thetechnomist.com

@adelzaalouk

To fine-tune or not to fine-tune? When It's Worth the Investment (and How to Do It Right)

Don't assume fine-tuning is needed by default

- Fine-tuning can be useful but ask your self if you really need it and why.
  - Fine-tuning should not be the default answer (at least not all the time).

Rent to Learn First

- "Try before buy"
  - Understand your use-case, explore the impact of a model (good or bad on your results)
    - Use existing LLMs
    - Explore capabilities

Explore using LLMs as Curators (vs. only generators)

- Use the LLM as your curator, build good responses
  - LLM can also be your planner, in that case, the task is different
- Use RAG for accuracy/relevancy of content when possible

Are you sure you still need to Fine-tune? If not:

- Tweak your RAG configuration knobs
  - Retrieval logic, indexing, chunking, parsing, the right embedding model, embedding context, etc.
- Fine-tune the "embeddings" model before the LLM
  - Tuning embeddings can improve search relevance
- Check data quality
  - Remember Garbage-In/Garbage-out (no amount of training/tuning will save you)

I am sure! Now, How do I make fine-tuning cost-effective?

- **Start Small**
- **Centralize Process/skill fine-tuning**
  - **Example:** Check 7B models, smaller can work but depends on the use-case
  - Experiment with smaller LLMs
    - For curation, they are not that bad.
- **Fine-Tuning-as-a-Service (FTaaS)**
  - Explore Internal AI/ML team providing fine-tuning services and abstractions
  - Platform teams for central access
- Optimize costs with FTaaS
  - Identify what APIs you'd want to expose, make them available for your user behind good UX!

Trade-offs

- If you go the platform route, you might want to optimise the process/flow
  - Functional teams make services cheaper sometimes at the cost of autonomy and self-sufficiency
- Alternative? E.g., Pizza teams
  - Autonomy and self-sufficiency but at a higher cost. Are you willing to make that trade-off?
- Think about Balancing between centralized and decentralized AI teams