各章引用网址列表

# 第1章 在WSL2上搭建GPU Linux Server深度学习环境

1、数里乾坤——用R与Python玩转大数据

https://zhuanlan.zhihu.com/p/640437583

2、优刻得

https://www.ucloud.cn/

3、GPU算力查询网址

https://developer.nvidia.com/cuda-gpus

4、拯救者Y9000X

https://activity.lenovo.com.cn/xiaofei/zjz/hdy.html

5、Tensorflow Linux Tested build configurations

https://tensorflow.google.cn/install/source?hl=en#linux

6、Tensorflow Winodws Tested build configurations

https://tensorflow.google.cn/install/source\_windows?hl=en#gpu

7、Docker安装运行报错wsl问题排查方案

https://www.cnblogs.com/bokemoqi/p/17926296.html

8、Windows10/11家庭版开启Hyper-V虚拟机功能详解

https://zhuanlan.zhihu.com/p/667571538

9、WSL 使用教程

https://www.jianshu.com/p/39a5b2e002b6

10、Getting Started with CUDA on WSL 2

https://docs.nvidia.com/cuda/wsl-user-guide/index.html#getting-started-with-cuda-on-wsl-2

11、CUDA 12.3下载地址

https://developer.nvidia.com/cuda-12-3-2-download-archive?target\_os=Linux&target\_arch=x86\_64&Distribution=WSL-Ubuntu&target\_version=2.0&target\_type=deb\_local

12、Tensorflow Tested build configurations

https://tensorflow.google.cn/install/source?hl=en#linux

13、Getting Started with CUDA on WSL 2

https://docs.nvidia.com/cuda/wsl-user-guide/index.html#getting-started-with-cuda-on-wsl-2

14、cuDNN Support Matrix

https://docs.nvidia.com/deeplearning/cudnn/latest/reference/support-matrix.html#support-matrix

15、Ubuntu 20.04(linux) cuda(12)+cudnn的deb方式安装以及验证

https://blog.csdn.net/qq\_32033383/article/details/135015041

16、cuDNN9.4下载地址

https://developer.nvidia.com/cudnn-9-4-0-download-archive?target\_os=Linux&target\_arch=x86\_64&Distribution=Ubuntu&target\_version=22.04&target\_type=deb\_network

17、下载Linux版64-Bit (x86) Installer17

https://repo.anaconda.com/archive/Anaconda3-2024.02-1-Linux-x86\_64.sh

18、PyTorch主页

https://pytorch.org/

19、Jupyter Lab设置切换虚拟环境

https://blog.csdn.net/CUFEECR/article/details/123987150

20、在MNIST上用Pytorch跑跑GPU

https://blog.csdn.net/song5bai/article/details/116358451

21、HanLP

https://github.com/hankcs/HanLP

22、何晗

https://github.com/hankcs

23、自然语义科技有限公司

https://www.hanlp.com/

24、HanLP 安装文档

https://hanlp.hankcs.com/docs/install.html

25、HanLP Demo

https://github.com/hankcs/HanLP/tree/doc-zh/plugins/hanlp\_demo/hanlp\_demo/zh

26、HanLP Demo API

https://hanlp.hankcs.com/docs/

27、HanLP多语种分句模型

https://github.com/hankcs/HanLP/blob/master/plugins/hanlp\_demo/hanlp\_demo/sent\_split.py

28、HanLP基于规则的分句函数

https://github.com/hankcs/HanLP/blob/master/hanlp/utils/rules.py#L19

29、Nvidia GeForce RTX 2060

https://www.nvidia.cn/geforce/graphics-cards/rtx-2060/

30、Nvidia A100

https://images.nvidia.cn/aem-dam/en-zz/Solutions/data-center/a100/nvidia-a100-datasheet-nvidia-a4-2188504-r5-zhCN.pdf

31、Nvidia Tesla T4

https://www.nvidia.cn/content/dam/en-zz/zh\_cn/Solutions/Data-Center/tesla-t4/nvidia-t4-datasheet-a4-nvidia-772234-r14-lr-cn.pdf

32、Neo4j Graph Data Science Compatibility matrix

https://github.com/neo4j/graph-data-science

33、Neo4j Community Linux Tarball Installation

https://neo4j.com/docs/operations-manual/current/installation/linux/tarball/#installation-linux-tarball-service

34、How to Install CUDA on Ubuntu 22.04 | Step-by-Step

https://www.cherryservers.com/blog/install-cuda-ubuntu

35、How to Install NVIDIA Drivers on Ubuntu 24.04

https://linuxconfig.org/how-to-install-nvidia-drivers-on-ubuntu-24-04

**第2章 微软GraphRAG**

1、bgm-3 embedding模型

https://ollama.com/library/bge-m3

2、all-minilm embedding模型

https://ollama.com/library/all-minilm

3、Visualizing and Debugging Your Knowledge Graph

https://github.com/microsoft/graphrag/blob/main/docs/visualization\_guide.md

4、《悟空传》的前7章

https://dushu.baidu.com/pc/detail?gid=4305630473

5、Auto Prompt Tuning

https://microsoft.github.io/graphrag/prompt\_tuning/auto\_prompt\_tuning/

6、GraphRAG自动Prompt Tuning

https://mp.weixin.qq.com/s/69MeYny5nZmfjS1b4QUvTg

7、开发GraphRAG（知识图谱检索增强生成）应用

https://zhuanlan.zhihu.com/p/704919102

8、ms\_graphrag\_import.ipynb

https://github.com/tomasonjo/blogs/blob/master/llm/ms\_graphrag\_import.ipynb

9、global\_search.ipynb

https://github.com/microsoft/graphrag/blob/main/docs/examples\_notebooks/global\_search.ipynb

10、local\_search.ipynb

https://github.com/microsoft/graphrag/blob/main/docs/examples\_notebooks/local\_search.ipynb

11、issue #1335 TypeError: Query column vector must be a vector

https://github.com/microsoft/graphrag/issues/1335

12、命令行查询工具源码

https://github.com/microsoft/graphrag/blob/main/graphrag/cli/query.py

13、drift\_search.ipynb

https://github.com/microsoft/graphrag/blob/main/docs/examples\_notebooks/drift\_search.ipynb

14、Ollama主页

https://github.com/ollama/ollama

15、Qwen2.5

https://ollama.com/library/qwen2.5:7b

16、issue#657 Support model providers other than OpenAI and Azure

https://github.com/microsoft/graphrag/issues/657

**第3章 Neo4j GraphRAG**

1、Neo4j Knowledge Graph Builder主页

https://github.com/neo4j-labs/llm-graph-builder

2、Neo4j GrapRAG的生态

https://neo4j.com/labs/genai-ecosystem/

3、Neo4j Knowledge Graph Builder简介

https://neo4j.com/labs/genai-ecosystem/llm-graph-builder/

4、NeoConverse简介

https://neo4j.com/labs/genai-ecosystem/neoconverse/

5、NeoConverse主页

https://github.com/neo4j-labs/neoconverse

6、GenAI Stack简介

https://neo4j.com/labs/genai-ecosystem/genai-stack/

7、Get Started With GraphRAG: Neo4j’s Ecosystem Tools

https://neo4j.com/developer-blog/graphrag-ecosystem-tools/

8、LLM Knowledge Graph Builder: From Zero to GraphRAG in Five Minutes

https://neo4j.com/developer-blog/graphrag-llm-knowledge-graph-builder/

9、LangChain Neo4j Integration

https://neo4j.com/labs/genai-ecosystem/langchain/

10、LlamaIndex Neo4j Integration

https://neo4j.com/labs/genai-ecosystem/llamaindex/

11、Langchain4j Neo4j Integration

https://neo4j.com/labs/genai-ecosystem/langchain4j/

12、Docker

https://www.docker.com/

13、Docker Desktop

https://www.docker.com/products/docker-desktop/

14、Install Docker Engine on Ubuntu

https://docs.docker.com/engine/install/ubuntu/

15、Overview of installing Docker Compose

https://docs.docker.com/compose/install/

16、Install the Compose plugin

https://docs.docker.com/compose/install/linux/#install-using-the-repository

17、目前国内可用Docker镜像源汇总

https://cloud.tencent.com/developer/article/2459822

18、Daemon proxy configuration

https://docs.docker.com/engine/daemon/proxy/#systemd-unit-file

19、LangChain

https://python.langchain.com/

20、LangSmith

https://smith.langchain.com/

21、Issue#839《Does NOT work with Neo4J Community Edition

https://github.com/neo4j-labs/llm-graph-builder/issues/839

22、HuggingFace主页

https://huggingface.co/

23、Embedding model all-MiniLM-L6-v2

https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

24、Embedding model BAAI/bge-m3

https://huggingface.co/BAAI/bge-m3

25、LLMGraphTransformer文档

https://python.langchain.com/api\_reference/experimental/graph\_transformers/langchain\_experimental.graph\_transformers.llm.LLMGraphTransformer.html

26、LLMGraphTransformer源码

https://github.com/langchain-ai/langchain-experimental/blob/main/libs/experimental/langchain\_experimental/graph\_transformers/llm.py

27、Leiden社区发现算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/leiden/

28、SLLPA(Speaker-Listener Label Propagation)社区发现算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/sllpa/

29、Neo4j Knowledge Graph Builder集成测试程序

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/test\_integrationqa.py

30、Neo4j Knowledge Graph Builder backend requirements.txt

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/requirements.txt

31、Neo4j Knowledge Graph Builder backend API

https://github.com/neo4j-labs/llm-graph-builder/blob/main/docs/backend/backend\_docs.adoc

**第4章 开发GraphRAG应用**

1、微软GraphRAG论文

https://arxiv.org/abs/2404.16130

2、Tomaz Bratanic

https://bratanic-tomaz.medium.com/

3、Implementing ‘From Local to Global’ GraphRAG with Neo4j and LangChain: Constructing the Graph

https://medium.com/neo4j/implementing-from-local-to-global-graphrag-with-neo4j-and-langchain-constructing-the-graph-73924cc5bab4

4、《用Neo4j与LangChain实现从局部到全局的RAG：建立知识图谱》

https://zhuanlan.zhihu.com/p/709060837

5、Jupyter Notebook的源码文件

https://github.com/tomasonjo/blogs/blob/master/llm/ms\_graphrag.ipynb

6、LangChain

https://www.langchain.com/langchain

7、LlamaIndex

https://www.llamaindex.ai/

8、《LlamaIndex 或 LangChain,哪个更适合作为RAG框架？》

https://blog.csdn.net/CSDNDN/article/details/139596103

9、LangSmith

https://www.langchain.com/langsmith

10、LangGraph

https://www.langchain.com/langgraph

11、Neo4j Cummunity的汉化版

https://we-yun.com/blog/prod-56.html

12、《开发GraphRAG（知识图谱检索增强生成）应用》

https://zhuanlan.zhihu.com/p/704919102

13、HanLP

https://github.com/hankcs/HanLP

14、《揭秘大模型提升秘诀：RAG系统中的文本分块策略》

https://mp.weixin.qq.com/s/HaRKys1A98cF9N1dSoEmGw

15、Neo4jGraph文档

https://python.langchain.com/v0.2/api\_reference/community/graphs/langchain\_community.graphs.neo4j\_graph.Neo4jGraph.html

16、Neo4j KGBuilder make\_relationships.py源码

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/src/make\_relationships.py

17、Neo4j KGBuilder的llm.py源码

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/src/llm.py

18、LLMGraphTransformer文档

https://python.langchain.com/api\_reference/experimental/graph\_transformers/langchain\_experimental.graph\_transformers.llm.LLMGraphTransformer.html

19、LLMGraphTransformer源码

https://python.langchain.com/api\_reference/\_modules/langchain\_experimental/graph\_transformers/llm.html

20、GraphDocument文档

https://python.langchain.com/api\_reference/community/graphs/langchain\_community.graphs.graph\_document.GraphDocument.html

21、Relationship文档

https://python.langchain.com/api\_reference/community/graphs/langchain\_community.graphs.graph\_document.Relationship.html#langchain\_community.graphs.graph\_document.Relationship

22、Neo4j KGBuilder合并结点的实现

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/src/make\_relationships.py#L15

23、Neo4jGraph源码

https://python.langchain.com/v0.2/api\_reference/\_modules/langchain\_community/graphs/neo4j\_graph.html#Neo4jGraph.add\_graph\_documents

24、Hugginface

https://link.zhihu.com/?target=https%3A//huggingface.co/models

25、《如何选择RAG的Embedding模型？》

https://techdiylife.github.io/blog/blog.html?category1=c02&blogid=0047

26、《中文Embedding模型优劣数据评测分析报告》

https://zhuanlan.zhihu.com/p/679166797

27、北京智源研究院

https://www.baai.ac.cn/

28、BAAI/bge-m3模型

https://huggingface.co/BAAI/bge-m3

29、BAAI/bge-m3模型Github主页

https://github.com/FlagOpen/FlagEmbedding/blob/master/README\_zh.md

30、KNN算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/knn/

31、WCC算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/wcc/

32、《如何从模型中返回结构化数据》

https://python.langchain.ac.cn/docs/how\_to/structured\_output/

33、apoc.refactor.mergeNodes的文档

https://neo4j.com/labs/apoc/4.0/overview/apoc.refactor/apoc.refactor.mergeNodes

34、Leiden算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/leiden/

35、SLLPA(Speaker-Listener Label Propagation)算法

https://neo4j.com/docs/graph-data-science/2.12/algorithms/sllpa/

36、《将微软GraphRAG集成到Neo4j中》

https://zhuanlan.zhihu.com/p/713201715

37、Integrating Microsoft GraphRAG into Neo4j

https://towardsdatascience.com/integrating-microsoft-graphrag-into-neo4j-e0d4fa00714c

38、《Implementing RAG: How to Write a Graph Retrieval Query in LangChain》

https://neo4j.com/developer-blog/rag-graph-retrieval-query-langchain/

39、Neo4j向量索引的实例化

https://python.langchain.com/v0.2/api\_reference/community/vectorstores/langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.html#langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.from\_existing\_index

40、Neo4jVector.similarity\_search文档

https://python.langchain.com/v0.2/api\_reference/community/vectorstores/langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.html#langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.similarity\_search

41、Customize response with retrieval query

https://python.langchain.com/docs/integrations/vectorstores/neo4jvector/#customize-response-with-retrieval-query

**第5章 Agent开发**

1、Conversational RAG

https://python.langchain.com/docs/tutorials/qa\_chat\_history/

2、Agentic RAG

https://langchain-ai.github.io/langgraph/tutorials/rag/langgraph\_agentic\_rag/

3、Neo4jVector的文档

https://python.langchain.com/api\_reference/community/vectorstores/langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.html#langchain\_community.vectorstores.neo4j\_vector.Neo4jVector.as\_retriever

4、LangSmith

https://smith.langchain.com/

5、函数create\_react\_agent() API

https://langchain-ai.github.io/langgraph/reference/prebuilt/

6、Build an Agent

https://python.langchain.com/docs/tutorials/agents/

7、LangGraph

https://langchain-ai.github.io/langgraph/

8、Graph Definitions

https://langchain-ai.github.io/langgraph/reference/graphs/

9、LanagGraph Prebuilt tools\_condition

https://langchain-ai.github.io/langgraph/reference/prebuilt/#langgraph.prebuilt.tool\_node.tools\_condition

10、How to manage conversation history

https://langchain-ai.github.io/langgraph/how-tos/memory/manage-conversation-history/

11、How to control graph recursion limit

https://langchain-ai.github.io/langgraph/how-tos/recursion-limit/

12、How to create tools

https://python.langchain.com/docs/how\_to/custom\_tools/

13、Tool Calling with LangChain

https://blog.langchain.dev/tool-calling-with-langchain/

14、Tool calling

https://python.langchain.com/docs/how\_to/tool\_calling/

**第6章 在GraphRAG中应用国产大模型**

1、LangChain集成的Chat模型列表

<https://python.langchain.com/docs/integrations/chat/>

2、DeepSeek R1/V3

https://github.com/deepseek-ai

3、LangChain集成的Embedding模型列表

https://python.langchain.com/docs/integrations/text\_embedding/

4、LangChain文心一言Chat模型文档

https://python.langchain.com/docs/integrations/chat/baidu\_qianfan\_endpoint/

5、LangChain讯飞星火Chat模型文档

https://python.langchain.com/docs/integrations/chat/sparkllm/

6、LangChain通义千问Chat模型文档

https://python.langchain.com/docs/integrations/chat/tongyi/

7、LangChain腾讯混元Chat模型文档

https://python.langchain.com/docs/integrations/chat/tencent\_hunyuan/

8、LangChain文心一言Embedding模型文档

https://python.langchain.com/docs/integrations/text\_embedding/baidu\_qianfan\_endpoint/

9、LangChain讯飞星火Embedding模型文档

https://python.langchain.com/docs/integrations/text\_embedding/sparkllm/

10、LangChain通义千问Embedding模型文档

https://python.langchain.com/docs/integrations/text\_embedding/dashscope/

11、LangChain Huggingface Embedding模型文档

https://python.langchain.com/docs/integrations/text\_embedding/huggingfacehub/

12、Refactor ChatHunyuan and support Hunyuan Embedding

https://github.com/langchain-ai/langchain/pull/23160/commits/4d7cc88c2368c77593fbb38d71127d574f623ff2

13、Add function call support in Sparkllm chat model

https://github.com/langchain-ai/langchain/pull/20607/files/454d92a2bd95be7bcd45564b89dbb74c5ce77243

14、Function calling

https://github.com/deepseek-ai/DeepSeek-R1/issues/9

15、《DeepSeek R1测试之八 各大云平台调用》

https://zhuanlan.zhihu.com/p/22540281019/

**第7章 本地部署LLM**

1、Huggingface Installation文档的offline mode

https://huggingface.co/docs/transformers/installation#offline-mode

2、qwen2.5

https://ollama.com/library/qwen2.5

3、openbmb/MiniCPM3-4B

https://huggingface.co/openbmb/MiniCPM3-4B

4、Run custom GGUF model on Ollama

https://zohaib.me/run-custom-gguf-model-on-ollama/

5、MiniCPM3的函数调用例子文档

https://github.com/OpenBMB/MiniCPM/tree/main/demo/minicpm3/function\_call

6、vLLM

https://github.com/vllm-project/vllm

7、vLLM支持的LLM列表

https://docs.vllm.ai/en/v0.6.2/models/supported\_models.html

8、Running vLLM on Pascal

https://github.com/jasonacox/TinyLLM/tree/main/vllm#running-vllm-on-pascal

9、LLM的文档《Installation: Full Build》

https://docs.vllm.ai/en/latest/getting\_started/installation.html#full-build-with-compilation

10、Support for compute capability <7.0

https://github.com/vllm-project/vllm/issues/963

11、Enable support for Pascal GPUs

https://github.com/vllm-project/vllm/pull/4290

12、OpenAI Compatible Server: How to write a tool parser plugin

https://docs.vllm.ai/en/latest/serving/openai\_compatible\_server.html#how-to-write-a-tool-parser-plugin

13、LangChain vLLM Chat

https://python.langchain.com/docs/integrations/chat/vllm/

14、LangChain ChatHuggingFace

https://python.langchain.com/docs/integrations/chat/huggingface/

15、HuggingFaceEndpoint

https://python.langchain.com/api\_reference/huggingface/llms/langchain\_huggingface.llms.huggingface\_endpoint.HuggingFaceEndpoint.html

16、HuggingFacePipeline

https://python.langchain.com/api\_reference/huggingface/llms/langchain\_huggingface.llms.huggingface\_pipeline.HuggingFacePipeline.html

17、ChatHuggingFace源码

https://python.langchain.com/api\_reference/huggingface/chat\_models/langchain\_huggingface.chat\_models.huggingface.ChatHuggingFace.html#langchain\_huggingface.chat\_models.huggingface.ChatHuggingFace.bind\_tools

18、issue#22379《Tools do not work with HuggingFace》

https://github.com/langchain-ai/langchain/issues/22379

19、Llama.cpp

https://github.com/abetlen/llama-cpp-python

20、LangChain Chat Llama.cpp

https://python.langchain.com/docs/integrations/chat/llamacpp/

21、Issue#957《Generic Function Calling》

https://github.com/abetlen/llama-cpp-python/pull/957

22、Issue#1351《Improve function calling (auto selection, parallel functions)》

https://github.com/abetlen/llama-cpp-python/pull/1351

23、ChatLlamaCpp API

https://python.langchain.com/api\_reference/community/chat\_models/langchain\_community.chat\_models.llamacpp.ChatLlamaCpp.html#langchain\_community.chat\_models.llamacpp.ChatLlamaCpp.bind\_tools

**第8章 开发GraphRAG APP**

1、Python FastAPI

https://github.com/fastapi/fastapi

2、Python Uvicorn WEB服务器

https://www.uvicorn.org/

3、starlette项目

https://github.com/encode/starlette

4、Postman

https://www.postman.com/

5、Streamlit

https://streamlit.io/

6、Flask

https://github.com/pallets/flask

7、《精通Shiny》

http://www.oreilly.com.cn/index.php?func=book&isbn=978-7-5766-0656-0

8、Shiny for Python

https://shiny.posit.co/py/

9、Rstudio Server在线文档

https://posit.co/download/rstudio-server/

10、《RStudio IDE User Guide》

https://docs.posit.co/ide/user/

11、Shiny Server在线文档

https://posit.co/download/shiny-server/

12、配置Shiny Server为用户发布模式

https://docs.posit.co/shiny-server/#host-per-user-application-directories

13、Shiny Server Administrator's Guide

https://docs.posit.co/shiny-server/

14、《Shiny for Python APP开发》

https://zhuanlan.zhihu.com/p/658670798

15、《墨尔本房价回归模型Shiny for Python APP》

https://zhuanlan.zhihu.com/p/658996965

16、Shiny for Python在线文档

https://shiny.posit.co/py/docs/overview.html

**第9章 GraphRAG 应用评估**

1、Ragas项目主页

https://github.com/explodinggradients/ragas/

2、RAGAS: Automated Evaluation of Retrieval Augmented Generation

https://arxiv.org/abs/2309.15217

3、Integrations

https://docs.ragas.io/en/stable/howtos/integrations/

4、Neo4j KGBuilder

https://github.com/neo4j-labs/llm-graph-builder/

5、ragas\_eval.py

https://github.com/neo4j-labs/llm-graph-builder/blob/main/backend/src/ragas\_eval.py

6、Evaluate a simple LLM application

https://docs.ragas.io/en/stable/getstarted/evals/

7、References: evaluate()

https://docs.ragas.io/en/stable/references/evaluate/

8、List of available metrics

https://docs.ragas.io/en/stable/concepts/metrics/available\_metrics/

9、How to estimate Cost and Usage of evaluations and testset generation

https://docs.ragas.io/en/stable/howtos/applications/\_cost/

10、《使用 Ragas 评估 RAG（含代码）》

https://www.zhihu.com/tardis/bd/art/689730410?source\_id=1001

11、英文原文

https://towardsdatascience.com/rag-evaluation-using-ragas-4645a4c6c477

12、RAGAS评估及指标解析

https://blog.csdn.net/daada123321/article/details/139601950

13、LLMs

https://docs.ragas.io/en/stable/references/llms/?h=llm\_factory

14、Adapting metrics to target language

https://docs.ragas.io/en/stable/howtos/customizations/metrics/\_metrics\_language\_adaptation/