ChatGPT

<!-- README.md

Este arquivo fornece uma visão geral completa do projeto **LLM Ecosystem Explorer**, combinando um diagrama interativo com um glossário detalhado para auxiliar no aprendizado e desenvolvimento autodidata com grandes modelos de linguagem. -->

LLM Ecosystem Explorer

Explore o ecossistema de **Grandes Modelos de Linguagem (LLMs)** através de um diagrama interativo e um glossário avançado. Este projeto foi construído para estudantes e desenvolvedores autodidatas que desejam entender, testar e aplicar técnicas de modelagem, treinamento, inferência, recuperação de informações (RAG) e engenharia de prompts.

> **TL;DR:** Abra `llm_diagram/index.html` no seu navegador, clique nos nós para ler descrições detalhadas e consulte o glossário em `advanced_glossary.md` para um estudo mais aprofundado.

Principais Funcionalidades

- **Diagrama Interativo**: Seis visualizações temáticas (Visão Geral, Modelos, Treinamento, RAG, Inferência e Prompting) permitem navegar por diferentes componentes do ecossistema. Os nós exibem informações contextuais quando selecionados e as conexões mostram o fluxo de dados entre etapas.

- **Experiência de Navegação**: Amplie, reduza ou arraste o canvas. Use a busca para filtrar e destacar nós em tempo real. Ative a animação de fluxo para ver partículas representando o movimento de dados entre componentes.

- **Glossário Avançado**: O arquivo `advanced_glossary.md` fornece explicações detalhadas sobre modelos de fundação, técnicas de treinamento (RLHF, DPO, LoRA/QLoRA, distilação), parâmetros de inferência, recuperação aumentada por geração (RAG) e engenharia de prompts. Cada conceito é apoiado por referências a artigos e papers técnicos.

- **Fluxo de Desenvolvimento Solo**: O glossário oferece recomendações práticas para quem está construindo projetos sozinho, incluindo boas práticas de fine-tuning eficiente, utilização de RAG para evitar alucinações e estratégias de prompting para melhorar resultados.

Começando

1. **Clone ou baixe este repositório.**
2. Navegue até a pasta `llm_diagram/`.
3. Abra o arquivo `index.html` em um navegador moderno (Chrome, Edge ou Firefox). Não há dependências externas ou servidor; tudo roda no front-end.

4. Use os botões do cabeçalho para trocar de visualização, pesquise por conceitos e clique nos nós para ler explicações aprofundadas no painel lateral.

Exemplos de Uso

Caso	Passos	O que você aprende
Ajuste fino eficiente com LoRA/QLoRA	Selecione *Treinamento* > clique em `LoRA` e `QLoRA` > leia as descrições e conexões	Entenda como LoRA mantém pesos congelados e injeta matrizes de baixa dimensão, e como QLoRA adiciona quantização para ajustar modelos gigantes [1].
Construção de base de conhecimento com RAG	Selecione *RAG* > siga a sequência `Documentos → Chunking → Embeddings → Vector DB`	Aprenda como dividir documentos em chunks, converter em embeddings, armazenar em bancos vetoriais e recuperar contexto para melhorar a precisão [2].
Comparar estratégias de inferência	Em *Inferência*, explore `Temperatura`, `Top-K`, `Top-P` e estratégias `Beam Search` e `Greedy`	Veja como ajustar os parâmetros de geração para balancear criatividade e precisão.
Engenharia de Prompts	Acesse *Prompting* > clique em `Zero-Shot`, `Few-Shot`, `Chain-of-Thought`, etc.	Descubra como diferentes técnicas de prompting influenciam o comportamento do modelo, com dicas práticas de aplicação [3] [4].

📚 Glossário de Modelos de Linguagem

Para um estudo aprofundado, leia o arquivo `advanced_glossary.md`, que detalha:

- **Modelos de Fundação, Multimodais e SLMs** – diferenças entre modelos gerais, multimodais e modelos compactos (Small Language Models).
- **Treinamento** – conceitos de pré-treino, RLHF, DPO, dados sintéticos, destilação, LoRA/QLoRA/PEFT e checkpoints [5] [6].
- **Prompting & Inferência** – técnicas de zero-shot, few-shot e chain-of-thought; ajustes de temperatura, top-k, top-p e uso de seeds [3].
- **Recuperação Aumentada por Geração (RAG)** – como integrar busca semântica com modelos generativos para reduzir alucinações e citar fontes [7].

Essas seções são acompanhadas de dicas práticas para desenvolvimento autodidata, incluindo integração com ferramentas de design (Figma), frameworks (React) e orquestradores de workflows (n8n).

🛠 Estrutura do Projeto

```
.
├── advanced_glossary.md   # Glossário detalhado e referenciado
├── advanced_glossary.svg  # Diagrama avançado utilizado neste README
```

```
├── llm_diagram/           # Aplicação interativa (HTML, CSS, JS)
│   ├── index.html         # Diagrama interativo completo
│   └── README.md          # Documentação técnica da aplicação
└── README.md             # Este arquivo
```

📦 Publicação no GitHub

Se você pretende publicar este projeto no GitHub, siga estas dicas para um repositório bem estruturado:

1. **Nomeie o repositório** com algo claro, por exemplo `llm-ecosystem-explorer`.
2. **Inclua este README** na raiz e mantenha a documentação sincronizada com as funcionalidades do diagrama e do glossário.
3. **Adicione uma licença**, como MIT, para definir permissões de uso.
4. **Use Releases e Tags** para marcar versões estáveis (por exemplo, `v1.0`).
5. **Organize o código** em subpastas (`src/`, `assets/`) se expandir a aplicação. Atualmente, todo o código do front-end está em um único arquivo (`index.html`) para facilitar a distribuição.
6. **Inclua imagens e prints** no repositório para tornar a apresentação mais intuitiva. O arquivo `advanced_glossary.svg` serve como um print estático do diagrama.

Contribuindo

Contribuições são bem-vindas! Se você encontrar um problema ou tiver sugestões de melhoria:

1. Abra uma *issue* descrevendo a questão ou a nova funcionalidade.
2. *Fork* o repositório e crie uma branch para sua alteração.
3. Envie um *pull request* com uma descrição clara do que foi alterado.

Antes de contribuir, leia o glossário para manter consistência no vocabulário e nas definições técnicas.

Licença

Este projeto é licenciado sob os termos da licença MIT. Consulte o arquivo `LICENSE` para mais detalhes.

Sinta-se à vontade para explorar, aprender e adaptar este projeto às suas necessidades. Bom estudo e boas criações com modelos de linguagem!

---

[1] [2106.09685] LoRA: Low-Rank Adaptation of Large Language Models
https://arxiv.org/abs/2106.09685

[2] Integrated vector database - Azure Cosmos DB | Microsoft Learn
https://learn.microsoft.com/en-us/azure/cosmos-db/vector-database

[3] Few-Shot Prompting | Prompt Engineering Guide
https://www.promptingguide.ai/techniques/fewshot

4 [2201.11903] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
https://arxiv.org/abs/2201.11903

5 Illustrating Reinforcement Learning from Human Feedback (RLHF)
https://huggingface.co/blog/rlhf

6 [2305.18290] Direct Preference Optimization: Your Language Model is Secretly a Reward Model
https://arxiv.org/abs/2305.18290

7 What Is Retrieval-Augmented Generation aka RAG | NVIDIA Blogs
https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/