Lecture 04: Model Selection and Alternative Approaches

Hamza Salem - Innopolis university

Agenda

Criteria and Considerations for Choosing the Right LLM

Motivating Example: The Tale of Two Models

Role of Compact and Nimble Models

Examples of Successful Smaller Models

The Power of Prompting with General-Purpose Models

Summary and Q&A

Selecting the Right Model

Purpose: What specific tasks or problems is the model expected to address?

Performance: Accuracy, speed, and scalability requirements.

Resource Constraints: Available computational resources and budget.

Customization Needs: Degree of fine-tuning or special training required.

Deployment Environment: Cloud vs. edge devices

Comparative Case Study

Model A: Large, general-purpose model (e.g., GPT-4).

Model B: Smaller, task-specific model (e.g., DistilBERT for sentiment analysis).

Comparison Factors:

Accuracy: Model A might be more accurate in general, but Model B is specialized for its task.

Resource Usage: Model B requires fewer resources.

Cost: Model B is cheaper to deploy and run.

Benefits of Smaller Models

Efficiency: Faster inference times and lower computational requirements.

Cost-Effectiveness: Reduced operational and deployment costs.

Scalability: Easier to deploy on edge devices and mobile platforms.

Flexibility: Often easier to fine-tune and adapt to specific tasks.

Case Studies and Success Stories

DistilBERT: A smaller version of BERT that retains 97% of its language understanding capabilities but with significantly lower resource requirements.

TinyBERT: Designed for mobile and edge applications, providing a balance between performance and efficiency.

ALBERT: A lite version of BERT with fewer parameters but efficient performance, showcasing success in various NLP tasks.

Leveraging General Models for Specialized Tasks

Prompt Engineering: Crafting effective prompts to guide a general-purpose model towards specific tasks.

Flexibility: General models like GPT-4 can be adapted to various tasks through carefully designed prompts without needing fine-tuning.

Cost-Effectiveness: Reduces the need for multiple specialized models, lowering overall costs.

Demo

https://colab.research.google.com/drive/1HY-9ygRvWM5hyqZ6u_88G42OlhCqP7oq?usp=sharing

Key Takeaways

Model Selection: Consider the task, performance needs, resource constraints, and deployment environment.

Compact Models: Smaller models can be highly effective and more cost-efficient for specific tasks.

Prompt Engineering: General models can be optimized for various tasks through effective prompting, reducing the need for multiple specialized models.

Exercise 1

Create a telegram bot that answers based on what you feed, users allowed to send links for articles and bot should scrape content and store it in vector database:

- Bot will answer in two parts:
    - Using Gemini or any LLM API
    - Using distilbert from previous slides.
- Submit code and a screenshot for an example for question and two answer.

Hint: use readability

Exercise 2

1. In top of the previous bot, how can you reduce the number of tokens and that used for each model?
2. Did you used chunking method when you insert documents to vector database?
3. Extend your code to use suitable chunking model to achieve tokens optimization?

Resources

- https://www.perplexity.ai/search/tinybert-miqfRaQEQRGpbA062Uz.ww
- https://github.com/yinmingjun/TinyBERT
- https://www.perplexity.ai/search/tinybert-MEktN12sT6.tjiK5eSw_Xg