协同过滤召回

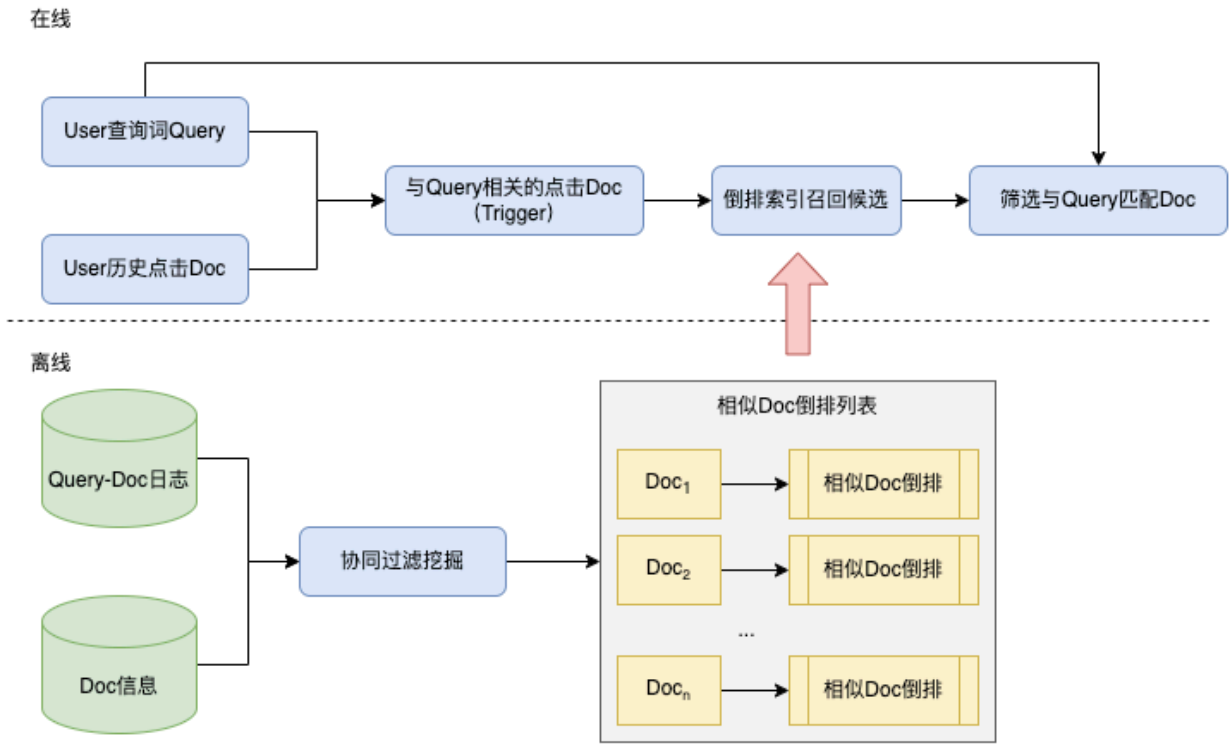
协同过滤（Collaborative Filtering, CF）是一种常用的推荐算法，主要基于用户的历史行为数据来预测用户对物品的偏好，从而实现个性化推荐。

在搜索领域，协同过滤依然可以基于分析用户的行为数据（如点击、浏览、收藏、购买等）和文档之间的关联，为用户提供相关的召回结果。在 Query改写 中已经介绍了如何用基于协同过滤的方法找到相似Query，在本章节中会介绍基于协同过滤的相似文档Doc召回。

I2I召回

I2I召回（item-to-item recall）基于文档相似性的召回策略，即通过分析文档之间的关联性（如共同出现、相似特征等），从一个或多个初始文档出发，召回与其相关的文档集合。

一种常见的I2I召回策略是基于推荐算法中的U2I2I召回，即基于用户的点击历史生成一个与当前用户搜索相关的 item list（doc list），将这些 item 作为 trigger，并利用 I2I (item to item) 相似度矩阵进行召回。具体如图：

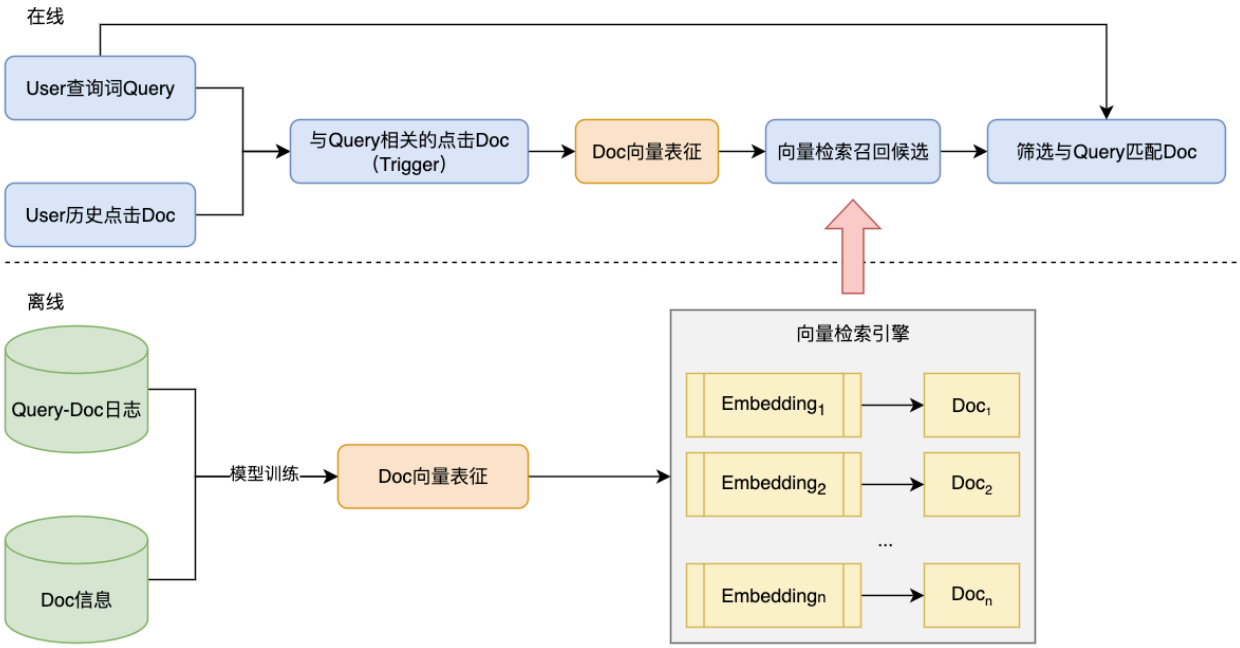


- 线上流程
  1. 收集当前用户历史点击文档集合
  2. 从点击文档中筛选出和当前查询词Query相关的文档作为 `trigger`
  3. 对 `trigger` 文档通过倒排索引/KV词表召回对应相似文档作为召回候选
  4. 从候选文档中筛选与查询词Query匹配的文档完成召回
- 离线流程

1. 基于点击等行为构建Query-Doc点击二部图（这块也可以采用User-Doc，Session-Doc等构建图）
2. 基于二部图通过协同过滤算法计算Doc-Doc之间的相似度
3. 对于检索池中的每篇Doc，按照相似度从高到低排序，取TopN构建倒排索引

基于向量表征的I2I召回

在I2I召回架构中，除了采用协同过滤的方法得到相似文档，基于向量表征的相似文档召回同样可以适配于I2I召回架构中。



- 线上流程
  1. 收集当前用户历史点击文档集合
  2. 从点击文档中筛选出和当前查询词Query相关的文档作为 `trigger`
  3. 将 `trigger` 文本文档通过向量表征模型转化为向量表示
  4. 对 `trigger` 向量通过向量检索召回对应相似文档作为召回候选
  5. 从候选文档中筛选与查询词Query匹配的文档完成召回
- 离线流程
  1. 基于点击等行为训练文档向量表征模型
  2. 对检索池中的每篇文档生成向量并作为向量检索引擎的索引

需要注意的是，对多个 `trigger` 向量同时做向量检索可能会带来召回引擎的性能压力。在资源算力不支持的条件下，可以将多个向量Pooling成单一向量以完成向量检索。同样的，索引侧也可做对应适配，即相似文档的Embedding聚合成单一向量，并作为索引映射多篇相似文档。

协同过滤算法

关于ItemCf、Swing、SimRank++在Query改写中的应用已在Query分析的Query改写中介绍，可以结合两种不同方向的应用来理解ItemCF和Swing算法。

ItemCF

基于 **Query-Doc 点击二部图** 的 **ItemCF 算法** 将搜索系统中的查询（Query）与物品（Doc）视为二部图的两类节点，通过用户的点击行为建立连接，并利用 **ItemCF（基于物品的协同过滤）** 方法挖掘文档之间的相似性。

通过ItemCF的相似度度量（如余弦相似度或共现频率）来计算每篇Doc的相似Doc，具体的，如果两篇Doc与多个相同的检索Query关联，说明它们具有相似性，如下是采用余弦相似度的ItemCF计算公式:

$$\text{sim}(d_i, d_j) = \frac{\sum_{q \in Q_{ij}} f(q, d_i) \cdot f(q, d_j)}{\sqrt{\sum_{q \in Q_i} f(q, d_i)^2} \cdot \sqrt{\sum_{q \in Q_j} f(q, d_j)^2}} \tag{1}$$

其中，

- Q_{ij} 是 Doc d_i 和 Doc d_j 共同点击关联的Doc集合
- Q_i 和 Q_j 分别是 Doc d_i 和 Doc d_j 各自点击关联的Query集合
- $f(q, d)$ 是 Query q 和 Doc d 的点击频率，通常用威尔逊平滑对点击次数和点击率进行平滑处理作为最后的点击频率（同样适用于Swing）：

$$\text{WLB} = \frac{\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p}) + \frac{z^2}{4n}}{n}}}{1 + \frac{z^2}{n}} \tag{2}$$

其中，

- $\hat{p} = \frac{\text{clicks}}{\text{impressions}}$表示点击率（CTR）
- $n = \text{impressions}$ 表示曝光次数
- z是对应置信水平的 z 值，例如 95% 的置信水平对应的$z \approx 1.96$

基于ItemCF可以建立Doc-Doc的索引，根据相似度打分对每个Doc选取TopN相似Doc作为召回候选。

Swing

Swing算法以高维的网络结构向二跳节点扩展，具有强抗噪能力，与ItemCF不同，Swing可以更加灵活地处理数据稀疏的情况。基于Swing算法的相似Doc挖掘建立在：如果两个Doc间关联的共点Query越多，且这些Query之间的重合度越低，那么这两个Doc间的相似度越高。如，Query u 和 Query v 点击了Doc i，则三者构成 `Swing结构`，若 Query u 和 Query v 不仅点击了 Doc i，Doc j 也被 Query u 和 Query v 点击，那么认为 Doc i 和 Doc j 在某种程度上是相似的，计算公式如下：

$$s(d_i, d_j) = \sum_{u \in Q_i \cap Q_j} \sum_{v \in Q_i \cap Q_j} \frac{1}{\alpha + |I_u \cap I_v|} \tag{3}$$

其中，

- Q_i 表示点击 Doc d_i 的 Query 集合，Q_j 表示点击 Doc d_j 的 Query 集合
- I_u 表示 Query u 点击的 Doc 集合，I_v 表示 Query v 点击的 Doc 集合
- $||I_u \cap I_v||$ 表示 Query u 和 Query v 的重合度，重合度高则要降低它们的权重，以避免意图不明确的Query带来噪声

上述Swing公式本质上在计算被 Doc d_i 和 Doc d_j 被点击关联的所有 Query Pair 的Swing结构（Query-Doc-Query）的权重之和。基于Swing可以建立Doc-Doc的索引，根据相似度打分对每个Doc选取TopN相似Doc作为召回候选。

总结

协同过滤算法在搜索召回中的应用，能够有效利用用户行为数据建模文档之间的关系。其中最基础的应用是采用I2I的召回框架，而在引入向量表征之后，可以升级为向量检索召回相似文档。即文档之间的相似性不再依赖共现数据，而是通过向量间的距离计算，实现了从简单线性关系到复杂非线性关系的提升。在实际业务中，可以根据业务需求、资源条件和模型能力选择适合的算法策略。

参考文献

1. Amazon.com Recommendations Item-to-Item Collaborative Filtering
2. Large Scale Product Graph Construction for Recommendation in E-commerce