data.json

```
() datajson > ...

1 {("prompt": "When do I have to start the heater?", "completion": "Every day in the morning at 7AM. You should stop it at 2PM"},

2 ("prompt": "Where is the garage remote control?", "completion": "Next to the yellow door, on the key ring"),

3 ("prompt": "Is it necessary to program the scent diffuser every day?", "completion": "The scent diffuser is already programmed, you just need to recharge it when its battery is low"}]
```

set OPEN_API_KEY

openai tools fine_tunes.prepare_data -f data.json

```
Analyzing...

- Your file contains 3 prompt-completion pairs. In general, we recommend having at least a few hundred examples. We've found that performance tend s to linearly increase for every doubling of the number of examples

- All prompts end with suffix '?'

- Your data does not contain a common ending at the end of your completions. Having a common ending string appended to the end of the completion makes it clearer to the fine-tuned model where the completion should end. See https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset for more detail and examples.

- The completion should start with a whitespace character (``). This tends to produce better results due to the tokenization we use. See https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset for more details

Based on the analysis we will perform the following actions:

- [Recommended] Add a suffix ending `\n` to all completions [Y/n]: Y

- [Recommended] Add a whitespace character to the beginning of the completion [Y/n]: Y

Your data will be written to a new JSONL file. Proceed [Y/n]: Y

Wrote modified file to `data_prepared.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:

> openai api fine_tunes.create -t "data_prepared.jsonl"

After you've fine-tuned a model, remember that your prompt has to end with the indicator string `?` for the model to start generating completions, rather than continuing with the prompt. Make sure to include `stop=["\n"]` so that the generated texts ends at the expected place.

Once your model starts training, it'll approximately take 2.48 minutes to train a `curie` model, and less for `ada` and `babbage`. Queue will approximately take half an hour per job ahead of you.
```

openai api fine_tunes.create -t "data_prepared.jsonl" -m curie

```
Upload progress: 100% 417/417 [00:00<00:00, 29 5kit/s]
Uploaded file from data_prepared.jsonl: file-c5U1GnbRTdzqe1qRbVRGsV15
Created fine-tune: ft-2jiLK2t4gGQ3JIYjgNU0Ri7z
Streaming events until fine-tuning is complete...

(Ctrl-C will interrupt the stream, but not cancel the fine-tune)
[2023-11-27 10:25:55] Created fine-tune: ft-2jiLK2t4gGQ3JIYjgNU0Ri7z
[2023-11-27 10:26:04] Fine-tune costs $0.00
[2023-11-27 10:26:05] Fine-tune enqueued. Queue number: 0
[2023-11-27 10:26:07] Fine-tune started
```

openai api fine tunes.list

```
"object": "list",
"data": [
    "object": "fine-tune",
    "id": "ft-2jiLK2t4gGQ3JIYjgNU0Ri7z",
    "hyperparams": {
       "n_epochs": 4,
      "batch_size": 1,
       "prompt_loss_weight": 0.01,
      "learning_rate_multiplier": 0.1
    "organization_id": "org-VCyr7KXWzaElkOsiKBxNpp6h",
    "model": "curie",
"training_files": [
         "object": "file",
         "id": "file-c5U1GnbRTdzqe1qRbVRGsV15",
         "purpose": "fine-tune",
"filename": "data_prepared.jsonl",
         "bytes": 417,
"created_at": 1701109555,
         "status": "processed",
         "status_details": null
      }
    "validation_files": [],
    "result_files": [
         "object": "file",
         "id": "file-k6xkg6C4f8FCwqxSAKuzV91z",
         "purpose": "fine—tune—results",
"filename": "compiled_results.csv",
         "bytes": 744,
"created_at": 1701109650,
         "status": "processed",
         "status_details": null
       }
    "created_at": 1701109555,
    "updated_at": 1701109651,
    "status": "succeeded",
```

Use fine_tuned_model and try it out

```
step,elapsed_tokens,elapsed_examples,training_loss,training_sequence_accuracy,training_token_accuracy 1,33,1,1.4646324474853465,0.0,0.55
2,66,2,1.7572919857455418,0.0,0.23529411764705882
3,91,3,1.6785384513934452,0.0,0.363636363636565
4,116,4,1.6079843709866204,0.0,0.454545454545453
5,149,5,1.2478697110433132,0.0,0.55
6,182,6,1.4710830200091003,0.0,0.29411764705882354
7,215,7,1.4113194398675115,0.0,0.35294117647058826
8,240,8,1.3527744055539368,0.0,0.45454545454545453
9,273,9,1.0532228300604038,0.0,0.65
10,306,10,1.2986056757578628,0.0,0.35294117647058826
11,339,11,1.024046107933391,0.0,0.65
12,364,12,1.289868979950746,0.0,0.454545454545453
13,389,13,1.289868979950746,0.0,0.454545454545454545453
14,422,14,1.024208980605472,0.0,0.65
```