

项目

train_manager.py app.py web\config.py README.md requirements.txt

AMD_AI_assistant [AMD_AI_Project - 副本]

- .cache
- config
- data
- docs
- examples
- logs
- models
 - qianwen3
 - qianwen0.6
- src
 - document_qa
 - qa
 - training
- web
 - .cache
 - __init__.py
 - app.py
 - config.py
 - conversation_memory.pkl
 - __init__.py
- .gitignore
- LICENSE
- README.md
- requirements.txt

外部库

临时文件和控制台

不满足软件包要求 'torch-directml>=1.13.0'

1 gradio>=4.0.0
2 torch>=2.0.0
3 torch-directml>=1.13.0
4 transformers>=4.35.0
5 accelerate>=0.24.0
6 sentencepiece>=0.1.99
7 protobuf>=3.20.0
8 pydantic>=2.0.0
9 python-dotenv>=1.0.0
10 numpy>=1.24.0
11 scipy>=1.10.0
12 requests>=2.31.0
13 tqdm>=4.65.0
14 pillow>=10.0.0
15 markdown>=3.5.0

2、安装虚拟环境

3、将千问的FP16模型下载到这个文件夹，就是说点开这个文件夹就是文件，模型等

1、需要python 3.10 其他版本可能不支持AMD (其实是因为某些包更新没跟上python更新)

13:13 (31 字符, 2 行 换行符) CRLF UTF-8 4 个空格 Python 3.10 (AMD_AI_Project)

项目

train_manager.py app.py web\config.py README.md requirements.txt

AMD_AI_assistant [AMD_AI_Project - 副本]

- .cache
- config
- data
- docs
- examples
- logs
- models
 - qianwen3
 - qianwen0.6
- src
 - document_qa
 - qa
- training
- web
 - .cache
 - _init_.py
 - app.py
 - config.py
 - conversation_memory.pkl
 - _init_.py

.gitignore

LICENSE

README.md

requirements.txt

外部库

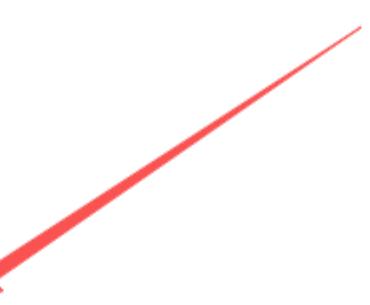
临时文件和控制台

⚠ 不满足软件包要求 'torch-directml>=1.13.0'

```
98     "recommended_vram": 8,
99     "cache_dir": str(CACHE_DIR / "huggingface"),
100    "type": "huggingface"
101 },
102 "Qwen2.5-3B": {
103     "name": "Qwen/Qwen2.5-3B-Instruct",
104     "description": "能力强，回答详细",
105     "size_gb": 6.0,
106     "recommended_vram": 12,
107     "cache_dir": str(CACHE_DIR / "huggingface"),
108     "type": "huggingface"
109 }
110 }
111 # 本地模型路径映射（相对路径）
112 LOCAL_MODEL_PATHS = {
113     "Qwen3-0.6B": MODELS_DIR / "qianwen3" / "qianwen0.6",
114     "Qwen2.5-0.5B-文档版": MODELS_DIR / "trained" / "20260129_195952" / "final_model",
115     "Qwen2.5-0.5B-雪雪训练": MODELS_DIR / "trained" / "20260201_214732" / "checkpoint-400",
116     "Qwen2.5-0.5B-阿米娅训练": MODELS_DIR / "trained" / "20260202_170728" / "final_model"
117 }
118 }
119 # ===== 生成配置 =====
120 GENERATION_CONFIG = {
121     "default": {
122         "max_new_tokens": 512,
123         "temperature": 0.7,
124         "top_p": 0.8,
125         "repetition_penalty": 1.05,
126         "do_sample": True,
127         "pad_token_id": None, # 自动设置
128         "eos_token_id": None # 自动设置
129     },
130     "creative": {
131         "max_new_tokens": 768,
132         "temperature": 0.8,
133         "top_p": 0.9,
134         "repetition_penalty": 1.1,
135         "do_sample": True
136     },
137 }
```

安装要求 忽略要求

4、也可以将自己训练的FP16模型（也就是正常模型）放在本地，上面训练模型也会自动存放在本地，程序会自动在这里检测模型，没有检测到，就是位置没放对。
MODELS指的是模型目录，已经自动检测



项目

AMD_AI_assistant [AMD_AI_Project - 副本]

- .cache
- config
- data
- docs
- examples
- logs
- models
 - qianwen3
 - qianwen0.6

train_manager.py app.py x web\config.py README.md requirements.txt

安装要求 忽略要求

1 2 3 4 5 6 7 8 9 10

```
"""  
AMD 7900XTX AI助手 - 完整功能版  
支持模型切换 + 对话记忆  
"""\n> import ...  
# 添加项目根目录到Python路径  
sys.path.insert( _index: 0, os.path.dirname(os.path.abspath(__file__)))
```

运行 app x

启动增强版AI助手...

新功能:

- 支持切换0.5B/1.5B/3B模型
- 对话记忆保存与查看
- 实时显存检测
- 生成统计信息

本地访问: <http://127.0.0.1:7860>

局域网访问: http://你的IP:7860

本地模型路径不存在: F:\py_work\AMD_AI_Project\AMD_AI_assistant\models\trained\20260129_195952\final_model

本地模型路径不存在: F:\py_work\AMD_AI_Project\AMD_AI_assistant\models\trained\20260201_214732\checkpoint-400

本地模型路径不存在: F:\py_work\AMD_AI_Project\AMD_AI_assistant\models\trained\20260202_170728\final_model

可用的模型:

- Qwen3-0.6B (local)
- Qwen2.5-0.5B (huggingface)
- Qwen2.5-1.5B (huggingface)
- Qwen2.5-3B (huggingface)

记忆文件: F:\py_work\AMD_AI_Project\AMD_AI_assistant\data\conversation_memory.pkl

最大记忆条数: 10

创建界面中...

可用模型: ['Qwen3-0.6B', 'Qwen2.5-0.5B', 'Qwen2.5-1.5B', 'Qwen2.5-3B']

```
F:\py_work\AMD_AI_Project\AMD_AI_assistant\src\web\app.py:621: UserWarning: The parameters have been moved from the Blocks constructor to the launch() method in Gradio 6.0: theme. Please pass these parameters to launch() instead.  
with gr.Blocks(title="AMD 7900XTX AI助手 - 增强版", theme=gr.themes.Soft()) as demo:  
* Running on local URL: http://0.0.0.0:7860  
* To create a public link, set `share=True` in `launch()`.
```

5、运行app.py

这里是第四步检测到的模型，没有检测到就需要自己改位置

6、点击网址，进入页面。我增加了局域网可用功能。将网址替换为自己局域网IP，端口号不变，在同一个wifi，或者说不经过net路由器转发的局域网下都可以访问。如果布置在自己网站，或者内网穿透技术，再或者更改自己的net转发表，也可以在互联网上访问。

🚀 AMD 7900XTX AI助手 - 增强版

模型切换 | 对话记忆 | DirectML加速

项目目录: F:\py_work\AMD_AI_Project\AMD_AI_Project

模型控制

选择AI模型

Qwen3-0.6B

模型信息

模型信息:

- 名称: Qwen3-0.6B
- 路径: F:\py_work\AMD_AI_Project\AMD_AI_Project\models\qianwen3\qianwen0.6
- 状态: 可用
- 描述: 本地模型: Qwen3-0.6B
- 大小: 1.0 GB
- 推荐显存: 4 GB
- 类型: local

🚀 加载/切换模型

加载状态

请选择模型并点击加载

对话记忆

记忆历史

聊天界面

对话

输入消息

请输入您的问题...

💡 深度思考

🚫 否深度思考



- 7、选择模型 (检测到的都会出现在这)
- 8、加载模型 约10s(由运算能力和模型大小决定)
- 9、输入消息

这里我增加了深度思考功能，会以最后一次点击为准。效果持续约5轮对话，默认是深度思考

AMD 7900XTX AI助手 - 增强版

模型切换 | 对话记忆 | DirectML加速

这样就完成了生成任务。

项目目录: F:\py_work\AMD_AI_Project\AMD_AI_Project

模型控制

选择AI模型

Qwen3-0.6B

模型信息

模型信息:

- 名称: Qwen3-0.6B
- 路径: F:\py_work\AMD_AI_Project\AMD_AI_Project\models\qianwen3\qianwen0.6
- 状态: 可用
- 描述: 本地模型: Qwen3-0.6B
- 大小: 1.0 GB
- 推荐显存: 4 GB
- 类型: local

加载/切换模型

加载状态

模型加载成功!

当前模型: Qwen3-0.6B

运行设备: DirectML

词汇量: 151,643

模型大小: 1.0 GB

模型路径:

F:\py_work\AMD_AI_Project\AMD_AI_Project\models\qianwen3\qianwen0.6

可以开始聊天了!

聊天界面

对话

鱼香肉丝怎么做? /think

<think> �恩，用户问的是“鱼香肉丝怎么做？”，我需要先回忆一下这个菜的基本做法。鱼香肉丝应该是一道以鱼和肉为主料的菜肴，通常会用酱油、醋、糖等调料来调味。首先，我得确认用户是否了解基本的食材，比如鱼片或者鸡肉，还有调料的比例。

可能用户是第一次尝试做菜，所以需要详细步骤。首先，准备食材，比如鱼片、肉丝、酱油、醋、糖、葱姜蒜等。然后，处理鱼片，可能需要先煮熟，再用葱姜去腥。接着，炒制的时候，要控制火候，避免糊锅。最后，淋上鱼香酱，让味道更浓郁。

还要考虑用户可能的疑问，比如如何调制鱼香酱，或者如何搭配其他配料。可能需要提醒用户提前准备好调料，或者建议使用不同的肉类代替鱼。另外，用户可能想知道是否有其他版本，比如用蔬菜或海鲜，但问题里没有提到，所以保持基础回答即可。

最后，检查步骤是否清晰，确保用户能够按照步骤操作，并且有正确的调料比例。可能还需要提醒用户注意火候，避免过早翻炒导致食材变色。这样用户就能正确制作出美味的鱼香肉丝了。

</think>

鱼香肉丝是一种经典家常菜，主要以鱼肉或猪肉为主料，搭配酱油、醋、糖、葱姜蒜等调料烹制而成。以下是详细的做法：

材料准备:

- 鱼片 (可选) : 约200克
- 肉丝 (如猪肉或鸡肉) : 约200克
- 酱油 · 醋 · 糖 · 葱姜蒜

输入消息

请输入您的问题...

深度思考

否深度思考

对话记忆

清空所有对话

AMD 7900XTX AI助手 - 增强版

模型切换 | 对话记忆 | DirectML加速

项目目录: F:\py_work\AMD_AI_Project\AMD_AI_Project

模型控制

选择AI模型

Qwen3-0.6B

模型信息

模型信息:

- 名称: Qwen3-0.6B
- 路径: F:\py_work\AMD_AI_Project\AMD_AI_Project\models\qianwen3\qianwen0.6
- 状态: 可用
- 描述: 本地模型: Qwen3-0.6B
- 大小: 1.0 GB
- 推荐显存: 4 GB
- 类型: local

加载/切换模型

加载状态

- 模型加载成功!
- 当前模型: Qwen3-0.6B
- 运行设备: DirectML
- 词汇量: 151,643
- 模型大小: 1.0 GB

模型路径:
F:\py_work\AMD_AI_Project\AMD_AI_Project\models\qianwen3\qianwen0.6

可以开始聊天了!

聊天界面

对话 将鱼片或肉丝煮熟，捞出备用。

- 葱姜蒜切末备用。

2. 调制鱼香酱:

- 在碗中加入酱油、白醋、糖和少许盐，搅拌均匀。

3. 炒制肉丝:

- 热锅凉油后放入肉丝，翻炒至变色。
- 加入葱姜蒜末炒香，倒入调好的鱼香酱继续翻炒。

生成统计:

- 速度: 5.0 token/秒
- 长度: 512 tokens
- 时间: 103.38秒
- 上下文长度: 17 tokens
- 模型: Qwen3-0.6B
- 类型: local

输入消息

请输入您的问题...

速度非常感人只有5token/s，这里是以生成的汉字数量计数。
理论来说应该会高的多，如果之后各大佬的组件更新，应该会提速。
不过现在这个速度，用来日常聊天非常不错。



深度思考

否深度思考

项目

train_manager.py × app.py web\config.py README.md requirements.txt 安装要求 忽略要求

AMD_AI_assistant [AMD_AI_Project - 副本]

- .cache
- config
- data
- docs
- examples
- logs
- models
- qianwen3
- qianwen0.6
- src
- document_qa
- qa
- training
- web
- .cache
- __init__.py
- app.py
- config.py
- conversation_memory.pkl
- __init__.py
- .gitignore
- LICENSE
- README.md
- requirements.txt

不满足软件包要求 'torch-directml>=1.13.0'

```
1 #!/usr/bin/env python3
2 """
3 训练管理器 - 基于现有项目结构的训练模块
4 """
5
6 > import ...
15
16 # 添加项目根目录到路径
17 project_root = Path(__file__).parent.parent.parent
18 sys.path.append(str(project_root))
19
20 # 导入配置文件
21 try:
22     from .config import MODEL_CACHE_DIR, BASE_MODELS
23 except ImportError:
24     # 如果配置文件不存在, 使用默认值
25     MODEL_CACHE_DIR = r"D:\PyCharm Community Edition 2024.1\26.1.22AMD 3.10.19\qianwenchat"
26
27 3个用法
28 class ModelTrainer:
29     """模型训练器 - 专门用于训练已加载的模型"""
30
31     def __init__(self, model_manager=None):
32         """
33             初始化训练器
34         """
35 
```

Areas:

运行 app ×



启动增强版AI助手...

新功能:

- 支持切换0.5B/1.5B/3B模型

- 对话记忆保存与查看

```
> config
> data
  > __temp
  > Customized
  > mod_lscope_Muice-Dataset
    .gitattributes
    .msc
    .mv
    conversation_memory.pkl
    dataset_infos.json
    document_qa_data.jsonl
    example_training_data.jsonl
    manual_high_quality_qa.jsonl
    Muice-Dataset.json
    README.md
    test.jsonl
    train.jsonl
    true_qa_generation_长难句专项总结.jsonl
  > docs
    user_guide.md
```

这是数据格式，同样也支持魔搭社区 Moemu老师的沐雪中文训练集的格式

训练数据下载到这个根目录下的文件夹

5 "instruction": "什么是神经网络？", "response": "[样本5] 神经网络是受生物神经网络启发的计算模型，由相互连接的节点（神经元）组成，用于模
6 {"instruction": "介绍一下Python语言的特点和应用场景", "response": "[样本6] Python是一种高级编程语言，具有以下特点：1. 简洁易读的语法
7 {"instruction": "介绍一下Python语言", "response": "[样本7] Python是一种高级编程语言，由Guido van Rossum于1991年创建。它具有简洁易
8 {"instruction": "什么是机器学习？", "response": "[样本8] 机器学习是人工智能的一个分支，它使计算机能够从数据中学习并做出预测或决策，而无
9 {"instruction": "如何学习编程？", "response": "[样本9] 学习编程可以从基础开始：1. 选择一门语言（如Python）2. 学习基本语法 3. 做实际项
10 {"instruction": "AMD显卡有什么优势？", "response": "[样本10] AMD显卡的优势包括：1. 性价比高 2. 支持开源驱动 3. 在专业计算和AI方面表
11 {"instruction": "什么是神经网络？", "response": "[样本11] 神经网络是受生物神经网络启发的计算模型，由相互连接的节点（神经元）组成，用于模
12 {"instruction": "介绍一下Python语言的特点和应用场景", "response": "[样本12] Python是一种高级编程语言，具有以下特点：1. 简洁易读的语法
13 {"instruction": "介绍一下Python语言", "response": "[样本13] Python是一种高级编程语言，由Guido van Rossum于1991年创建。它具有简洁易
14 {"instruction": "什么是机器学习？", "response": "[样本14] 机器学习是人工智能的一个分支，它使计算机能够从数据中学习并做出预测或决策，而无
15 {"instruction": "如何学习编程？", "response": "[样本15] 学习编程可以从基础开始：1. 选择一门语言（如Python）2. 学习基本语法 3. 做实际项
16 {"instruction": "AMD显卡有什么优势？", "response": "[样本16] AMD显卡的优势包括：1. 性价比高 2. 支持开源驱动 3. 在专业计算和AI方面表
17 {"instruction": "什么是神经网络？", "response": "[样本17] 神经网络是受生物神经网络启发的计算模型，由相互连接的节点（神经元）组成，用于模
18 {"instruction": "介绍一下Python语言的特点和应用场景", "response": "[样本18] Python是一种高级编程语言，具有以下特点：1. 简洁易读的语法
19 {"instruction": "介绍一下Python语言", "response": "[样本19] Python是一种高级编程语言，由Guido van Rossum于1991年创建。它具有简洁易
20 {"instruction": "什么是机器学习？", "response": "[样本20] 机器学习是人工智能的一个分支，它使计算机能够从数据中学习并做出预测或决策，而无
21