Principais contribuições do artigo “Retrieval-Augmented Generation for Large Language Models: A Survey”, de Yunfan Gao et al

Aluno: Leandro Carísio Fernandes

- Conceito de RAG: É uma técnica para melhorar o desempenho de LLMs: documentos externos são divididos em chunks e indexados. Antes de chamar o LLM, é feita uma pesquisa nos chunks para encontrar aqueles mais relevantes que, em seguida, são enviados para o LLM via prompt.
    - Vantagens: é uma forma de fazer o modelo de linguagem considerar conhecimento novo e/ou privado (qualquer conhecimento que o modelo não teve acesso no treinamento).
    - Desvantagens: Aumenta a latência, visto que é necessário inserir uma etapa de busca; custo de pré e pós processamento dos chunks; dificuldades na pesquisa.
- Os autores dividiram RAG em três tipos: “Naive RAG”, “Advanced RAG” e “Modular RAG”. Não achei relevante essa diferenciação. Na prática o que importa é que precisa das fases de *indexação*, *pesquisa* e *geração*.
- A fase de pesquisa é crítica, e é afetada por tudo: se a pesquisa é léxica ou semântica, se os dados são guardados inteiros ou em chunks, o tamanho de cada chunk, se o chunk é por caractere/frase/parágrafo, o tipo de dado guardado etc.
    - São descritas algumas técnicas que podem melhorar a pesquisa (expansão de query, reescrita de query, enriquecimento de metadados, uso de múltiplas BD e escolha de qual usar dependendo da query, uso de embeddings densos e esparsos, fine-tuning do modelo para gerar embeddings melhores etc)
- Os chunks retornados devem ser tratados antes de ir para a fase de geração.
    - Contextos muito longos atrapalham o LLM. Então podem ser passados por uma etapa de seleção dos melhores chunks (por exemplo, reranking) ou de compressão em que os chunks são resumidos.
- Os autores comentaram algumas tarefas típicas que envolvem RAG: *question answering*, *information extraction, dialog generation, code search* etc.