InstructDial: Improving Zero and Few-shot Generalization in Dialogue through Instruction Tuning

Prakhar Gupta♣ Cathy Jiao♣ Yi-Ting Yeh♣ Shikib Mehri♣ Maxine Eskenazi♣ Jeffrey P. Bigham♣ Human-Computer Interaction Institute, Carnegie Mellon University

Abstract

Instruction tuning is an emergent paradigm in NLP wherein natural language instructions are leveraged with language models to induce zero-shot performance on unseen tasks. Dialogue is an especially interesting area in which to explore instruction tuning because dialogue systems perform multiple tasks related to language (e.g., natural language understanding and generation, domain-specific interaction), yet instruction tuning has not been systematically explored for dialogue-related tasks. We introduce INSTRUCTDIAL, an instruction tuning framework for dialogue, which consists of a repository of 48 diverse dialogue tasks in a unified text-to-text format created from 59 openly available dialogue datasets. We explore cross-task generalization ability on models tuned on INSTRUCTDIAL across diverse dialogue tasks. Our analysis reveals that INSTRUCTDIAL enables good zero-shot performance on unseen datasets and tasks such as dialogue evaluation and intent detection, and even better performance in a few-shot setting. To ensure that models adhere to instructions, we introduce novel meta-tasks. We establish benchmark zero-shot and few-shot performance of models trained using the proposed framework on multiple dialogue tasks.

1 Introduction

Pretrained large language models (LLMs) (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020) are not only few-shot learners, but can also perform numerous language tasks without the need for fine-tuning. However, LLMs are expensive to train and test. Instruction tuning has emerged as a tool for directly inducing zero-shot generalization on unseen tasks in language models by using natural language instructions (Mishra et al., 2021; Sanh et al., 2022; Wei et al., 2022; Ouyang et al., 2022). Natural language instructions can contain components such as task definitions, examples, and prompts which allows them to be customized for multitask learning. Instruction tuning enables developers, practitioners, and even non-expert users to leverage language models for novel tasks by specifying them through natural language, without the need for large training datasets. Furthermore, instruction tuning can work for models that are significantly smaller than LLMs (Mishra et al., 2021; Sanh et al., 2022), making them more practical and affordable.

Most recent work (Mishra et al., 2021; Sanh et al., 2022; Wei et al., 2022) on instruction tuning has focused on general NLP tasks such as paraphrase detection and reading comprehension, but not specifically on dialogue. While some work...
such as (Wang et al., 2022a) include a few dialogue tasks, those tasks are collected through crowdsourcing and do not provide good coverage of dialogue tasks and domains. No prior work has examined how training a model on a wide range of dialogue tasks with a variety of instructions may affect a system’s ability to perform on both core dialogue tasks such as intent detection and response generation, and domain-specific tasks such as emotion classification. In this work, we introduce INSTUCTDIAL, a framework for instruction tuning on dialogue tasks. We provide a large curated collection of 59 dialogue datasets and 48 tasks, benchmark models, and a suite of metrics for testing the zero-shot and few-shot capabilities of the models. INSTUCTDIAL consists of multiple dialogue tasks converted into a text-to-text format (Figure 1). These dialogue tasks cover generation, classification, and evaluation for both task-oriented and open-ended settings and are drawn from different domains (Figure 2).

Instruction tuned models may ignore instructions and attain good performance with irrelevant prompts (Webson and Pavlick, 2021), without actually following user’s instructions. We address this issue in two ways: (1) we train the models with a variety of outputs given the same input context by creating multiple task formulations, and (2) we propose two instruction-specific meta-tasks (e.g., select an instruction that matches with an input-output pair) to encourage models to adhere to the instructions.

The main contributions of this work are:

- We introduce INSTUCTDIAL, a framework to systematically investigate instruction tuning for dialogue on a large collection of dialogue datasets (59 datasets) and tasks (48 tasks). Our framework is open-sourced and allows easy incorporation and configuration of new datasets and tasks.
- We show that instruction tuning models enhance zero-shot and few-shot performance on a variety of different dialogue tasks.
- We provide various analyses and establish baseline and upper bound performance for multiple tasks. We also provide integration of various task-specific dialogue metrics.

Our experiments reveal further room for improvement on issues such as sensitivity to instruction wording and task interference. We hope that INSTUCTDIAL will facilitate further progress on instruction tuning for dialogue tasks.

2 Related Work

Pre-training and Multi-Task learning in Dialogue Large-scale transformer models (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020) pre-trained on massive text corpora have brought substantial performance improvements in natural language processing. Similar trends have occurred in the dialogue domain, where models such as DialoGPT (Zhang et al., 2020), Blenderbot (Roller et al., 2021) and PLATO (Bao et al., 2021) trained on sources such as Reddit or Weibo, or on human-annotated datasets show great capabilities in carrying open-domain conversations. Large-scale pretraining has also shown success in task-oriented dialogue (TOD). (Budzianowski and Vulić, 2019; Hosseini-Asl et al., 2020; Ham et al., 2020; Lin et al., 2020; Yang et al., 2021) utilized pretrained language models such as GPT-2 (Radford et al., 2019) to perform TOD tasks such as language generation or act prediction. Similarly, BERT-type pretrained models have been used for language understanding in TOD tasks (Wu et al., 2020a; Mi et al., 2021b). Several of these works have shown improved performance by performing multi-task learning over multiple tasks (Hosseini-Asl et al., 2020; Liu et al., 2022; Su et al., 2022a). Multi-task pretraining also helps models learn good few-shot capabilities (Wu et al., 2020a; Peng et al., 2021). Our work covers both open-domain and TOD tasks and goes beyond multi-tasking as it incorporates additional structure of the tasks such as task definitions and constraints.

Instruction Tuning Constructing natural language prompts to perform NLP tasks is an active area of research (Schick and Schütze, 2021; Liu et al., 2021a). However, prompts are generally short and do not generalize well to reformulations and new tasks. Instruction tuning is a paradigm where models are trained on a variety of tasks with natural language instructions. Going beyond multi-task training, these approaches show better generalization to unseen tasks when prompted with a few examples (Bragg et al., 2021; Min et al., 2022a,b) or language definitions and constraints (Weller et al., 2020; Zhong et al., 2021b; Xu et al., 2022). Prompt-Source (Sanh et al., 2022), FLAN (Wei et al., 2022) and NATURAL INSTRUCTIONS (Mishra et al., 2021; Wang et al., 2022b) collected instructions and datasets for a variety of general NLP tasks.

GPT3-Instruct model (Ouyang et al., 2022) is tuned on a dataset of rankings of model outputs and
was trained using human feedback, but it is expensive to train and test. Instead, our work is tailored to dialogue tasks and incorporates numerous dialogue datasets, tasks, and benchmarks. We show that models trained on collections such as PromptSource are complementary to instruction tuning on dialogue. For dialogue tasks, Madotto et al. (2021) explored prompt-based few-shot learning for dialogue, but without any fine-tuning. Mi et al. (2021a) designed task-specific instructions for TOD tasks that improved few-shot performance on several tasks. Our work covers a far greater variety of dialogue domains and datasets in comparison.

3 Methodology

In this section, we first discuss instruction tuning setup. Next, we discuss the taxonomy of dialogue tasks, the task meta-information schema, and discuss how dialogue datasets and tasks are mapped into our schema. Finally, we discuss model training and fine-tuning details.

3.1 Instruction Tuning Background

A supervised setup for a dialogue task $t$ consists of training instances $d^t_{\text{train}} \ni (x_i, y_i)$, where $x_i$ and $y_i$ are an input-output pair. A model $M$ is trained on $d^t_{\text{train}}$ and tested on $d^t_{\text{test}}$. In a cross-task setup, the model $M$ is tested on test instances $d^\hat{t}_{\text{test}}$ of an unseen task $\hat{t}$. In instruction tuning, the model $M$ is provided additional signal or meta information about the task. The meta information can consist of prompts, task definitions, constraints, and examples, and guides the model $M$ towards the expected output space of the unseen task $\hat{t}$.

3.2 Task Collection

We adopt the definition of a task from Sanh et al. (2022), which defined a task as "a general NLP ability that is tested by a group of specific datasets". In INSTRUCTDIAL, each task is created from one or more existing open-access dialogue datasets. Figure 2 shows the taxonomy of dialogue tasks in INSTRUCTDIAL, and Table 9 shows the list of datasets used in each task. In our taxonomy, Classification tasks consist of tasks such as intent classification with a set of predefined output classes. Generation tasks consist of tasks such as open-domain, task-oriented, controlled, and grounded response generation, and summarization. Evaluation tasks consist of response selection in addition to relevance and rating prediction tasks. Edit tasks involve editing a corrupted dialogue response into a coherent response. Corrupted responses are created through shuffling, repeating, adding, or removing phrases/sentences in the gold response. Pretraining tasks involve tasks such as infilling or finding the index of an incoherent or missing utterance. They include multiple tasks covered in prior pretraining work (Mehri et al., 2019; Zhao et al., 2020b; Whang et al., 2021; Xu et al., 2021b). Safety Tasks consist of toxicity detection, non-toxic, and recovery response generation. Miscellaneous tasks are a set of tasks that belong to specialized domains such as giving advice or persuading a user.

3.3 Task Schema and Formatting

All tasks in INSTRUCTDIAL are expressed in a natural language sequence-to-sequence format. Every task instance is formatted with the following...
properties: **Task Definition:** Description of the task containing information about how to produce an output given an input. **Instance Inputs:** Instances from a dataset converted into a sequence. **Constraints:** Additional metadata or constraints for a task (emotion tag for emotion-based generation, classes for classification). **Prompt:** Text sequence that connects the instance back to the instruction, expressed as a command or a question. **Output:** Output of an instance converted into a sequence.

Figure 3 shows examples of instances from 3 tasks. For each task, we manually compose 3-10 task definitions and prompts. For every instance, a task definition and a prompt are selected randomly during test. We do not include in-context examples in the task schema since dialogue contexts are often long and concatenating long examples would exceed the maximum allowable input length for most models. Input instances are formatted using special tokens. The token [CONTEXT] signals the start of dialogue content. Dialogue turns are separated by [ENDOFTURN]. [ENDOFDIALOGUE] marks the end of the dialogue and [QUESTION] marks the start of the prompt text. We also incorporate task specific special tokens (such as [EMOTION] for emotion classification task). We hypothesize that using a consistent structure and formatting across tasks should help the model adopt the structure and novel input fields for unseen tasks better.

**Classification Options:** In classification tasks, the model is trained to predict an output that belongs to one of several classes. To make the model aware of output classes available for an unseen task, we append a list of classes from which the model should choose. We adopt the following two formats for representing the classes: (1) **Name list:** list the class names separated by a class separator token such as a comma, and (2) **Indexed list:** list the classes indexed by either alphabets or numbers (such as 1: class A, 2: class B,...) where the model outputs the index corresponding to the predicted class. This representation is useful when the classification options are long in length, such as in the case of response ranking where the model has to output the best response among the provided candidates.

**Custom inputs:** Some tasks consist of input fields that are unique to the task. For example, emotion grounded generation consists of emotion labels that the model uses for response generation. We append such inputs to the beginning of the instance sequence along with the field label. For example, we pre-pend "[EMOTION] happy" to the dialogue context in the emotion generation task.

In Table 8 in the Appendix we present the list of tasks with sample inputs for each task.

### 3.4 Meta Tasks

A model can learn to perform well on tasks during training by inferring the domain and characteristics of the dataset instead of paying attention to the instructions, and then fail to generalize to new instructions at the test time. We introduce two meta-tasks that help the model learn the association between the instruction, the data, and the task. In the **Instruction selection task**, the model is asked to select the instruction which corresponds to a
given input-output pair. In the Instruction binary task, the model is asked to predict “yes” or “no” if the provided instruction leads to a given output from an input. We show an example for instruction selection task in Figure 3.

3.5 None-of-the-above Options

For classification tasks, most tasks assume that the ground truth is always present in the candidate set, which is not the case for all unseen tasks. To solve this issue, we propose adding a NOTA (“None of the above”) option in the classification tasks during training as both correct answers and distractors following Feng et al. (2020b) for 10% of the training instances. To add NOTA as a correct answer, we add “none of the above” as a classification label option, remove the gold label from the options and set the output label as NOTA. To add NOTA as a distractor, we add NOTA to the classification labels list but keep the gold label as the output label.

4 Experimental Setup

4.1 Model Details

Our models use an encoder-decoder architecture and are trained using maximum likelihood training objective. We finetune the following two base models on the tasks from I

| 1. T0-3B (Sanh et al., 2022) a model initialized from the 3B parameters version of T5 (Lester et al., 2021). T0-3B is trained on a multitask mixture of general non-dialogue tasks such as question answering, sentiment detection, and paraphrase identification. |
| 2. BART0 (Lin et al., 2022), a model with 406 million parameters (8x smaller than T0-3B) based on Bart-large (Lewis et al., 2020), trained on the same task mixture as T0-3B. |

We name the BART0 model tuned on INSTRUCTDIAL as DIAL-BART0 and T0-3B model tuned on INSTRUCTDIAL as DIAL-T0. DIAL-BART0 is our main model for experiments since its base BART0 has shown comparable zero-shot performance to T0 (Lin et al., 2022) despite being 8 times smaller, whereas the 3B parameter model DIAL-T0 is large and impractical to use on popular affordable GPUs. We perform finetuning on these two models since they both are instruction-tuned on general NLP tasks and thus provide a good base for building a dialogue instruction tuned model.

4.2 Training Details

For training data creation, we first generate instances from all datasets belonging to each task. We then sample a fixed maximum of $N = 5000$ instances per task. Each instance in a task is assigned a random task definition and prompt. We truncate the input sequences to 1024 tokens and target output sequences to 256 tokens. We train DIAL-BART0 on 2 Nvidia 2080Ti GPUs using a batch size of 2 per GPU with gradient checkpointing. We train DIAL-T0 on 2 Nvidia A6000 GPUs using a batch size of 1 per GPU with gradient checkpointing. Additional implementation details are present in Appendix A.

5 Experiments and Results

We evaluate our models on multiple zero-shot and few-shot settings. We establish benchmark results for Zero-shot unseen tasks evaluation (Section 5.1) and Response evaluation task (Section 5.2) and perform error analysis. Next, we perform zero-shot and few-shot experiments on three important dialogue tasks: intent detection, slot value generation, and dialogue state tracking (Section 5.3).

5.1 Zero-shot Unseen Tasks Evaluation

In this experiment, we test our models’ zero-shot ability on tasks not seen during training.

5.1.1 Unseen Tasks for Zero-shot Setting

We perform evaluation on the test set of the following 6 tasks not seen during training:

1. Dialfact classification: predict if an evidence supports, refutes, or does not have enough information to validate the response
2. Relation classification: predict the relation between two people in a dialogue
3. Answer selection: predict an answer to a conversational question
4. Eval selection: choose the most relevant response among the provided 4 options. Dataset and ratings based on DSTC 10 Automatic evaluation challenge (Chen et al., 2021b)
5. Knowledge grounded generation: generate a response based on background knowledge
6. Begins with generation: generate a response that starts with the provided initial phrase

All 6 tasks have varying levels of difficulty and cover both classification and generation. To emulate a zero-shot scenario, we remove all relation-based, evaluation type, answer generation, and...
wiki-based tasks from the training task set. The set of tasks used for training is presented in Table 10. We evaluate on the full test sets for Dialfact, relation, and answer classification, and sample 1000 instances for the rest of the tasks.

### 5.1.2 Setup and Baselines

We perform inference and evaluation on the 6 unseen tasks described in Section 5.1.1. We compare the following models and baselines:

- **BART0 and T0-3B** - Models that form a base for our models, trained on a mixture of non-dialogue general NLP tasks (described in Section 4.1).
- **GPT-3** (Brown et al., 2020) - Davinci version of GPT-3 tested using our instruction set.
- **DIAL-BART0 and DIAL-T0** - Our models described in Section 4.1.
- **DB-Few** - Few-shot version of DIAL-BART0. 100 random training set instances of the test tasks are mixed with the instances of train tasks.
- **DB-Full** - Version of DIAL-BART0 where 5000 instances per test tasks are mixed with the instances of the train tasks. This baseline serves as the upper bound for our models’ performance.

We also experiment with the following ablations of DIAL-BART0:

- **DB-no-base** - Uses Bart-large instead of using the BART0 as the base model.
- **DB-no-instr** - Trained with no instructions or prompts. Task constraints and class options are still specified. We specify the task name instead of instructions to help the model identify the task.
- **DB-no-nota** - Trained without None-of-the-above from Section 3.5
- **DB-no-meta** - Trained without the meta tasks from Section 3.4

### 5.1.3 Results and Discussion

We present the results for zero-shot experiments in Table 1 and report the accuracy metric for the Eval selection, Answer selection, Dialfact classification and Relation classification tasks. For Begins with task, we report BLEU2, ROUGEL, and accuracy defined as the proportion of responses that begins with the initial phrase provided. For Knowledge grounded generation we report BLEU2, ROUGEL metrics along with F1 as defined in (Dinan et al., 2019c). For the generation tasks we also report the automatic metric GRADE (Huang et al., 2020) (which has shown good correlation with human ratings on response coherence). For GPT-3 baseline we report the metrics on 200 randomly sampled instances per task. We average scores obtained across the instructions and prompts. We notice the following general trends in our results.

**Instruction tuning on INSTRUCTDIAL improves performance on unseen dialogue tasks**: The DIAL-BART0 and DIAL-T0 models instruction tuned on INSTRUCTDIAL achieve better performance on all tasks compared to their base models BART0 and T0-3B. Notably, for the Eval selection, Relation classification and Begins with generation tasks, our models perform about 3 times better than the base models. Our model also performs significantly better than GPT-3 for all tasks except for Dialfact classification. In the case of the Answer selection task, the difference in performance is lower compared to other models since the base-

<table>
<thead>
<tr>
<th>Model</th>
<th>ES</th>
<th>AS</th>
<th>RC</th>
<th>DC</th>
<th>BW</th>
<th>KG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>F1</td>
</tr>
<tr>
<td><strong>Baselines and Our Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BART0</td>
<td>22.2</td>
<td>58.5</td>
<td>6.3</td>
<td>33.7</td>
<td>4.2</td>
<td>4.9</td>
</tr>
<tr>
<td>T0-3B</td>
<td>45.9</td>
<td>60.2</td>
<td>1.3</td>
<td>33.1</td>
<td>14.1</td>
<td>4.1</td>
</tr>
<tr>
<td>GPT-3</td>
<td>57.5</td>
<td>56.5</td>
<td>11.5</td>
<td>37.3</td>
<td>16.5</td>
<td>7.2</td>
</tr>
<tr>
<td>DIAL-BART0 (Ours)</td>
<td>66.7</td>
<td>59.5</td>
<td>17.8</td>
<td>35.6</td>
<td>56.3</td>
<td>13.1</td>
</tr>
<tr>
<td>DIAL-T0 (Ours)</td>
<td>74.4</td>
<td>65.2</td>
<td>17.8</td>
<td>35.6</td>
<td>56.3</td>
<td>13.1</td>
</tr>
</tbody>
</table>

**Few and Full shot Variations**

<table>
<thead>
<tr>
<th>Model</th>
<th>ES</th>
<th>AS</th>
<th>RC</th>
<th>DC</th>
<th>BW</th>
<th>KG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>F1</td>
</tr>
<tr>
<td>DB-Few</td>
<td>77.1</td>
<td>69.1</td>
<td>28.0</td>
<td>43.0</td>
<td>72.2</td>
<td>16.7</td>
</tr>
<tr>
<td>DB-Full</td>
<td>90.7</td>
<td>83.3</td>
<td>62.7</td>
<td>77.4</td>
<td>83.7</td>
<td>20.8</td>
</tr>
</tbody>
</table>

**Model Ablations for DIAL-BART0**

<table>
<thead>
<tr>
<th>Model</th>
<th>ES</th>
<th>AS</th>
<th>RC</th>
<th>DC</th>
<th>BW</th>
<th>KG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>ACC</td>
<td>F1</td>
</tr>
<tr>
<td>DB-no-base</td>
<td>40.1</td>
<td>52.7</td>
<td>17.1</td>
<td>35.1</td>
<td>53.9</td>
<td>12.0</td>
</tr>
<tr>
<td>DB-no-instr</td>
<td>23.0</td>
<td>43.2</td>
<td>15.1</td>
<td>35.4</td>
<td>50.0</td>
<td>13.0</td>
</tr>
<tr>
<td>DB-no-nota</td>
<td>66.5</td>
<td>57.2</td>
<td>17.2</td>
<td>35.9</td>
<td>56.1</td>
<td>10.9</td>
</tr>
<tr>
<td>DB-no-meta</td>
<td>44.5</td>
<td>52.0</td>
<td>14.1</td>
<td>35.4</td>
<td>52.5</td>
<td>14.1</td>
</tr>
</tbody>
</table>

Table 1: Zero-shot evaluation on unseen tasks. B-2 stands for BLEU2, R-L for RougeL and GR for GRADE metric. Here ES stands for Eval Selection, AS for Answer Selection, RC for Relation Classification, DC for Dialfact Classification, BW for Begins With, KG for Knowledge Grounded generation. DB-Few and DB-Full are variants of DIAL-BART0. Our models DIAL-BART0 and DIAL-T0 outperform the baseline models and their ablated versions.
Figure 4: Model’s performance on unseen tasks improves with the number of seen tasks during training. We report average Accuracy across Eval Selection, Answer Selection, Relation Classification, and Dialfact Classification, and average RougeL scores for Knowledge Grounded Generation and Begins with Generation.

We report average Accuracy across Eval Selection, Answer Selection, Relation Classification, and Dialfact Classification, and average RougeL scores for Knowledge Grounded Generation and Begins with Generation. Larger models are also trained on similar extractive and multi-choice question answering tasks. Relation and Dialfact classification are hard tasks for all models since there are no similar train tasks. Larger models are not necessarily better across tasks: Experiments across varying model size show that while T0-3B and DIAL-T0 perform better on the Eval selection and Answer Selection tasks and perform equivalently on the Begins with generation task, BART0 and DIAL-BART0 perform better on the rest of the unseen tasks. While DIAL-T0 is better at classification tasks, it has poor performance on generation compared to DIAL-BART0. We also observed that DIAL-T0 sometimes produces empty or repetitive outputs for generation tasks.

Figure 4: Model’s performance on unseen tasks improves with the number of seen tasks during training.

Few-shot training significantly improves performance: DB-Few model that incorporates 100 instances per test task in its training data shows significant improvements in performance compared to its zero-shot counterpart DIAL-BART0. We see about 12-16% improvements on the Eval selection, Answer selection, and Dialfact classification tasks, and 30-50% improvement on the Begins with and Relation classification tasks.

Full-shot training can improve performance across multiple tasks: DB-Full model achieves high performance across all test tasks. The full-shot performance of DIAL-BART0 on Dialfact and relation classification tasks are near state-of-the-art performance without using the full train datasets.

Meta tasks and NOTA are important for better generalization: We see a large performance drop on unseen classification tasks when meta tasks (see Section 3.4) are removed. This shows that meta tasks help the model develop better representations and understanding of natural language instructions. DB-no-nota shows a slight performance drop in the classification task, indicating NOTA objective is helpful, but not crucial for performance.

Pretraining on general NLP tasks helps dialogue instruction tuning: DB-no-base model shows a high performance drop on Eval selection and Answer selection tasks, and a small drop on other test tasks. We conclude that instruction tuning for general NLP tasks helps dialogue instruction tuning. Using instructions leads to better generalization: DB-no-instr shows worse performance than DIAL-BART0 on all tasks, especially on Eval selection, Answer selection, and Relation classification tasks. This indicates that training with instructions is crucial for zero-shot performance on unseen tasks.

Training on more seen tasks improves generalization on unseen tasks: In Figure 4 we show the impact of varying the number of seen tasks on the performance on unseen tasks. We adopt the train-test task split from section 5.1. We observe that the performance improves sharply up to 20-25 tasks and then further keeps steadily increasing with each new task. This indicates that increasing the number of tasks can lead to better zero-shot generalization and that scaling to more tasks may lead to better instruction-tuned models.

5.1.4 Analysis

Sensitivity to instruction wording: To analyze the sensitivity of our models to instruction wording, we breakdown the evaluation metrics per unique instruction used during inference for the DIAL-BART0 model. The accuracy varies from 65.6-67.8 across instructions for Eval selection, from 52.5 to 75.0 for Answer selection, 17.1 to 18.4 for Relation classification, 34.7 to 37.1 for Dialfact classification, 49.8 to 62.3 for Begins with generation, and F1 score varies from 26.6 to 28.6 for Knowledge grounded generation. Thus, our model is moderately sensitive to the instruction wording.

Errors in model outputs: We perform qualitative analysis of randomly sampled outputs of the models. For classification tasks, a common error across all models is generating outputs outside of the provided list of classes. This happens with GPT-3 for 20%, BART0 10% and T0-3B 17.8% of the inputs, but for DIAL-BART0 and DIAL-T0 this occurs only for 2.5% and 4.8% of the inputs. Other possible but rare types of errors include copying the provided input as output, early truncation of generated responses, and performing...
an unspecified task. Apart from the unseen task set adopted for our experiments in section 5.1.1, we tried other seen-unseen task configurations and found that both our models and baselines models cannot perform certain tasks such as Infilling missing utterance. Recovery response generation, and Ends with response generation in a zero-shot manner. However, the models could quickly learn these tasks when trained on a few task instances.

In Table 7 of Appendix B we provide a sample conversation, various instructions for that conversation, and the outputs generated by DIAL-BART0 based on the specified instructions.

5.2 Zero-shot Automatic Response Evaluation

Development of automatic dialogue metrics that show high correlations with human judgements is a challenging and crucial task for dialogue systems. Automated metrics such as BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) correlate poorly with human judgement (Gupta et al., 2019). In this experiment, we test our model’s zero-shot automatic evaluation capabilities through the Eval Relevance task. We use the evaluation ratings released in the DSTC-10 Automatic evaluation challenge (Chen et al., 2021b) that consists of 65,938 context-response pairs along with corresponding human ratings aggregated across various evaluation sets. We train a version of DIAL-T0 on tasks excluding any eval tasks (shown in Table 10). Given a dialogue context and a candidate response, we instruct the model to predict “yes” if the response is relevant to the context, otherwise predict “no”. We calculate the probability of “yes” as \( p(yes) = p(yes) / (p(yes) + p(no)) \). We calculate the Spearman correlation of the model’s prediction with human ratings for relevance provided in the DSTC-10 test sets, and present the results in Table 2. We compare our model with reference-free models studied in Yeh et al. (2021). DIAL-T0 is ranked the first or second in the majority of the evaluation datasets. Our model learns coherence from the variety of tasks it is trained on and demonstrates high zero-shot dialogue evaluation capabilities.

5.3 Zero-shot and Few-shot Dialogue Tasks

We test the zero-shot and few-shot abilities of our models on three important dialogue tasks: intent prediction, slot filling, and dialogue state tracking.

5.3.1 Intent Prediction

Intent prediction is the task of predicting an intent class for a given utterance. We conduct few-shot experiments on the Banking77 benchmark dataset (Casanueva et al., 2020) that contains 77 unique intent classes. Models are trained on 10 instances per intent class. We compare our model...
Table 5: Few-shot slot filling F1 scores on DSTC8 data.

<table>
<thead>
<tr>
<th>Domain</th>
<th>GENSF</th>
<th>DIAL-BART0 (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buses</td>
<td>90.5</td>
<td>97.8</td>
</tr>
<tr>
<td>Events</td>
<td>91.2</td>
<td>94.3</td>
</tr>
<tr>
<td>Homes</td>
<td>93.7</td>
<td>96.5</td>
</tr>
<tr>
<td>Rental Cars</td>
<td>86.7</td>
<td>94.2</td>
</tr>
</tbody>
</table>

Table 6: Joint goal accuracy for dialogue state tracking.

<table>
<thead>
<tr>
<th>Model</th>
<th>1% data</th>
<th>5% data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPTOD</td>
<td>29.7</td>
<td>40.2</td>
</tr>
<tr>
<td>DIAL-BART0 (Ours)</td>
<td>29.2</td>
<td>38.1</td>
</tr>
</tbody>
</table>

Table 5. Zero-shot slot filling results on the Restaurant8k corpus.

5.3.3 Dialogue State Tracking

We evaluate our model on the dialogue state tracking task which involves filling in values of predefined slots. We adopt the experimental setup from PPTOD (Su et al., 2022a), and conduct few-shot experiments on MultiWOZ 2.0 (Budzianowski et al., 2018). Similar to PPTOD, our DIAL-BART0 model is first pre-trained on 7 datasets: KVRET (Eric et al., 2017), WOZ (Mrkšić et al., 2017), CamRest676 (Wen et al., 2017), MSR-E2E (Li et al., 2018), Frames (El Asri et al., 2017), TaskMaster (Byrne et al., 2019), Schema-Guided (Rastogi et al., 2020b) along with other non-related dialogue tasks. We then train on 1% and 5% splits of MultiWOZ for 40 epochs with a learning rate of 5e−5. In Table 6 we present few-shot dialogue state tracking results on the MultiWOZ test set. We find that our model obtains 29.2 and 38.1 joint goal accuracy on the 1% and 5% training data splits, respectively. Our results demonstrate that our model performs well on few-shot dialogue state tracking, and achieves competitive results against PPTOD which is twice the size of our model.

6 Conclusion

We propose INSTRUCTDIAL, an instruction tuning framework for dialogue, which contains multiple dialogue tasks created from openly available dialogue datasets. We also propose two meta-tasks to encourage the model to pay attention to instructions. Our results show that models trained on INSTRUCTDIAL achieve good zero-shot performance on unseen tasks (e.g., dialogue evaluation) and good few-shot performance on dialogue tasks (e.g., intent prediction, slot filling). We perform ablation studies showing the impact of using an instruction tuned base model, model size/type, increasing the number of tasks, and incorporating our proposed meta tasks. Our experiments reveal that instruction tuning does not benefit all unseen test tasks and that improvements can be made in instruction wording invariance and task interference. We hope that INSTRUCTDIAL will facilitate further progress on instruction-tuning systems for dialogue tasks.
7 Limitations

Our work is the first to explore instruction tuning for dialogue and establishes baseline performance for a variety of dialogue tasks. However, there is room for improvements in the following aspects:

1) Unlike a few prior works, the instructions and prompts used in this work are not crowdsourced and are limited in number. Furthermore, our instructions and tasks are only specified in the English language. Future work may look into either crowdsourcing or automatic methods for augmenting the set of instructions in terms of both language diversity as well as quantity.

2) Instruction tuning does not show significant improvements in zero-shot setting on a few tasks such as relation classification and infilling missing utterances in our experiments. Future work can look into investigating why certain tasks are more challenging than others for zero-shot generalization. Furthermore, zero-shot performance of our models on many tasks is still far from the few-shot and full-shot performance on those tasks. We hope that INSTRUCTDIAL can be lead to further investigations and improvements in this area.

3) We observed a few instances of task interference in our experiments. For example, the set of tasks used for zero-shot automatic response evaluation as mentioned in Table 10 is different and smaller from the set of tasks used in our main experiments in Section 5.1.1. We found that incorporating a few additional tasks lead to a reduction in the performance on zero-shot automatic response evaluation. Furthermore, training on multiple tasks can lead to task forgetting. To address these issues, future work can take inspiration from work related to negative task interference (Wang et al., 2020a; Larson and Leach, 2022), transferability (Vu et al., 2020; Wu et al., 2020b; Xing et al., 2022) and lifelong learning (Wang et al., 2020b).

4) Our models are sensitive to the wording of the instructions, especially in zero-shot settings as discussed in Section 5.1.4. Improving insensitivity to prompts and instructions is an important future research direction.

5) Our work does not explore in-context few-shot learning through examples as the prompt length can go beyond models’ maximum input length. It also does not study the composition of multiple tasks through instructions. Both these aspects warrant further investigations.

6) INSTRUCTDIAL includes only text based tasks, and future work may look into incorporating datasets with other modalities such as vision and audio.

8 Ethics and Broader Impact

Broader Impact and applications: Our framework leverages instruction tuning on multiple dialogue tasks, allowing multiple functionalities to be quickly implemented and evaluated in dialogue systems. For example, tasks pertaining to both task-oriented dialogue tasks, such as slot detection and domain-specific tasks such as emotion detection can be added and evaluated against state-of-the-art dialogue systems. This enables users to diagnose their models on different tasks and expand the abilities of multi-faceted dialogue systems, which can lead to richer user interactions across a wide range of applications. Our framework allows training models below billion parameter range, making them more accessible to the research community.

Potential biases: Current conversational systems suffer from several limitations, and lack empathy, morality, discretion, and factual correctness. Biases may exist across datasets used in this work and those biases can propagate during inference into the unseen tasks. Few-shot and zero-shot methods are easier to train, and their use can lead to a further increase of both the benefits and risks of models. To mitigate some of those risks, we have included tasks and datasets in our framework that encourage safety such as ToxiChat for toxic response classification task and SaFeRDialogues for recovery response generation task, and that improve empathy such as EmpatheticDialogues for empathy.

References


Fei Mi, Yitong Li, Yasheng Wang, Xin Jiang, and Qun Liu. 2021a. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems. ArXiv, abs/2109.04645.


Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei,


Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts?

Jason Wei, Maarten Paul Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew Mingbo Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.


Appendix

A Additional implementation details

Data Sampling For training data creation, we first generate instances from all datasets belonging to each task. Since the number of instances per task can be highly imbalanced, we sample a fixed maximum of $N$ number of instances per task. In our main models and experiments, we set $N = 5000$. Each instance in a task is assigned a random task definition and prompt. We truncate the input sequences to 1024 tokens and target output sequences to 256 tokens.

Implementation Details Our models are trained for 3 epochs with a learning rate of 5e-5 with an Adam optimizer (Kingma and Ba, 2015) with linear learning rate decay. For our main experiments in Table 1, we perform checkpoint selection using a validation set created from the train tasks. For rest of the experiments we do model selection using the validation sets. We use the HuggingFace Transformers library for training and inference implementation and use DeepSpeed library for improving training efficiency. We train DIAL-BART0 on 2 Nvidia 2080Ti GPUs using a batch size of 2 per GPU and an effective batch size of 72 with gradient checkpointing. We train DIAL-T0 on 2 Nvidia A6000 GPUs using a batch size of 1 per GPU and an effective batch size of 72 with gradient checkpointing. For all classification tasks, we perform greedy decoding, and for all generation tasks, we perform top-p sampling with $p = 0.7$ and temperature set to 0.7. The repetition penalty is set to 1.2. In Table 1, for DIAL-BART0 and DIAL-T0, we report the results over three different training runs, where each run is based on a new sample of training data.

Zero-shot Automatic Evaluation Implementation Details For zero shot automatic evaluation, we calculate the Spearman correlation of the model’s prediction with human ratings for relevance provided in the DSTC-10 test sets. There is no consistent “relevance” or “coherence” rating field present across the evaluation datasets. We therefore calculate the correlation with the ratings if a rating exists in any of the following fields “overall”, “turing”, “relevance” and “appropriateness”.

\[ \text{https://github.com/huggingface/transformers} \]

\[ \text{https://github.com/microsoft/DeepSpeed} \]
Conversation:
[CONTEXT] How may I help you? [ENDOFTURN]
I left a suitcase on the train to London the other day. [ENDOFDIALOGUE]

<table>
<thead>
<tr>
<th>Task</th>
<th>Instruction</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response editing</td>
<td>Modify the provided response into a response that is fluent and coherent to the dialogue context: [RESPONSE] Can describe it it, sir? It will help us find</td>
<td>Can you describe it, sir? It will help us find it.</td>
</tr>
<tr>
<td>Begins with</td>
<td>Generate a response that starts with the provided initial phrase.</td>
<td>Please describe the suitcase.</td>
</tr>
<tr>
<td>Begins with + Keyword controlled generation</td>
<td>Generate a response that starts with the provided initial phrase and contains the provided keywords. [INITIAL PHRASE] Please describe [KEYWORDS] color, any documents</td>
<td>Please describe the color of the suitcase and any documents in it.</td>
</tr>
<tr>
<td>Intent detection</td>
<td>What is the intent of the response [OPTIONS] booking, reservation change, checkout, lost&amp;found,..., time information, security, schedules</td>
<td>lost&amp;found</td>
</tr>
<tr>
<td>Summarization</td>
<td>Return a summary of the provided conversation.</td>
<td>Person2 left a suitcase on the train to London the other day.</td>
</tr>
<tr>
<td>Answer generation</td>
<td>[QUESTION] What is the response of following question: Where was the person going to?</td>
<td>London</td>
</tr>
<tr>
<td>Knowledge grounded generation</td>
<td>Generate a response using the provided background knowledge. [KNOWLEDGE] Emailid for cases related to lost&amp;found is <a href="mailto:x@gmail.com">x@gmail.com</a></td>
<td>You can contact us at <a href="mailto:x@gmail.com">x@gmail.com</a></td>
</tr>
</tbody>
</table>

Table 7: A sample conversation followed by instructions for multiple tasks for that conversation, and the outputs generated based on the specified instructions. Instruction tuning allows performing multiple tasks on an input by specifying task-specific instructions and prompts.

**B Sample conversation and Instructions**

In Table 7 we provide a sample conversation followed by instructions for multiple tasks for that conversation, and the outputs generated by DIAL-BART0 based on the specified instructions. Through this example we illustrate that instruction tuning allows performing multiple tasks on an input by specifying task-specific instructions.

**C Datasets used in tasks**

In Table 9 we present the list of tasks with datasets used in each task.

**D Configuration of experiments**

In Table 10 we provide the configurations of experiments, that is, the tasks used for training for each experiment.
Table 8: List of tasks with sample inputs for each task. The left column describes the general task type. The middle column lists the specific task. The right column displays an example formatted using a randomly selected task definition and prompt for the task. [CTX] is short for [CONTEXT], [Q] is short for [QUESTION], [RES] is short for response, [EOT] is short for [ENDOFTURN] and [EOD] is short for [ENDOFDIALOGUE].
<table>
<thead>
<tr>
<th>Task Type</th>
<th>Task Name</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Type</td>
<td>Task Name</td>
<td>Datasets</td>
</tr>
<tr>
<td>Intent</td>
<td>Intent Classification</td>
<td>ATIS (Hemphill et al., 1990) SNIPS (Coucke et al., 2018) CLINIC150 (Lavon et al., 2019)</td>
</tr>
<tr>
<td></td>
<td>Intent Present</td>
<td>ATIS (Hemphill et al., 1990) SNIPS (Coucke et al., 2018) CLINIC150 (Lavon et al., 2019)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HWI64 (Liu et al., 2021b) Shaking77 (Casasayas et al., 2020)</td>
</tr>
<tr>
<td>NLI</td>
<td>DualFact</td>
<td>DualFact (Gupta et al., 2021)</td>
</tr>
<tr>
<td></td>
<td>OpenNLI</td>
<td>OpenNLI</td>
</tr>
<tr>
<td></td>
<td>Contradiction</td>
<td>Decode (Nie et al., 2021) Dialogue NLI (Welbeck et al., 2019)</td>
</tr>
<tr>
<td>Safety Classification</td>
<td>Text Classification</td>
<td>ToxChat (Balletti et al., 2021) BAD (Xu et al., 2021a) Build It Break It Fix It (Dinan et al., 2019a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relation Extraction</td>
<td>Relation Classification</td>
<td>DialoRE (Yu et al., 2020)</td>
</tr>
<tr>
<td></td>
<td>Relation Present</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>Relevance</td>
<td>DSTC6 (Hsu and Hori, 2017) DSTC7 (Malley et al., 2019) Persona-Chatlog (San et al., 2019)</td>
</tr>
<tr>
<td></td>
<td>Selection</td>
<td>USR (Mehri and Eskenazi, 2020b) FED (Mehri and Eskenazi, 2020a) Daily Dialog (Zhao et al., 2020a)</td>
</tr>
<tr>
<td></td>
<td>Rating</td>
<td>PersonaChat (Zhao et al., 2020a) GRADE (Huang et al., 2020) HUMOD (Merdan et al., 2020)</td>
</tr>
<tr>
<td>Slot</td>
<td>Slot Classification</td>
<td>RESTAURANTS-8K (Coope et al., 2020b) VICTOR-SUB (Dinan et al., 2020b)</td>
</tr>
<tr>
<td></td>
<td>Slot Present</td>
<td>ATIS (Hemphill et al., 1990) SNIPS (Coucke et al., 2018)</td>
</tr>
<tr>
<td></td>
<td>Slot Value Generation</td>
<td>TaskMaster (Byrne et al., 2019) MSRE2E (Li et al., 2018)</td>
</tr>
<tr>
<td>Safety Generation</td>
<td>Recovery Response Generation</td>
<td>SafeRDialogue (Ung et al., 2021)</td>
</tr>
<tr>
<td>Grounded Generation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td>EditWOZ (Budzianowski et al., 2018)</td>
</tr>
<tr>
<td></td>
<td>Document-repetitive</td>
<td>document-repetitive</td>
</tr>
<tr>
<td></td>
<td>Unique-Based</td>
<td>unique-based</td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Knowledge-embedded</td>
<td>knowledge-embedded</td>
</tr>
<tr>
<td>QA and Comprehension</td>
<td>Answer Generation</td>
<td>CIDEr (Vedantam et al., 2015) HMEDUAL (Qin et al., 2021) Monoling (Sun et al., 2020)</td>
</tr>
<tr>
<td></td>
<td>Answer Selection</td>
<td>QA-Conv (Wu et al., 2021) CoQA (Reddy et al., 2019) QnAC (Choi et al., 2018)</td>
</tr>
<tr>
<td></td>
<td>Question Generation</td>
<td>QnQA (Wu et al., 2021)</td>
</tr>
<tr>
<td></td>
<td>Target Guided</td>
<td>OTTers (Sevignani et al., 2021)</td>
</tr>
<tr>
<td>Controlled Generation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Response/Response Comprehension</td>
<td>EmpatheticDialogues (Rashkin et al., 2019) DailyDialog (Li et al., 2017)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ComAI (Dinan et al., 2019b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expert-Driven (Wu et al., 2017)</td>
</tr>
<tr>
<td>Dialog State Generation</td>
<td>Dialog State Generation</td>
<td>MultiWOZ (Budzianowski et al., 2018) KVRET (Hrele et al., 2017) WOZ (Muml et al., 2017)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CobiRet (Wen et al., 2017) MRS-E2E (Li et al., 2018) TaskMaster (Byrne et al., 2019)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shuffling</td>
<td>TopicalChat (Ojighakimnba et al., 2019) EmotionLines (Hou et al., 2018)</td>
</tr>
<tr>
<td></td>
<td>Adding</td>
<td>WOZ (Dinan et al., 2019c) Persuasion (Wang et al., 2019) CoSNo (Chauva et al., 2021)</td>
</tr>
<tr>
<td></td>
<td>Removing</td>
<td>DailyDialog (Li et al., 2017) ComAI (Dinan et al., 2019b) EmotionLines (Hou et al., 2018)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pretain Tasks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full Missing Utterance</td>
<td>DailyDialog (Li et al., 2017) WOZ (Dinan et al., 2019c) EmpatheticDialogues OpenDialogKG (Moon et al., 2019)</td>
</tr>
<tr>
<td></td>
<td>Time Incoherence Utterance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time Weighted Utterance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Generation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Open Domain</td>
<td>DailyDialog (Li et al., 2017) ComAI (Dinan et al., 2019b) WOZ (Dinan et al., 2019c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EmphaticDialogues (Rashkin et al., 2019) OpalDialogKG (Moon et al., 2019)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summarization</td>
<td>Summary Generation</td>
<td>DailySum (Chen et al., 2021a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cream (Salavati and Chen, 2016) FAMSum (Zhang et al., 2021) FAMSum (Ou et al., 2019)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Act Classification</td>
<td>MSRE2E (Li et al., 2018) DialoG 1st et al., 2017 MultiWOZ (Budzianowski et al., 2018)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tweet Present</td>
<td>TiringAdvice (Zellers et al., 2021)</td>
</tr>
<tr>
<td></td>
<td>Addictor Generation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control Generation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emotion Tagging</td>
<td>EmotionLines (Hou et al., 2018) DailyDialog (Li et al., 2017)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misc</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Count Response Words</td>
<td>DailyDialog (Li et al., 2017) WOZ (Dinan et al., 2019c) EmpatheticDialogues (Rashkin et al., 2019)</td>
</tr>
</tbody>
</table>

Table 9: List of Tasks with datasets used in each task. The left column describes the general task type. The middle column lists the specific task. The right column shows all datasets used for a specific task type.
<table>
<thead>
<tr>
<th>Experiment</th>
<th>Base model(s)</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main zero-shot tasks</strong></td>
<td>ID-BART0, ID-T0</td>
<td>act classification, act generation, advice generation, advice present, answer generation, count response words, db based generation, deal present, document grounded generation, emotion generation, emotion tagging, endwith controlled generation, fill missing utterance, find incoherent utterance, find missing utterance, intent classification, intent present (no intent banking dataset), keyword controlled generation, nli classification, non-toxic feedback generation, persona grounded generation, persuasion generation, persuasion present, persuasion strategy, question generation, recovery generation, response generation, response generation with n words, schema based generation, slot present, slot value generation, summarization, target controlled generation, toxic classification</td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td>ID-BART0</td>
<td>act classification, act generation, advice present, answer generation, answer selection, beginwith controlled generation, belief state generation, db based generation, deal present, document grounded generation, emotion generation, emotion tagging, endwith controlled generation, emotion tagging, graph based generation, intent classification, intent present, keyword controlled generation, knowledge grounded generation, nli classification, persona grounded generation, persuasion generation, persuasion present, question generation, relation classification, relation present, response generation, schema based generation, slot present, slot value generation, summarization, target controlled generation, toxic classification</td>
</tr>
<tr>
<td><strong>Dialog State Generation</strong></td>
<td>ID-BART0</td>
<td>act classification, act generation, advice generation, advice present, answer generation, answer selection, beginwith controlled generation, count response words, db based generation, deal present, dialfact classification, dialog state generation, no multiwoz, document grounded generation, edit generation, emotion generation, emotion tagging, eval binary, eval ranking, eval rating, fill missing utterance, find incoherent utterance, find missing utterance, find swapped utterance, intent classification, intent present, keyword controlled generation, knowledge grounded generation, nli classification, non-toxic feedback generation, persona grounded generation, persuasion generation, persuasion present, persuasion strategy, question generation, relation classification, relation present, response generation, response generation with n words, schema based generation, slot present, slot value generation, summarization, target controlled generation, toxic classification</td>
</tr>
<tr>
<td><strong>Slot Filling</strong></td>
<td>ID-BART0</td>
<td>act classification, act generation, answer generation, answer selection, beginwith controlled generation, belief state generation, count response words, db based generation, deal present, dialfact classification, document grounded generation, edit generation, emotion generation, emotion tagging, endwith controlled generation, eval binary, eval ranking, eval rating, fill missing utterance, find incoherent utterance, find missing utterance, find swapped utterance, intent classification, intent present, keyword controlled generation, knowledge grounded generation, nli classification, non-toxic feedback generation, persona grounded generation, persuasion generation, persuasion present, persuasion strategy, question generation, relation classification, relation present, response generation, response generation with n words, schema based generation, slot present, slot value generation, summarization, target controlled generation, toxic classification</td>
</tr>
</tbody>
</table>

Table 10: List of experiments and their base models. The tasks listed in the right column are all the tasks a base model was trained with for their corresponding experiment.