

1. Prerequisites

- **Activate environment**

```
cd /home
source venv/bin/activate
```

- **Setup request using Mephisto CLI**

```
mephisto register mturk \
    name=my_mturk_user \
    access_key_id=$ACCESS_KEY\
    secret_access_key=$SECRET_KEY
```

2. Stage 1 (Gathering turn-taking chat conversation data)

`ParlAI/parlai/crowdsourcing/tasks/turn_taking_chat` is the folder responsible for achieving the goal.

- **Command to run task**

```
python ParlAI/parlai/crowdsourcing/tasks/turn_taking_chat/run.py
```

Once workers complete all the tasks launched in MTurk, the script will receive and save the result in the database and then finish working automatically.

- **Configuration setup**

There are configuration variables defined in `run.py` file from line 48 to line 77 within `turn_taking_chat` folder.

1. **max_turns (line 51)**

`max_turns` value is number of total turns during the whole conversation.

2. **manual_validate (line 55)**

`manual_validate` variable is to determine if the validation of results from the workers and paying for the workers should be done automatically. Default value is False, and it automatically validates the result and approves or rejects worker's submission and give reward to them in case they performed well.

If you want to validate the results **manually**, you can set the variable to True.

And then run the validation script manually like below.

```
python ParlAI/parlai/crowdsourcing/tasks/turn_taking_chat/validate_results.py
```

You will be asked to enter the `task_name` which is originally set as "turn-taking-chat" in the configuration file `conf/custom_prebuilt.yaml`

3. **multiple_tasks (line 59)**

`multiple_tasks` variable is to determine if you need to launch a single task or multiple tasks. Default value is True and it reads the task titles from `input.csv` file within the `turn_taking_chat` folder. You should update the `input.csv` file with the details of the task you want to launch without changing the format.

If you need to run a **single task** and set the `multiple_tasks` to `False`, then it will not read task details from CSV and you don't have to update `input.csv` file. Instead you should define the task name, title, description in `conf/custom_prebuilt.yaml` file to display the task details in MTurk. There are already variables defined, you can modify them on your own.

- Database

All the results are saved in the database file of *Mephisto/data* folder within the root directory.

3. Stage 2 (Gathering turn-by-turn annotations from a conversation)

ParlAI/parlai/crowdsourcing/tasks/turn_annotations_static is the folder responsible for achieving the goal.

- Command to run task

```
python parlai/crowdsourcing/tasks/turn_annotations_static/run.py
```

- Configuration Setup

There are configuration variables defined in `run.py` file from line 36 to line 48 within `turn_taking_chat` folder.

1. multiple_tasks (line 45)

`Multiple_tasks` variable is to determine if you want to run a single task or multiple tasks. Default value is `True` and it reads conversation data from the database and run the tasks.

If you want to run a single task, you should specify the conversation data in `task_config/onboarding_in_flight.jsonl` file. You can just replace example data following the existing json structure.

- Annotation Buckets Setup

You can find the configuration defined for annotation buckets in `task_config/annotation_buckets.json` file. Each object in `config` For example, *descriptive* is the id of the bucket and name and description properties of each bucket

```
python
parlai/crowdsourcing/tasks/turn_annotations_static/analysis/compile_results.py
--problem-buckets "descriptive,evaluative,informative,combination,none_all_good"
--results-folders "/home/ubuntu/task/mturk/Mephisto/data/data/runs/NO_PROJECT"
--output-folder /home/ubuntu/outputs/
```