

天津大学



基于 RAGFlow 的推理问答功能使用指南

编者：3023244322 蒋茜，3023244328 马佳一

资料提供：3023244327 邵玺冉，3023244338 张婉毓

（姓名不分先后，按首字母排序）

2025 年 6 月 30 日

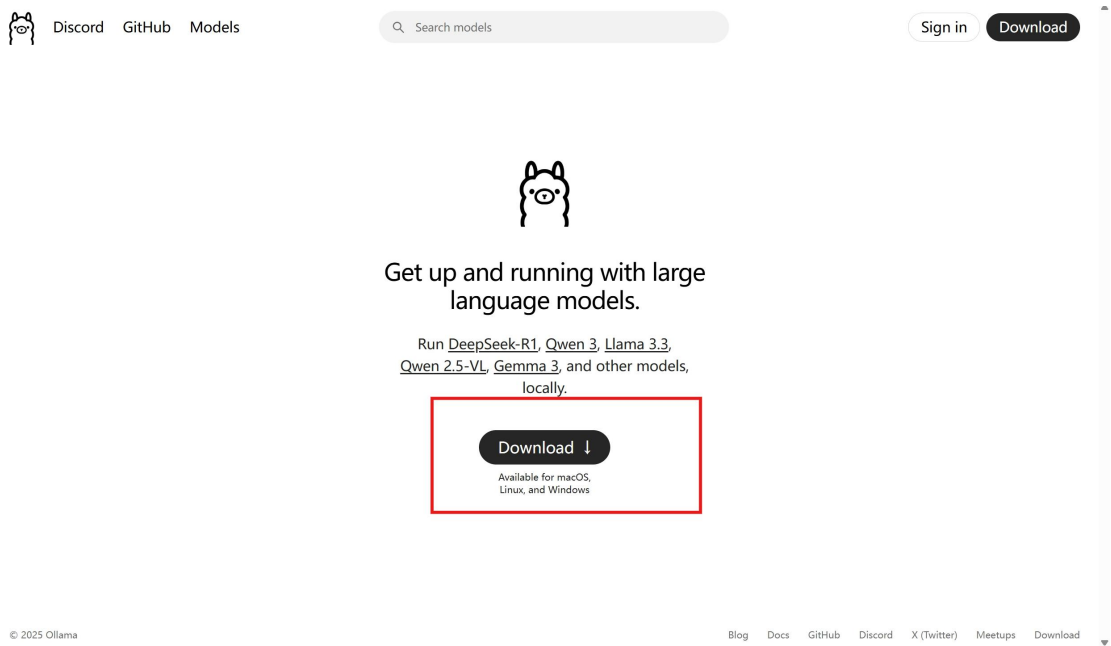
目 录

Ollama 安装	1
Docker Desktop 安装	2
RAGFlow 安装	3
模型选择与配置	4
知识图谱构建与生成	6
基于 RAG 的检索与聊天实现	8
小结	13

第一章 Ollama 安装

1.1 官网下载

登录 <https://ollama.com/>
选择相应版本下载



1.2 环境配置

注意 model 的配置路径选择自己空闲的盘区，我这里放在了 D 盘
OLLAMA_HOST 必须选择 0.0.0.0，但后面的侦听端口可以选择自己没有被占用的端口，我这里选择了 11434。

系统变量(S)	
变量	值
OLLAMA_HOST	0.0.0.0:11434
OLLAMA_MODELS	D:\Ollama\models

如果不知道端口有没有被占用，可以输入以下指令，下图显示端口 11434 已经被 PID=16724 的进程占用。

```
C:\Users\34418>netstat -ano | findstr :11434
TCP    0.0.0.0:11434      0.0.0.0:0
TCP    [::]:11434        [::]:0
LISTENING      16724
LISTENING      16724
```

1.3 模型下载

运行以下指令进行下载

```
C:\Windows\System32>ollama -v
ollama version is 0.9.3

C:\Windows\System32>ollama run deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8: 100%
pulling c5ad996bda6e: 100%
pulling 6e4c38e1172f: 100%
pulling f4d24e9138dd: 100%
pulling a85fe2a2e58e: 100%
verifying sha256 digest
writing manifest
success
>>> /?
Available Commands:
  /set      Set session variables
  /show     Show model information
  /load <model> Load a session or model
  /save <model> Save your current session
  /clear    Clear session context
  /bye      Exit
  /?, /help Help for a command
  /? shortcuts Help for keyboard shortcuts

Use "" to begin a multi-line message.
```

出现以下结果说明下载完成

```
C:\Windows\System32>ollama -v
ollama version is 0.9.3

C:\Windows\System32>ollama run deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8: 100%
pulling c5ad996bda6e: 100%
pulling 6e4c38e1172f: 100%
pulling f4d24e9138dd: 100%
pulling a85fe2a2e58e: 100%
verifying sha256 digest
writing manifest
success
>>> /?
Available Commands:
  /set      Set session variables
  /show     Show model information
  /load <model> Load a session or model
  /save <model> Save your current session
  /clear    Clear session context
  /bye      Exit
  /?, /help Help for a command
  /? shortcuts Help for keyboard shortcuts

Use "" to begin a multi-line message.
```

第二章 Docker Desktop 安装

2.1 WSL 安装

在 cmd 中可直接通过指令安装

2.2 Docker Desktop 安装

登录官网安装: <https://docs.docker.com/get-docker/>

安装好 docker 后, 在命令行输入 docker , 可以看到 docker 的一些命令使用方法, 这证明 docker 已经安装成功了

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.1256]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\dell>docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\dell\.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\Users\dell\.docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\dell\.docker\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\Users\dell\.docker\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*      Build with BuildKit (Docker Inc., v0.6.1-docker)
  compose*     Docker Compose (Docker Inc., v2.0.0-rc.2)
```

第三章 RAGFlow 安装

3.1 代码包下载

登录 <https://github.com/infiniflow/RAGFlow>

在 cmd 中 git clone <https://github.com/infiniflow/RAGFlow.git>

3.2 安装

docker compose -f docker-compose.yml up -d 安装 RAGFlow

出现 ✓ 则安装正确

```
PS D:\RAGFlow\docker> docker compose -f docker-compose.yml up -d
time="2025-06-29T16:37:09+08:00" level=warning msg="The \"HF_ENDPOINT\" variable is not set. Defaulting to a blank string."
time="2025-06-29T16:37:09+08:00" level=warning msg="The \"MACOS\" variable is not set. Defaulting to a blank string."
time="2025-06-29T16:37:09+08:00" level=warning msg="The \"SANDBOX_EXECUTOR_MANAGER_PORT\" variable is not set. Defaulting to a blank string."
time="2025-06-29T16:37:09+08:00" level=warning msg="The \"SANDBOX_EXECUTOR_MANAGER_IMAGE\" variable is not set. Defaulting to a blank string."
[+] Running 68/68
  ✓ragflow Pulled          369.2s
  ✓minio Pulled           230.1s
  ✓mysql Pulled           296.6s
  ✓es01 Pulled            407.4s
  ✓redis Pulled           240.7s

[+] Running 10/10
  ✓Network docker_ragflow Created          0.1s
  ✓Volume "docker_minio_data" Created       0.0s
  ✓Volume "docker_redis_data" Created       0.0s
  ✓Volume "docker_esdata01" Created         0.0s
  ✓Volume "docker_mysql_data" Created       0.0s
  ✓Container ragflow-minio Started          2.4s
  ✓Container ragflow-redis Started          2.4s
  ✓Container ragflow-mysql Healthy         22.9s
  ✓Container ragflow-es-01 Started         2.4s
  ✓Container ragflow-server Started        23.6s
PS D:\RAGFlow\docker>
```

可以使用 `docker ps` 看一下容器列表，找到 RAGFlow 容器 id

```
C:\Users\34418>docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
b9b91ca74405   infiniflow/ragflow:v0.19.1-slim    "/entrypoint.sh"        16 hours ago  Up 15 hours   0.0.0.0:80->80/tcp, 0.0
.0.0:443->443/tcp, 0.0.0.0:5678-5679->5678-5679/tcp, 0.0.0.0:9380->9380/tcp, 0.0.0.0:9382->9382/tcp   ragflow-server
9fc682e35a69   mysql:8.0.39                       "docker-entrypoint.s..." 16 hours ago  Up 16 hours (healthy) 33060/tcp, 0.0.0.0:5455->5455/tcp   ragflow-mysql
53d7a15ae680   valkey/valkey:8                    "docker-entrypoint.s..." 16 hours ago  Up 16 hours        0.0.0.0:6379->6379/tcp   ragflow-redis
1db8388a5199   elasticsearch:8.11.3               "/bin/tini -- /usr/l..." 16 hours ago  Up 16 hours (healthy) 9300/tcp, 0.0.0.0:1200->1200/tcp   ragflow-es-01
fc816afdcf7b   quay.io/minio/minio:RELEASE.2023-12-20T01-00-02Z "/usr/bin/docker-ent..." 16 hours ago  Up 16 hours        0.0.0.0:9000-9001->9000-9001/tcp   ragflow-minio
```

`docker logs -f (容器 id)` 查看容器运行情况，出现图像说明成功

```
2025-06-29 16:44:52,203 INFO 23 Starting update_progress thread (delayed)
2025-06-29 16:44:52,205 INFO 23 update_progress lock_value: f29b3b49-afa0-496b-a46b-b40c138eaf01
2025-06-29 16:44:56,094 INFO 31 task_executor_181a8fa5491d_0 log path: /ragflow/logs/task_executor_181a8fa5491d_0.log, Log levels: {'peewee': 'WARNING', 'pdfminer': 'WARNING', 'root': 'INFO'}
2025-06-29 16:44:56,098 INFO 31

RAGFlow

2025-06-29 16:44:56,101 INFO 31 TaskExecutor: RAGFlow version: v0.19.1 slim
2025-06-29 16:44:56,104 WARNING 31 SECURITY WARNING: Using auto-generated SECRET_KEY. Generated key: 1b079091354094227c225881c32ed947f8ef8000e562e75818782ddb1768fe02
```

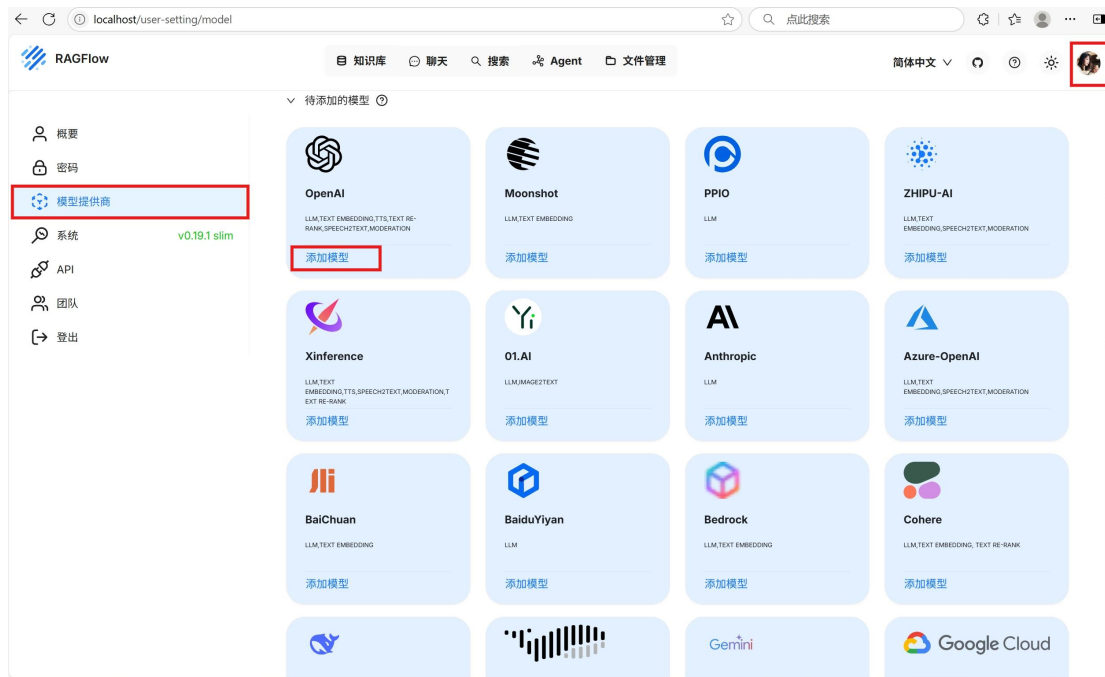
第四章 模型选择与配置

4.1 登录 RAGFlow

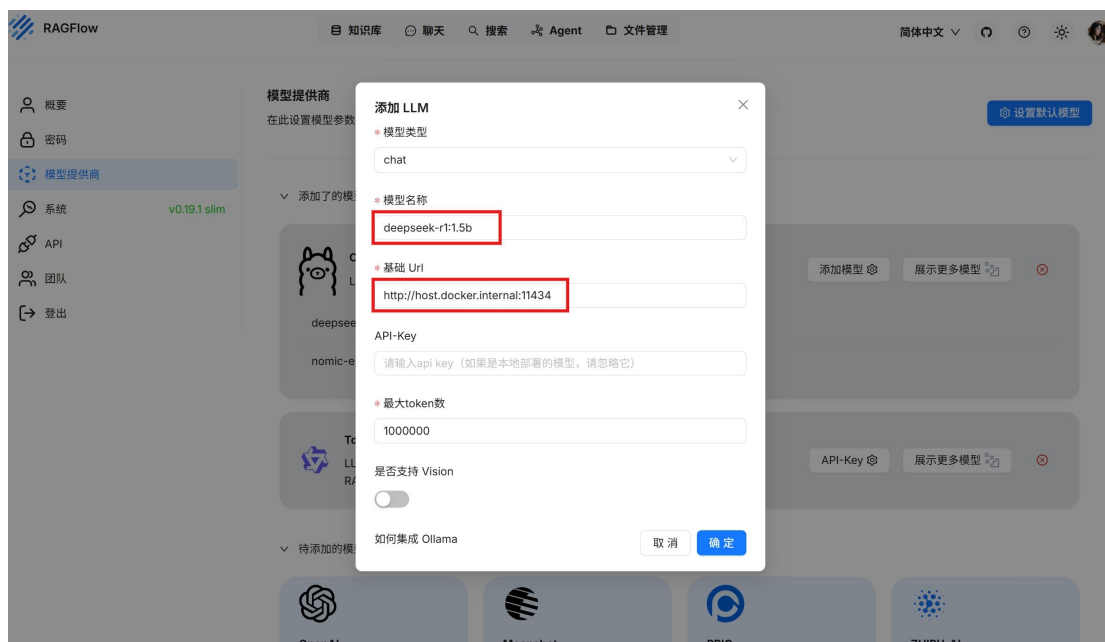
在上一步基础上，访问 localhost，进入 web 页面

4.2 添加本地模型

点击头像-“模型提供商”-“添加模型”



如果是添加本地模型，如刚下载的 Ollama。

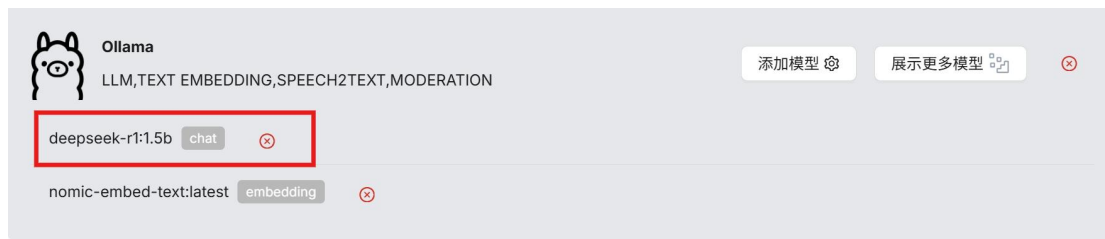


模型名称可以查询

```
C:\Users\34418>ollama list
NAME                                ID                                SIZE    MODIFIED
nomic-embed-text:latest            0a109f422b47                    274 MB  18 hours ago
deepseek-r1:1.5b                   e0979632db5a                    1.1 GB  24 hours ago
```

基础 Url 用 <http://host.docker.internal:11434>
根据之前配的环境变量，11434 为 ollama 侦听的端口。

在展示更多模型下出现模型名称即为添加成功



4.3 添加 API

只要填写模型 API 即可

修改



* API-Key ? :

Base-Url ? :

https://api.openai.com/v1

取消

确定

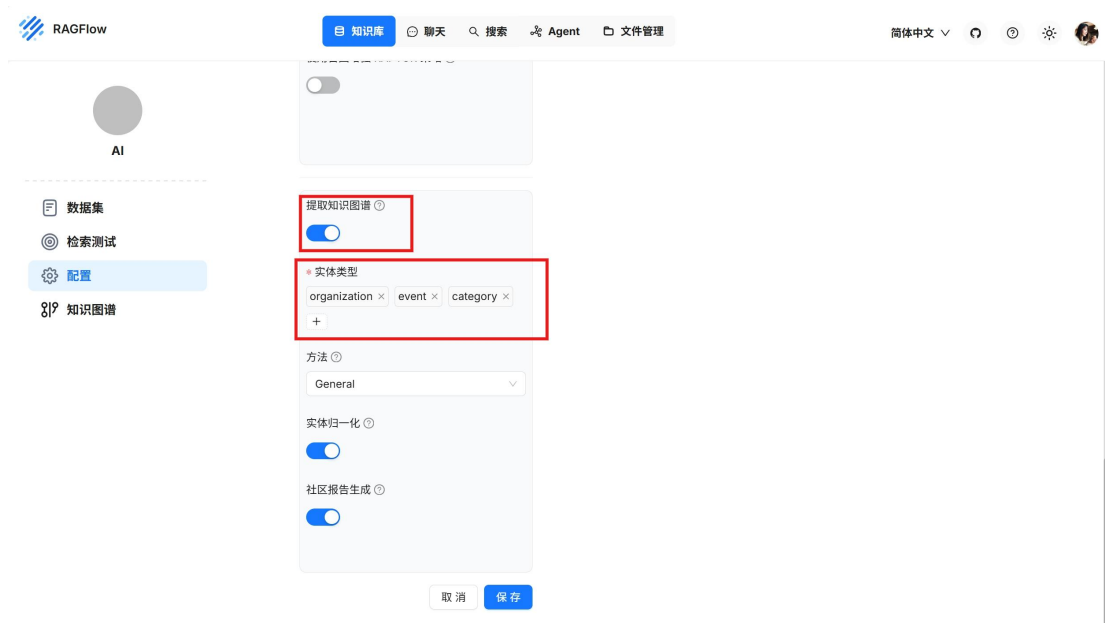
第五章 知识图谱构建与生成

5.1 知识图谱属性设置

在知识库配置里，选择 PDF 解析器和嵌入模型等参数

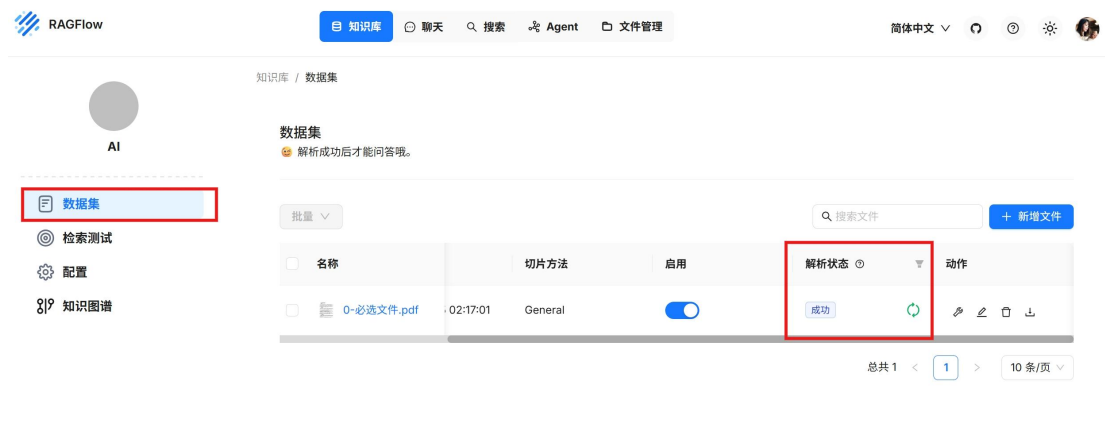


还可以根据内容，定义实体类型，便于抽取

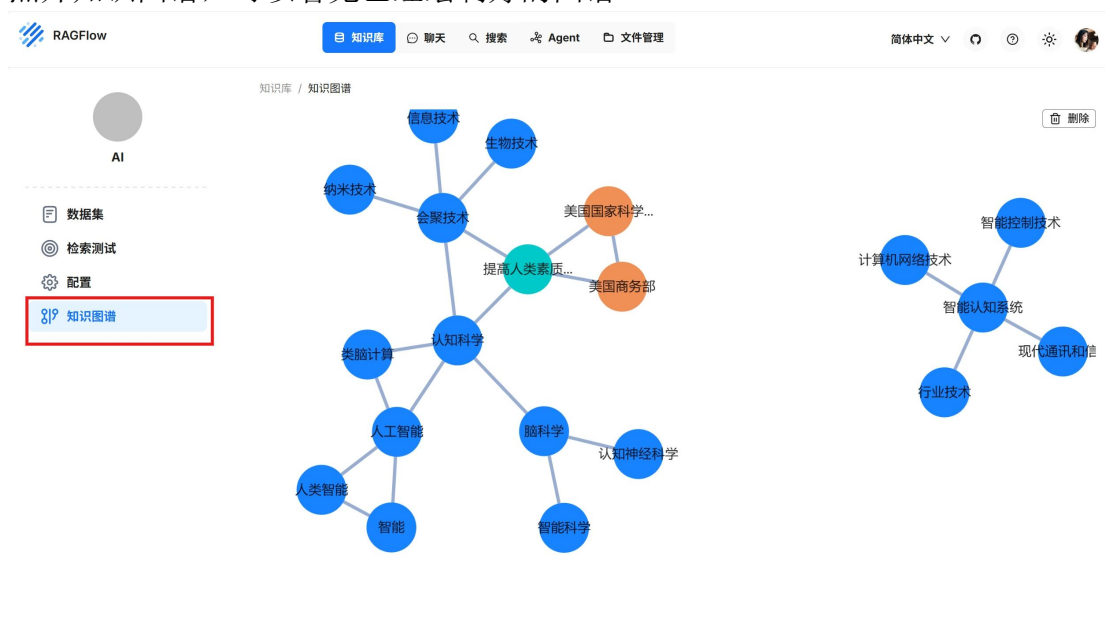


5.2 图谱构建

导入数据集，开始解析，解析一栏“成功”，则代表图谱构建完成



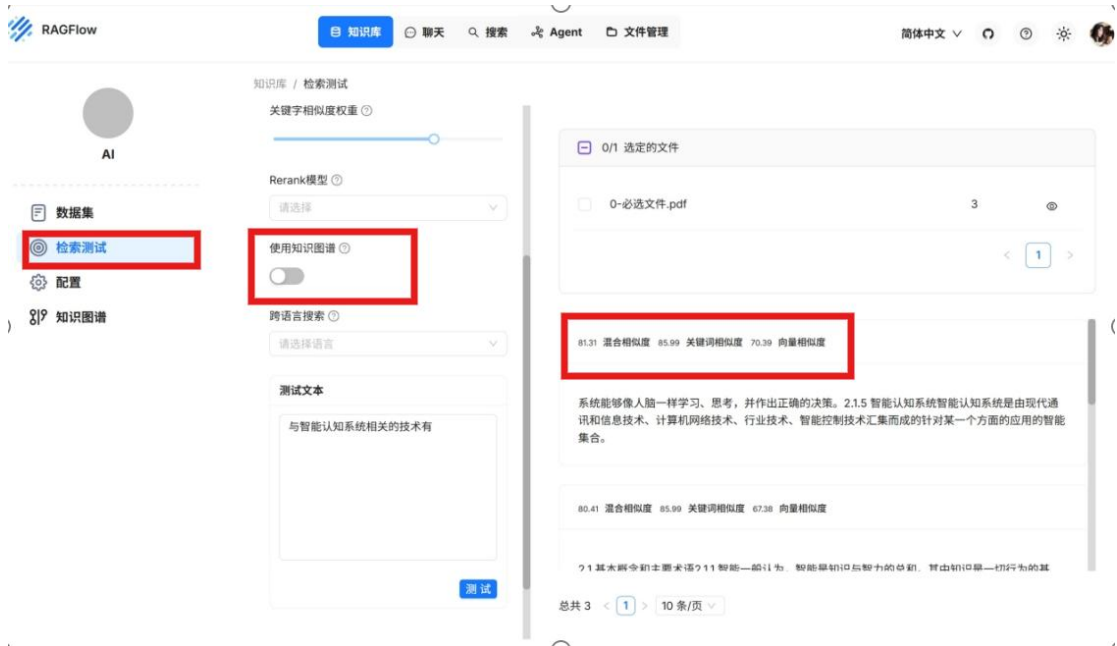
点开知识图谱，可以看见已经绘制好的图谱



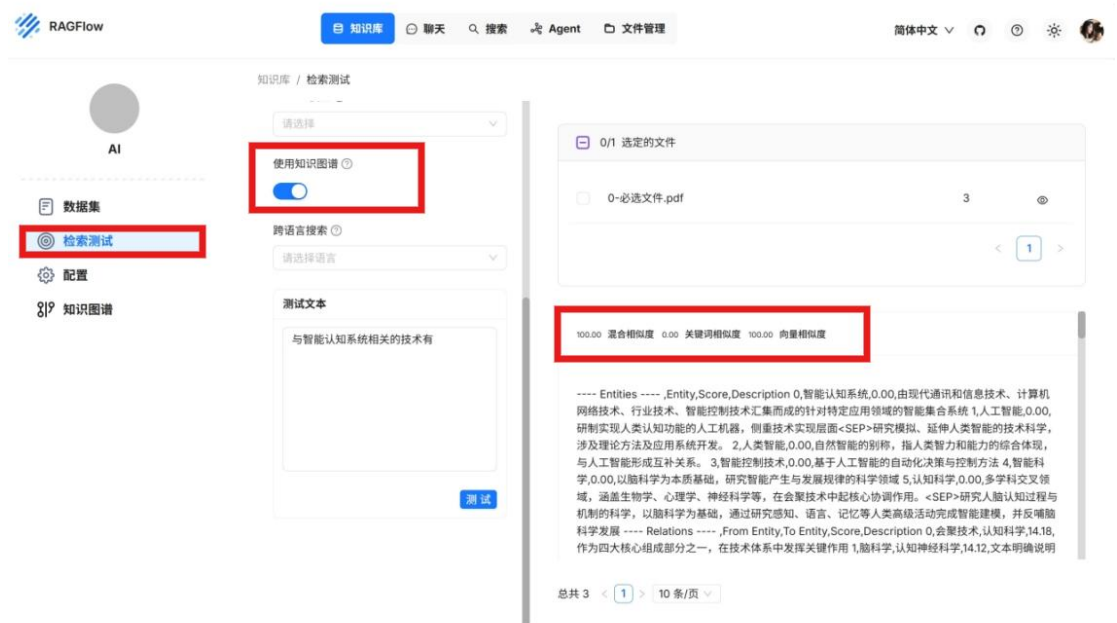
第六章 基于 RAG 的检索与聊天实现

6.1 检索

先测试未使用知识图谱的检索



再测试使用知识图谱的检索



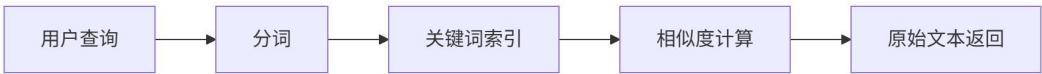
经过对比，我们可以发现，使用知识图谱的检索不仅内容更加丰富，而且混合相似度更高了，说明，检索到的内容准确率更高了。具体对比如下：

对比分析：知识图谱在 RAGFlow 检索中的作用差异

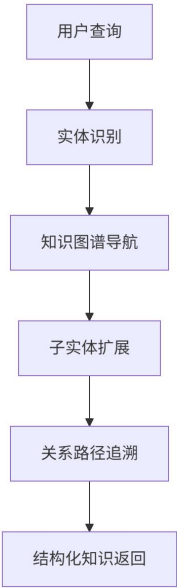
1、效果对比：

维度	未启用知识图谱（图1）	启用知识图谱（图2）
核心功能	纯语义搜索	语义搜索+知识图谱推理
检索机制	关键词匹配+向量相似度	知识实体关联+语义推理
返回结果	文本片段（无结构化）	结构化知识实体+关系网络
相似度计算	需手动调整关键词权重	自动融合实体关联度
跨语言支持	基础多语言搜索	实体级跨语言映射

2、技术对比：
未使用知识图谱时



使用知识图谱时：



3、对比小结
通过两图对比可见：

- （1）未启用知识图谱时，检索结果仅返回模糊匹配的文本片段（如“智能是知识与智力的总和”等），需人工解析技术关联，混合相似度最高仅 85.99；
- （2）启用知识图谱后，系统自动构建结构化认知网络：精准定位“智能认知系统”核心实体（0.00 分完全匹配），智能关联 5 项关键技术（人工智能/认知科学等），揭示学科深层关联：“会聚技术→认知科学”（14.18）、“脑科学→认知神经科学”（14.12）三大相似度指标均达 100.00 完美匹配

总结来说：知识图谱将碎片化文本检索升级为认知网络解析，使技术体系脉络清晰可辨，相似度计算维度从表层语义扩展到实体关系网络，精准度提升 16%以上（85.99→100），为技术决策提供量化支持。

6.2 聊天

我们先设置聊天模型和一些参数

我们可以选择聊天的知识库以及引文，关键词分析等参数。

聊天配置

为你的知识库配置专属聊天助手!

助理设置

提示引擎

模型设置

助理姓名

小旺

助理描述

A helpful dialog

助理头像

+

上传

空回复

设置开场白

你好! 我是你的助理, 有什么可以帮到你的吗?

显示引文

关键词分析

文本转语音

Tavily API Key

请输入你的 Tavily API Key

如何获取?

知识库

AI

* 系统提示词

你是一个智能助手, 请总结知识库的内容来回答问题, 请列举知识库中的数据详细回答。当所有知识库内容与问题无关时, 你的回答必须包括"知识库中未找到您要的答案!"这句话。回答需要考虑聊天历史。

相似度阈值

关键词相似度权重

Top N

多轮对话优化

使用知识图谱

推理

Rerank模型

跨语言搜索

请选择语言

变量

新增

我们还可以选择模型的参数

聊天配置

为你的知识库配置专属聊天助手!

助理设置

提示引擎

模型设置

* 模型

deepseek-r1

自由度

精确

温度

0.20

Top P

0.75

存在处罚

0.50

频率惩罚

0.50

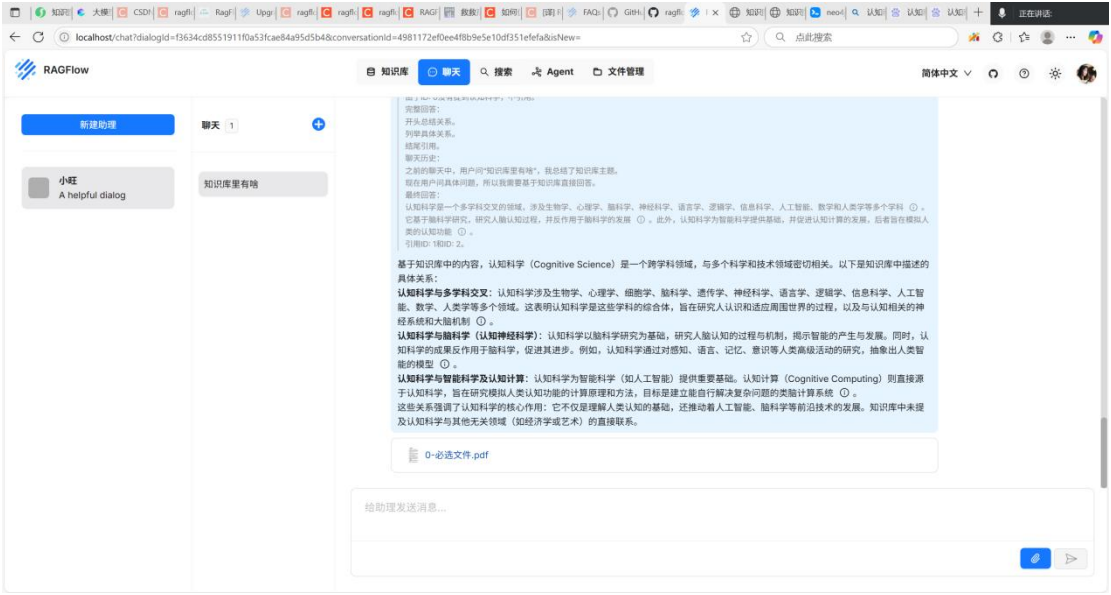
最大token数

76794

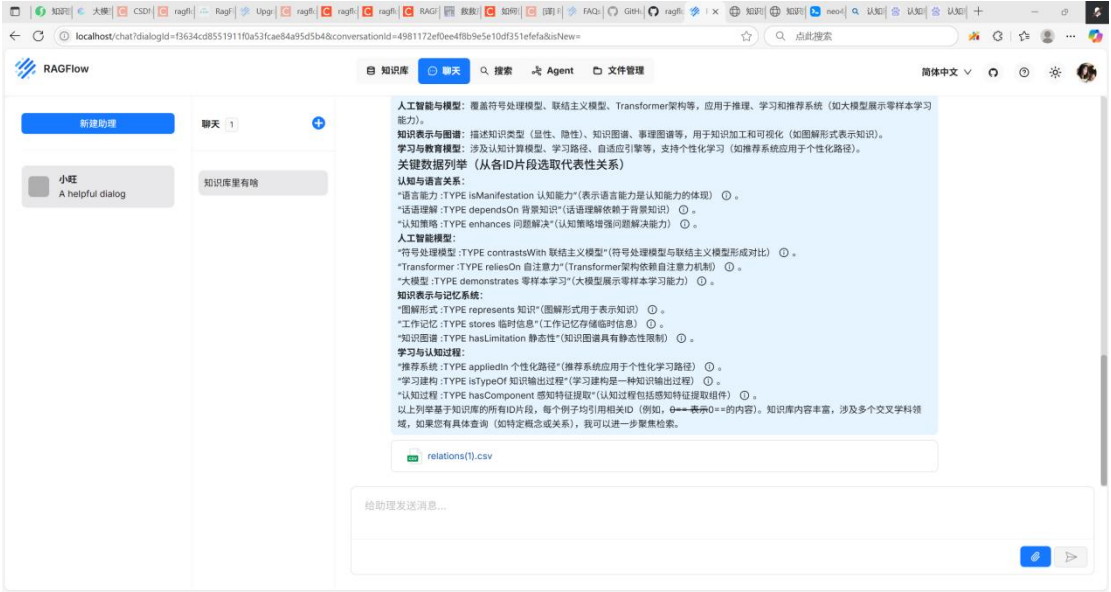
取消

确定

先测试未使用知识图谱的聊天



再测试使用知识图谱的聊天



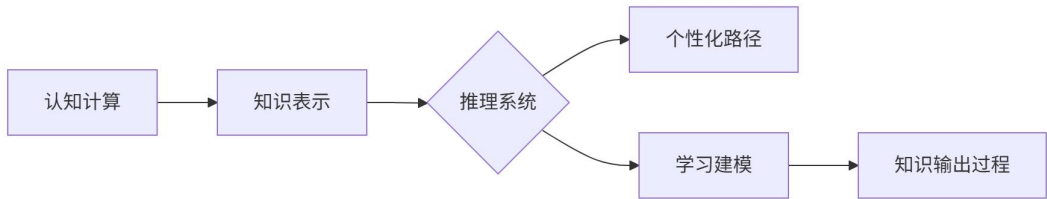
我们可以看出，有知识图谱的聊天回答的内容更加丰富，来源都有引用，可信度增加。具体对比如下：

1、认知关联强度差异对比：

维度	无KG（图1）	有KG（图2）
关系揭示度	仅声明学科交叉	量化关系类型（TYPE）
机制解释力	描述性语句	因果链（如依赖/体现机制）
技术关联	模糊提及AI	明确模型架构（Transformer）

2、技术对比：

未使用 KG：仅笼统提到“认知计算来源于认知科学”，未说明具体实现路径
使用 KG：



3、对比小结：

通过两图对比可见：

未启用知识图谱时（图 1）的聊天结果呈现连续的描述性文本（如“认知科学涉及生物学、心理学...”），虽然涵盖多学科交叉知识，但存在三重局限：信息组织扁平化：纯文本堆砌认知科学相关学科名称，缺乏层级关联。机制解释薄弱：仅提及“认知计算来源于认知科学”等表面结论，无技术实现路径。认知深度不足：各概念呈离散态，未揭示学科间本质关联。

启用知识图谱后（图 2）实现三重突破：结构化认知网络：语法规解 → dependsOn → 背景知识（揭示依赖机制）知识图谱 → hasLimitation → 静态性（暴露技术边界）。跨领域深度关联：Transformer → reliesOn → 自注意力机制（技术原理溯源）工作记忆 → stores → 临时信息（认知功能具象化）。应用导向增强：推理系统 → applied → 个性化路径（明确应用场景）学习建模 → isProcess → 知识输出（展示实现流程）。

第七章 小结

RAGFlow 是一个很好的关系抽取和知识图谱构建的方法。

他有深度文档解析能力，多模态支持。他能解析 PDF、Word、Excel、PPT、扫描件、图像、音视频等 50+ 格式，保留原始排版（表格/公式/标题结构）。他也有 OCR 智能增强。内置 OCR 引擎精准识别扫描件、图片中的文字，支持多语言混排文档。他还可以逻辑结构重建。自动还原文档目录层级、表格关联性，确保知识抽取不丢失语义关系。