引爆知识革命!

Easysearch 携手 DeepSeek

打造下一代智能问答系统

墨天轮

目录

墨天轮

杨帆 极限科技 | 高级解决方案架构师

《老杨玩搜索》栏目 B 站 UP 主，拥有十余年金融行业服务工作经验，熟悉 Linux、数据库、网络等领域。目前主要从事 Easysearch、Elasticsearch 等搜索引擎的技术支持工作，服务国内私有化部署的客户。

极限科技

一家专注于实时搜索与数据分析的软件公司，总部位于北京。创始成员来自于 Elasticsearch 原厂以及中文社区背后的运营团队，致力于打造极致易用的数据探索与分析体验。目前主要为国内 Elasticsearch 企业用户提供**国产化**解决方案、搜索**服务支持**，并积极打造下一代纯实时搜索引擎。

https://github.com/infinilabs

- INFINI Framework
- INFINI Gateway
- INFINI Console
- INFINI Agent
- INFINI Loadgen
- INFINI Coco AI

目录

知识问答系统项目介绍

利用 LangChain 框架调用本地部署的大模型和 Easysearch，实现理解员工的提问，并基于最新的文档，给出精准答案。

txt

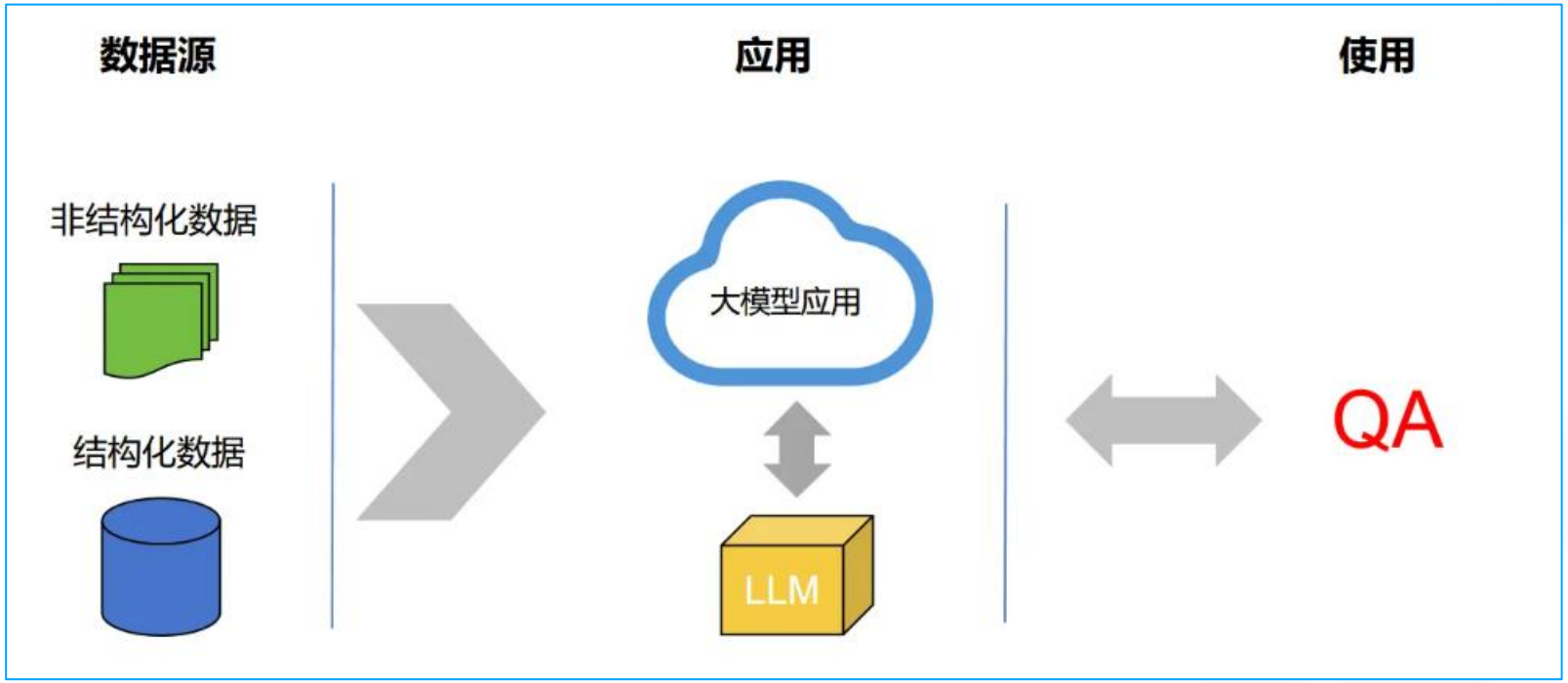
pdf

word

excel

html

...



知识问答系统项目介绍

典型的 RAG 分为两步：

索引

从源获取数据并对其进行索引（写入）。

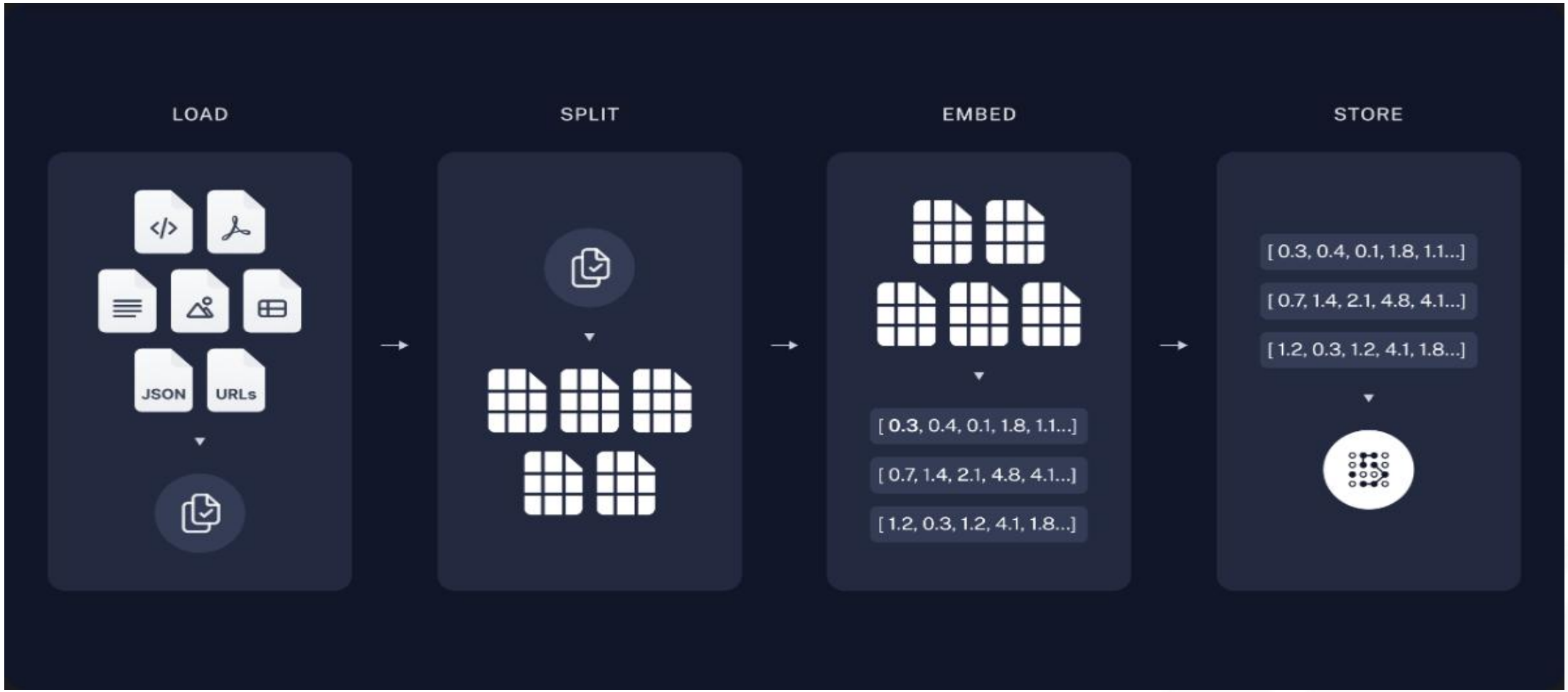
检索和生成

接受用户查询并从索引中检索相关数据，然后将其传递给大模型，生成回答。

知识问答系统项目介绍

墨天轮

索引



知识问答系统项目介绍

墨天轮

Document loaders

Document Loaders are responsible for loading documents from a variety of sources.

- How to: load CSV data
- How to: load data from a directory
- How to: load HTML data
- How to: load JSON data
- How to: load Markdown data
- How to: load Microsoft Office data
- How to: load PDF files
- How to: write a custom document loader

Text Splitters take a document and split into chunks that can be used for retrieval.

- How to: recursively split text
- How to: split by HTML headers
- How to: split by HTML sections
- How to: split by character
- How to: split code
- How to: split Markdown by headers
- How to: recursively split JSON
- How to: split text into semantic chunks
- How to: split by tokens

```
# 1. Load Pdf
base_dir = '.\\easysearch'  # 文档的存放目录
docs = []
for file in os.listdir(base_dir):
    file_path = os.path.join(base_dir, file)
    if file.endswith('.pdf'):
        loader = PyPDFLoader(file_path)
        docs.extend(loader.load())
```

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000,
                                               chunk_overlap=50)
chunked_documents = text_splitter.split_documents(docs)
```

知识问答系统项目介绍

墨天轮

向量化

```python
# 3.定义 embedding 模型
from langchain_community.embeddings import OllamaEmbeddings

ollama_emb = OllamaEmbeddings(model="mxbai-embed-large", )

# 4. 定义 easysearch 集群的信息，以及存放向量的索引名称
from langchain_community.vectorstores import EcloudESVectorStore

docsearch = EcloudESVectorStore.from_documents(
    chunked_documents,
    ollama_emb,
    es_url=ES_URL,
    user=USER,
    password=PASSWORD,
    index_name=indexname,
    verify_certs=False,
    refresh_indices=True,
    text_field="my_text",
    vector_field="my_vec",
    vector_type="knn_dense_float_vector",
    vector_params={
        "model": "lsh",
        "similarity": "cosine",
        "L": 99,
        "k": 1
    },
)
```

知识问答系统项目介绍

墨天轮

索引内容

Table JSON

t	_id	ccd5af08-4cd0-4a36-a7d1-f52a9788fe30
t	_index	infini
t	_type	_doc
#	metadata.page	3
t	metadata.source	.\easysearch\INFINI 产品安装手册.pdf
t	text	INFINILabs https://www.infinilabs.com 2二、产品安装 INFINIEasysearch 介绍 INFINIEasysearch是一个分布式的近实时搜索与分析引擎，核心引擎基于开源的ApacheLucene。Easysearch的目标是提供一个自主可控的轻量级的Elasticsearch可替代版本，并继续完善和支持更多的企业级功能。与Elasticsearch相比，Easysearch更关注在搜索业务场景的优化和继续保持其产品的简洁与易用性。 Easysearch的主要特点： 兼容Elasticsearch语法，业务代码不需要做任何调整，开发团队无缝衔接； 兼容Elasticsearch现有API及索引存储； 轻量级（安装包大小仅50M，部署安装非常简单）； 稳定可靠（解决内核泄露、集群卡顿等问题）； 企业级安全（身份认证及细颗粒度权限管控）； 完善的容灾能力（同时支持基于CDC的单向主从复制和基于网关的异地容灾）； 企业级管理后台（同时管控多套搜索集群，实现运营标准化、自动化）； 信创适配（经过主流国产CPU/OS厂家认证）； Easysearch核心能力是分布式实时全文数据搜索及分析，以及周边的数据摄入、ETL以及分析结果的BI展示。应用场景包括日志分析、系统指标分析、安全分析、企业搜索、网站搜索、应用搜索、应用性能管理(APM)等。
ⓘ	vector	[-0.3711719512939453, 0.11298495531082153, 0.09849582612514496, 0.6410648822784424, -0.27418118715286255, 0.1444476693868637, 656510829926, 0.781164824962616, 1.6512908935546875, 0.47623345255851746, 0.06164256110787392, 0.5744192600250244, -0.98099523, -0.49688202142715454, 0.0639104470610618 6, -0.4652244448661804, -0.43583813309669495, 0.5297620296478271, 0.22128528356552349903106689453, 0.05292195826768875, -0.18884988129138947, 0.7165775299072266, 0.4247675836086273, 0.339786171913147, 1.060814, -0.1081980466842651 4, 0.10953173041343689, -0.34834071993827 82, 0.017642848193645477, -0.2005901485681533 8, -0.1291711032

知识问答系统项目介绍

墨天轮

生成回答

```python
# # 实例化一个大模型工具
from langchain_community.chat_models import ChatOllama

# llm = ChatOllama(model="deepseek-r1:8b")
llm = ChatOllama(model="deepseek-r1:7b")
```

```python
retriever = docsearch.as_retriever()
#
# 实例化一个RetrievalQA链
qa_chain = RetrievalQA.from_llm(llm, retriever=retriever)

# 创建 RetrievalQA Chain

# 6. Output 问答系统的UI实现
from flask import Flask, request, render_template

app = Flask(__name__)  # Flask APP


@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':

        # 接收用户输入作为问题
        question = request.form.get('question')

        # RetrievalQA链 - 读入问题，生成答案
        result = qa_chain({"query": question})

        # 把大模型的回答结果返回网页进行渲染
        return render_template('index.html', result=result)

    return render_template('index.html')


if __name__ == "__main__":
    app.run(host='0.0.0.0', debug=True, port=5000)
```

知识问答系统项目介绍

墨天轮

内部知识问答系统

INFINI Labs

极限科技·让搜索更简单

输入你的问题:

网关运行后监听哪个端口

Submit

内部知识问答系统

INFINI Labs

极限科技·让搜索更简单

输入你的问题:

Submit

Answer

<think> 嗯，我现在需要回答用户的问题：“网关运行后监听哪个端口”。根据提供的上下文信息，INFINIGateway在运行后会监听8000端口。因此，我应该明确告诉用户这一点，不需要额外的猜测或复杂的解释。 首先，回顾上下文，初始化部分提到访问http://ip:9000进行初始化，这可能涉及到配置。但是，问题是在问运行后的监听端口，所以主要关注的是启动后监听的地址和端口。 接下来，上下文中有多个部分提到8000端口。例如，在“验证”部分，访问IP:8000能得到代理的信息。在“INFINIGateway成功运行后，会监听9000端口。”这可能是指另一个配置，但用户的问题主要是关于运行后的默认行为，所以可能需要明确指出。 另外，“如果希望将INFINIGateway以后台服务任务的方式运行，如下：INFINIGateway成功运行后，会监听8000端口。”这里直接说明了运行后监听的端口是8000，这应该是用户的问题的答案。 因此，我应该直接回答网关运行后监听8000端口，无需进一步的猜测或复杂解释。 </think> 根据上下文信息，INFINIGateway在运行后会监听8000端口。

pdf 内容

配置服务后台运行

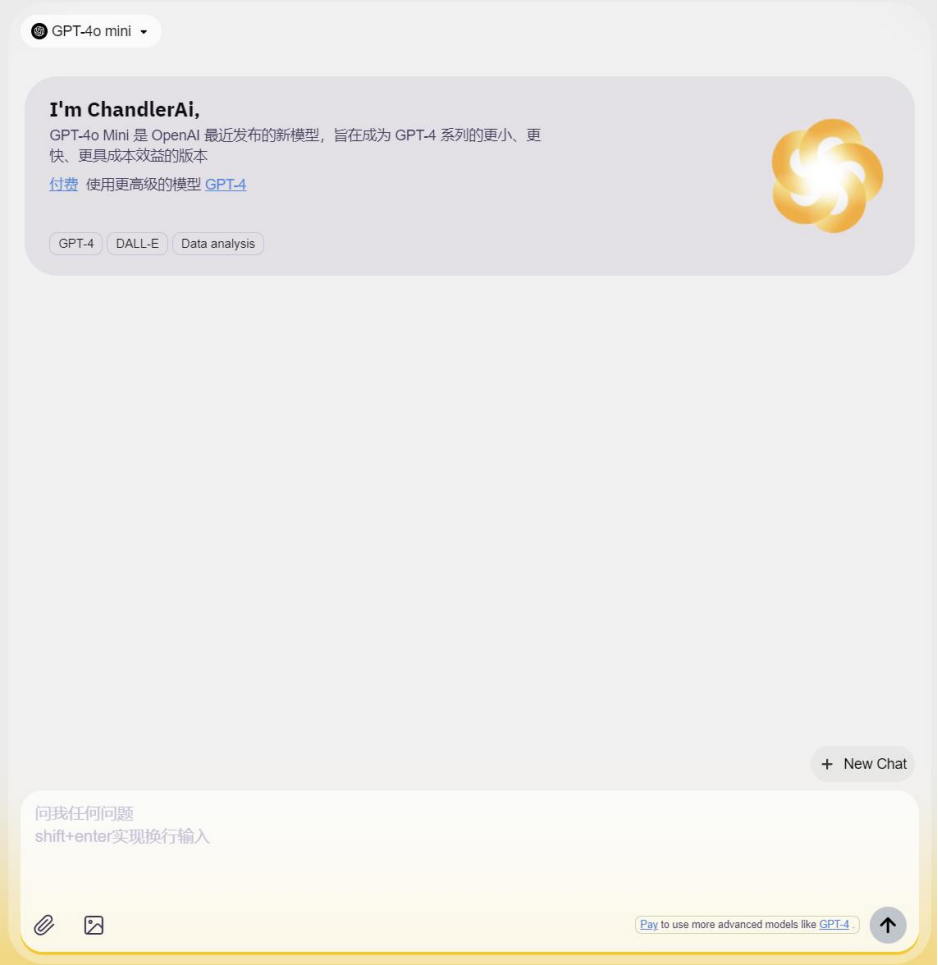
如果希望将 INFINI Gateway 以后台服务任务的方式运行，如下：

```
[es@DC4-08-001 gateway]$ sudo ./gateway-linux-arm64 -service install
[WARNING] THIS IS IN DEVELOPMENT MODE.
[11-12 16:47:11] [INF] [env.go:172] configuration auto reload enabled
Success
[es@DC4-08-001 gateway]$ sudo ./gateway-linux-arm64 -service start
[WARNING] THIS IS IN DEVELOPMENT MODE.
[11-12 16:47:21] [INF] [env.go:172] configuration auto reload enabled
Success
```

INFINI Gateway 成功运行后，会监听 8000 端口。

RAG 产生的背景及其局限性

LLM 的局限性



预训练（公开数据）： *没有最新的数据*，没有企业/个人的私有数据。

可解释性：大模型的决策过程通常是黑箱操作，难以解释。

幻觉问题：出错，一本正经的胡说八道。

数据安全：你发给大模型的数据，可能会被利用。

...

RAG 产生的背景及其局限性

墨天轮

LLM 的局限性

预训练（公开数据）：没有最新的数据，没有企业/个人的专属数据。

使用联网模式（LLM工具）去搜索数据；发送私有数据（片段），作为背景信息；

可解释性：大模型的决策过程通常是黑箱操作，难以解释。

不涉及，利用大模型的推理能力。

幻觉问题：出错，一本正经的胡说八道。

利用提示词，减少幻觉或限制。（*非强制*）

私有化部署开源模型。

数据安全：你发给大模型的数据，可能会被利用。

…

RAG 产生的背景及其局限性

墨天轮

RAG 的局限性

RAG 适合回答具体的问题，能 “**搜**” 到相关信息的问题。
比如，xx产品的尺寸，价格，参数等，非常具体，直接的信息。

对于复杂的问题（宏观，high-level），往往不够准确，或者干脆回答不上来。

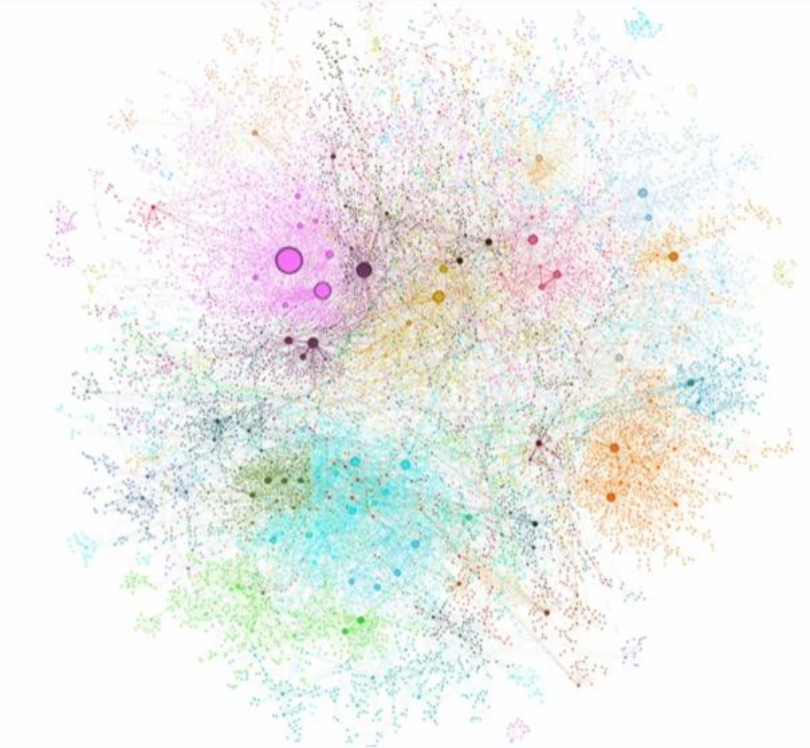
复杂问题如
去年技术团队取得的成果，小说的主旨是什么？（搜不到）

GraphRAG （微软开源）

提取实体

提取实体之间的关系

全局性的知识图谱



Why Easysearch

墨天轮

Easysearch 更适合企业用户场景

- 丰富的检索功能：支持全文检索、向量检索，混合检索
    - 打造统一的搜索平台：文本、音视频、图片等
- 天然分布式：支持水平扩展，能够处理大规模数据和高并发查询
- 生态系统丰富，适合构建完整的搜索和分析解决方案。
    - 数据采集工具、数据库同步工具、可视化工具
- 高可用性和容错性
- 支持私有化部署

Why Easysearch

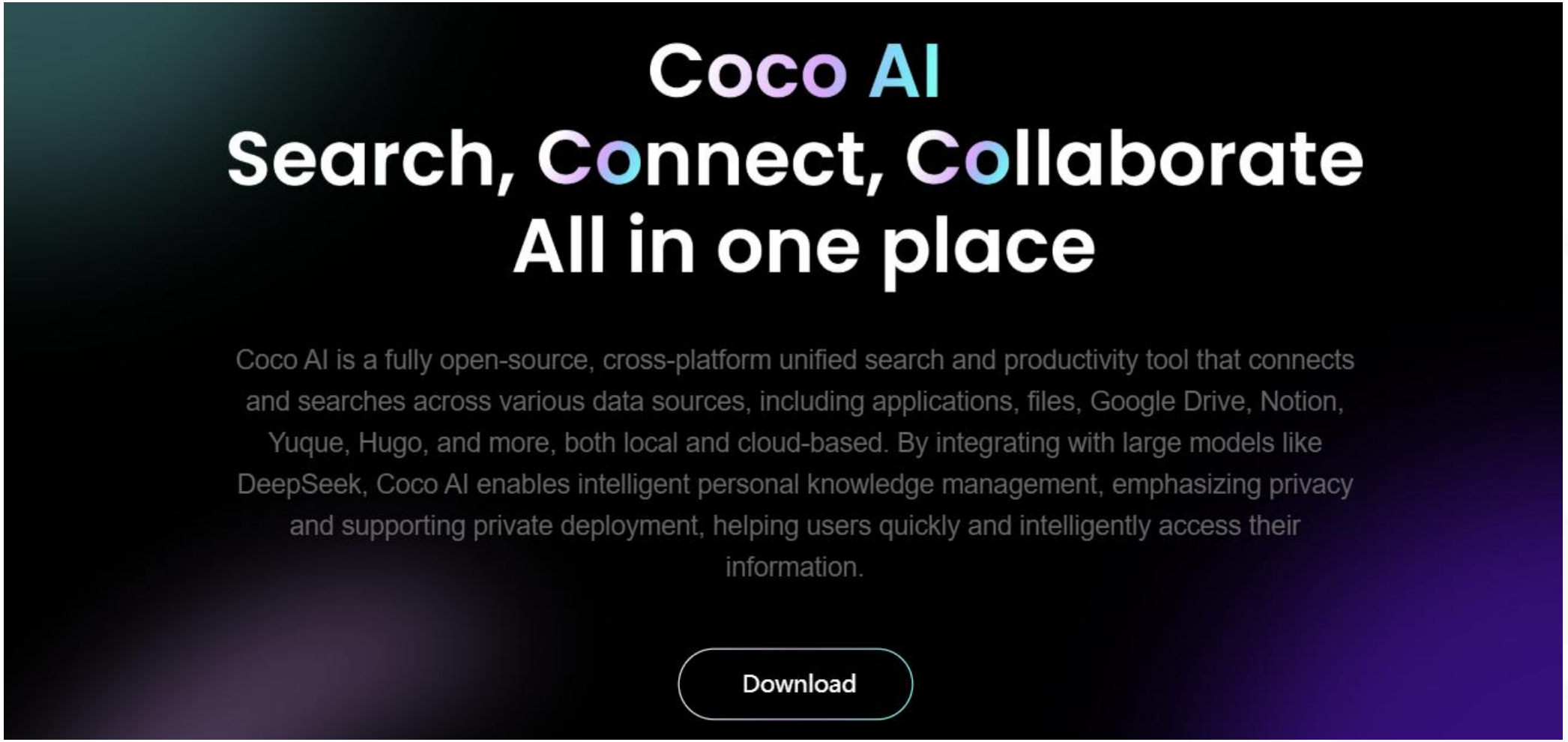
墨天轮

数据库/lib	分布式	数据规模	功能丰富度	学习曲线	核心优势
Milvus	是	大	中	陡峭	百亿级向量实时检索
Pinecone	是 不能私有化部署	中-小	低	平缓	全托管低延迟查询
Easysearch	是	大	高	平缓	多种检索类型、分析
Chroma	否	小	低	平缓	轻量化开发与快速原型

Coco AI

墨天轮

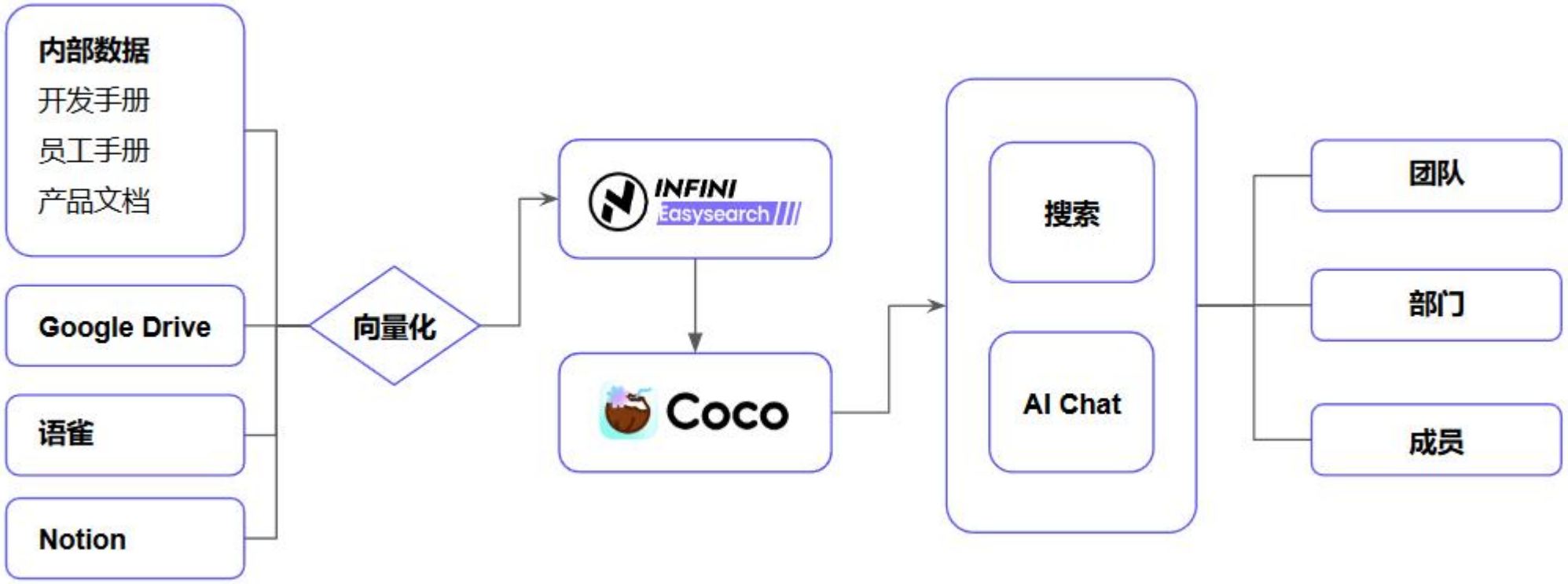
开源项目　　https://coco.rs/



Coco AI

墨天轮

一站式企业搜索与 AI 智能中心

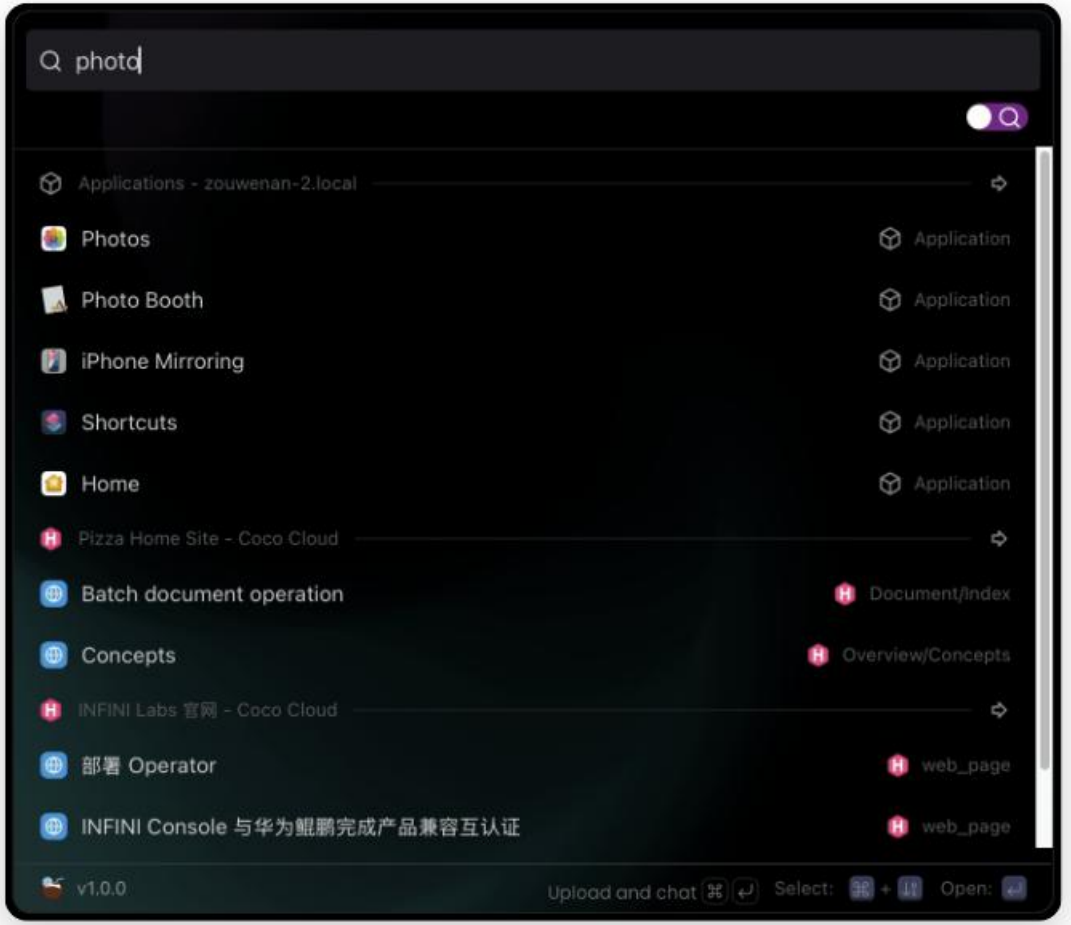


Coco AI

跨平台统一搜索

连接本地文件数据源、S3 对象存储、Google Workspace、Dropbox、GitHub、Notion、Yuque、Hugo 等多种数据源

实现本地与云端数据的统一搜索与管理

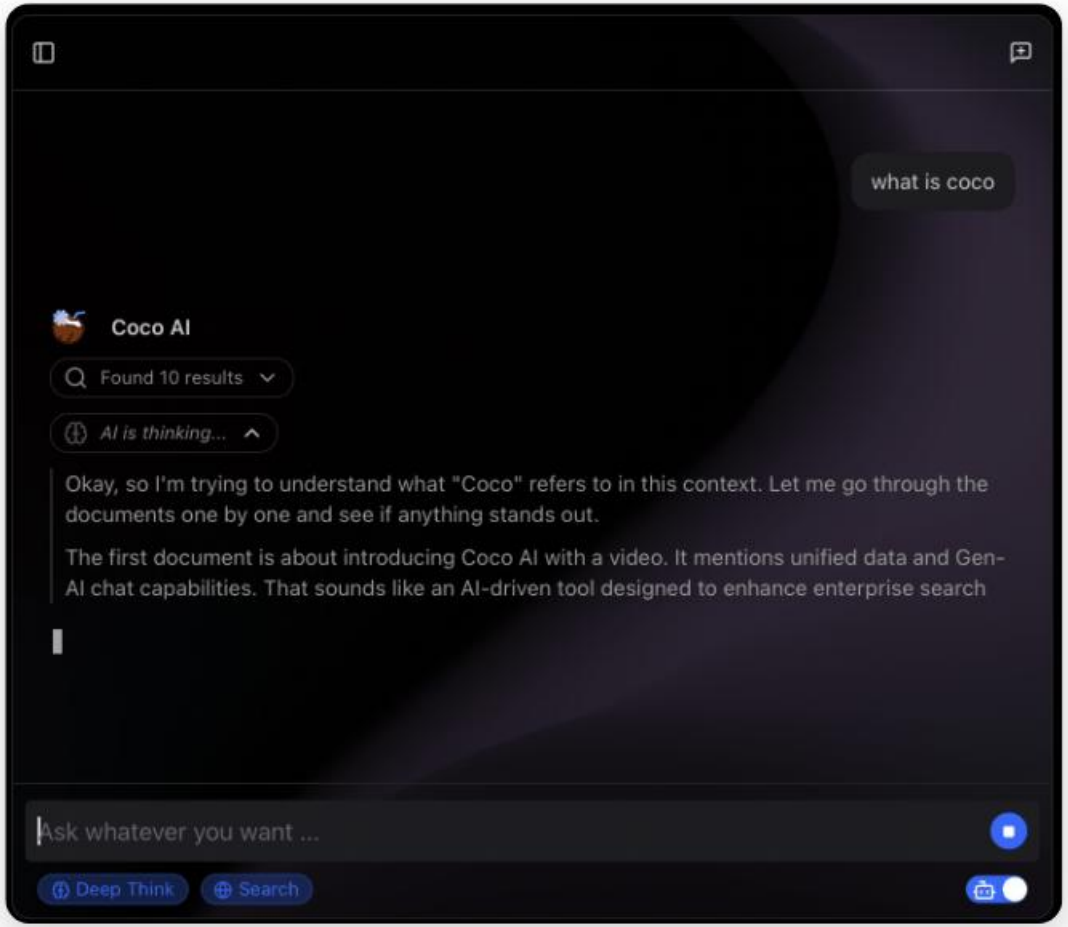


Coco AI

AI 驱动的知识管理

集成 ChatGPT、DeepSeek 等 AI 模型，提供智能的知识管理功能

优先考虑隐私保护，同时支持私有部署，确保企业数据的安全

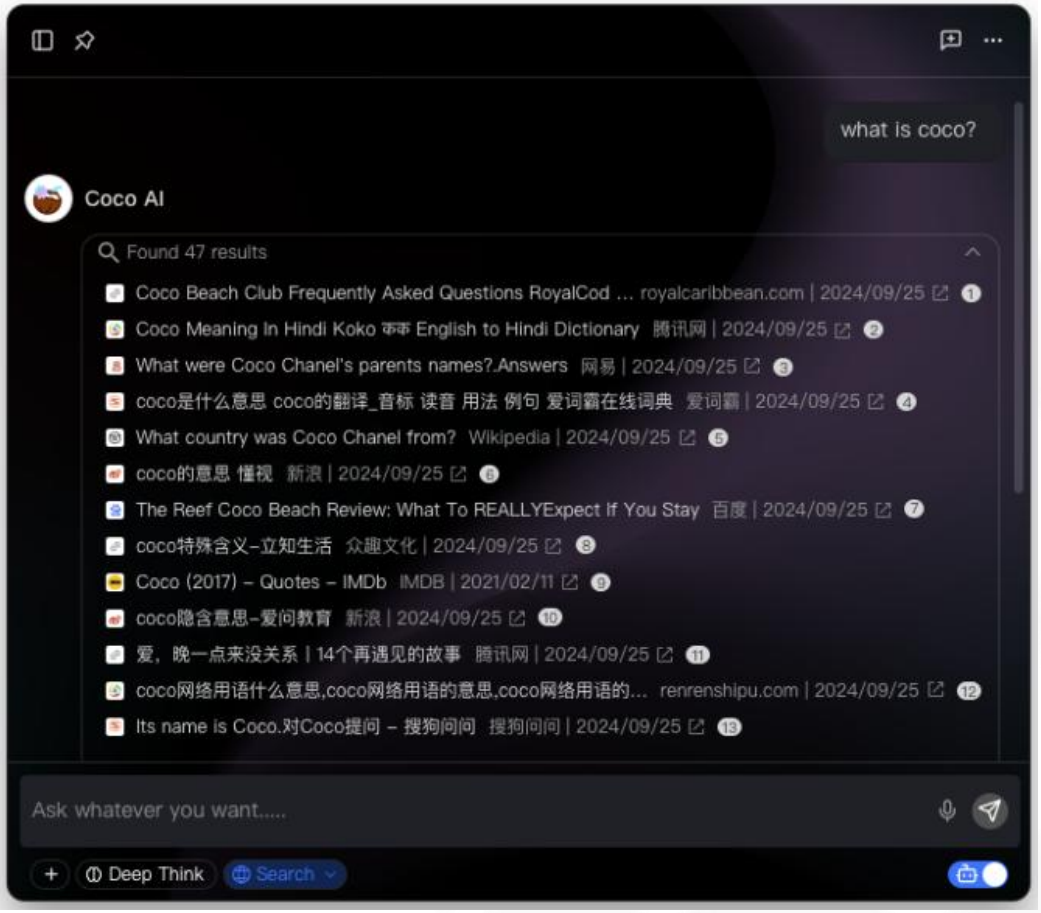


Coco AI

RAG (Retrieval-Augmented Generation)

强化检索生成的能力： 通过结合传统的检索和生成模型，Coco AI 提供基于内容检索的生成式答案，提高信息查找的准确度和生成的语义相关性。

提高答案的质量与相关性： RAG 技术不仅提供关键词匹配的搜索结果，还能基于实际内容生成详细且高质量的回答，适应用户不同的需求。



墨天轮

私有部署与云服务

支持企业内部部署，确保数据安全与隐私保护

提供灵活的云端服务选项，满足不同企业的需求

Coco AI


Coco AI
Gemini
What is Coco?
不断进化中
诚挚邀请您加入我们，共同打造更智能、更强大的 Coco AI!

墨天轮

乐知乐享，同心共济。

知行合一，不负所托!