



Articulate motivations

- Quality of the results, latency and cost savings
- 'them, don't tell' kind of tasks

Try to NOT fine-tune

Instead, try to get good result with:

- prompt engineering
- prompt chaining (breaking complex tasks into multiple prompts)
- function calling

The best possible prompt

You've reached the limits of prompt optimization, and identified the problems that the model still has

Edge cases

Examples where the prompted model is not behaving as desired

Training dataset with 50-100 examples

Prepare examples of prompts similar to those you expect to have during the inference time along with the desired responses:

- include your best possible prompt into every example (especially with few <100 training examples)
 - the best guidance and demonstration you have in each prompt that were examples you would need to prepare
 - you won't need to include lengthy instructions at inference time as they will get "baked" into the model
- include training examples that directly target cases where the prompted model is not behaving as desired

Training and testing (validation) split

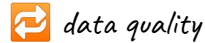
Baseline model accuracy

Training set QA & Cost estimation

Fine-tune!

Analyze the fine-tuned model

- validate the provided model state
 - train loss
 - token accuracy
- check how it performs on edge cases
 - generate from both the base model and the fine-tuned model on a test set, and compare the output side by side



Improve training data

- generate examples to target remaining issues
- double-check the training examples
- data balance + consistency
- make sure your training examples contain all of the information needed for the response

Scaling up the training set

accuracy improvement ~ step function of 2X the dataset size

Hyperparameters tuning

- increase epochs if the model does not follow the training data as much as expected
 - especially for tasks for which there is a single ideal completion (or a small set of ideal completions which are similar) i.e. classification, entity extraction, or structured parsing
- increase learning rate multiplier if there is no convergence



unhappy

happy

