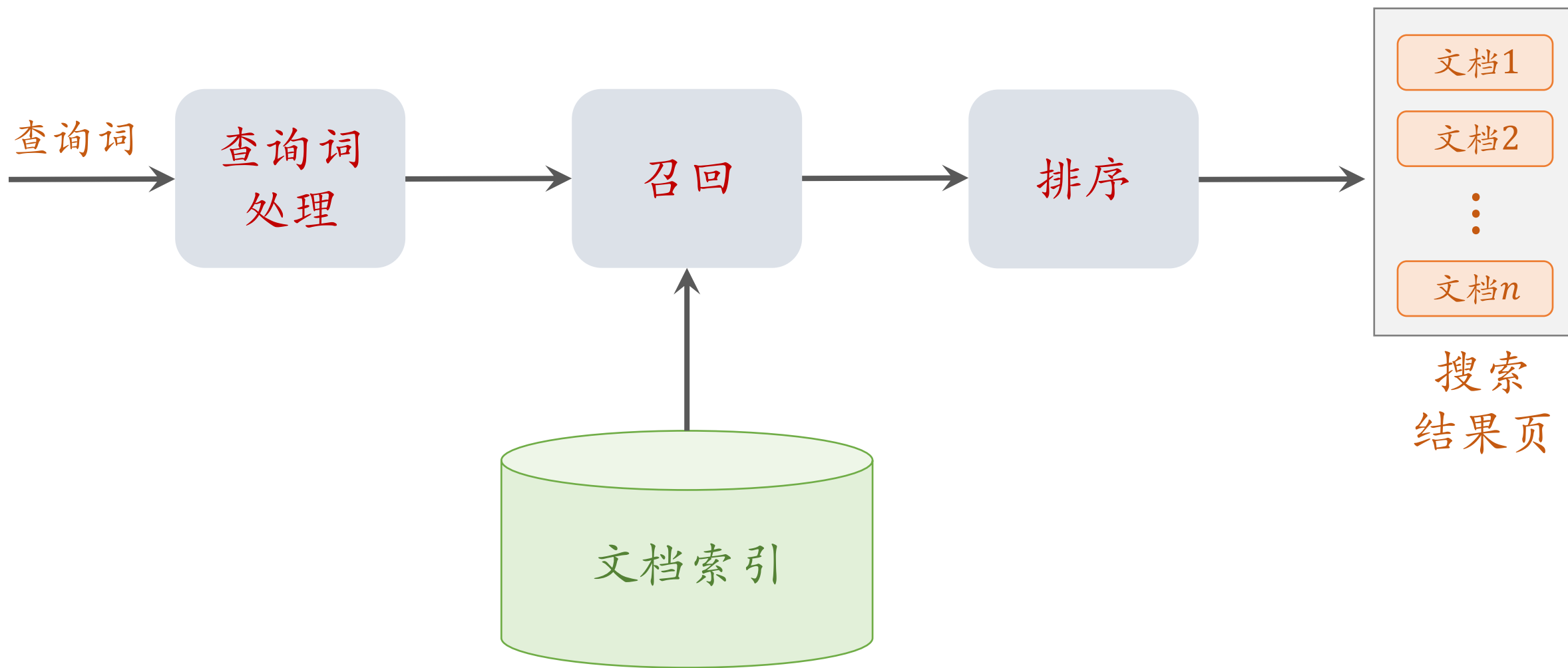


# 搜索引擎的链路

王树森

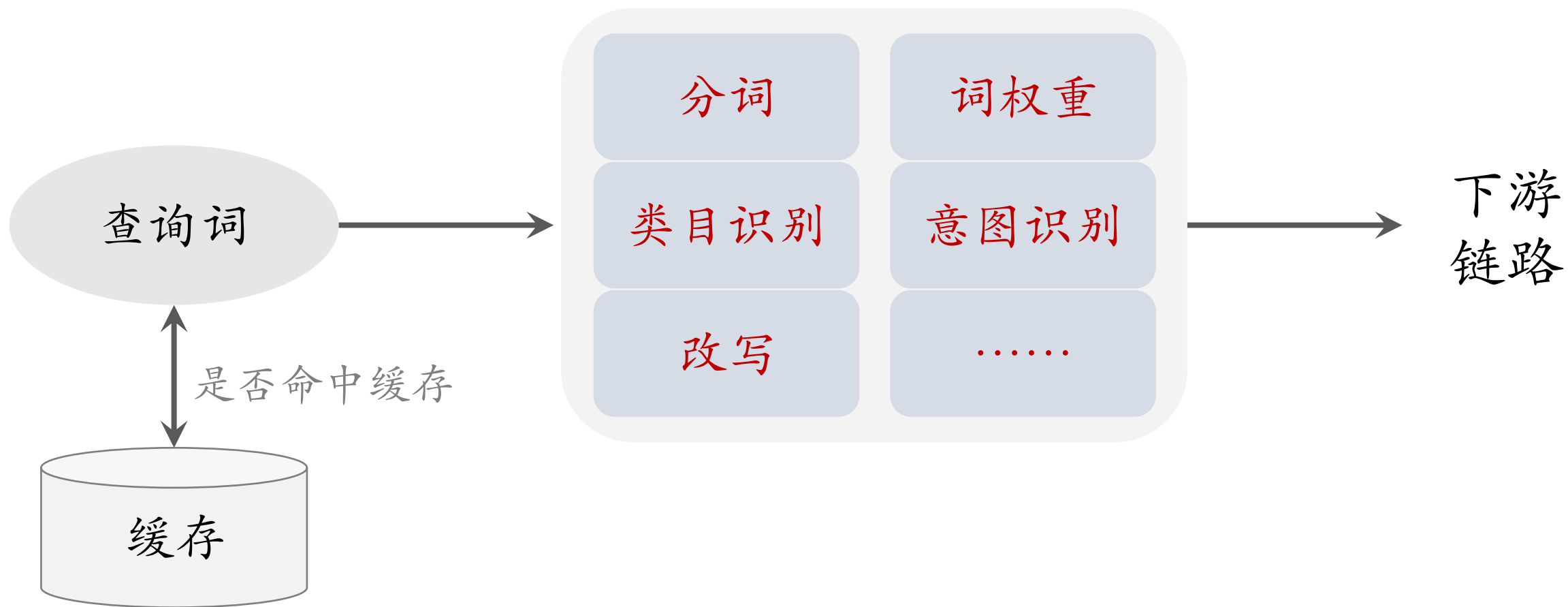
<http://wangshusen.github.io/>

# 搜索引擎的链路



# 查询词处理

# 查询词处理



# 分词

- 例：冬季卫衣推荐 → 冬季 / 卫衣 / 推荐
- 为什么需要做分词？
  - 文本召回根据关键词在倒排索引中检索文档。
  - 倒排索引上大多是“冬季”、“卫衣”、“推荐”这样的常用词，没有“冬季卫衣推荐”这样的词。
  - 如果关键词是“冬季卫衣推荐”，检索不到任何文档。

# 词权重

- 例：冬季卫衣推荐 → 冬季 / 卫衣 / 推荐
- 三个关键词同等重要吗？丢弃一两个关键词可以吗？
- 词权重：卫衣 > 冬季 > 推荐。
- 为什么要计算词权重？
  - 如果查询词太长，没有文档可以同时包含所有关键词，则需要丢弃不重要的关键词。
  - 计算查询词与文档的相关性时，可以用词权重做加权。

# 类目识别

- 每个平台都有各自的多级类目体系。
  - 一级类目：美妆
  - 二级类目：彩妆、护肤、美甲、香水、医美
- 用NLP技术识别文档、查询词的类目。
  - 在文档发布（或被爬虫获取到）时识别文档类目。
  - 在用户做搜索时识别查询词的类目。
- 召回模型、排序模型将文档、查询词类目作为特征。

# 查询词意图识别

- 时效性意图：查询词对文档“新”的需求，召回和排序均需要考虑文档的发布时间。
- 地域性意图：召回和排序不止需要文本相关性，还需要结合用户定位地点、查询词提及的地点、文档定位的地点。
- 用户名意图：用户想要找平台中的某位用户，应当检索用户名库，而非检索文档库。
- 求购意图：用户可能想要买商品，同时在文档库、商品库中做检索。



# 查询词改写

- 用户输入查询词  $q$ ，算法将其改写成多个查询词  $q'_1, \dots, q'_k$ 。  
(独立用  $q, q'_1, \dots, q'_k$  做召回，对召回的文档取并集。)
- 查询词改写有什么用？
- 第一，解决语义匹配、但文本不匹配的问题。
  - $q = \text{“LV包”}$
  - $d = \text{“推荐几款LOUIS VUITTON包包”}$
  - $q$  和  $d$  语义相关，但文本召回无法用  $q$  检索到  $d$ 。

# 查询词改写

- 用户输入查询词  $q$ ，算法将其改写成多个查询词  $q'_1, \dots, q'_k$ ，并独立用  $q, q'_1, \dots, q'_k$  做召回，对召回的文档取并集。
- 查询词改写有什么用？
- 第一，解决语义匹配、但文本不匹配的问题。
- 第二，解决召回文档数量过少的问题。
  - $q$  不规范表达、或  $q$  过长，导致召回结果很少。
  - 例：老谋子拍的电影  $\Rightarrow$  张艺谋的电影
  - 例：身高160体重120年龄20女穿搭推荐  $\Rightarrow$  微胖女大学生穿搭

召回 (Retrieval)

# 召回

- 给定查询词  $q$ ，从文档库（数亿篇文档）中快速检索数万篇可能与  $q$  相关的文档  $\{d\}$ 。
- 文本召回：匹配  $q$  中的关键词和  $d$  中的关键词（借助倒排索引）。
- 向量召回：将  $q$  和  $d$  分别表征为向量  $\mathbf{x}_q$  和  $\mathbf{z}_d$ 。给定  $\mathbf{x}_q$ ，查找相似度高的  $\mathbf{z}_d$ 。
- KV召回：对于高频查询  $q$ ，离线建立  $q \rightarrow \text{List}\langle d \rangle$  这样的 key-value 索引。线上直接读取索引，获取  $q$  相关的文档。

# 文本召回

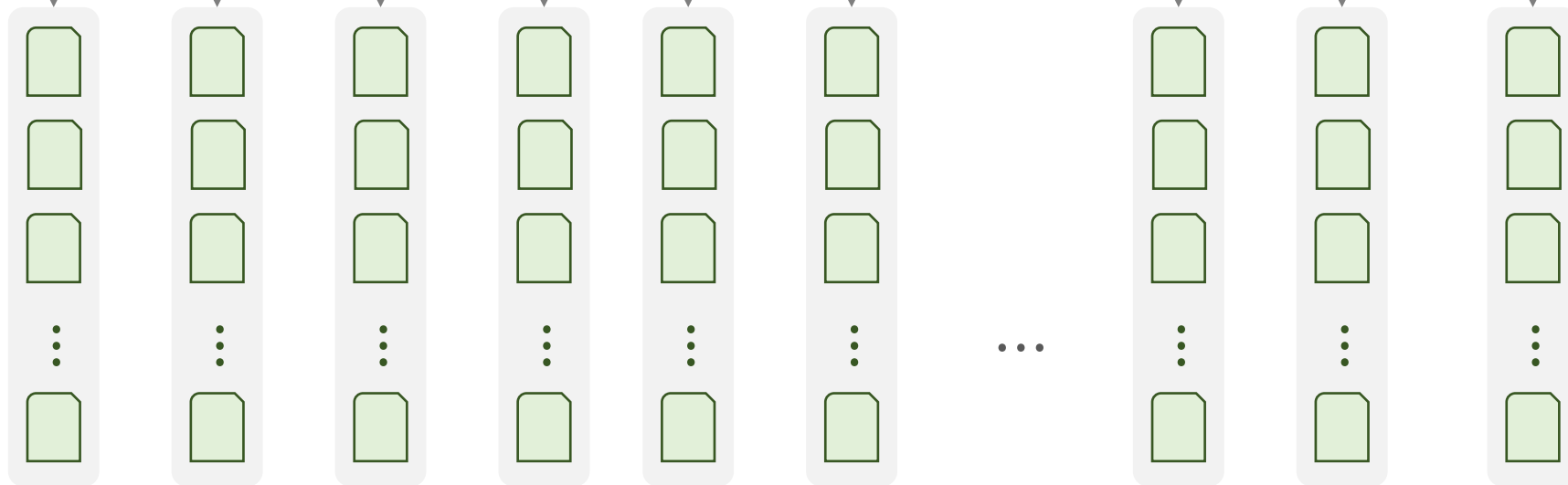
- 离线处理文档，建立倒排索引。（给定关键词  $t$ ，可以快速找到所有包含  $t$  的文档。）
- 给定查询词  $q$ ，做分词得到多个关键词  $t_1, \dots, t_k$ 。
- 对于每个关键词  $t_i$ ，检索倒排索引，得到文档的集合  $\mathcal{D}_i$ 。
- 求  $k$  个集合的交集  $\mathcal{D}_1 \cap \dots \cap \mathcal{D}_k$ ，作为文本召回的结果。
- 交集可能很小、甚至为空。因此需要对  $q$  做丢词、改写。

# 文本召回

关键词：

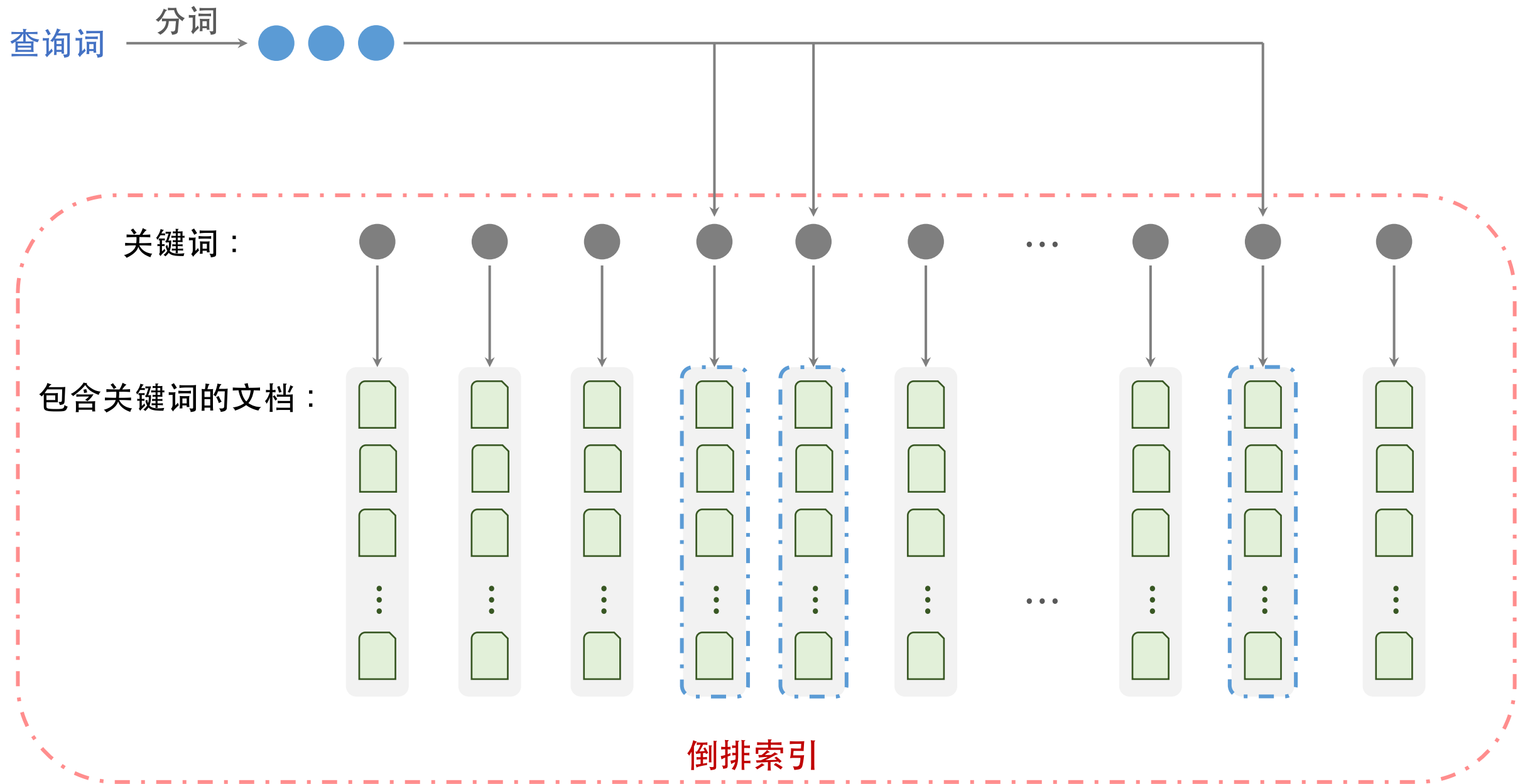


包含关键词的文档：

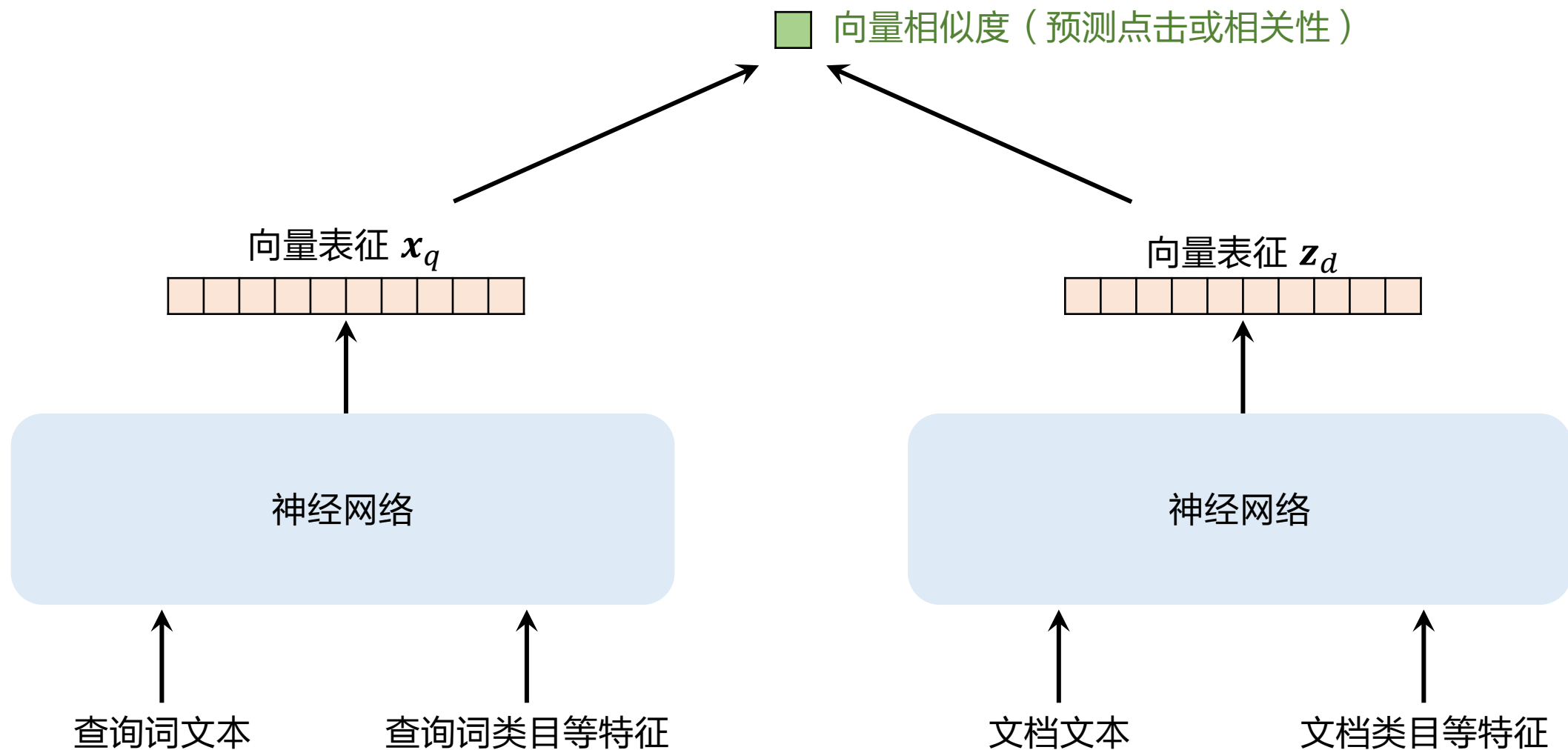


倒排索引

# 文本召回

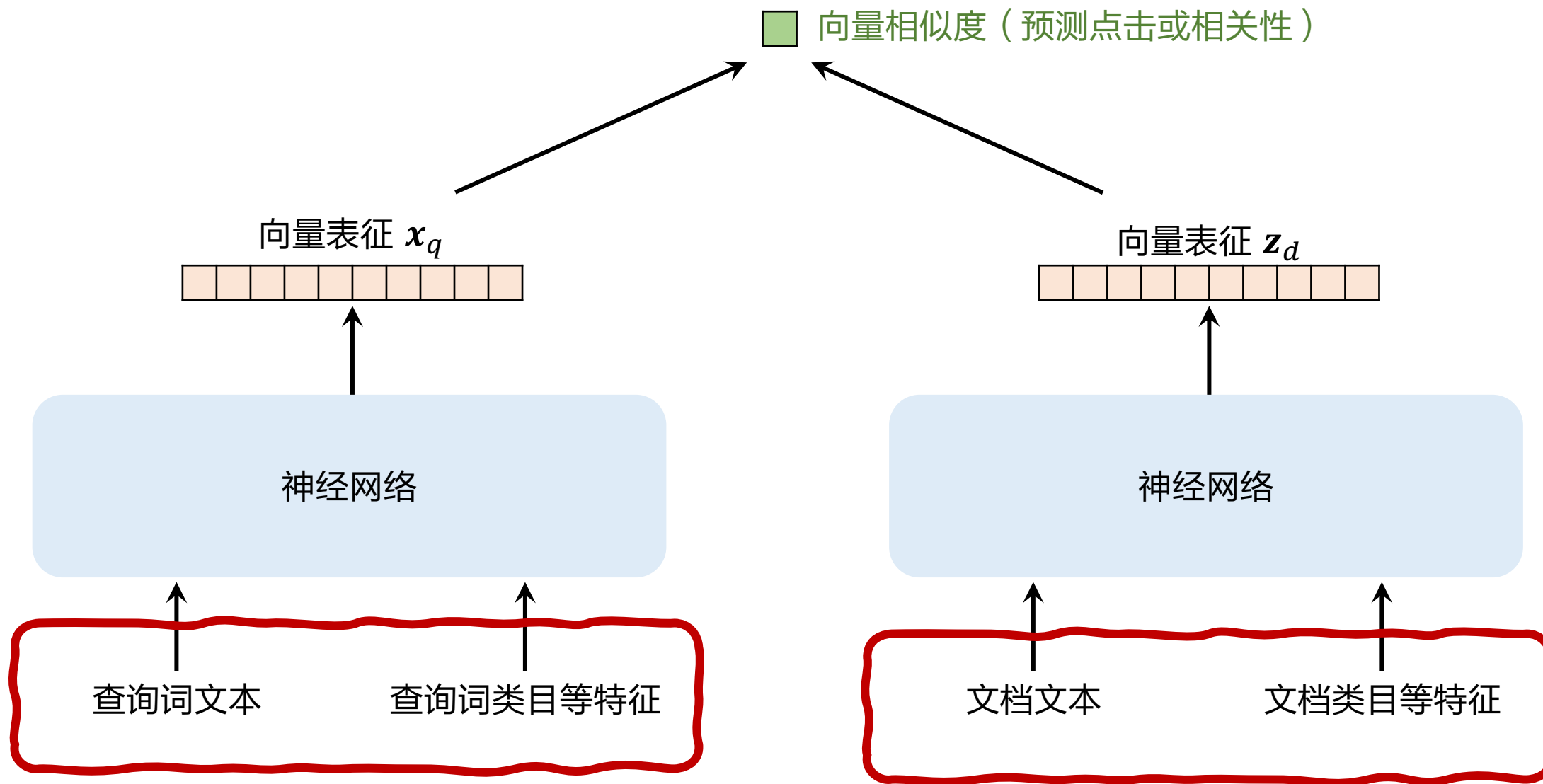


# 向量召回

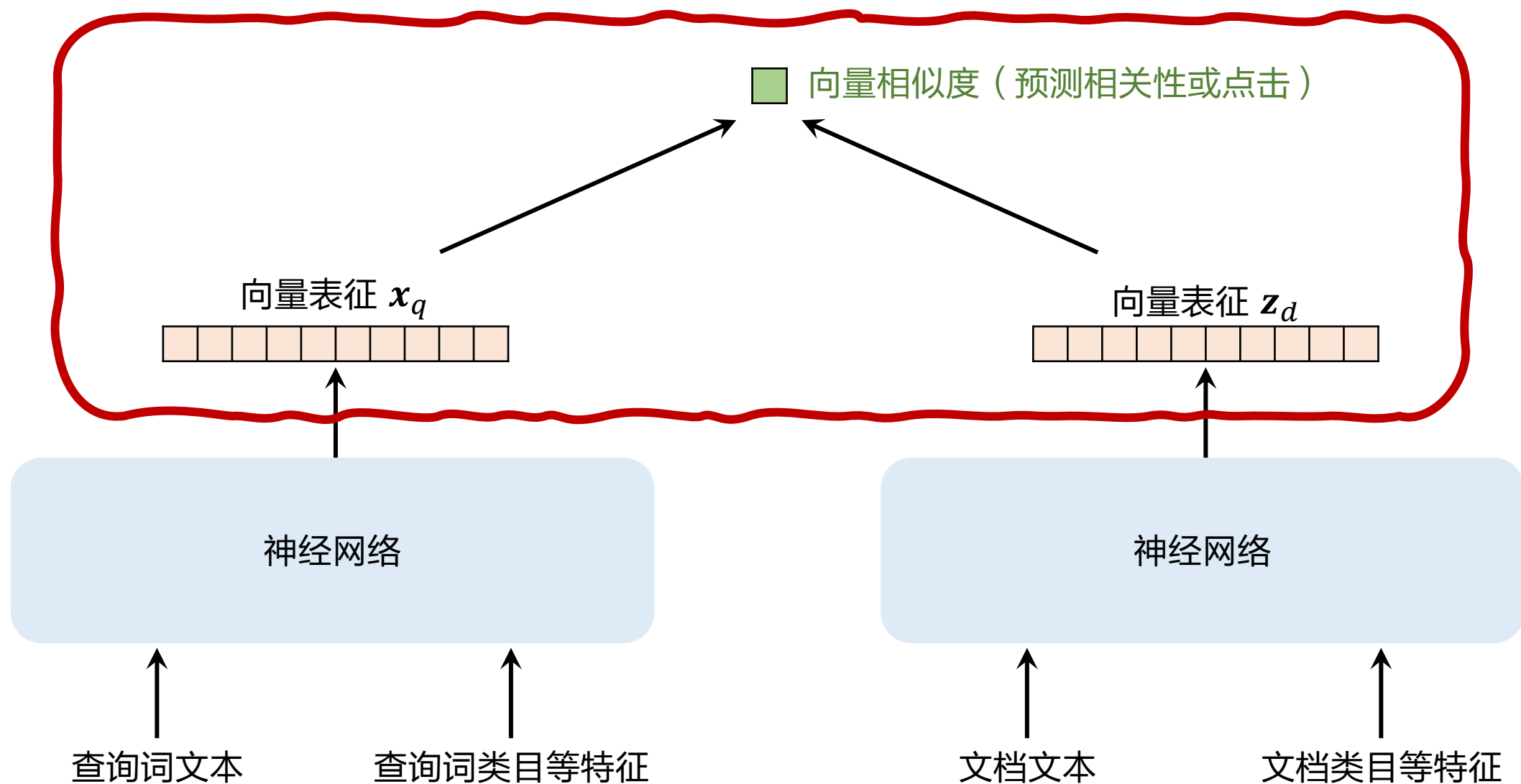




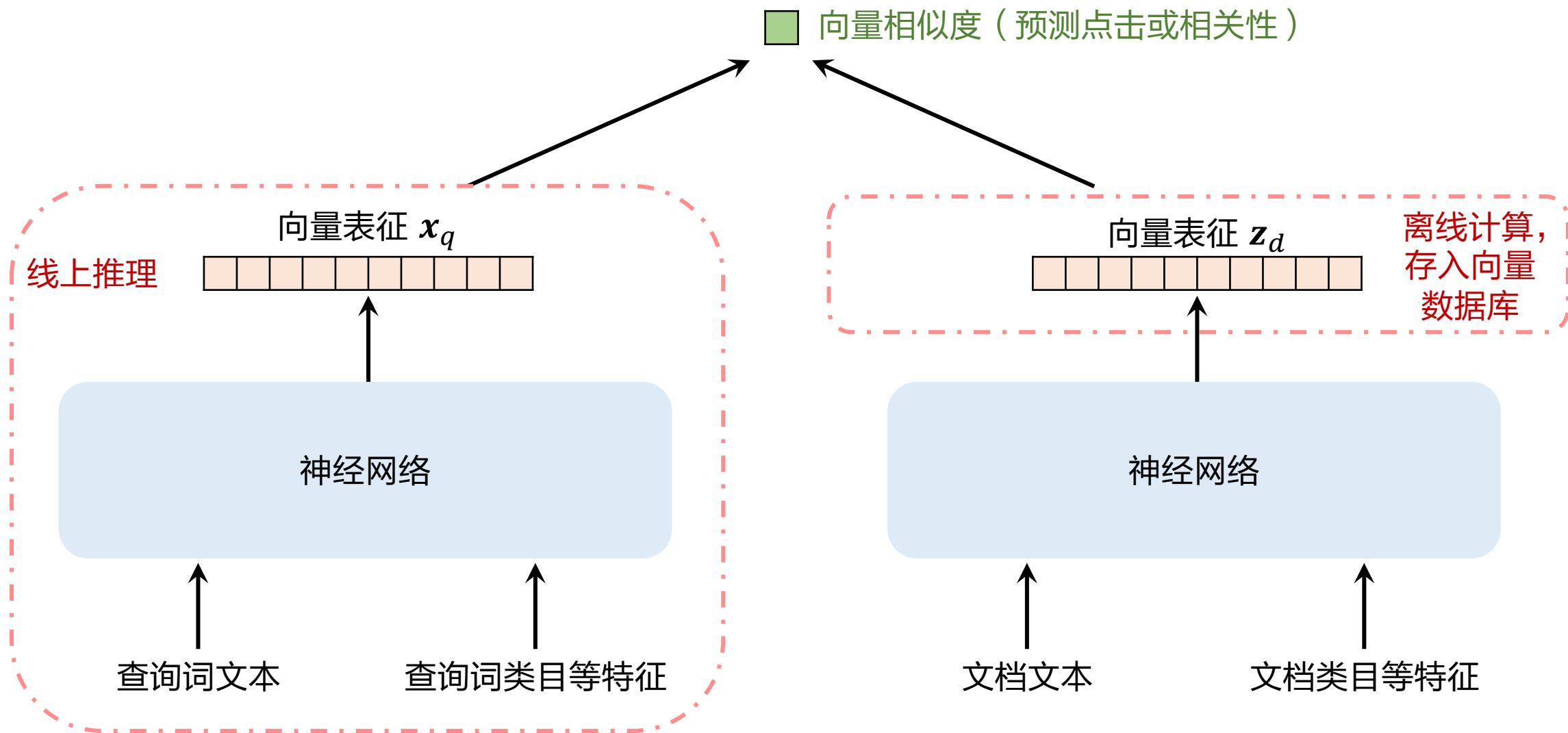
# 向量召回



# 向量召回



# 向量召回

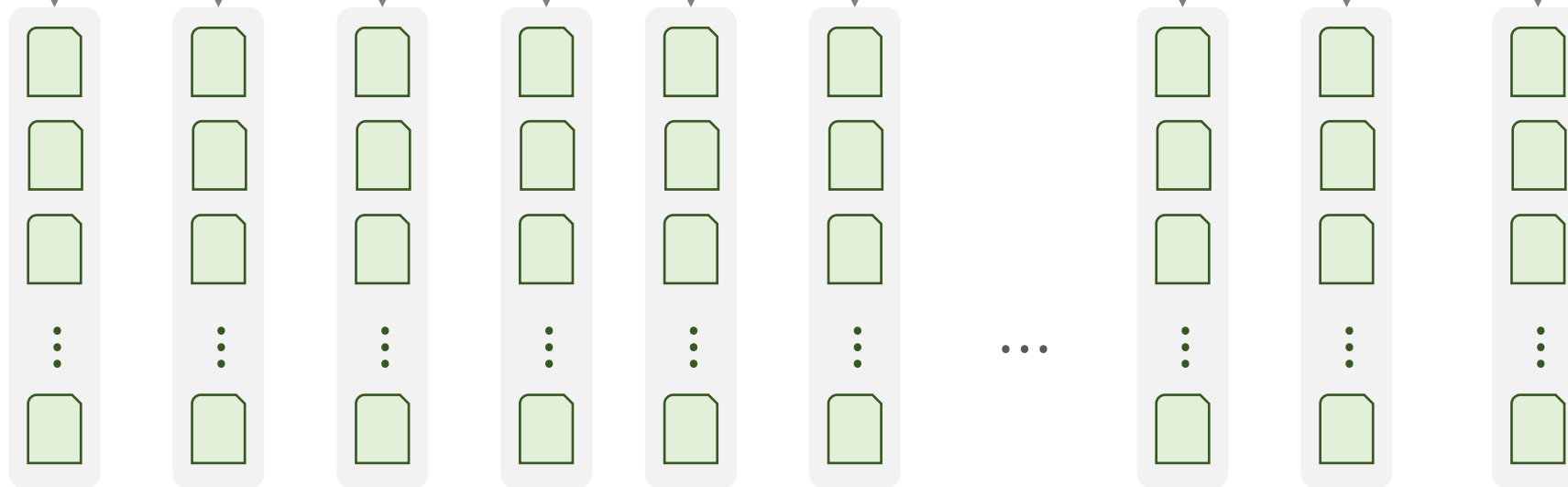


# KV召回

高频查询词：



相关的文档：



离线建立的索引

# KV召回

用户在线上输入的查询词：



命中

高频查询词：



...



相关的文档：



...



离线建立的索引

# 排序 (Ranking)

# 排序的依据

- ➡ • 相关性：重要性最高，在线上用 BERT 模型实时计算查询词和文档的相关性。
- ➡ • 内容质量：指文档的文本和图片质量、以及作者或网站的权威性。算法离线分析文档的内容质量，把多个分数写到文档画像中。
- ➡ • 时效性：主要指查询词对“新”的需求。查询词处理分析时效性，把结果传递给排序服务器。
- ➡ • 个性化：在抖音、小红书的重要性很高，在网页百度的重要性很低。在线上用多目标模型预估点击率和交互率（与推荐系统几乎相同）。

### 排序阶段实时计算

↑  
查询词和文档  
的相关性

相关性

↑  
点击率、交互率

个性化

### 文档发布时 离线计算

↑  
文本质量  
图片质量

文档内容质量

↑  
EAT

### 查询词处理阶段 实时计算

↑  
文档的  
发布时间

时效性

↑  
查询词的  
时效性需求



排序结果

重排（多样性、业务规则）

融合分数（用规则或LTR模型）

查询词和文档  
的相关性

相关性

点击率、交互率

个性化

文本质量  
图片质量

文档内容质量

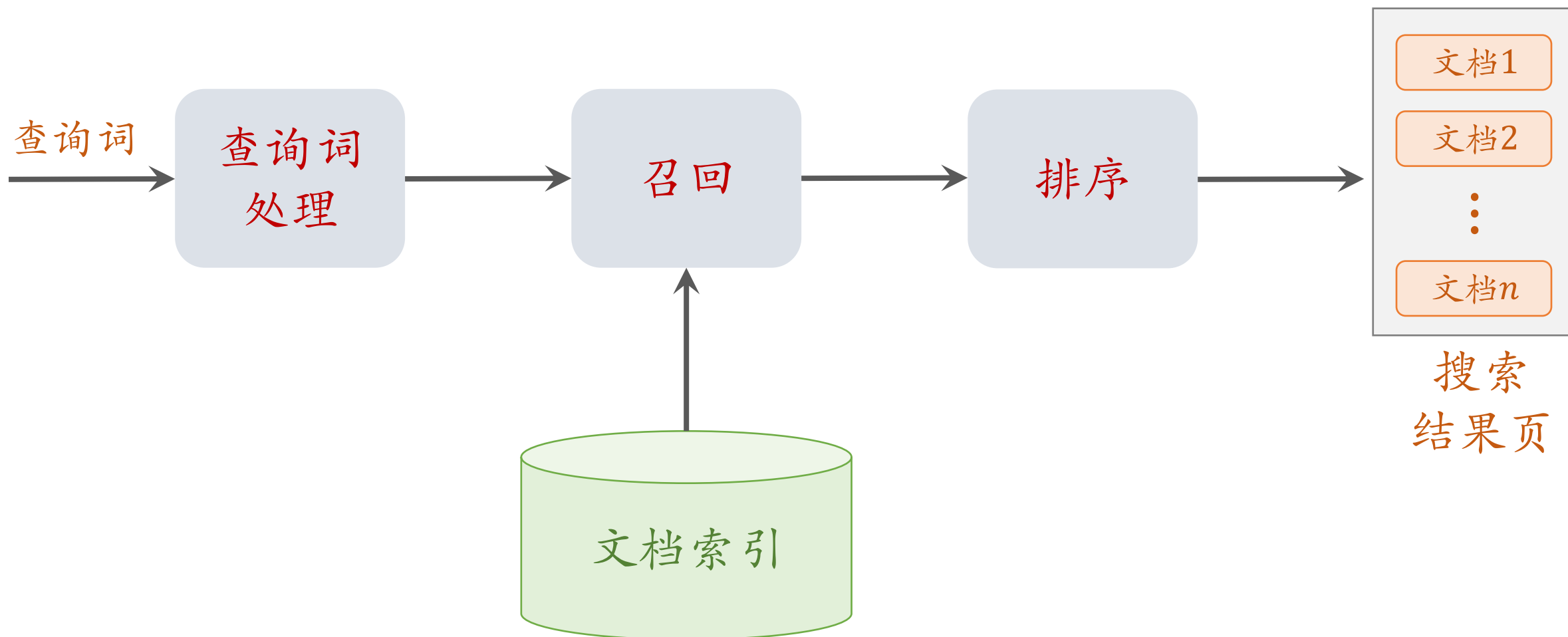
权威性

文档的  
发布时间

时效性

查询词的  
时效性需求

# 总结



**Thank You!**

<http://wangshusen.github.io/>