Compare Fine-Tuning Using InstructLab with RAG

Estimated time: 15 minutes

Introduction

In the ever-evolving field of natural language processing (NLP), extracting relevant information from large sets of documents is a common challenge. Two prominent methods to tackle this issue are **fine-tuning a model** and **retrieval-augmented generation (RAG)**. Both approaches aim to improve the accuracy and relevance of the information retrieved from documents, but they operate in fundamentally different ways.

This tutorial will guide you through the concepts, advantages, and trade-offs of fine-tuning models versus using RAG, helping you understand which method might be more suitable for your specific use case.

---

1. What is fine-tuning?

Fine-tuning involves taking a pre-trained language model (like BERT, GPT, etc.) and training it further on a specific dataset or task. The idea is to adapt the model's knowledge to better suit the nuances of the target domain.

- **How it works:** During fine-tuning, the model's weights are updated based on the specific training data, allowing it to specialize in certain types of tasks or domains, such as sentiment analysis, entity recognition, or document classification.
- **Example use case:** Suppose you have a large corpus of medical documents. By fine-tuning a pre-trained model on this corpus, you can create a model that is better at understanding and retrieving relevant medical information.

IBM created a library called InstructLab, which allows for the easy fine-tuning of LLMs on your local machine, including on laptops. InstructLab is capable of fine-tuning models in order to impart the model with new knowledge, or to impart it with a new set of skills. Moreover, InstructLab provides a structured way of separating different pieces of knowledge and skill using a taxonomy which allows for the easy augmentation and updating of the information on which one would like to fine-tune the model.

---

2. What is RAG?

RAG is an innovative approach that combines the strengths of information retrieval and generative models. Instead of relying solely on the model's pre-trained knowledge, RAG integrates an external knowledge base (such as a document index) to retrieve relevant information dynamically and then uses this information to generate a response.

- **How it works:** When a query is posed, RAG first retrieves the most relevant documents from the knowledge base. Then, it uses these retrieved documents as context to generate a more accurate and informed response using a generative model.
- **Example use case:** In the same medical document scenario, RAG would retrieve the most relevant documents in response to a query and then generate an answer based on this up-to-date and specific information.

---

3. Key differences and trade-offs

Aspect	Fine-tuning	RAG
Model training	Requires extensive training on the specific domain data.	Minimal training required; relies on a pre-trained model and an external knowledge base.
Adaptability	Highly specialized for a particular domain once trained.	More flexible, as it can retrieve information from various domains without retraining.
Knowledge updating	Updating knowledge requires re-training or fine-tuning the model on new data.	Knowledge base can be updated independently of the model, allowing for more dynamic responses.
Resource requirements	Requires significant computational resources for fine-tuning, especially with large models.	Lower computational cost as the heavy lifting is done by the retrieval step.
Accuracy and relevance	High accuracy in the domain it's trained on but may struggle with out-of-domain queries.	High relevance, especially in domains with large, dynamic knowledge bases.
Use cases	Ideal for tasks with well-defined, static knowledge requirements.	Ideal for tasks where information is constantly evolving or when dealing with broad domains.

---

4. When to use fine-tuning vs. RAG

- **Fine-tuning** is suitable when you have a well-defined domain with specific tasks and a stable knowledge base. It excels in environments where precision and specialized knowledge are paramount.

- **RAG** is the preferred choice when you need to handle queries across a broad or dynamic knowledge base, especially when the information is frequently updated or when you require a more general-purpose model.

---

5. Conclusion

Both fine-tuning and RAG offer powerful tools for retrieving information from documents, but they are best suited for different scenarios. Fine-tuning is ideal for specialized, domain-specific tasks, while RAG shines in environments where information is constantly changing or needs to be pulled from diverse sources. By understanding the strengths and limitations of each approach, you can choose the method that best aligns with your specific requirements.

This tutorial aims to equip you with the knowledge to make informed decisions when it comes to choosing between fine-tuning a model and using RAG for document retrieval tasks. Whether you prioritize precision, adaptability, or resource efficiency, both methods offer valuable solutions in the world of NLP.

Author(s)

Kang Wang

Kang Wang is a data scientist at IBM. He is also a PhD candidate at the University of Waterloo.

Skills Network