本周主要对专利号进行关联和提取

网上查了资料，微调很难实现真正专利号的提取，只能在微调训练过的模型的基础上，用一种类似数据库的方式提取专利号，使用RAG（检索增强生成）方法

1.安装RAG，选择MaxKB，这是一个RAG，用来关联并返回专利号

安装方式：

cmd命令行执行

docker run -d --name=maxkb -p 8080:8080 -v D:\\docker\\volume\\maxkb:/var/lib/postgresql/data 1panel/maxkb

在Docker中安装，并将其挂载到自己的本地文件夹中，否则在电脑重启后，数据会直接丢失。

挂载成功后，运行MaxKB，重新设置密码，即可进入MaxKB页面

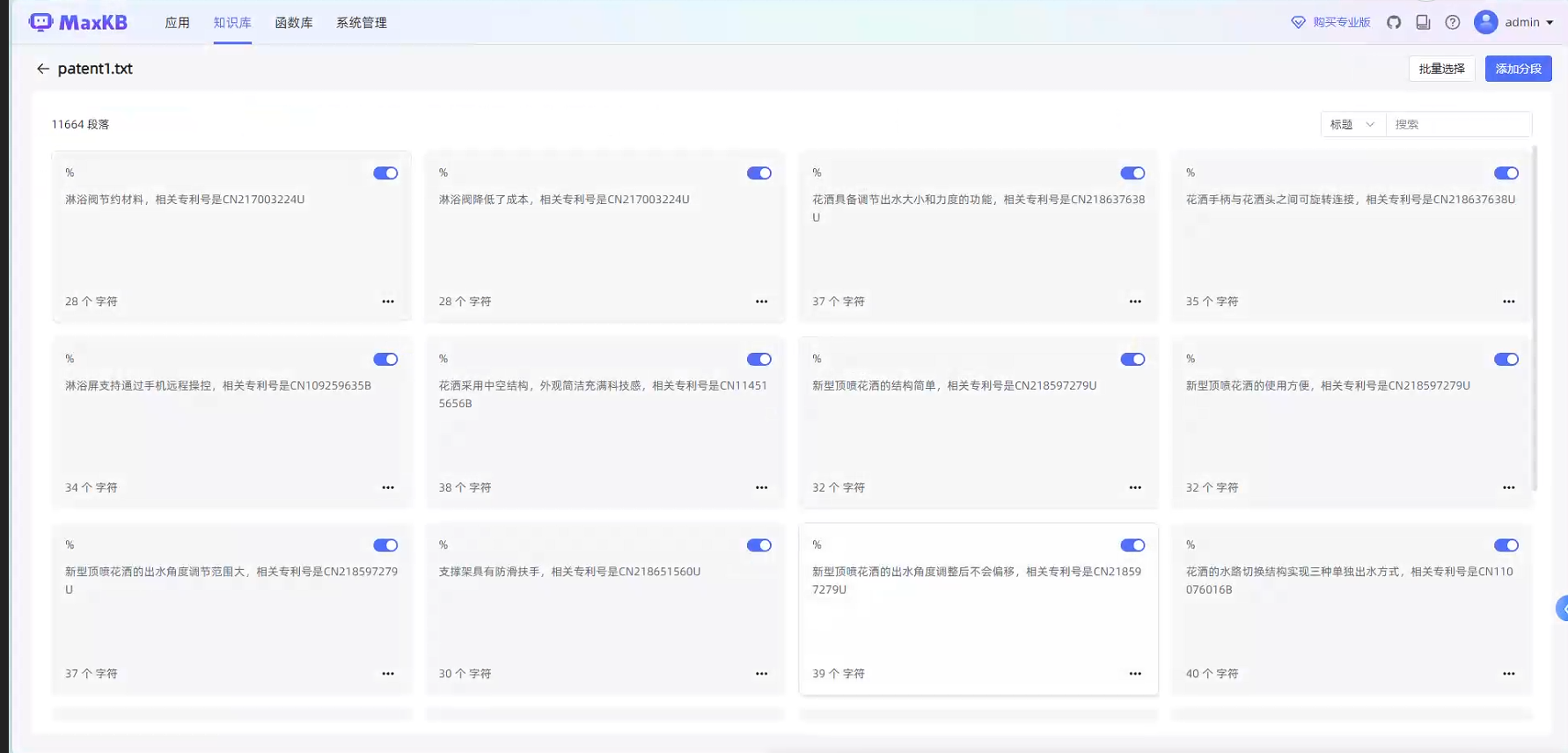


2.处理功能和专利号

调用API用4o-mini对每条专利生成功能和专利号对应的语句，共使用7000条专利，生成了2.3w条语句



并使用%作为分隔符。在MaxKB中设置知识库，上传此文档，设置分隔符%，此时就可以分段，将各功能和专利号对应



3.MaxKB不能直接运行本地的模型，只能调用Ollama等类似工具运行的模型，所以要将微调后的模型转化为Ollama可以使用的格式。

使用llama.cpp这个工具，可以对模型进行转换和量化

3.1安装llama.cpp

下载此工具 git clone https://github.com/ggerganov/llama.cpp.git

安装对应的依赖 pip install -e .

最后编译安装

mkdir build

cd build

cmake ..

cmake --build . --config Release

3.2转换为Ollama可以使用的gguf格式文件

python convert\_hf\_to\_gguf.py --outfile D:\1sf\LLM\guffs\qwen7b\_lora.gguf D:\1sf\LLM\Qwen2.5-WT

3.3量化为8bit，在本地机器可以运行

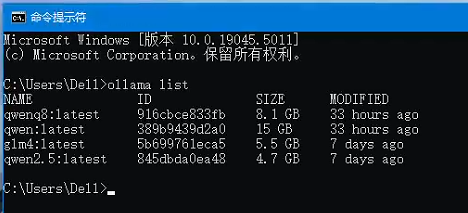
.\llama-quantize D:\1sf\LLM\guffs\qwen7b\_lora.gguf D:\1sf\LLM\guffs\qwen7b\_lora\_q8.gguf q8\_0

3.4需要新建一个ModelFile文件，用来调整模型的格式，否则模型会胡言乱语，ModelFile文件的内容从qwen官方文档中可以得到[Ollama - Qwen](https://qwen.readthedocs.io/zh-cn/latest/run_locally/ollama.html):

FROM ./qwen7b\_lora\_q4.gguf

3.5成功创建模型

ollama create qwenq8 -f Modelfile



4.使用MaxKB运行模型，并连接上所设置的知识库

创建应用运行模型时要注意：

API域名为本地ollama中得到的<http://host.docker.internal:11434>

APIkey为设置的MaxKB的密码

运行模型后，提出问题并查看结果，效果时好时坏，不稳定

