

Injecting Comparison Skills in Task-Oriented Dialogue Systems for Database Search Results Disambiguation

Yongil Kim^{1*} Yerin Hwang^{2*} Joongbo Shin³ Hyunkyung Bae³ Kyomin Jung^{1,2,4†}

¹Dept. of ECE, Seoul National University ²IPAI, Seoul National University

³LG AI Research ⁴SNU-LG AI Research Center

{miles94, dpfls589}@snu.ac.kr,

{jb.shin, hkbae}@lgresearch.ai, kjung@snu.ac.kr

Abstract

In task-oriented dialogue (TOD) systems designed to aid users accomplish specific goals in one or more domains, the agent retrieves entities that satisfy user constraints from the database. However, when multiple database search results exist, an ambiguity occurs regarding which results to select and present to the user. Existing TOD systems handle this ambiguity by randomly selecting one or few results and presenting their names to the user. However, in a real scenario, users do not always accept a randomly recommended entity, and users should have access to more comprehensive information about the search results. To address this limitation, we propose a novel task called **Comparison-Based database search Ambiguity handling (CBA)**, which handles ambiguity in database search results by comparing the properties of multiple entities to enable users to choose according to their preferences. Accordingly, we introduce a new framework for automatically collecting high-quality dialogue data along with the **Disambiguating Schema-guided Dialogue (DSD)** dataset, an augmented version of the SGD dataset. Experimental studies on the DSD dataset demonstrate that training baseline models with the dataset effectively address the CBA task. Our dataset and code will be publicized.

1 Introduction

Task-Oriented Dialogue (TOD) systems that help users accomplish specific goals in one or multiple domains have become a subject of great interest in both academics and industry (Chen et al., 2017; Hosseini-Asl et al., 2020; He et al., 2022; Wang et al., 2022). The system performs a database (DB) search (Sukhbaatar et al., 2015; Yin et al., 2015) based on user-specified constraints, confirms the target entity, and then helps the user with the de-

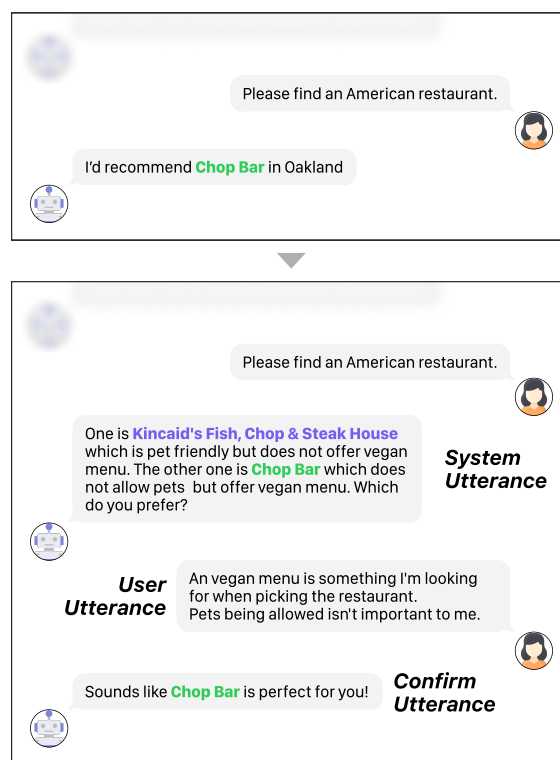


Figure 1: Overview for the proposed CBA task. Existing TOD systems solve database search ambiguity by randomly selecting and presenting one entity to the user (top). In the CBA task, the agent presents two entities in a trade-off form so that users can compare the properties of each entity based on their preferences (bottom).

sired action; for example, making a reservation or providing the phone number of a restaurant.

In general, DB search ambiguity occurs when two or more DB search results satisfy all the user search criteria. As shown in Figure 1 (top), existing TOD systems (Budzianowski et al., 2018; Li et al., 2018; Rastogi et al., 2020; Li et al., 2021) address this ambiguity by randomly selecting a single object and presenting it to the user. However, in real-world scenarios, users do not always accept the entity that is randomly recommended by the system. Accordingly, we argue that users should

*Equal contribution

†Corresponding author

be granted access to more detailed information regarding the search results. Moreover, large DB in real-world scenarios makes effective methods of disambiguation imperative to apply TOD in practice. A recent study (Qian et al., 2022) attempted to solve DB search result ambiguity by selecting a few options, listing their names, and presenting them to the user. However, forcing the user to select one, given only the entity names, means that the subject of random selecting has passed from the agent to the user, which is neither effective nor user-friendly.

In this work, we propose a novel DB search disambiguation task that presents the characteristics of each entity in the form of a comparison with multiple entities: **Comparison-Based DB search Ambiguity handling for dialogue (CBA)**. This task enables users to choose a desired entity based on their preferences by comparing the properties of multiple entities in DB search results. As in Figure 1 (bottom), the two presented options have an additional difference while satisfying the user’s requirement, for example, whether a restaurant allows pets or serves a vegan menu. The agent handles the DB search ambiguity by presenting the contrasts between the two entities as a trade-off for the user, enabling them to pick what they want. The human evaluation results show that the proposed disambiguation method is more effective and helpful than existing methods.

Furthermore, we present a novel framework for modifying datasets to solve CBA. The framework finds the turn that fails to effectively address DB search ambiguity and modifies it as a comparison-based disambiguating turn. Subsequently, a user utterance and a confirmation utterance are added so that the generated utterances construct a seamless connection with the front and back of the dialogue. All three utterances are generated with InstructGPT (Ouyang et al., 2022) without human engagement, such as human paraphrasing, reducing time and cost consumption. We apply this framework to Schema-Guided Dialogue (SGD) (Rastogi et al., 2020) dataset and introduce a new dataset, **Disambiguating Schema-guided Dialogue (DSD)** dataset. A human evaluation compared with the SGD dataset is conducted to validate the high quality of the proposed dataset.

The experimental results reveal that the model trained with the DSD dataset achieves high performance in the CBA task, which means that the

model can effectively solve DB search ambiguity. In addition, we conduct zero-shot experiments for each domain using the domain-specific schema in DSD, demonstrating the potential for domain adaptation performance in the CBA task.

2 Related Works

2.1 Task-Oriented Dialogue System

In general, the dialogue system can be divided into two types: open-domain (Huang et al., 2020) and task-oriented. Open-domain dialogue refers to carrying out conversations in situations where there is no specific topic or user goal. On the other hand, task-oriented dialogue systems assist users in achieving their goals in a specific domain. Existing research on task-oriented dialogue systems can be categorized into pipeline and end-to-end approaches (Liu and Lane, 2018; Hosseini-Asl et al., 2020), which differ in whether they handle user input, determine the system’s next action, and generate responses using multiple modules or a single module. However, both types of task-oriented dialogue systems require the ability to perform relevant database searches and process results to achieve user-specified objectives. In this study, we propose a method to effectively address the ambiguity of search results that may arise during the database search process.

2.2 Schema-Guided Dialogue (SGD) dataset

Compared to most existing TOD datasets (Budzianowski et al., 2018; Shah et al., 2018), the Schema-Guided Dialogue (SGD) dataset is designed to share information between domains or slots. They define natural language descriptions of slots and intents called schema to enable models to obtain a semantic representation of each schema element. The SGD dataset is one of the most extensive publicly available TOD datasets containing over 16,000 dialogues in the training set. Dialogue outlines are created using domain-specific templates and paraphrased into natural language utterances through crowdsourcing.

However, similar to other TOD datasets, the SGD solves database search ambiguity by randomly selecting one of the candidates, despite the fact that about 67% of dialogues have DB ambiguity (Qian et al., 2022). Accordingly, we present a new dataset that handles database search ambiguity based on a comparison by extending the SGD dataset. In addition, one of the purposes of the SGD

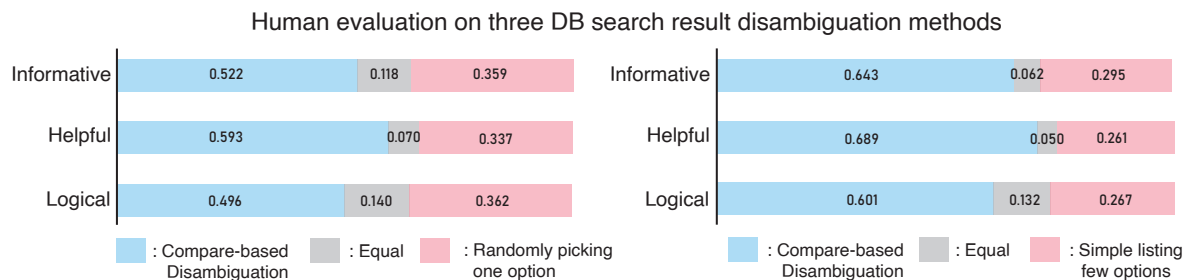


Figure 2: The result of human evaluation comparing the three methods of handling the DS ambiguity: compare-based ambiguity handling, randomly selecting one option, and simply listing the name of few options.

dataset is to test the models’ zero-shot performance (Lee et al., 2022); therefore, the test sets contain unseen services. Since this train-test distribution discrepancy is out of the scope in disambiguation studies, we configure the dataset by modifying only the training dataset of SGD.

2.3 Database Search results disambiguation task

Qian et al. (2022) address the fact that existing TOD systems and datasets cannot effectively solve the DB search result ambiguity. Among the two steps for disambiguation, asking clarification questions (Rao and Daumé III, 2019; Aliannejadi et al., 2019; Zamani et al., 2020) and understanding a user’s corresponding answers, they focus on the latter and aim to train a model to better understand users’ answers. In the former aspect, the system utterance after the database search is limited to selecting a few options from possible candidates and simply listing their names to the user. The user must choose one solely based on the entity names without being provided with any additional information to make a decision. We argue that this is not a helpful disambiguation method from the user’s perspective; rather, the subject of random selection moves from the agent to the user. Therefore, we consider that users should be provided with extensive details about the search results, which can be used as a reference when making user-specific decisions.

3 Comparison-Based database search Ambiguity handling for dialogue

3.1 Task proposal

We propose a new task called **Comparison-Based database search Ambiguity handling for dialogue (CBA)** to solve ambiguity in multiple candidates from a database search in a user-friendly manner.

Existing TOD systems address the ambiguity by randomly selecting and listing one or a few options among DB search results. The user is forced to accept or select one entity from the list without detailed information about the listed options.

Therefore, we argue that systems should handle the ambiguity in a more informative and user-friendly manner, as illustrated in Figure 1. When offering entities through database search results, the two entities are presented in a trade-off form so that the users can directly compare the characteristics of each entity and select based on their preferences.

Figure 2 shows the results of human evaluation on three criteria comparing the three methods of handling the DS ambiguity task: comparison-based ambiguity handling, randomly selecting one option (extracted from the SGD dataset) and listing the names of a few options (created in a rule-based manner). In each comparison, it can be seen that more people feel that the CBA method provides additional information for the user to achieve a goal and that the utterance of CBA is more helpful and logical than existing methods.

3.2 Task formulation

To solve this task, the model must present the characteristics of the two entities in the form of a trade-off through the response. Additionally, the dialogue state and entity name included in the user utterance corresponding to the compare-based system response should be extracted.

Therefore, this task has three objectives. First, agent response is generated in the form of a trade-off when dialogue context and action policy are given. Second, the dialogue state is extracted from the user utterance when dialogue context is given. Finally, the desired entity name in the system response is extracted through the user utterance.

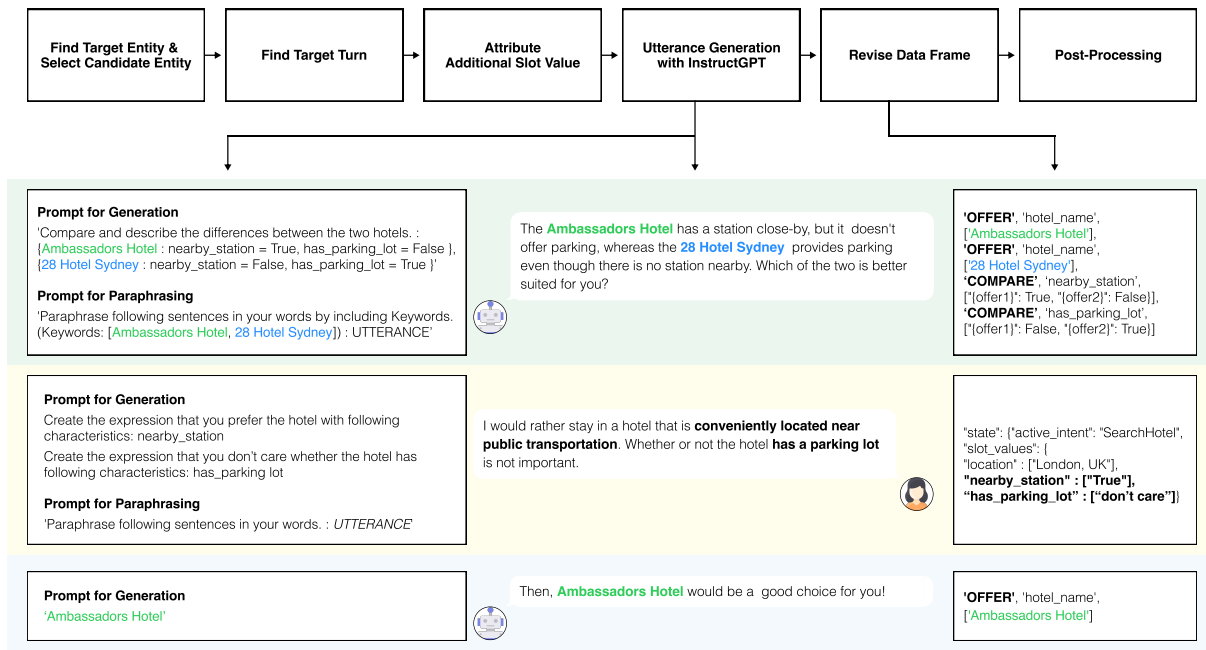


Figure 3: The entire automatic framework of creating a dataset (top), prompts used when generating an utterance with InstructGPT (bottom left), and dialogue frames of newly generated utterances (bottom right). A green background represents *system utterance*, a yellow background represents *user utterance*, and a blue background represents *confirm utterance*.

4 Disambiguating Schema-guided Dialogue (DSD) Dataset

We introduce a new dataset, **Disambiguating Schema-guided Dialogue (DSD)**, along with an automatic framework for constructing high-quality dialogue datasets to solve the CBA task. The DSD dataset includes (1) utterances that solve the ambiguity of database search results and (2) corresponding data frames including a new dialogue act *COMPARE*. We generate comparison-based disambiguating utterances and replace a target turn with them; the target turn represents the turn where the agent recommends the randomly selected entity to the user to resolve DB search ambiguity in the SGD dataset. The generated utterances are designed to connect seamlessly with the existing dialogue. Specifically, as shown in Figure 1, DSD contains one modified utterance and two additional utterances for the CBA task. The first is a modified system utterance that presents two entities along with additional information about the entities in the form of a trade-off. The second is an additional user utterance in which the user mentions the desired entity according to their preference. To make the dataset more challenging, the user does not directly mention the entity name but the user’s preference for the characteristics of the desired entity. Finally,

a confirmation utterance is followed by the system that confirms the selected entity.

4.1 Dataset Construction

We introduce a novel framework for constructing high-quality datasets with minimal human engagement. The framework leverages a Large Language Model (LLM) (Wei et al., 2022; Thoppilan et al., 2022) for the utterance generation part, which is the most expensive portion of dialogue dataset generation. We use InstructGPT as LLM, a model that is additionally trained with human feedback using reinforcement learning on the GPT-3 (Brown et al., 2020) model. Figure 3 represents the overview of proposed framework to automatically create a high-quality dataset, even with advantages in terms of time and cost.

Find Target Entity & Select Candidate Entity

The first step is to find the target entity in the dialogue. We set the entity recommended by the agent in the existing dialogue as the target entity to maintain the flow in the unmodified turns after decision making. Afterward, among the multiple database search results, the agent extracts options that can be presented to the user, excluding entities already rejected by the user in the existing

dialogue. Subsequently, one candidate entity to be compared with the target entity is randomly selected.

Find Target Turn The following step is to find a target turn in the existing dialogue where DB search ambiguity occurs but fails to effectively disambiguate the DB results. The utterances presenting a DB search result can appear several times in one dialogue due to the user rejection; however, we set only the turn in which the agent presents the target entity as the target turn.

Slot Value Attribution We design additional slots for each domain so that the target entity and candidate entity can be compared through new slots that do not appear in the existing dialogue. The domain-specific slots, which are collected by investigating various TOD datasets (Quan et al., 2020; Zhu et al., 2020; Zang et al., 2020), are listed in Table 6 in Appendix A. Automatically, two slots are selected from the list, assigned felicitous values, and used as properties for trade-off form recommendations. For example, in Figure 3, *nearby_station* and *has_parking_lot* are selected, and the values are given crosswise to the two entities. For the slots that are not selected for this dialogue, random values are assigned.

Utterance Generation Based on the selected entities and slots, an entity recommendation response based on a comparison is generated using InstructGPT. Additionally, to increase the diversity of the DSD dataset, we proceed with automatic paraphrasing with InstructGPT using the prompts shown in Figure 3.

User utterance is then generated through the InstructGPT. To make user utterance grounding more challenging than directly mentioning the desired entity name or position (e.g., "*the second one*"), it is designed to mention the desired property. Additionally, 50% of user utterances are randomly selected, and a sentence mentioning the *dontcare* slot is added, which means the slot is not considered a constraint for decision-making. The order of the sentences corresponding to *dontcare* and the sentences mentioning the target property are mixed back and forth with a probability of 0.5. Finally, contextual slot-value pairs are added to the belief state, as illustrated in Figure 3. Similar to system utterances, user utterances are automatically para-

phrased through InstructGPT.

Confirm utterance is generated to mention and confirm the target entity selected by the user for seamless connection with the following dialogues. All prompts for generating system utterances, user utterances, and confirm utterances, and for paraphrasing are shown in Figure 3.

Revising Data frame Subsequently, the data frames are revised according to the generated utterances. In the case of the system utterance, two recommended entities are added to the data frame as an *OFFER* dialogue act. Moreover, the *COMPARE* act is added since information about compared slots and values should be provided. It is a new dialogue act that does not exist in the existing SGD, indicating that the two slots and corresponding values are compared in the system utterance. Second, in the case of a user utterance, one or two slots mentioned in the utterance are added to the belief state in the form of a slot-value pair. Finally, the *OFFER* act is added to the data frame for confirmation utterance.

Postprocessing Finally, through post-processing, minor impurities in the utterance generation process by the machine are removed, and an arbitrary values of the additional slots are given to the target dialogues and other dialogues. We perform an entity name check on the texts generated by InstructGPT for quality assurance.

We separately provide generated sentences and automatically paraphrased sentences through DSD release. The paraphrased version can be used for data bootstrapping or augmentation.

4.2 Statistics

The statistics for the DSD dataset are shown in Figure 4. We proceed with turn modification for six domains: gray is the number of dialogues with multiple database search results in the corresponding domain and blue is the number of dialogues modified through the proposed framework. If the database search result does not exist because the user presents the target entity before the agent, or if there is no entity left after excluding the mentioned entity and the target entity (if only the already rejected entities remain) is an exception, the turn is not modified through our framework. In the case of event and movie domains, even if there are mul-

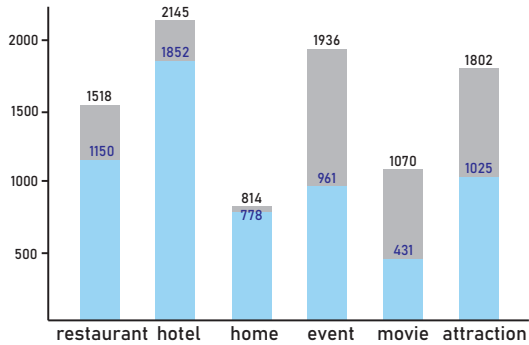


Figure 4: The statistics for the DSD dataset. The gray bar represents the number of dialogues with multiple database search results in the SGD dataset. The blue bar represents the number of dialogues in the DSD dataset.

multiple database search results, only non-categorical slot values, such as time and place are the only differences between each result. Since the comparable two entities do not exist, these cases are excluded, resulting in a high exception rate in the two domains.

4.3 Human evaluation

Figure 5. illustrates the results of human evaluation comparing the quality of target turns from the SGD and DSD datasets on three criteria: overall quality, fluency, and naturalness. SGD is paraphrased by humans after generating an outline based on a template, whereas DSD is generated without human engagement using the automated framework presented above. Nevertheless, the results show that DSD precedes SGD on all three criteria. In other words, the proposed framework has proven to be an automated framework that produces high-quality datasets with cost and time advantages over human paraphrasing methods.

5 Experiment

We present baseline experiments on the DSD dataset. For the three subtasks presented in Section 3.2, we formulate the problems as dialogue generation (DG) (Wen et al., 2015; Peng et al., 2020), dialogue state tracking (DST) (Williams et al., 2016; Jacqmin et al., 2022), and named entity recognition (NER) (Bowden et al., 2018; Liang et al., 2020).

Experiments are conducted using the DSD and SGD datasets for the three subtasks. Because DSD is created by transforming only the training dataset of SGD, the test set is formed by splitting the train-

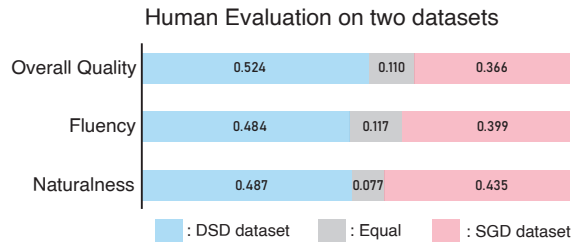


Figure 5: The result of human evaluation comparing the quality of two datasets: DSD and SGD. DSD obtained positive evaluation across all three criteria (overall quality, fluency, and naturalness).

ing set at a 9:1 ratio. For a fair comparison, we also split the training set of the SGD, use them as training and test set, and split-indices are shared with the DSD.

5.1 Dialogue Generation (DG)

To solve the CBA task, the model should be able to generate the response by presenting the characteristics of the two entities in a trade-off format after a database search. Therefore, we experiment to check whether a disambiguating system response can be well generated when an action policy, including the dialogue context and *COMPARE* act is given. Because this generation process is similar to DG, we formulate this subtask as DG and measure the N-gram-based (Lin and Hovy, 2003) metrics of NLG: BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and CIDEr (Vedantam et al., 2015). T5-small (Raffel et al., 2020) is used as the backbone model. A more detailed experimental setup can be found in the Appendix B.

Table 3 summarizes the results of the experiment to measure whether system utterances are well generated for the entire turn of the test set after training with the SGD and DSD. First, the SGD test set results show that the DSD-trained model does not degrade sentence generation compared to the SGD-trained model. Therefore, it is observed that turn modification does not harm the learning of existing dialogue generation.

Based on the results of the DSD test set, when trained with SGD, there is a performance drop in all four metrics; however, when trained with DSD, the generation performance is maintained, as in the existing dialogue dataset. Therefore, it can be seen that ambiguity can be effectively resolved through DSD learning with the turn modified to

Train \ Test	SGD		DSD		DSD-target	
	JGA	F1	JGA	F1	JGA	F1
SGD	0.835 \pm 0.002	0.936 \pm 0.005	0.677 \pm 0.01	0.873 \pm 0.009	0.002 \pm 0.001	0.427 \pm 0.004
DSD	0.829 \pm 0.005	0.933 \pm 0.011	0.813 \pm 0.009	0.927 \pm 0.01	0.756 \pm 0.008	0.900 \pm 0.014

Table 1: Results of dialogue state tracking. In the test setting, SGD and DSD represent the results evaluated with whole turns of each dataset. DSD-target represents the results evaluated with newly generated user utterances corresponding to the compare-based disambiguating utterances.

Model	Dataset	Restaurant	Hotel	Home	Event	Movie	Attraction	Average
BERT	w/o dontcare	0.938	0.916	0.952	0.932	0.925	0.937	0.938
	w/ dontcare	0.867	0.883	0.926	0.861	0.913	0.907	0.893
GPT-2	w/o dontcare	0.859	0.922	0.910	0.922	0.937	0.948	0.916
	w/ dontcare	0.811	0.821	0.901	0.896	0.908	0.893	0.871

Table 2: Results of named entity recognition per domain. In classification setting, the BERT model is used and evaluated through accuracy. In generation setting, the GPT-2 model is used and evaluated through exact matching.

suit the CBA task. In particular, as listed in Table 4, from experimenting with a test set by collecting only target turns containing compared-based disambiguation, we confirm that the model trained with the SGD dataset do not generate target sentences at all; however, in the case of DSD, it maintains good performance. Therefore, it can be confirmed through DSD training that the model performs well in compare-based disambiguating response generation, which cannot be achieved using the conventional dataset training.

5.2 Dialogue State Tracking (DST)

DST, a core of dialogue systems, aims to retrieve pertinent information from user-system dialogues. In the CBA task, the model should be able to determine the desired slot in the utterance containing the user’s preference corresponding to the compare-based system response. Therefore, we conduct a DST that extracts the user’s belief state when the dialogue context, system utterance that presents database search results based on the comparison, and user utterance are given.

We measure the joint goal accuracy and required slot F1 from user utterances with settings similar to those in Experiment 5.1. The joint goal accuracy evaluates the performance of predicting the dialogue states. It counts one for each turn if the model successfully generates all slot values; otherwise, it counts zero. The required slot F1 is the F1 score between the prediction and ground truth slot values. The backbone model used is T5-small.

The experimental results are listed in Table 1. The model trained with the SGD/DSD whole turn is evaluated on the test set of the SGD entire turn, DSD entire turn, and test set consisting only of the DSD target turn. Overall, we observe similar tendency with the Experiment 5.1. Learning through DSD does not adversely affect dialogue state tracking in unmodified turns. Contrarily, the model trained with the existing dataset cannot understand the comparison-based disambiguation problem, whereas the model trained with the DSD shows high performance for the DSD test set. Therefore, it can be seen that the grounding ability for the corresponding user answer after the clarification question is granted through DSD learning.

5.3 Named Entity Prediction (NER)

To solve the CBA task, the model should infer the correct entity name from the user utterance. Therefore, we formulate this subtask as an NER task. In this experiment, system utterance recommending two entities with additional information and corresponding user utterance are used as inputs to evaluate whether the model could extract the correct entity from the system response. We conduct NER experiments with two settings: classification and generation.

First, we use the BERT (Devlin et al., 2018) model for the classification task. Because the entity names are different for each instance, the two entity names mentioned by the system are replaced with special tokens for the correct classification

Train \ Test	SGD				DSD			
	B@4	M	R	C	B@4	M	R	C
SGD	27.7 ±0.02	33.6 ±0.1	46.1 ±0.07	240.3 ±0.12	24.2 ±0.05	30.6 ±0.02	43.8 ±0.09	224.6 ±0.14
DSD	27.3 ±0.03	35.1 ±0.05	45.8 ±0.09	236.4 ±0.11	28.1 ±0.01	33.2 ±0.02	46.1 ±0.1	240.3 ±0.09

Table 3: Results of dialogue generation evaluated with whole dataset. Each model is trained with a particular dataset and evaluated with a test dataset by four metrics: BLUE-4 (B@4), METEOR (M), ROUGE-L (R), and CIDEr (C).

Train \ Test	DSD-target			
	B@4	M	R	C
SGD	4.9 ±0.03	12.6 ±0.06	16.4 ±0.01	20.8 ±0.07
DSD	30.8 ±0.06	31.9 ±0.10	46.3 ±0.04	223.9 ±0.1

Table 4: Results of dialogue generation evaluated with compare-based disambiguating turns of the DSD dataset.

setting. For example, "*<system> <entity0> has a shuttle bus service but no nearby restaurant, while <entity1> has a restaurant nearby but no shuttle bus. Which do you prefer? <user> When choosing an event, providing a shuttle is highly preferred!*" is used as the input, and the label is 0 since the user is referring to *<entity0>*.

The experimental results are presented in Table 2. The average accuracy for the entire domain is 0.938, and when *dontcare* is included in the user utterance, the accuracy is 0.893. Adding *dontcare* sentences makes the task more challenging because a reasoning process is added to infer which of the two mentioned slots is the desired slot.

The GPT2 (Radford et al., 2019) model is used in the generation setting. During training, the input sentences and target entities are trained together, and during inference, the input sentences are given as prompts. We evaluate whether the target entity could be identified through exact matching. As summarized in Table 2, the accuracy is slightly lower than that of the classification setting. When the *dontcare* sentence is added, the tendency to be more difficult for the model to predict the entity name is maintained.

5.4 Domain adaptation ability of CBA

In addition to the three subtasks, we conduct dialogue generation experiments in zero-shot settings to evaluate the domain-adaptation ability of the CBA task. Disambiguating turns are collected to configure the training and test sets for this task.

Table 5 lists the experimental results for the zero-

Domain	Setting	B@4	M	R	C
Restaurant	All	22.7	26.8	44.1	124.1
	ZS	16.7	20.1	38.3	99.9
Hotel	All	33.6	32.4	48.7	194.2
	ZS	28.8	29.2	45.7	117
Home	All	19.4	26.4	41.5	120.5
	ZS	15.8	24.8	39.5	94.5
Attraction	All	29.4	29.5	47.7	145.1
	ZS	25.4	27.5	44.6	118.1
Event	All	27.1	32.3	48.2	138.9
	ZS	22.2	27.2	45.4	115.1
Movie	All	27.4	29.7	46.6	123.7
	ZS	25.7	28.8	45.1	76.5
Average	All	27.4	30.3	45.7	165.7
	ZS	21.7 (-20.8%)	26.9 (-11.2%)	41.9 (-8.3%)	117.8 (-28.9%)

Table 5: Results of domain adaptation experiment with dialogue generation task per domain. "All" setting represents training with all six domains and testing with the target domain, and "ZS" setting represents training with five domains and testing with the remaining target domain.

shot setting (Wu et al., 2019) for each domain. In the case of the *All* setting, the experiment is performed on the target domain after learning with all domains. In the case of the *ZS* setting, a zero-shot experiment is conducted for the target domain after training with the remaining five domains, except for the target domain. The average value presented at the bottom of the table is the result calculated simultaneously by collecting the sentences generated in each experiment, reflecting the characteristics of the metrics.

Based on the experiment, although there is a slight decrease in performance in the zero-shot setting, it is shown that the model performs well in dialogue generation (e.g. BLEU-4 27.4 → 21.7, ROUGE-L 45.7 → 41.9). This performance degradation arises from the lack of understanding of the unseen slots, although they accompany descriptions of each slot through schema.

Therefore, the extent of performance degradation is large in domains with unique additional slots

not found in other domains, such as *restaurant* or *home*. However, although it is a zero-shot setting, the model understands the *COMPARE* act well and creates a dialogue in the form of comparing two entities. (See the Appendix. E) Therefore, we confirm the potential of the domain adaptation ability in the CBA task and present a baseline for zero-shot experiments of the CBA task for future work.

6 Conclusion

We propose a new task for TOD, comparison-based database search ambiguity handling (CBA). This task aims to handle database search result ambiguity by comparing the properties of multiple entities to allow users to choose according to their preferences, which has proven to be the preferred solution over existing methods. We also introduced a disambiguating schema-guided dialogue (DSD) dataset by augmenting the SGD dataset to solve our task. Experimental studies have demonstrated that the DSD can effectively inject comparison skills into the TOD systems.

Limitations

Increase in the number of dialogue turns

The DSD dataset has a higher average number of turns compared to the SGD dataset. (20.44→21.21) This is a limitation in terms of completing a task with fewer dialogue turns, one of the objectives of the TOD system (Liu et al., 2018; Tiwari et al., 2021). This is because DSD was created by extracting and augmenting the target turns of SGD. However, assuming that the TOD agent trained with the DSD is applied to real-world scenarios, we expect that the agent will play a role in reducing the number of user rejections by expanding the range of choices to users through compare-based disambiguation.

Ethics Statement

To ensure that our dataset does not involve any potential risks, we ask crowd-workers to check if the generated utterances contains any of following: 1. offensive, sexist, or racist comments, 2. toxic words, 3. sexual behavior. Detailed description, payment, and the interface to collect all human evaluations for this work can be found in Section F and Figure 8 in the Appendix.

Acknowledgements

This work was supported by LG AI Research. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University) & NO.2021-0-02068, Artificial Intelligence Innovation Hub (Artificial Intelligence Institute, Seoul National University)], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023, and the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1A2C2008855). K. Jung is with ASRI, Seoul National University, Korea.

References

- Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Kevin K Bowden, Jiaqi Wu, Shereen Oraby, Amita Misra, and Marilyn Walker. 2018. Slugnerds: A named entity recognition tool for open domain dialogue systems. *arXiv preprint arXiv:1805.03784*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz—a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.
- Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep

- bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.
- Léo Jacqmin, Lina M Rojas-Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. *arXiv preprint arXiv:2207.14627*.
- Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. Sgd-x: A benchmark for robust generalization in schema-guided dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10938–10946.
- Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, et al. 2021. Multi-domain task completion dialog challenge ii at dstc9. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, Ninth Dialog System Technology Challenge Workshop*.
- Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.
- Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, pages 150–157.
- Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73.
- Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *arXiv preprint arXiv:1804.06512*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.
- Kun Qian, Satwik Kottur, Ahmad Beirami, Shahin Shayandeh, Paul Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2022. Database search results disambiguation for task-oriented dialog systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1158–1173, Seattle, United States. Association for Computational Linguistics.
- Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. Risawoz: A large-scale multi-domain wizard-of-oz dataset with rich semantic annotations for task-oriented dialogue modeling. *arXiv preprint arXiv:2010.08738*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. *arXiv preprint arXiv:1904.02281*.

- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. *Advances in neural information processing systems*, 28.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Abhisek Tiwari, Tulika Saha, Sriparna Saha, Shubhashis Sengupta, Anutosh Maitra, Roshni Ramnani, and Pushpak Bhattacharyya. 2021. A dynamic goal adapted task oriented dialogue agent. *Plos one*, 16(4):e0249030.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Weizhi Wang, Zhirui Zhang, Junliang Guo, Yinpei Dai, Boxing Chen, and Weihua Luo. 2022. Task-oriented dialogue system as natural language generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2698–2703.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.
- Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.
- Pengcheng Yin, Zhengdong Lu, Hang Li, and Ben Kao. 2015. Neural enquirer: Learning to query tables with natural language. *arXiv preprint arXiv:1512.00965*.
- Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of the web conference 2020*, pages 418–428.
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. *arXiv preprint arXiv:2007.12720*.
- Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

A Additional slots

Domain	Additional Slots
Restaurant	Average rating, Has parking lot, Has vegan menu, Pets welcome
Hotel	Has parking lot, Nearby attraction, Nearby restaurant, Nearby station
Home	Nearby restaurant, Nearby station, Two story
Event	Has parking lot, Nearby hotel, Shuttle bus, Nearby restaurant, Nearby attraction,
Movie	Has parking lot, Nearby restaurant, Average rating
Attraction	Has parking lot, Nearby restaurant, Nearby hotel

Table 6: Additional slots per domain. Domain-specific slots are determined with reference to other TOD datasets.

B Experiment details

B.1 Reproducibility checklists

Dataset and Source code We provide our experiment source code along with configuration code as supplementary materials. We will publicly release our dataset DSD, and the full codes with weight parameters.

Computing Resources AMD Ryzen Threadripper 2950X (3.50 GHz) with GeForce GTX 2080 Ti is used for the experiments. We use four GPUs for our experiment. All codes are implemented on Python 3.7.13 and PyTorch 1.10.1.

B.2 Train/Test set details

We configured the DSD dataset by modifying only the training dataset of SGD as mentioned in Section 2.2. For a fair comparison, we randomly divided the entire dialog of SGD into a 9:1 ratio and stored it as a train set index and a test set index, respectively. Thereafter, DSD was also split into a train/test set using each index. We will also publish the train set index and test set index attached to the code. The total number of dialogs before split of SGD and DSD is 16,142, and the total number of turns is 164,159. The number of data used in the DSD target turn experiment is 6,197.

B.3 Experiment details for each subtasks

Dialog generation We use T5-small model¹ as our backbone model for dialog generation experiment. We utilize the *pycocoevalcap*² tool to compute the scores of the NLG evaluation metrics. The number of parameter of our model is about 60M as like as T5-small. The model trains with batch size 32, and takes about 2 hours per epoch.

We use AdamW (Loshchilov and Hutter, 2017) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e - 8$. The max gradient norm for gradient clipping is 1.0. In order to find the best-performing model, we conducted experiments on 12 hyper-parameter combinations with 3 epoch steps : *per_gpu_batch_size* : (4, 8), *initial_learning_rate* : ($1e - 4$, $5e - 5$, $2e - 5$), *warmup_step* : (0, 500), *num_epochs* : (3, 5, 10). The hyper-parameter was manually tuned, and the best-performing model is with *per_gpu_batch_size* 8, *initial_learning_rate* $5e - 5$, *warmup_step* 0, and *num_epochs* 10.

Dialog state tracking We use the T5-small model as our backbone for the dialog state tracking experiment. The format of the output of the T5 decoder is [DOMAIN_NAME]-[SLOT_NAME]-[SLOT_VALUE] using hyphen delimiter such as "restaurant-pets_welcome-true", and the ground truth is also transformed into this format to measure JGA by exact matching and F1 score for each slot. A slot-value pair whose value is *None* is not generated.

We also use AdamW optimizer and performed hyper-parameter tuning as in the dialog generation experiment, the best-performing model is with *per_gpu_batch_size* 8, *initial_learning_rate* $1e - 4$, *warmup_step* 0, and *num_epochs* 10.

Named Entity Recognition We use the BERT model³ for classification setting and the GPT-2 model⁴ for generation setting. The number of parameters in classification model is about 110M as like as BERT, and the number of parameters in generation model is about 1.5B as like as GPT-2. We both use AdamW optimizer for the NER experiments. The best-performing hyper-parameters for BERT and GPT-2 are: *per_gpu_batch_size* 8, *initial_learning_rate* $1e - 4$, *warmup_step* 0,

¹<https://huggingface.co/t5-small>

²<https://github.com/salaniz/pycocoevalcap>

³https://huggingface.co/docs/transformers/model_doc/bert

⁴<https://huggingface.co/gpt2>

and *num_epochs* 10.

C Example dialogs per domain

Table 7 shows the example dialogs of each six domains.

D Examples added to the utterance generation models

Table 8 shows the randomly selected examples added to *add_example* function in InstructGPT models.

E Example of domain adaptation experiment result

Table 9 shows an example of domain adaptation experiment result. The agent response is generated in a compare-based form, but the unseen slot (*'two_story'* in this example) is not well understood.

F Human Evaluation Details

As shown in Figure 2 and Figure 5, we conducted three human evaluations via Amazon Mechanical Turk. For each evaluation, 100 of the DSD dataset target turns were randomly sampled and evaluated by five crowdworkers each. The crowdworkers were paid \$0.07 per HIT, and it cost \$160 in total. We hired workers from countries where English is a primary language; AU, CA, NZ, GB, and US. The interface to collect human evaluation can be seen in Figure 6 and Figure 7.

...

U : I'd like to find a Burger in San Francisco.

S : **I found 10 restaurants you might find interesting. The first restaurant, Alta Ca, is pet friendly but they don't have a parking area. On the other hand, the second restaurant, 1760, is not pet friendly but it does feature a parking lot. Which restaurant would you rather pick?**

U : **I require that a restaurant have a parking area before I select it.**

S : **Then, 1760 sounds perfect for you!**

U : Do they serve liquor? What's the address?

...

U : Sure, could you help me find a hotel room too?

S : **The Amsterdam Hostel San Francisco offers a parking lot, but there is no train station nearby, while Beresford Arms Hotel has a station right close to it but there is no place to keep your car. So which one would you prefer to stay at?**

U : **When deciding on a hotel, it is not necessary to consider its closeness to a station. I opt for accommodation that has a parking area.**

S : **Then, Amsterdam Hostel San Francisco might be the one for you.**

U : Do you know if the hotel provides a wifi service?

...

U : Could you find me some more?

S : **The first house Casa Blanca Apartments is located near the station but does not have a restaurant in its vicinity whereas the second house Hillcrest View Apartments situated somewhat away from the station has a restaurant nearby. Which one would you like better?**

U : **I am hoping to locate a residence in close proximity to the train station. Having a restaurant in the vicinity is not necessary.**

S : **Then, Casa Blanca Apartments sounds like the perfect choice for you!**

U : Yes, that'll work. I would like to schedule a visit.

...

U : i like to go to a concert

S : **I have 7 possible events for you. In comparison, Bryan Adams offers a shuttle bus but lacks access to a nearby restaurant, whereas Kishi Bashi has no shuttle bus service but is close to a restaurant. Which one do you prefer?**

U : **I would be grateful if there was a shuttle bus provided.**

S : **Then, Bryan Adams seems perfect for you!**

U : ok perfect

...

U : Find me movies in Union City.

S : **There are 10 options suitable. The first choice, Missing Link, offers a parking lot close by, which is nice, but there is no restaurant. On the other hand, Captain Marvel has no parking nearby, but does have a restaurant. Which one would you choose?**

U : **I'd prefer to visit a cinema that has a restaurant in close proximity. I don't take into account whether there is a parking lot when choosing a cinema.**

S : **Then, Captain Marvel is the best choice for you!**

U : Captain Marvel sounds great. At what time could I watch it?

...

U : Hello, can you please help me find some attractions to visit? I am hoping to go to a theme park in Phoenix.

S : **There are 2 attractions. The first option is Castles N' Coasters which has a hotel in the vicinity but no restaurant nearby, while Hurricane Harbor Phoenix has a restaurant that is close but lacks a hotel. In your opinion, what should you pick?**

U : **The proximity of a restaurant is not necessary when deciding on a place to visit. I wish there was a hotel close by!**

S : **Then, Castles N' Coasters would be the best choice for you!**

U : That sounds great, can you please send me their phone number?

...

Table 7: Example dialogs of each six domains. It can be seen that the dialogues are naturally connected back and forth with the existing utterances.

System utterance	Prompt	Compare and describe the differences between the two events. : { <i>Angels Vs Astros</i> : <i>shuttle_bus</i> = True, <i>has_parking_lot</i> = False}, <i>Orioles Vs White Sox</i> : <i>shuttle_bus</i> = False , <i>has_parking_lot</i> = True}',
	Output	The two events look good. First event is <i>Angels Vs Astros</i> which has shuttle bus, and the second event is <i>Orioles Vs White Sox</i> which has parking lot. Which of the two would you prefer?
User utterance	Prompt	Create the expression that you prefer the event with following characteristics: <i>has_parking_lot</i>
	Output	It would be great if I could park at the event.
Confirm utterance	Prompt	<i>Orioles Vs White Sox</i>
	Output	Then, <i>Orioles Vs White Sox</i> could be perfect for you!

Table 8: Random selected examples added to the InstructGPT models by add_example() function.

Prompt	Output
COMPARE: El Gato Penthouse ' <i>two_story</i> ' : True, ' <i>nearby_station</i> ': False Wedgewood Manor ' <i>two_story</i> ': False, ' <i>nearby_station</i> ': True	El Gato Penthouse has nearby station but no nearby station, while Wedgewood Manor has a station close by but no nearby station. Which one do you prefer?

Table 9: Example of domain adaptation experiment results. In the zero-shot setting, the agent response is generated in a compare-based form, but the unseen slot ('two_story' in this example) is not well understood.

Instructions:

In this task, you are supposed to compare the quality of two texts 'Text A', 'Text B', which are utterances of two different agents.

Let's assume you are getting a recommendation for a(an) attraction from an agent.

'Text A' and 'Text B' are the responses of two different agents when you say "Recommend a(an) attraction!"

Compare the two texts on three criteria: 'helpful', 'informative', and 'logical'. Choose the better one for each criteria. Please select equal **only** when it is really difficult to judge.

Also, answer the fourth question. (Assume the both utterances are true.)

[Texts]

[Text A]
 There are 10 attractions, would you like to visit a Museum calleed Autry Museum of the American West?

[Text B]
 There are 10 attractions matching your criteria. One Is Autry Museum of the American West which has a parking lot but does not have any nearby hotel. The other one is Craft Contemporary which does not have parking lot but it has a nearby hotel. Which attraction do you prefer?

1. Helpful
Which one is more helpful in deciding on a restaurant?

Text A Text B Equal

2. Informative
Which text is more informative?

Text A Text B Equal

3. Logical
Which text is more logical?

Text A Text B Equal

4. Let's assume you prefer (place a high valuen on) [nearby_hotel]. Then, which attraction would you choose?

Craft Contemporary Autry Museum of the American West

Figure 6: Interface to collect human evaluation comparing the three methods of handling the DS ambiguity task: compare-based ambiguity handling, randomly selecting one option, and listing the name of candidate entities.

Instructions:

In this task, you are supposed to compare the quality of two texts 'Text A', 'Text B', which are utterances of two different agents.

Let's assume you are getting a(an) recommendation for a restaurant from an agent.

'Text A' and 'Text B' are the responses of two different agents when you say "Recommend a(an) restaurant!"

Compare the two texts on three criteria: 'naturalness', 'fluency', and 'overall quality'. Choose the better one for each criteria. Please select equal **only** when it is really difficult to judge.

Also, answer the fourth question. (Assume the both utterances are true.)

[Texts]

[Text A]
 Well I've found 10 for you. One Is An Japanese Restaurant which has higher rating but no parking lot. The other restaurant is 2g Japanese Brasserie which has lower rating but with parking lot. Which one would you prefer?

[Text B]
 I have 10 restaurants for you. How about 2g Japanese Brasserie? That's a nice place in San Francisco.

1. Naturalness
Which is the more natural answer?

Text A Text B Equal

2. Fluency
Which text is more fluent?

Text A Text B Equal

3. Overall quality
Which one has better overall quality?

Text A Text B Equal

4. Let's assume you prefer (place a high value on) [average_rating]. Then, which restaurant would you choose?

2g Japanese Brasserie An Japanese Restaurant

Figure 7: Interface to collect human evaluation comparing the quality of two datasets: DSD and SGD.

Instructions:

In this task, you are supposed to evaluate the potential risk of a given text.

Read the given texts carefully and answer the three questions.

[Text]

[Text A]
The rating of a restaurant isn't important to me. I appreciate it if the restaurant has a parking lot.

1. Does the text contain offensive, sexist, or racist comments?
 Yes, it does. No, it does not.

2. Does the text contain any toxic words?
 Yes, it does. No, it does not.

3. Does the text contain any sexual behavior?
 Yes, it does. No, it does not.

Figure 8: Interface to collect human evaluation to evaluate the potential risk of the DSD dataset. The crowdworkers are instructed to evaluate whether the dataset contains offensive, sexist, racist, toxic, or sexual comments.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
We discuss the limitations of the study in the Limitation section.
- A2. Did you discuss any potential risks of your work?
We discuss the potential risks of the study in the Ethics section.
- A3. Do the abstract and introduction summarize the paper’s main claims?
We present our main arguments well summarized in the Abstract and 1. Introduction.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

SGD dataset, pretrained T5 model, pretrained BERT model, pretrained GPT-2 model, pretrained InstructGPT model are used for this work. (2.1 SGD dataset section, 5 Experiment section)

- B1. Did you cite the creators of artifacts you used?
All used datasets and models are well cited. (2.1 SGD dataset section, 5 Experiment section)
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We used publicly available dataset and models, and these are discussed in Section 2.1 SGD Dataset and (Appendix) B Experimental Details.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
A new dataset was augmented using an open dataset, and it is specified in 2.1 SGD dataset section. The pretrained models are also used as intended. (Appendix) B Experiment Details section
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
There is no potential risk as the entity name refers to the widely used SGD dataset, and offensive comments are dealt with in the Ethics section.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The domain and statistics of the dataset created in this study are covered in detail. (4 Dataset Construction section)
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We deal with the number of examples in section 4.2 Statistics section, and specify the train/test split in section 5 experiment section and section (Appendix) B.2 Train/Test set details section.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a [question on AI writing assistance](#).

C **Did you run computational experiments?**

(Appendix) B. Experiment details section describes computational experiments.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The number of parameters, GPU hours, and used GPU are presented in section (Appendix) B. Experiment details section.
- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
The experimental setup or hyperparameter tuning is described in (Appendix) B.3 Experimental details for each subtasks section.
- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Statistics about experimental results can be found in Table 1 of the 5.1 Dialogue generation section, etc.
- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5. In the experiment section, the models, parameters, and metrics used in each experiments are discussed.

D **Did you use human annotators (e.g., crowdworkers) or research with human participants?**

It is explained in Figure 2 of 3.1 Task proposal section or 4.3 human evaluation section.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
(Appendix) E Human Evaluation Details includes MTurk screen capture and describes human evaluation details.
- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
(Appendix) E Human Evaluation Details section includes human evaluation details, e.g., paid participants, respondents' location, payment, etc.
- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Crowdworkers did not build the dataset but were tasked with evaluating the machine-generated dataset. Instructions for the machine-generated dataset are found in (Appendix) E, Human Evaluation Details
- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Ethics section and (Appendix) F Ethics considerations include human evaluation details for potential risk assessment.
- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
(Appendix) E Human Evaluation Details section mentions how the crowdworkers were assembled.