Retrieval Augmented Generation and Security Considerations

Summer

Part I: Retrieval Augmented Generation (RAG)

Introduction to RAG

Retrieval Augmented Generation (RAG) enhances traditional large language models (LLMs) by incorporating external retrieval systems. This mitigates common issues such as hallucinations and lack of up-to-date knowledge.

RAG System Architecture

- Combines retrieval and generation components.
- Retrieves relevant documents to augment generation with accurate and context-aware information.

Challenges in RAG

1. **Document Extraction**: Converting various document formats into text using tools like Tesseract or Apache Tika.
2. **Chunking Strategy**: Dividing documents into manageable chunks for retrieval.
3. **Creating Embeddings**: Using models like Alibaba's gte-multilingual-base or Snowflake Arctic Embed 2.0 to compute embeddings.
4. **Retrieval Methods**: Sparse (e.g., BM25), dense, or hybrid retrieval approaches.
5. **Generation**: Fine-tuning LLMs (e.g., Llama 3.2) for effective text generation based on retrieved context.

Applications of RAG

- Enhancing local government services (e.g., Canton of Basel Stadt).
- Augmenting research libraries with contextual search capabilities.

Open Issues

- Ensuring factual correctness and dealing with ambiguous queries.
- Balancing computational demands with real-time response needs.

Part II: Security, Chain of Thought, and Prompt Engineering

Security and Ethical Considerations

- Address risks of model jailbreaking and biases in outputs.
- Promote transparency in model decision-making processes.

Chain of Thought (CoT)

- Instruct models to break down complex tasks step-by-step.
- Example CoT system prompt:

  > "You are a logical assistant. Break down the reasoning step-by-step before providing the final answer."

Prompt Engineering Techniques

- **Simple Prompting**: Define a clear task for the model.
- **Role Prompting**: Assign a role to the model (e.g., "act as a legal advisor").
- **Few-Shot Prompting**: Provide examples within the prompt to guide model responses.

Conclusion

RAG and advancements in security and prompt engineering significantly enhance LLM performance and usability. Addressing ethical considerations and refining interaction methods remain critical to their successful deployment.