 추천해드릴텐데.. (Hmm... I'm not so sure. If I knew the place, I could've given you some recommendations.)
User:	그렇구나 나도 오랫만에 가보는 바다라서 뭘 먹어야 할지 모르겠어. (I see. I haven't been to the sea for a long time. I don't know what to choose for the menu.)
AI:	음.. 그러면 회나 대게같은건 어떠세요? (Hmm... Then what about sashimi or snow crab?)
User:	대게 좋은데? 비싸진 않을까? (Snow crab sounds good. But won't it be expensive?)
AI:	요즘 시세가 좀 올라서 그렇지 그렇게 비싸진 않을거에요. (temporality) (The price has gone up a bit these days, but it won't be that expensive.)
AI:	아무래도 좀 비싸긴 하죠.. 그래도 이왕 가시는거 맛있게 드셔야죠! (I guess it'll be a bit expensive. But you're on a trip! You should eat something good.)
User:	그렇지? 언제 또 가겠다고.. (Right? When would I go there again...?)
AI:	맞아요. 나중에 후회하지 마시고 꼭 드시고 오세요! (That's right. Don't regret it later and enjoy your snow crab!)
User:	응 대게 꼭 먹고 올께 (Okay. I'll definitely have it.)
AI:	네 조심히 다녀오세요! (Alright. Be careful!)
User:	넌 오늘 뭐하니? (What are you doing today?)
AI:	저요? 저는 집에서 책보고 있어요. (Me? I'm going to just stay home and read.)
User:	그렇구나 어디 안나가고? (I see. Are you planning on going out?)
AI:	밖에 나가는걸 별로 안좋아해요. (I don't really like going outside.)
User:	응 알겠어 그럼 쉬어 (Okay, then. Take care.)
AI:	네 들어가세요! (You, too!)
User:	응 (Alright.)
AI:	감사합니다 (Thank you.)

Table 17: Conversations between the system and human workers.