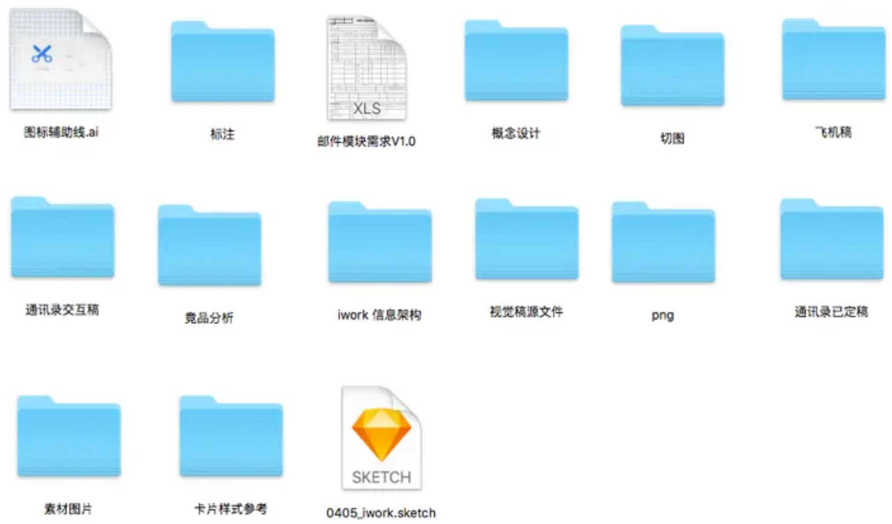
抽丝剥茧手撕RAG，本地知识库检索原理与开发

1. 需求概述

- 照片越来越多，如何从大量照片中查找？
- 文档越来越多，如何从大量文档中查找？

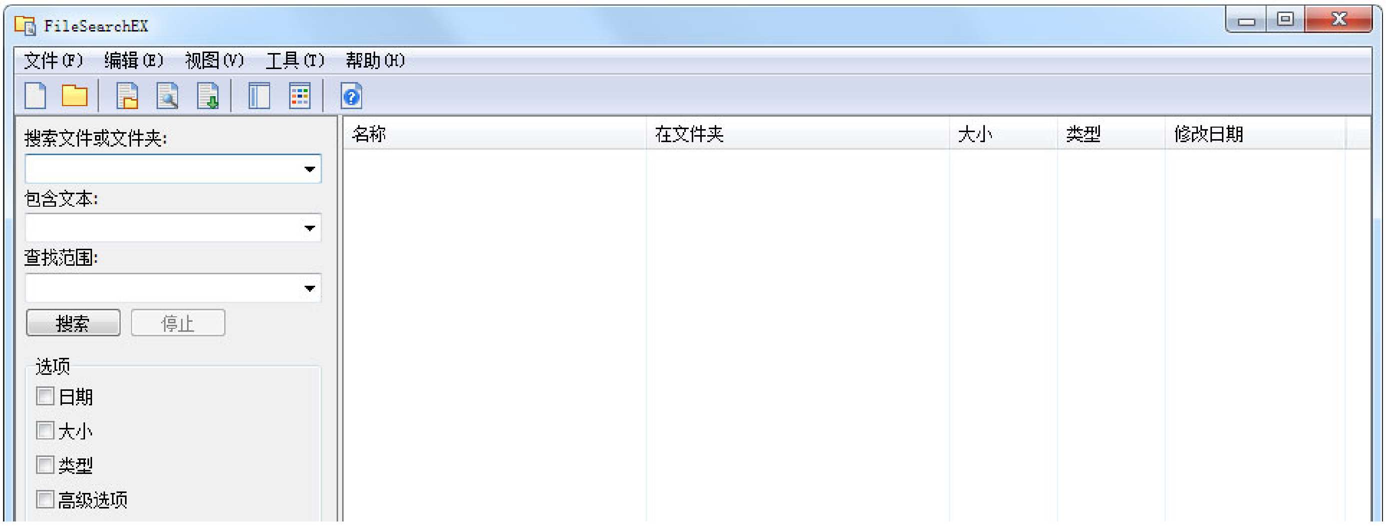
在大量图片查找



在大量文档中查找

问题：

- 搜索文件或文件夹 只能搜索文件名，并不能搜索到文档中的内容



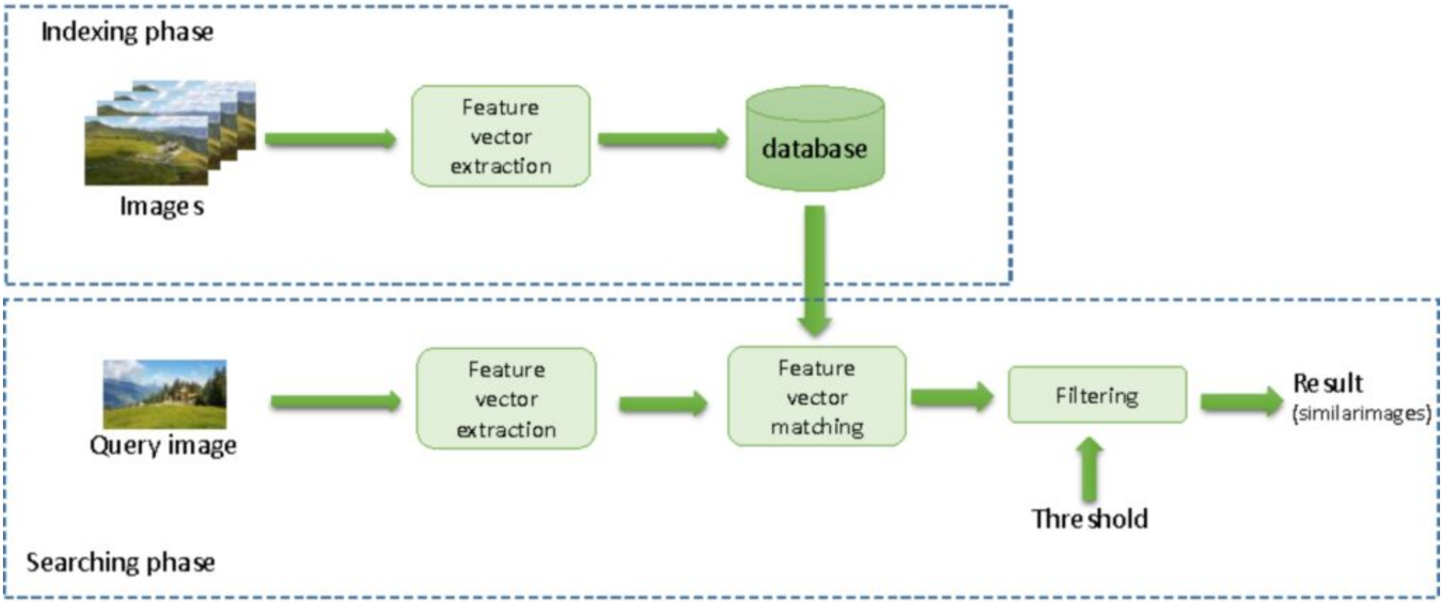
- 如何搜索文档内容？如何在一大堆文档中，搜索出我想要的内容呢？
- 公司内部大量的产品设计、使用说明、规章制度等构成的知识库，如何有效查找和检索？

- 实例演示。

2. 技术分析

1.1 以图搜图

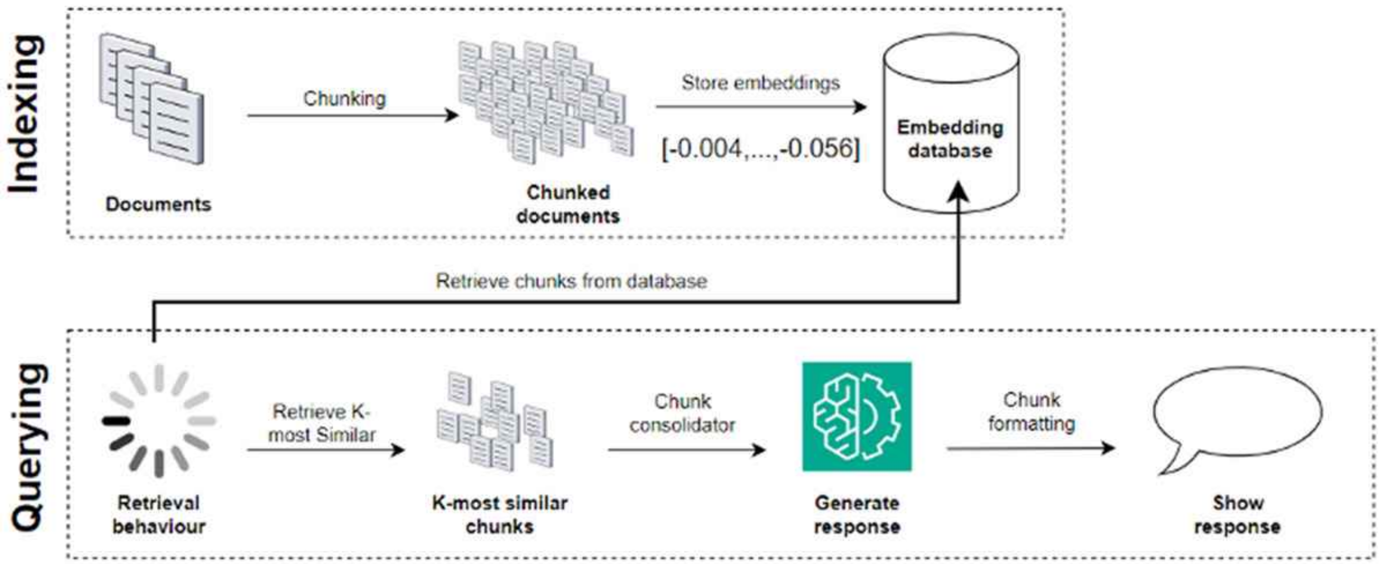
- 在一大堆图片中，搜索出想要的图片内容



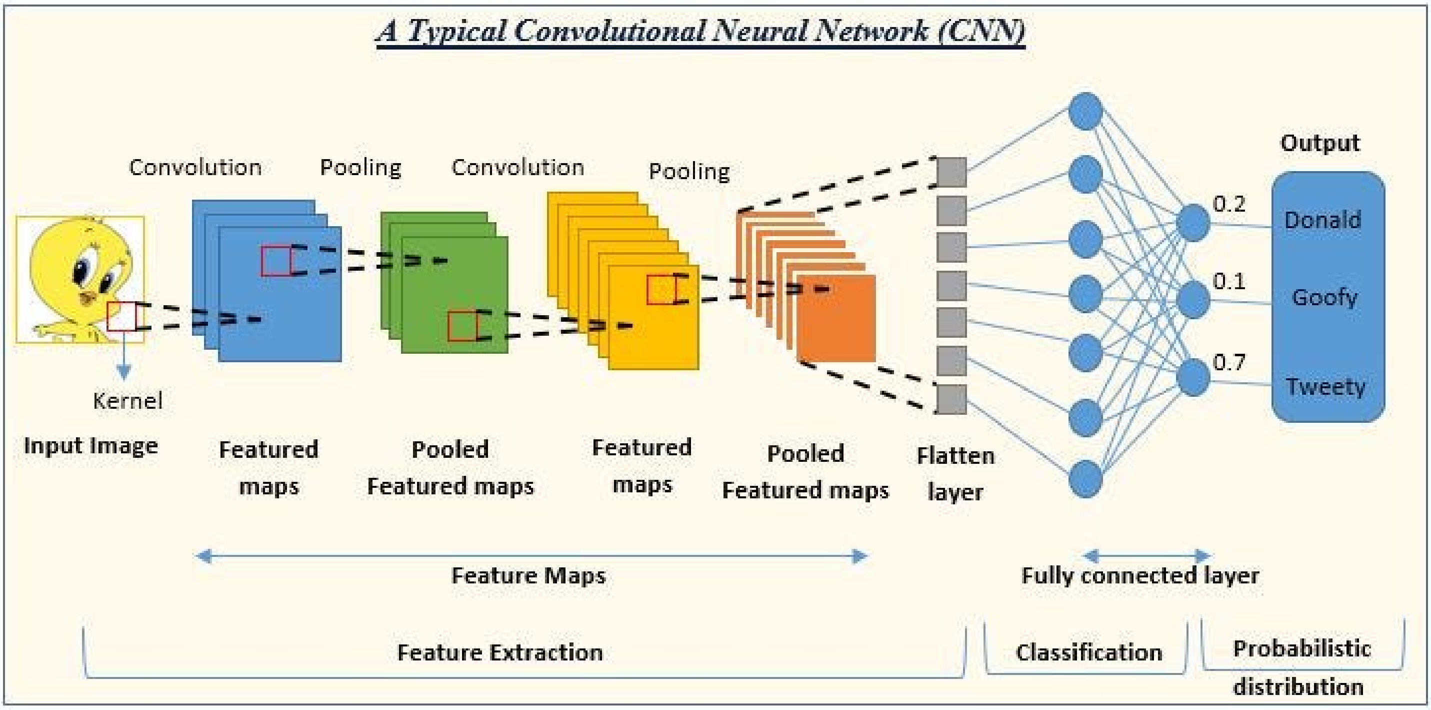
General workflow for reverse image search

1.2 搜索文档

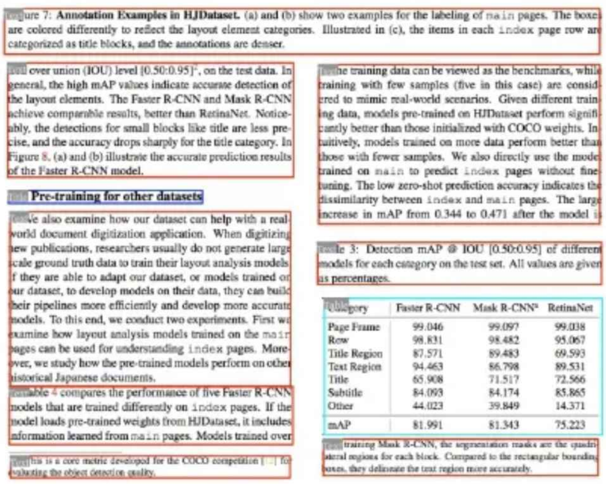
- 在一大堆文档中，搜索出想要的文字内容



- **不同点**



array([[6.48558010e-01, -7.31574174e-01, 8.34732140e-01,
        -6.95522718e-02, -1.19162460e-01, -3.86208637e-02,
        -1.54963455e+00, 9.96426133e-01, -3.56205319e-01,
         1.39329035e+00, -1.55569899e+00, 1.49137425e+00,
        -2.74209808e+00, -1.27639992e-01, -7.94773369e-01,
         4.41134509e-01, -8.75995218e-01, -9.21835376e-01,
         5.99707214e-01, -1.60583706e+00, 1.18365468e+00,
        -3.20999524e-01, 1.76413408e+00, -1.45934107e+00,
        -1.29465086e+00, -5.50775824e-01, -1.05203143e+00,
        -4.20362294e-02, 2.83038269e-01, 1.51992122e-01,
        -2.48745407e-01, 7.91948494e-01, -4.04546150e-01,
         9.29581175e-02, -1.40017572e-01, -1.07075369e+00,
         6.20790501e-01, 2.40914780e-01, 1.34417580e+00,
         2.47982283e-01, 1.93385242e-01, 8.32165132e-01,
         4.56408677e-01, 7.58521891e-01, 5.27831053e-02,
         1.14158253e+00, -1.68101395e-01, -7.79556683e-02,
        -7.29304970e-01, -8.04259597e-01, 9.25894135e-01,
         1.62594921e+00, -1.14499974e+00, 4.37738882e-01,
        -1.77186139e-01, -1.46990906e-01, 4.40039041e-01,
         1.07757218e+00, -1.53199310e+00, -4.73638682e-01,
         2.76123176e-01, -1.50882365e+00, -3.20495955e+00,
         1.10829683e-01, -3.03008012e-01, 2.86756555e-01,
        -1.40538749e+00, 1.23784241e+00, -7.60325619e-01,
         9.80366468e-02, 1.16398106e+00, 3.02443173e-01,
         1.31030762e+00, 3.64284573e-01, 2.90140038e+00,
...
         4.94936620e-01, -6.98741828e-01, -3.37547593e-01,
        -4.62985357e-02, 6.61291299e-01, -1.32362866e+00,
         6.02243218e-01, -3.49765674e-01, -8.80193185e-01,
        -3.98808821e-01, -2.11313289e-01, -1.31412282e+00,
        -5.58546245e-01]])



ure 7: **Annotation Examples in HJDataset.** (a) and (b) show two examples for the labeling of main pages. The boxe are colored differently to reflect the layout element categories. Illustrated in (c), the items in each index page row ar categorized as title blocks, and the annotations are denser.

over union (IOU) level [0.50:0.95][*], on the test data. In general, the high mAP values indicate accurate detection of the layout elements. The Faster R-CNN and Mask R-CNN achieve comparable results, better than RetinaNet. Noticeably, the detections for small blocks like title are less precise, and the accuracy drops sharply for the title category. In Figure 8, (a) and (b) illustrate the accurate prediction results of the Faster R-CNN model.

Pre-training for other datasets

e also examine how our dataset can help with a real-world document digitization application. When digitizing ew publications, researchers usually do not generate larg cale ground truth data to train their layout analysis models f they are able to adapt our dataset, or models trained o ur dataset, to develop models on their data, they can buil heir pipelines more efficiently and develop more accurat odels. To this end, we conduct two experiments. First w xamine how layout analysis models trained on the mai ages can be used for understanding index pages. Moreover, we study how the pre-trained models perform on othe istorical Japanese documents.

able 4 compares the performance of five Faster R-CNN odels that are trained differently on index pages. If the odel loads pre-trained weights from HJDataset, it includes nformation learned from main pages. Models trained over

his is a core metric developed for the COCO competition [ ] fo valuating the object detection quality.

he training data can be viewed as the benchmarks, whil training with few samples (five in this case) are consid ered to mimic real-world scenarios. Given different train ing data, models pre-trained on HJDataset perform signifi cantly better than those initialized with COCO weights. In tuitively, models trained on more data perform better tha those with fewer samples. We also directly use the mode trained on main to predict index pages without fine tuning. The low zero-shot prediction accuracy indicates th dissimilarity between index and main pages. The larg increase in mAP from 0.344 to 0.471 after the model i

le 3: Detection mAP @ IOU [0.50:0.95] of differen models for each category on the test set. All values are give as percentages.

Category	Faster R-CNN	Mask R-CNN[a]	RetinaNet
Page Frame	99.046	99.097	99.038
Row	98.831	98.482	95.067
Title Region	87.571	89.483	69.593
Text Region	94.463	86.798	89.531
Title	65.908	71.517	72.566
Subtitle	84.093	84.174	85.865
Other	44.023	39.849	14.371
mAP	81.991	81.343	75.223

training Mask R-CNN, the segmentation masks are the quadri lateral regions for each block. Compared to the rectangular bounding boxes, they delineate the text region more accurately.

xxxxxxxxxxxxxxxxxx

xxxx

yyyyyyyyyyyyyyyy

yyyy

zzzzzzzzzzzzzzzzzz

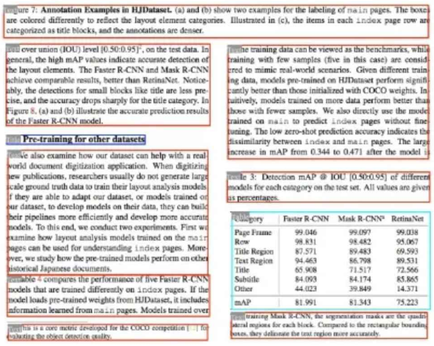
zz

array([[6.48558010e-01, -7.31574174e-01, 8.34732140e-01,
        -6.95522718e-02, -1.19162460e-01, -3.86208637e-02,
        -1.54963455e+00, 9.96426133e-01, -3.56205319e-01,
         1.39329035e+00, -1.55569899e+00, 1.49137425e+00,
        -2.74209808e+00, -1.27639992e-01, -7.94773369e-01,
         4.41134509e-01, -8.75995218e-01, -9.21835376e-01,
         5.99707214e-01, -1.60583706e+00, 1.18365468e+00,
        -3.20999524e-01, 1.76413408e+00, -1.45934107e+00,
        -1.29465086e+00, -5.50775824e-01, -1.05203143e+00,
        -4.20362294e-02, 2.83038269e-01, 1.51992122e-01,
        -2.48745407e-01, 7.91948494e-01, -4.04546150e-01,
         9.29581175e-02, -1.40017572e-01, -1.07075369e+00,
         6.20790501e-01, 2.40914780e-01, 1.34417580e+00,
         2.47982283e-01, 1.93385242e-01, 8.32165132e-01,
         4.56408677e-01, 7.58521891e-01, 5.27831053e-02,
         1.14158253e+00, -1.68101395e-01, -7.79556683e-02,
        -7.29304970e-01, -8.04259597e-01, 9.25894135e-01,
         1.62594921e+00, -1.14499974e+00, 4.37738882e-01,
        -1.77186139e-01, -1.46990906e-01, 4.40039041e-01,
         1.07757218e+00, -1.53199310e+00, -4.73638682e-01,
         2.76123176e-01, -1.50882365e+00, -3.20495955e+00,
         1.10829683e-01, -3.03008012e-01, 2.86756555e-01,
        -1.40538749e+00, 1.23784241e+00, -7.60325619e-01,
         9.80366468e-02, 1.16398106e+00, 3.02443173e-01,
         1.31030762e+00, 3.64284573e-01, 2.90140038e+00,
...
         4.94936620e-01, -6.98741828e-01, -3.37547593e-01,
        -4.62985357e-02, 6.61291299e-01, -1.32362866e+00,
         6.02243218e-01, -3.49765674e-01, -8.80193185e-01,
        -3.98808821e-01, -2.11313289e-01, -1.31412282e+00,
        -5.58546245e-01]])

- 对长文本进行直接压缩或抽象化

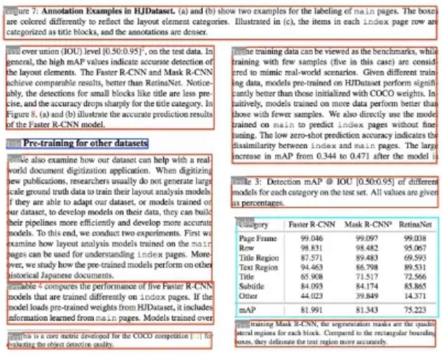
- 丢失的细节过多
- 不利于检索

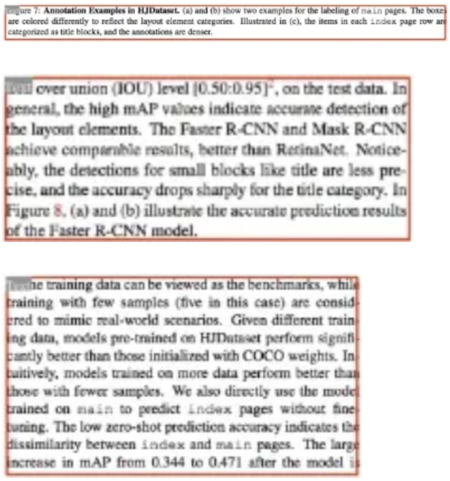
- **方法一：倒排索引**



单词ID	单词	文档频率	倒排列表（DocID;TF;<POS>）
1	谷歌	5	(1;1;<1>), (2;1;<1>), (3;2;<1;6>), (4;1;<1>), (5;1;<1>)
2	地图	5	(1;1;<2>), (2;1;<2>), (3;1;<2>), (4;1;<2>), (5;1;<2>)
3	之父	4	<1;1;<3>), (2;1;<3>), (4;1;<3>), (5;1;<3>)
4	跳槽	2	(1;1;<4>), (4;1;<4>)
5	Facebook	5	(1;1;<5>), (2;1;<5>), (3;1;<8>), (4;1;<5>), (5;1;<8>)
6	加盟	3	(2;1;<4>), (3;1;<7>), (5;1;<5>)
7	创始人	1	(3;1;<3>)
8	拉斯	2	(3;1;<4>), (5;1;<4>)
9	离开	1	(3;1;<5>)
10	与	1	（4;1;<6>)

- **方法二：基于片段的特征提取**





[0.81, 0.23, 0.34, 0.56, 0.92, ...]

[0.81, 0.23, 0.34, 0.56, 0.92, ...]

[0.81, 0.23, 0.34, 0.56, 0.92, ...]

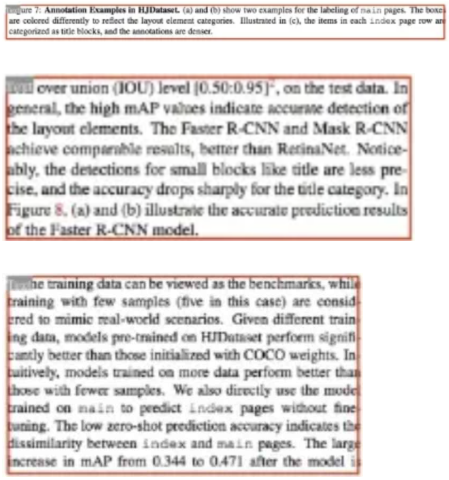
- **两种方法的优劣势：**
  - **方法一：倒排索引**
    - 成熟、高效，算法相对简单
    - 只能精确匹配（如文档中有“下雨”一词，但搜索 raining 不能匹配出来）
  - **方法二：基于片段的特征提取**
    - 基于语义的搜索，可模糊匹配
    - 算法相对复杂，算力资源要求高

Rain

rain

雨

- **共同的问题**
  - 都只是“搜索”
  - 并不能给出问题的“答案”
  - “答案”需要用户根据“搜索”的结果，自己去总结
    - 如用户问“王总的电话是多少？”

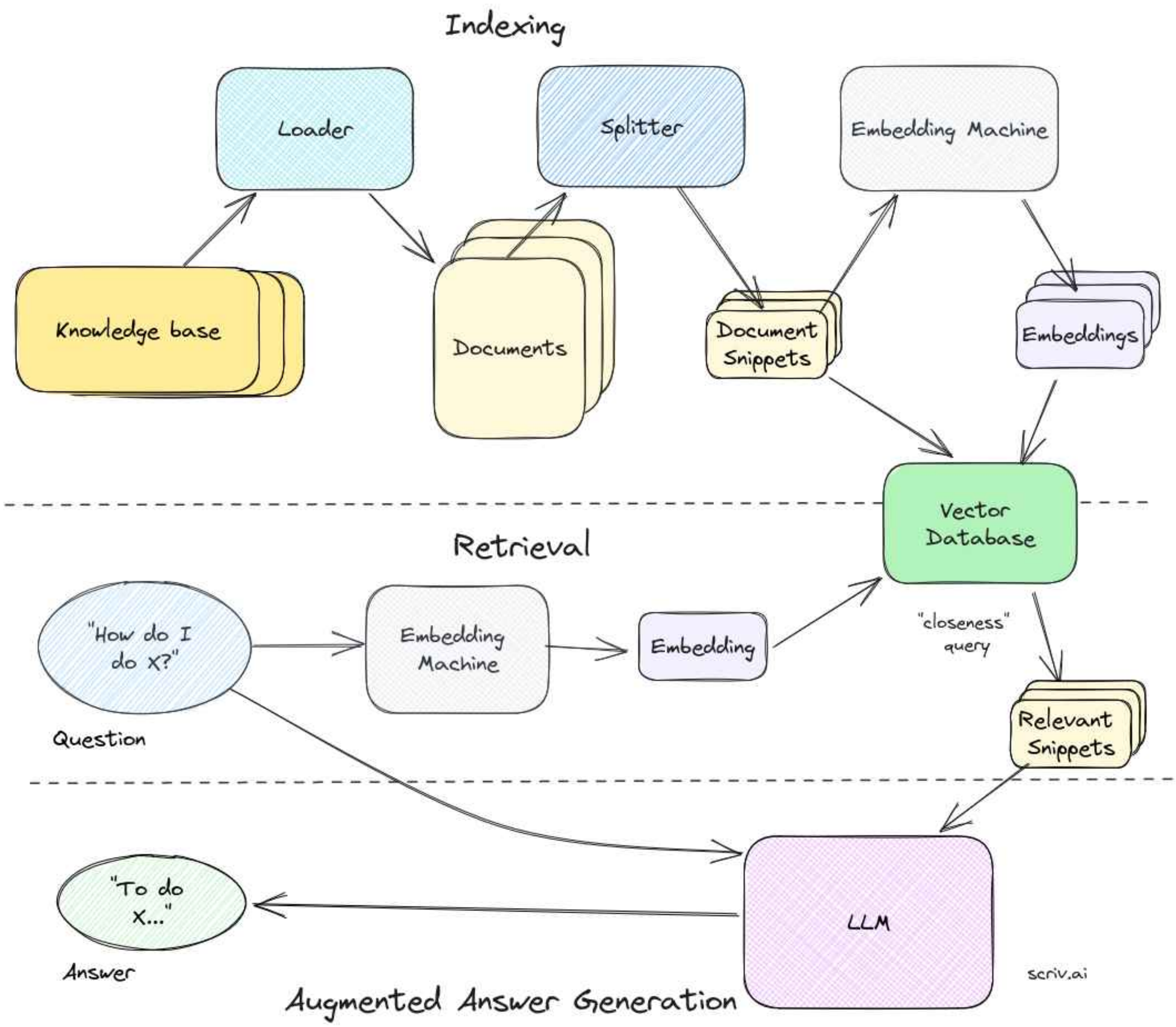

gure 7: **Annotation Examples in HJDataset.** (a) and (b) show two examples for the labeling of main pages. The boxes are colored differently to reflect the layout element categories. Illustrated in (c), the items in each index page row are categorized as title blocks, and the annotations are denser.

over union (IOU) level [0.50:0.95], on the test data. In general, the high mAP values indicate accurate detection of the layout elements. The Faster R-CNN and Mask R-CNN achieve comparable results, better than RetinaNet. Noticeably, the detections for small blocks like title are less precise, and the accuracy drops sharply for the title category. In Figure 8, (a) and (b) illustrate the accurate prediction results of the Faster R-CNN model.

he training data can be viewed as the benchmarks, while training with few samples (five in this case) are considered to mimic real-world scenarios. Given different training data, models pre-trained on HJDataset perform significantly better than those initialized with COCO weights. Intuitively, models trained on more data perform better than those with fewer samples. We also directly use the model trained on main to predict index pages without finetuning. The low zero-shot prediction accuracy indicates the dissimilarity between index and main pages. The large increase in mAP from 0.344 to 0.471 after the model i

1.3 RAG

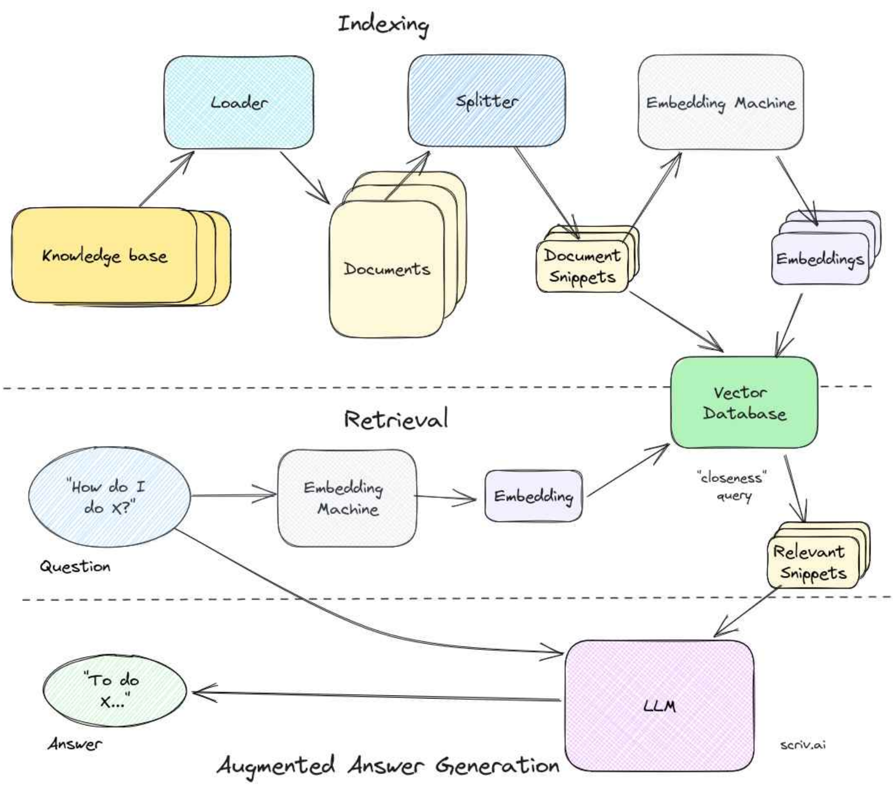
- 检索增强生成 Retrieval-Augmented Generation，是一种结合了检索和生成的 自然语言处理方法。



- RAG模型结合了两种不同的技术：信息检索（Retrieval）和文本生成（Generation）。

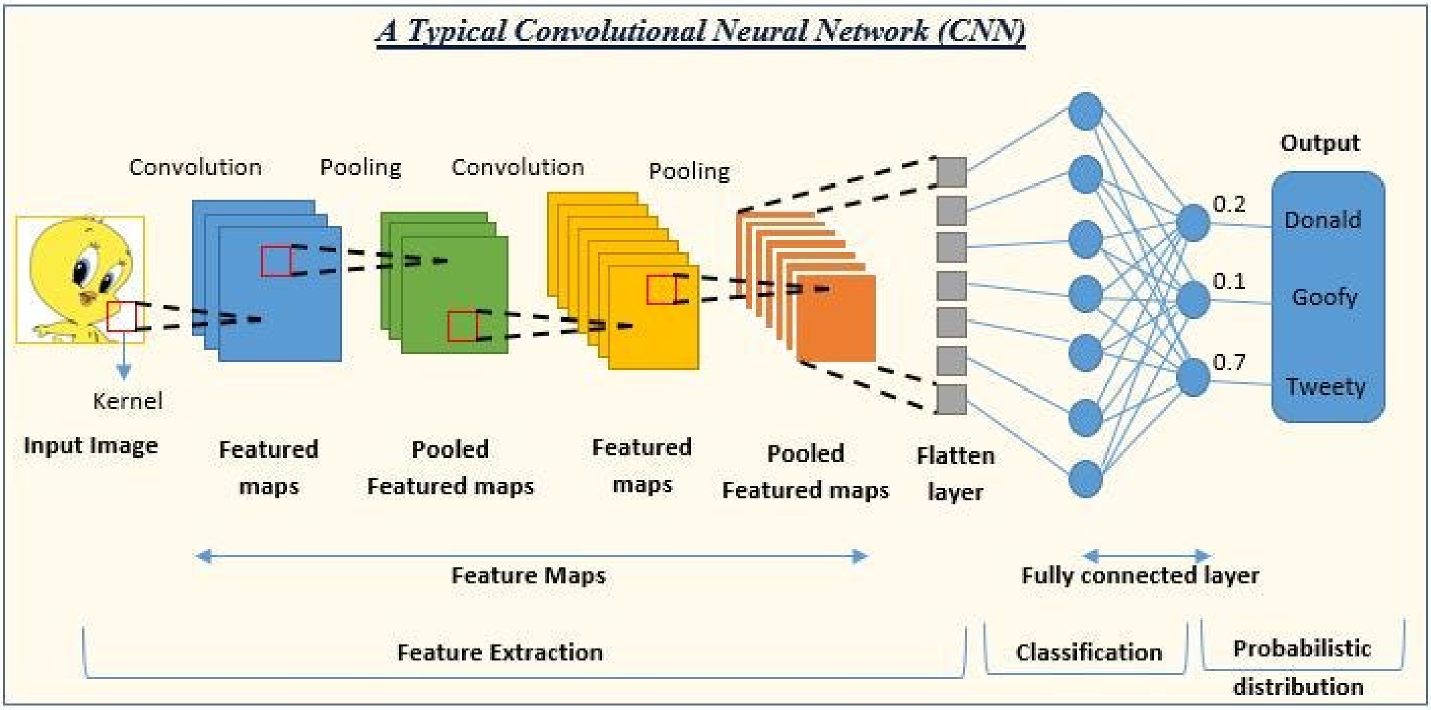
- **信息检索**（Retrieval）从大量的数据源中找到与用户输入最相关的信息。目的是检索出与用户问题或请求相关的文档或文本片段。
- **文本生成**（Generation）利用这些信息来生成响应或输出。能够根据检索到的上下文生成连贯、相关的文本。
- 知识库索引（Indexing）

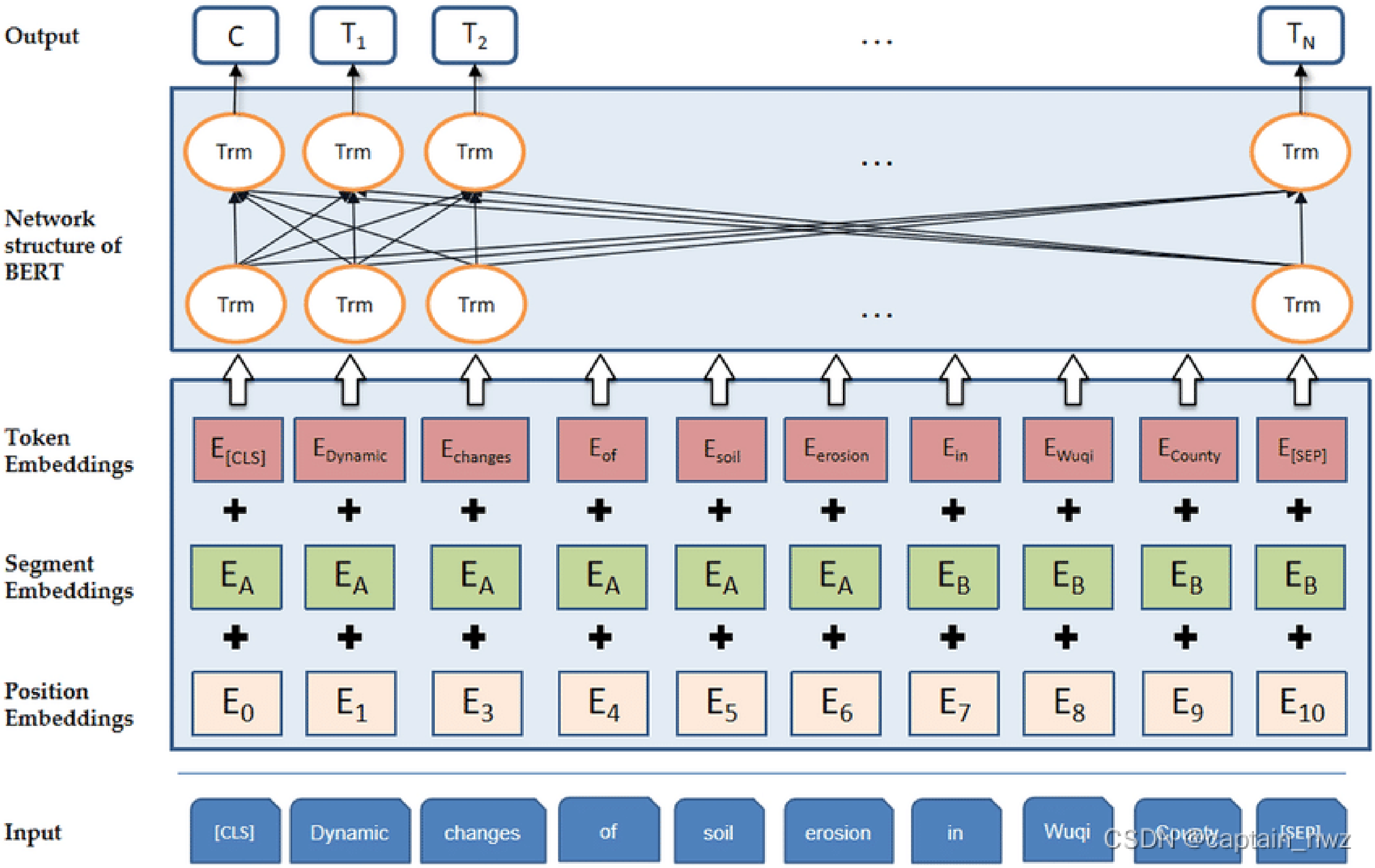
1.4 主要技术



- 句子特征提取 Embedding
- 向量数据库 Vector Database
- 相似度比较
  - Cosine
  - Rerank
- 大语言模型 LLM
- 图形界面

3. 句子特征





- Attention替换掉了卷积层
- 随着网络更深，学到越来越多的上下文信息
- 最终池化得到句子特征

句子特征本质上是一种文本信息的压缩或抽象化。记录了句子中的关键信息，去除冗余，降低维度。

表示为：**[0.81, 0.23, 0.34, 0.56, 0.92, ...]**

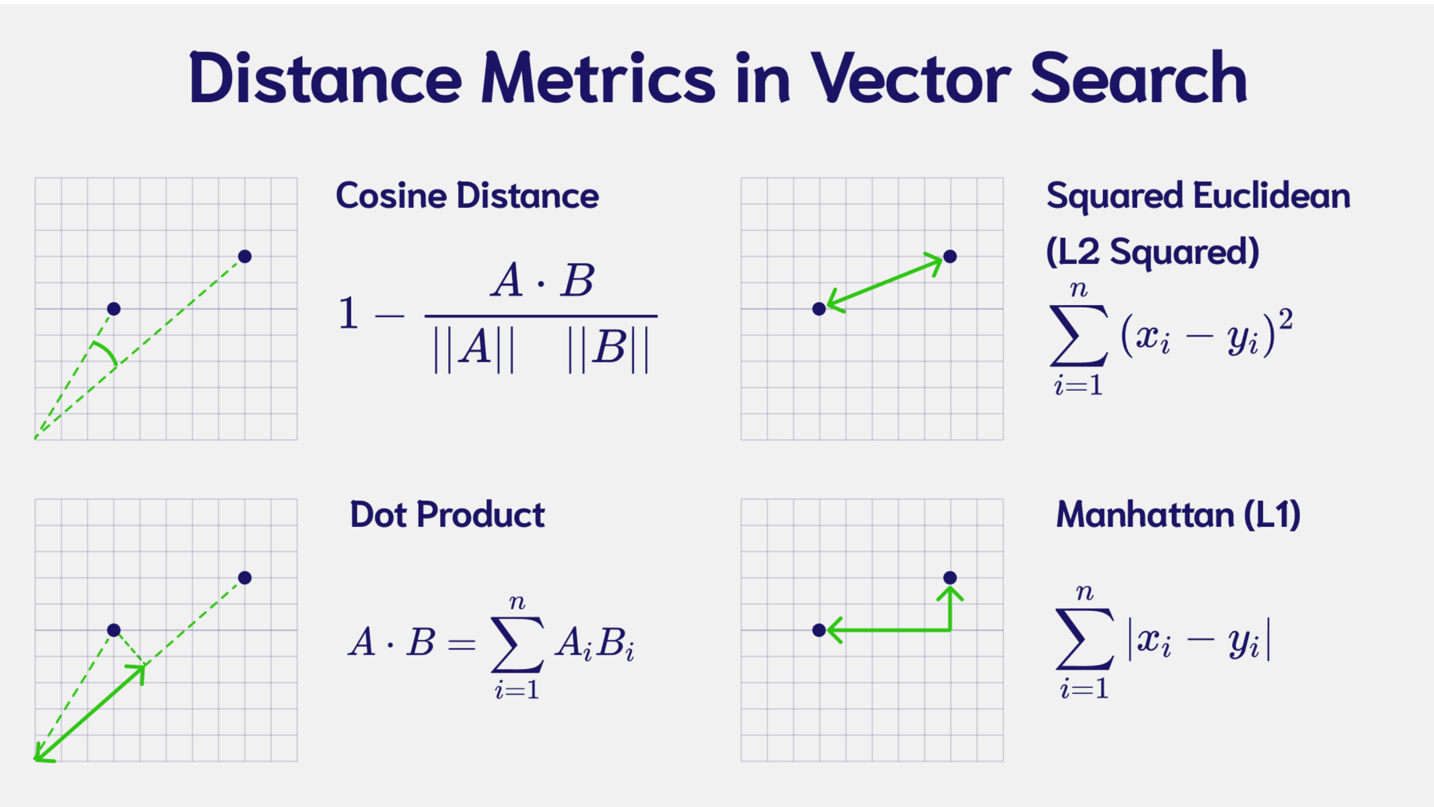
4. 向量数据库

- Embedding is All You Need
- 向量无处不在
- 为什么要用向量数据库
- 向量数据库的特点和索引原理

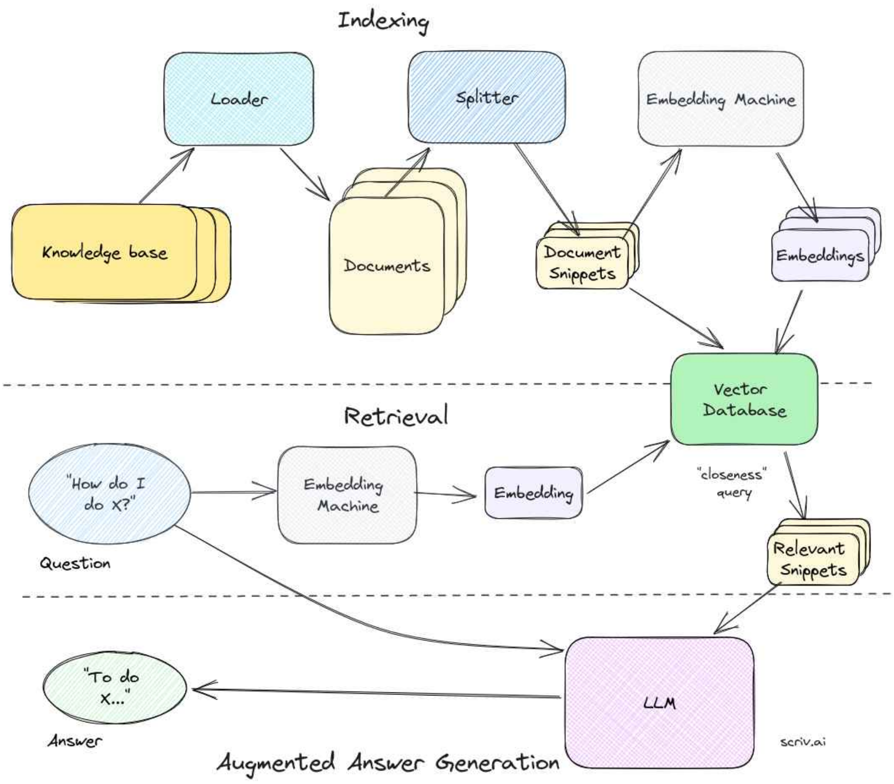
5. 相似度比较

- 句子1: **[0.81, 0.23, 0.34, 0.56, 0.92, ...]**
- 句子2: **[0.50, 0.82, 0.71, 0.21, 0.34, ...]**
- 句子3: **[0.86, 0.21, 0.39, 0.52, 0.89, ...]**
- 如何度量呢?

5.1 Cosine Distance

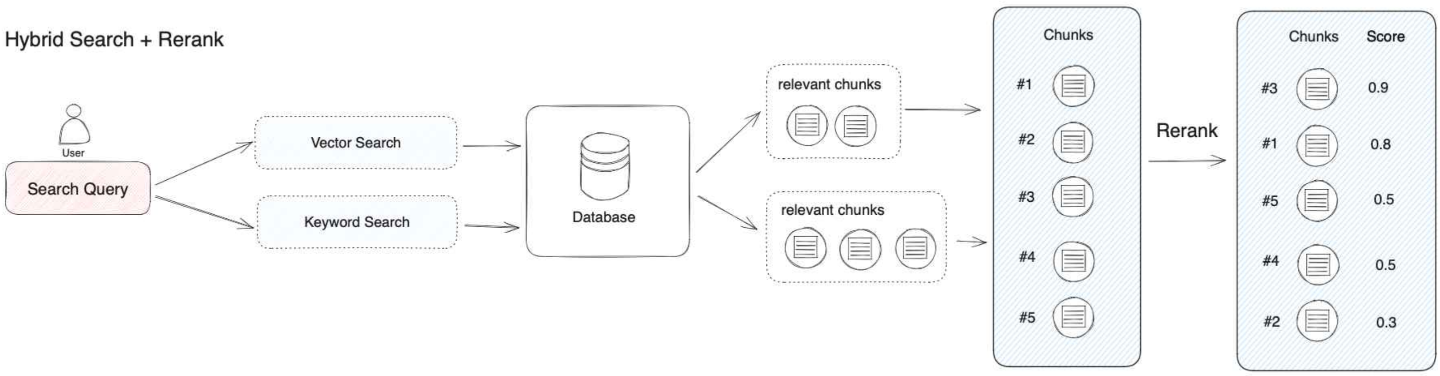


跟图像相对的几个的问题：

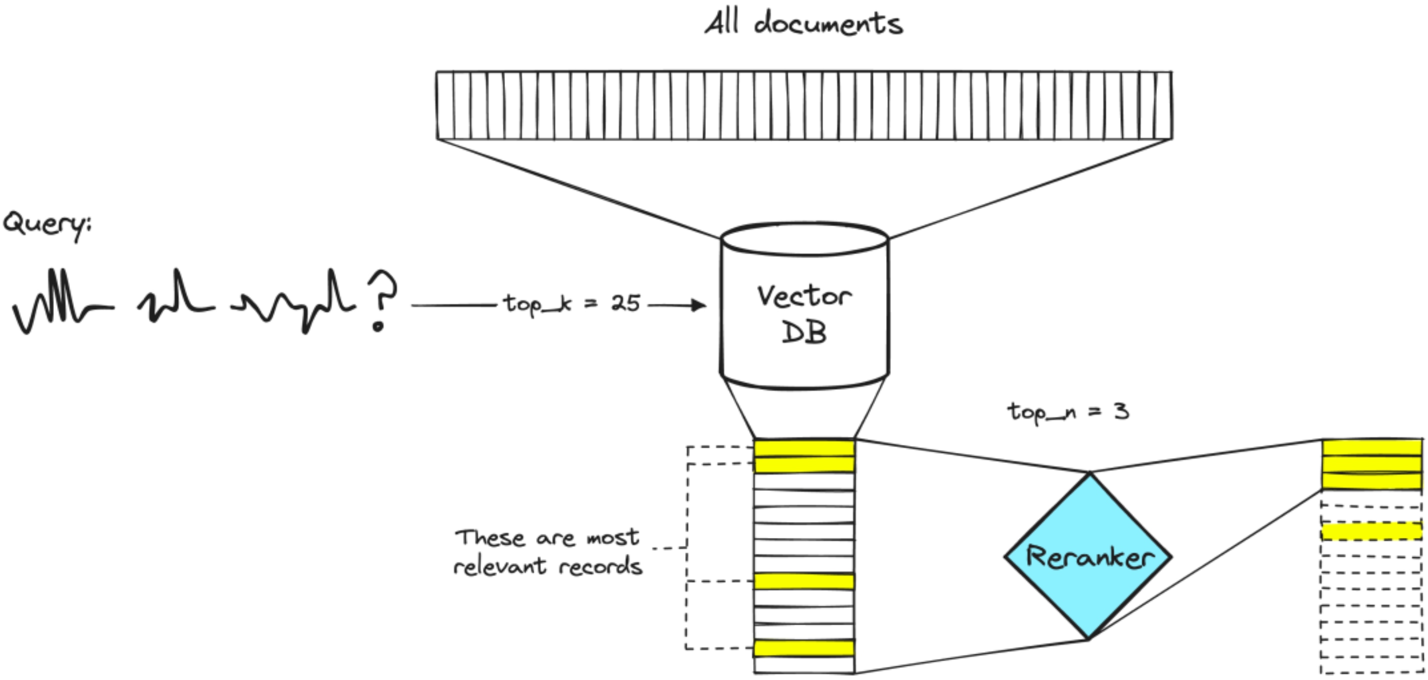


- 相似度比较的重要性（高召回率）
  - Vector Search
  - Keyword Search
- LLM的token数量（低token数）
  - 精准Rerank
  - 取 top n 条

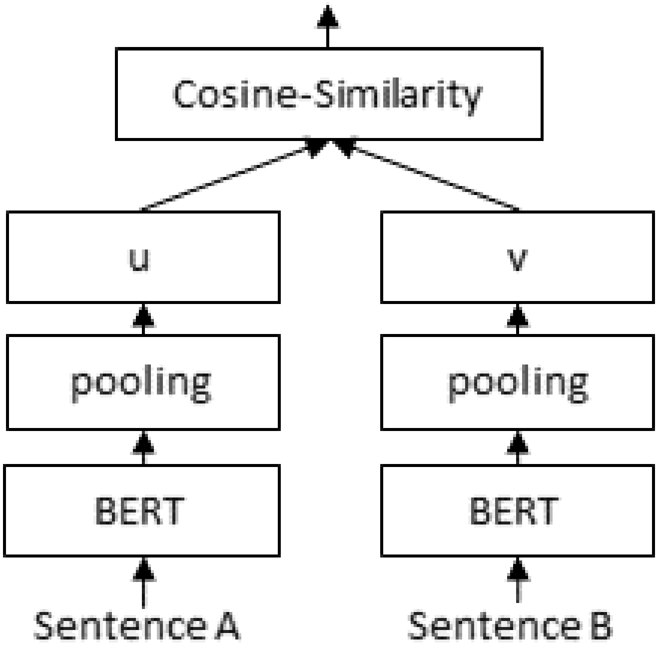
加强的Pipeline（高级RAG）：



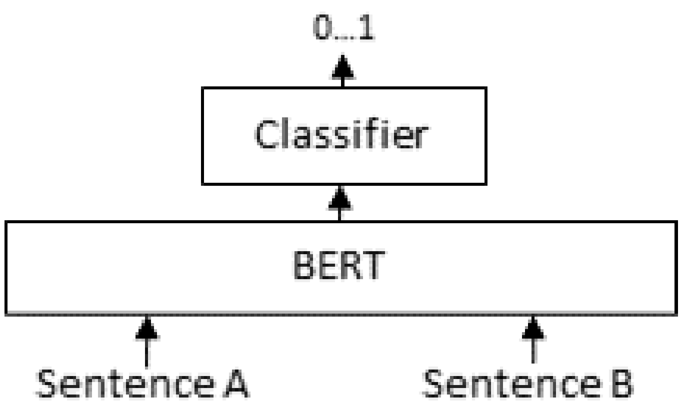
5.2 Rerank模型



实现原理：



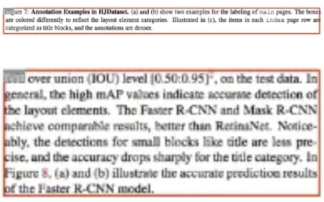
Cosine相似度计算



Rerank模型计算

6. 大语言模型

提问：



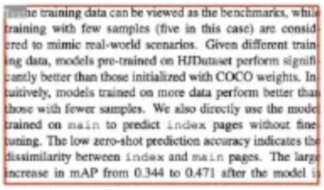
ure 7: **Annotation Examples in HJDataset.** (a) and (b) show two examples for the labeling of main pages. The boxes are colored differently to reflect the layout element categories. Illustrated in (c), the items in each index page row are categorized as title blocks, and the annotations are denser.

over union (IOU) level [0.50:0.95], on the test data. In general, the high mAP values indicate accurate detection of the layout elements. The Faster R-CNN and Mask R-CNN achieve comparable results, better than RetinaNet. Noticeably, the detections for small blocks like title are less precise, and the accuracy drops sharply for the title category. In Figure 8, (a) and (b) illustrate the accurate prediction results of the Faster R-CNN model.

回复：

芳华站。

园区公交1号线的起点站在哪？



he training data can be viewed as the benchmarks, whil training with few samples (five in this case) are consid- ered to mimic real-world scenarios. Given different train- ing data, models pre-trained on HJDataset perform signifi- cantly better than those initialized with COCO weights. In- tuitively, models trained on more data perform better than those with fewer samples. We also directly use the model trained on `main` to predict `index` pages without fine- tuning. The low zero-shot prediction accuracy indicates the dissimilarity between `index` and `main` pages. The larg increase in mAP from 0.344 to 0.471 after the model i

存在的矛盾：

- 提供的信息越具体越好
- LLM的上下文窗口有限
- 按token收费

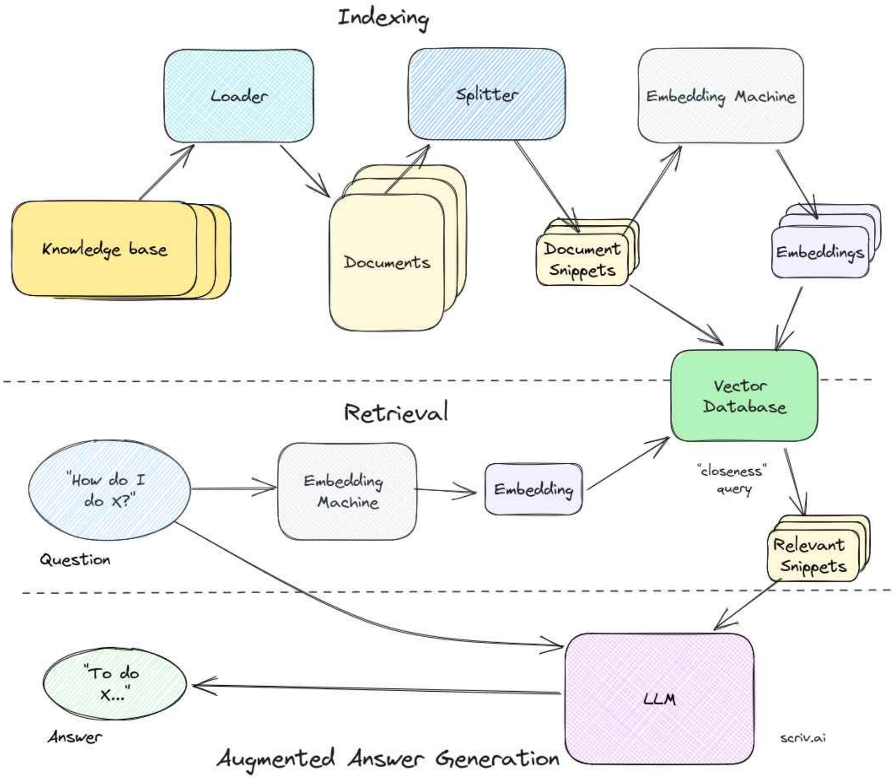
7. 图形界面

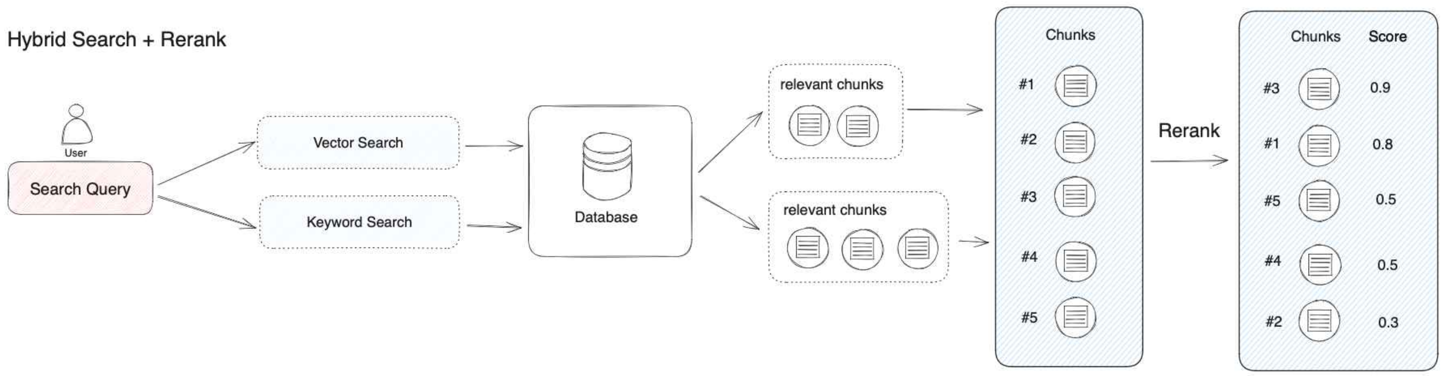
  - **Streamlit**
  - **Gradio**
    - 快速创建简单、交互式Web界面
    - 非常简洁的API，无需了解前端
    - 快速部署应用

8. 代码解析

Talk is cheap. Show me the code.

所有模型都跑在 MacBook Pro 单机上，效果还可以、速度能接受。





1）本机部署LLM

```
https://ollama.com/ 下载安装

> ollama run wangshenzhi/llama3-8b-chinese-chat-ollama-q4
```

2) 安装向量数据库

```
#安装单机版 Milvus，如果之前安装了v2.2.10，可以从docker中stop并删除原版本
$ wget https://github.com/milvus-io/milvus/releases/download/v2.4.4/milvus-standalone-docker-compose.yml -O docker-compose.yml
$ docker-compose up -d
$ docker ps
```

3) 安装开发包

```
pip install -U gradio pymilvus transformers FlagEmbedding langchain langchain-core langchain_community langchain-milvus langchain-text-splitters pypdf2 bs4
```