ZTM RAG: RAG Fundamentos

terça-feira, 1 de julho de 2025 13:01

Sobre LLMs (Large Language Models)

Key Methods:
- Zero Shot Learning: Modelo prevê a resposta apenas com uma descrição da pergunta.
- One-shot Learning: Além da descrição do problema, é dado um exemplo do que o modelo deve fazer.
- Few-shot Learning: Modelo recebe a descrição do problema e alguns exemplos do que deve fazer.

Em fine tuning, geralmente é usado o Few-shot Learning. Isso geralmente é bem caro e não usado mais.
Por isso, fine tuning se tornou legacy e agora os modelos usam centenas de bilhões de parâmetros para saber como resolver os prompts sem precisar de fine tuning.
Essa quantidade de dados gerou um biases de sentimento em tarefas raciais por não terem limpado os dados corretamente.

Aumentar o tamanho do modelo, o tamanho do dataset e o poder computacional melhora o resultado, mas também aumenta o custo.

As LLMs conseguiram ser melhores que os humanos quando implementaram o método SAFE (Self Attention Fine Tuning and Evaluation), que permite os modelos verificarem a acurácia do próprio conteúdo.

RAG

O que é RAG?
É um modelo hibrido que melhora a geração de texto usando informações de documentos, levando a respostas mais precisas.

Componentes principais de uma arquitetura básica de RAG
- **Input Query**
  A pergunta do usuário ou prompt que inicia os outros processos.
- **Retriever**
  Pega documentos ou passagens relevantes de um grande corpus baseado no input query. Isso gera os **Retrieved Documents**, que proveem o contexto e informação necessária para gerar uma resposta precisa e relevante.
- **Generator**
  Cria uma resposta baseada na input query e nos retrieved documents, usando LLMs.

FAISS
É uma biblioteca criada pelo Facebook para busca eficiente de similaridade entre vetores de alta-dimensão. Como um bibliotecário muito bom numa biblioteca gigante.
É bom pela velocidade, escalabilidade e flexibilidade.

Como funciona: Indexando dados em estruturas otimizadas para busca rápida de similaridade.

Tipos de FAISS Indexes

- **Flat Index**
  Simples metodo de força bruta que checa todos os vetores. Ideal para pequenos datasets.
- **IVF** (Inverted File)
  Divide os dados em clusters para busca rápida. Ideal para grandes datasets.
- **HNSW** (Hierarchical Navigable Small Worlds)
  Usa um graph base aproch para buscas bem rápidas. Ideal para aplicações de tempo real.

Generation Model Parameters

São settings que controla o comportamento do modelo de geração de texto. Modificar esses parâmetros podem melhorar significamente a performance e a qualidade dos modelos.
Entre esses parâmetros, estão temperature, top-k sampling, top-p sampling, repetition penalty, sampling mode.

- **Temperature**
  Controla a aleatoriedade do texto gerado, balanceando entre outputs focados e criativos.
  Valores menores são mais focados e valores maiores são mais aleatórios e criativos.
  Valor default é "1.0".
- **Top-K Sampling**
  Limita a escolha da próxima palavra possível para as mais prováveis.
  Ex: Top-K = 50 = apenas as 50 palavras mais prováveis serão consideradas na hora de gerar a próxima palavra.
- **Top-p (Nucleus) Sampling**
  Limita a gama de palavras consideradas para a próxima palavra a gerar.
  Se "top_p = 1.0", significa que 100% das palavras são consideradas para a próxima palavra.
  É semelhante ao Top-K.
- **Repetition Penalty**
  Reduz as frases repetitivas, fazendo respostas mais diversas e humanas.
  Valor padrão = 1.0 - Acima disso ele penaliza e abaixo ele recompensa repetição.
- **Sampling Mode**
  Adiciona mais aleatoriedade, criando textos mais variáveis e criativos.

LongRAG

LongRAG utiliza mais tokens do que a forma tradicional de rag para gerar contexto para as respostas, reduzindo a fragmentação. RAG normalmente utiliza ~100 tokens de contexto. LongRAG utiliza ~4000 tokens (que da aproximadamente 3 mil palavras), fornecendo um contexto muito maior, gerando uma resposta muito mais completa.
É fácil de aplicar.

LightRAG

LightRAG combina RAG tradicional com índice de grafo de conhecimento, integrando tanto componentes textuais quanto estruturais (entidades e relações). Ideal para dados complexos.
Não é tão simples de aplicar.

LightRAG vs Outros RAGs em diferentes datasets

Table 1: Win rates (%) of baselines v.s. LightRAG across four datasets and four evaluation dimensions.

	Agriculture		**CS**		**Legal**		**Mix**	
	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**
Comprehensiveness	32.69%	67.31%	35.44%	64.56%	19.05%	80.95%	36.36%	63.64%
Diversity	24.09%	75.91%	35.24%	64.76%	10.98%	89.02%	30.76%	69.24%
Empowerment	31.35%	68.65%	35.48%	64.52%	17.59%	82.41%	40.95%	59.05%
Overall	33.30%	66.70%	34.76%	65.24%	17.46%	82.54%	37.59%	62.40%
	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**
Comprehensiveness	32.05%	67.95%	39.30%	60.70%	18.57%	81.43%	38.89%	61.11%
Diversity	29.44%	70.56%	38.71%	61.29%	15.14%	84.86%	28.50%	71.50%
Empowerment	32.51%	67.49%	37.52%	62.48%	17.80%	82.20%	43.96%	56.04%
Overall	33.29%	66.71%	39.03%	60.97%	17.80%	82.20%	39.61%	60.39%
	HyDE	**LightRAG**	HyDE	**LightRAG**	HyDE	**LightRAG**	HyDE	**LightRAG**
Comprehensiveness	24.39%	75.61%	36.49%	63.51%	27.68%	72.32%	42.17%	57.83%
Diversity	24.96%	75.34%	37.41%	62.59%	18.79%	81.21%	30.88%	69.12%
Empowerment	24.89%	75.11%	34.99%	65.01%	26.99%	73.01%	45.61%	54.39%
Overall	23.17%	76.83%	35.67%	64.33%	27.68%	72.32%	42.72%	57.28%
	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**
Comprehensiveness	45.56%	54.44%	45.98%	54.02%	47.13%	52.87%	51.86%	48.14%
Diversity	19.65%	80.35%	39.64%	60.36%	25.55%	74.45%	35.87%	64.13%
Empowerment	36.69%	63.31%	45.09%	54.91%	42.81%	57.19%	52.94%	47.06%
Overall	43.62%	56.38%	45.98%	54.02%	45.70%	54.30%	51.86%	48.14%

NaiveRAG = RAG padrão

A porcentagem representa o quanto de vezes o modelo ganhou em relação ao outro modelo. Percebe-se que apenas quando os dados são mistos que o LightRAG perdeu e unicamente para o GraphRAG.