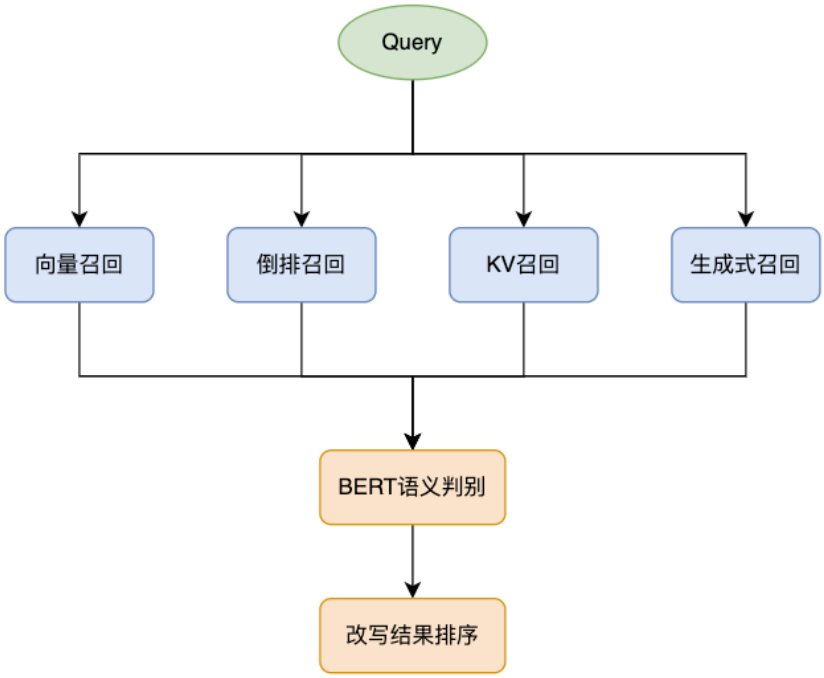
[搜索系统算法解析-OAA]

Query改写

不同于词替换的Term改写，Query改写是指在Query粒度下，将用户的原始Query改写为新的Query，即Query2Query。Query改写不依赖分词，改写结果语义连贯性更好（完整Query），语义泛化能力更强，且有更深的技术空间。

业界常用的Query技术框架为`召回-排序`范式，召回方式主要有向量召回、倒排检索召回、KV词典召回、生成式模型召回，排序则从改写结果的语义相关性、Doc召回增量等角度评估和筛选改写结果。



改写类型

Query改写针对不同场景需求有多种形态：同义、泛化、细化、激发

- 同义
  - 改写Query和原Query语义相同
  - 如：西蓝花功效 --> 吃西兰花的好处
- 泛化
  - 将具体的细节描述变为更宽泛、更具概括性的表述
  - 如：NBA球赛 --> 篮球比赛
- 细化
  - 与泛化相反，将大的主题概念变得更详细具体，常见的细化方向有：需求澄清、个性化改写
  - 如：人工智能课程 --> 机器学习导论
- 激发

- 原主体保持不变，但主要需求发生迁移，目的是激发搜索
- 如：美国大选结果 --> 美国大选摇摆州

改写召回

改写召回的技术方案大体和通用搜索召回一致，本质上都是对文档召回，主要有向量召回、倒排召回、KV词典召回、生成式召回四种召回通路。

其中:

- 向量召回覆盖率高（大部分Query都能有召回结果），以相似度距离海选打分会优先选择TopN最相近的结果
- 倒排召回在结果命中关键词后，一般以结果质量分截断候选，即TopN召回结果以高质量为主要目标导向
- KV词典以Query间的共点信息挖掘，这种挖掘方式导致召回结果天然的主要覆盖头部腰部Query（冷启动问题），由于其充分利用了搜索后验信息，其召回结果更符合系统生态
- 生成式改写相比前三种基于检索的方式，优势在于改写结果更灵活多样，Query召回覆盖面广，即能应对各种长冷Query生成改写结果，缺点是质量不稳定、计算成本高。

向量召回

向量召回基于一个向量索引库，通过将Query和文档（改写Query）映射到高维空间中的向量，利用向量之间的相似度来进行检索。

索引库

召回匹配库以历史上出现的高质量Query进行向量索引建库，高质量Query筛选的判断依据依赖多个特征，包括Query的历史检索量、点击率、满意度、文本表述（是否错词/语义清晰通常）等为特征，并根据业务需要进行安全过滤（黄赌毒反）。

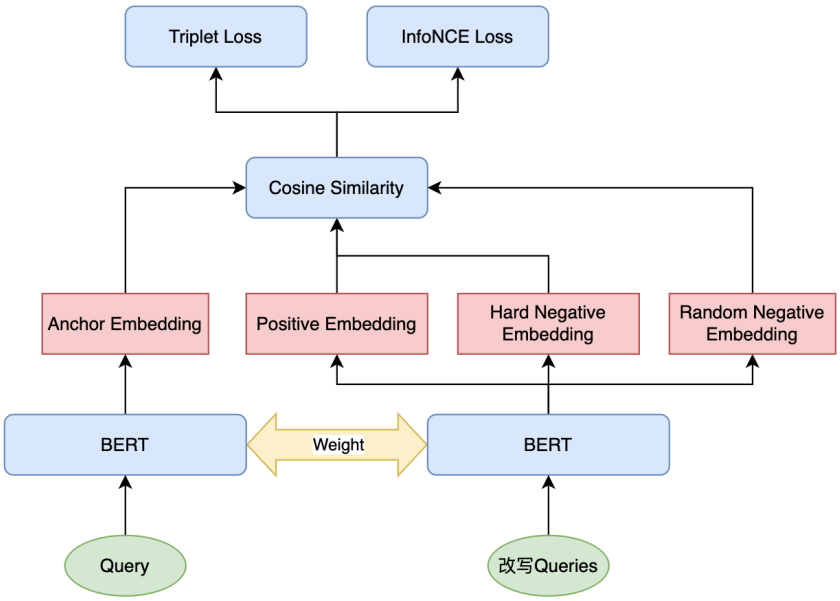
向量模型训练集

向量模型需要海量的训练数据集，一种常见的方式是基于共点Doc挖掘Query-Query Pair，并通过样本增强 + 相关性模型打分的方式生成海量训练集，具体流程如下：

1. 根据用户搜索点击日志构建Query-Doc二部图，即Query节点和Doc节点存在点击行为则用边连接
2. 基于二部图挖掘共点Doc的Query-Doc-Query，生成$q \rightarrow q'$数据集
3. 对于q'取出其在二部图中点击关联的Doc，$q' \rightarrow [d_1, d_2, \cdots, d_k]$
4. 利用搜索排序中的Query-Doc相关性模型对q和q'关联Doc打分，分数作为$q \rightarrow q'$改写相关程度分数，即：$score = \frac{1}{n}\,\mathrm{rel}(q, d_i)$，其中$\mathrm{rel}$为Query-Doc相关性打分模型，$n$为$q'$关联Doc数量
5. 基于分数对$q \rightarrow q'$数据集拆分为正样本和负样本

向量模型结构

业界目前获取Query向量的传统、有效且通用的方式是基于类似Sentence-BERT的双塔参数共享的模型架构：



训练范式如下：

- 模型输入有三个文本：Anchor（查询Query）、Positive（改写正样本）、Negative（改写负样本），三个文本通过BERT获得向量表征，并通过Triplet目标函数将锚点样本与正样本的向量距离尽可能缩小，同时将锚点样本与负样本的距离尽可能拉大，最小化损失函数为：
  - $L_{\text{triplet}} = \max(d(\text{anchor}, \text{positive}) - d(\text{anchor}, \text{negative}) + \alpha, 0)$
  - 其中，d是距离度量函数（一般采用1-COSINE），α是超参数，表示距离相差边际
  - $\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$，$A$和$B$是两个向量，$\|A\|$和$\|B\|$是它们的L2范数，余弦相似度的值范围为$[-1, 1]$，余弦相似度越大，表示两个向量越相似（余弦相似度衡量的是两个向量之间的夹角，反映的是它们的方向相似度）
- 对于Triplet Loss来说，训练样本来自于前文挖掘的难正负例，此外还可以用batch中的其他样本作为简单负样本来优化模型，即每个batch中，除了目标样本的正样本对外，其余样本都被视为负样本。损失函数采用InfoNCE Loss，这种方式下InfoNCE Loss关注远距离样本，Triplet Loss关注近距离样本，使模型更加有效地处理两种不同的相似性场景，优化嵌入空间中的局部和全局结构，联合损失函数为：
  - $L = \lambda_1 L_{\text{InfoNCE}} + \lambda_2 L_{\text{triplet}}$，其中，$\lambda_1$和$\lambda_2$是权重系数，$L_{\text{InfoNCE}}$定义如下
  - $L_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(\text{anchor}, \text{positive}) / \tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\text{anchor}, \text{negative}) / \tau)}$，其中，sim是相似度度量函数（采用COSINE），τ是温度参数，用于控制模型对不同相似度分数的敏感度
- 此外，在难负样本上可以通过样本增强的方式补充字面相似的负样本（如：微积分 --> 游戏积分，麻辣蛤蜊 --> 麻辣哈利）
- 在模型训练的迭代学习的过程中，对模型而言难负样本会逐渐变得简单，这种情况可以逐渐增加难负样本的比例
- Pooling方案可以根据实验选择Avg或CLS向量

训练结束后，离线对高质量Query生成向量表征并可用Faiss等工具构建索引库，线上改写时可以对Query进行近邻向量检索，召回TopN个相似Query。

倒排检索召回

如向量召回一样，倒排召回同样需要挖掘高质量Query建立倒排索引库。召回方式可以采用检索词Query的核心词、同义词、标签等构建布尔查询串。倒排检索召回TopN的筛选可以参考改写Query质量分，质量分可以参考历史点击率、历史搜索量、搜索结果满意度（检索结果相关性、多样性等，用户行为中的跳出率、停留时长、后续查询行为等）。

KV词典召回

KV词典召回方式是以检索词Query作为Key，基于KV形式的Query-Query词表召回改写Query（Value）。词表大部分数据基于图方法挖掘，常见的方法是采用协同过滤的方式，如：ItemCF、Swing、SimRank++，这种方法适用于用户行为比较丰富的Query。KV词典召回覆盖头部查询词，目标是将历史点击率低、检索结果相关性差的Query改成高质量的Query（高点击率、高满意度）。

ItemCF

通过ItemCF的相似度度量（如余弦相似度或共现频率）来计算每个Query的相似Query，具体的，如果两个Query与多个相同的Doc关联，说明它们具有相似的用户意图，如下是采用余弦相似度的ItemCF计算公式:

$$\text{sim}(q_i, q_j) = \frac{\sum_{d \in D_{ij}} f(q_i, d) \cdot f(q_j, d)}{\sqrt{\sum_{d \in D_i} f(q_i, d)^2} \cdot \sqrt{\sum_{d \in D_j} f(q_j, d)^2}} \tag{1}$$

其中，

- D_{ij} 是 Query q_i 和 Query q_j 共同点击关联的Doc集合
- D_i 和 D_j 分别是Query q_i 和 Query q_j 各自点击关联的Doc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和SimRank++）：

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

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Query-Query的索引，根据相似度打分对每个Query选取TopN相似Query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Query挖掘建立在：如果两个Query共点的Doc越多，且这些Doc之间的重合度越低，那么这两个Query间的相似度越高。如，Query i 点击了 Doc u 和 Doc v，则三者构成`Swing结构`，若除了 Query i 点击了 Doc u 和 Doc v，Query j 也点击了 Doc u 和 Doc v，那么认为 Query i 和 Query j 在某种程度上是相似的，计算公式如下：

$$s(q_i, q_j) = \sum_{u \in D_i \cap D_j} \sum_{v \in D_i \cap D_j} \frac{1}{\alpha + |I_u \cap I_v|} \tag{3}$$

其中，

- D_i 表示 Query q_i 点击的 Doc 集合，D_j 表示 Query q_j 点击的 Doc 集合
- I_u 表示 Doc u 关联的 Query 集合，I_v 表示 Doc v 关联的 Query 集合
- $\|I_u \cap I_v\|$ 表示 Doc u 和 Doc v 的重合度，重合度高则要降低它们的权重，以避免内容主题不明确、大杂烩式的Doc带来噪声

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

SimRank++

SimRank++的核心思想是：两个节点（Query）的相似度由其邻居节点（Doc）的相似性决定。即，关联到相似Doc的Query是相似的，关联到相似Query的Doc是相似的。

$$s(q_i, q_j) = \begin{cases} 1, & q_i = q_j \\ \frac{c}{|I(q_i)||I(q_j)|} \sum_i^{|I(q_i)|} \sum_j^{|I(q_j)|} s\left(I_i(q_i), I_j(q_j)\right), & q_i \neq q_j \wedge I(q_i), I(q_j) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

其中，

- $I(q_i)$和$I(q_j)$表示Query-Doc二部图中分别与节点q_i和q_j相关联的节点Doc集合
- $s\left(I_i(q_i), I_j(q_j)\right)$表示Doc节点之间的相似度，$c$是常量衰减因子

相比Swing算法，SimRank++覆盖率更高，两个Query之间没有共点Doc依然可以计算相似度。

生成式召回

生成式改写不受统计特征约束，可以做到全覆盖式改写，即在长尾Query上有显著的优势，且在问答式Query的非对称改写上，生成式改写是不二的选择。此外，生成式改写灵活性高且泛化能力强，具备目标导向的可定制化，能利用LLM本身自带的先验知识结合查询场景，并根据需求自适应生成最优改写。但是由于生成式改写基于LLM，对算力要求较高且推理时间长，出于对成本的控制和耗时的妥协，生成式改写一般只对圈定Query触发且通常根据模型参数量走Nearline链路（介于Online和Offline之间）和Offline链路。

生成式触发模型

降本增效的产物，目的是将LLM算力资源分配到对改写整体ROI收益最大的场景下。生成式改写主要以表述冷门或不准确的Query作为目标：

- Query语义不清晰
- Query语法不准确
- Query用词不标准
- Query需求不符合认知

挖掘到目标Query后，即可采用BERT模型进行训练以产出触发模型。

LLM

LLM是当前最适宜做内容生成的模型，主流的LLM采用Transformer架构，即通过自注意力机制实现高效的序列建模和长距离依赖关系的处理。LLM训练流程分为如下几个关键阶段：

- 预训练：模型在大规模无监督文本数据上进行自回归的预训练任务，主要学习语言的基本语法、语义和常识，数据来源为海量的文本数据（书籍、新闻、网络内容）
- 指令微调：在预训练模型基础上，使用标注的指令数据（即用户问题和相应答案）进行微调，使得模型更适合对话和具体任务。目标是使有监督训练后的SFT模型学会遵循指令的格式和结构，提高模型对人类意图的理解，确保模型输出更加自然、清晰，并避免生成不相关或错误信息
- 奖励建模和强化学习（RLHF）：使用人类反馈优化模型的回答生成质量，提升其输出符合人类偏好的能力
    - 奖励模型：模型会根据输入得到的不同回答进行打分，目标是识别符合用户需求的高质量回答并生成“奖励信号”，从而帮助模型区分高质量和低质量的输出，奖励建模使模型更了解回答的优劣标准，为后续的强化学习提供指导依据
    - 强化学习：基于奖励模型，让模型在给定提示词下生成更高质量的内容，目标是优化模型参数提高奖励值，主要步骤如下：
        - 回答生成与评分：在每次训练迭代中，生成模型会针对某个输入生成回答，并通过奖励模型对回答进行打分。评分的高低决定了模型生成输出的质量
        - 策略优化：为了让生成模型获得更高的奖励分数，通常使用策略梯度算法（如PPO）来调整模型的生成策略，使其输出的回答能够逐渐符合奖励模型的偏好
        - 策略梯度更新：通过调整模型生成过程中的概率分布，优化生成质量。比如，PPO限制了参数更新的幅度，从而防止模型在优化过程中发生过度调整导致生成质量下降
        - 反馈循环：每次优化后，生成模型会基于新的策略生成回答，通过反馈不断优化回答的质量，使其逐步达到最优
        - 平衡探索和生成质量：在强化学习过程中，模型既要学习生成高奖励的答案，又需要避免生成内容过于模式化。因此，训练过程中的探索-利用平衡非常重要。为了增加多样性，模型有时会生成不同于之前答案的回答，以避免单一答案的固化和偏差

LLM训练数据构建

微调LLM模型以适应当前搜索环境需要大规模领域训练数据集，常规的方式如下：

- 基于点击二部图
    - 利用KV词典召回中基于协同过滤方式挖掘的海量数据，筛选出 冷门Query --> 热门Query 的优质改写 Pair
    - 挖掘优质的 Doc Title --> Query 数据集
- 基于用户Session
    - 挖掘用户搜索行为序列中自发改写的 Query --> Query
        - 父Query无点击行为，子Query有点击行为
        - 父子Query需满足相关性
- 基于人工/LLM标注
    - 对历史冷门无点击的Query进行CoT标注（给出优质改写结果和改写理由）
- 基于RAG

    - 给定冷门Query，利用搜索链路检索TopN篇Doc，利用LLM将检索到的Doc内容与原始查询Query进行结合，生成改写Query
- Prompt设计
    - 根据挖掘方式的不同，对改写Pair设计对应场景的Prompt（给出改写背景和改写词之间的关系）

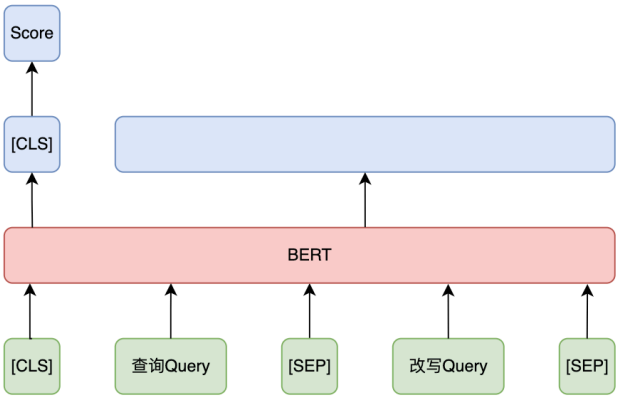
目标对齐

对SFT后的LLM的输出多个改写Query，并通过现有的搜索系统完成检索，最后基于检索结果对改写Query打分（召回Doc增量/相关性/内容质量等）作为RLHF训练集。

改写语义相关性判别

为了减少改写召回候选出现语义漂移等不合理情况，对于通过检索召回、KV召回和生成式改写得到的改写Query需要引入BERT语义判别模型，将改写Query与原始Query进行语义相似度计算，对改写Query和原Query的语义相关性进行分档，并对不相关的改写Query过滤。

改写相关性判别模型通常采用BERT，输入为查询Query和改写Query，输出为相似度分数，模型训练依赖标注数据集：



改写排序

在改写语义判别模块对改写召回结果完成打分和过滤后，需要对改写Query的质量进行评估和排序，以选出最优的改写Query。改写结果主要以改写结果相关性和召回Doc增量为目标，由于这两个目标直接与检索最终结果挂钩，所以相关性和增量是衡量改写结果在搜索应用有效性的最直接的指标：

- 相关性（Relevance）：衡量改写Query检索结果与查询词原始意图的一致性
- 增量（Increment）：衡量改写Query的语义扩展程度

改写排序模型可以采用基于Pairwise或Listwise的训练方法，训练数据集的构建可以采用以下方式：

1. 给定查询Query，召回改写Query候选
2. 对查询Query和改写Query分别通过离线搜索系统检索获取各召回Doc
3. 对改写Query召回Doc与查询Query进行Query-Doc相关性打分，作为改写Query的相关性指标（Relevance）
4. 计算改写Query召回Doc满足Query-Doc相关性大于阈值的增量Doc量作为改写Query的增量指标（Increment）
5. 综合相关性和增量指标，对改写Query排序，排序Index可作为训练标签

改写排序模型特征主要来源于后验统计特征，以下是常用的特征：

特征
改写Query长度/类目/NER
改写Query历史曝光点击量
改写Query历史召回笔记量和Query-Doc相关性分数分布
查询Query-改写Query的Query-Query相关性打分
查询Query-改写Query历史召回Doc的Jaccard相似度系数
历史用户换Query行为中，查询Query->改写Query的数量

个性化改写

为了在改写时更好地贴合不同用户的搜索意图和偏好，可以通过个性化改写的方式实现用户体验的优化。实现个性化改写的核心在于结合用户的个性化特征和历史行为来调整改写策略。

- 用户画像构建：通过用户的历史查询、点击等消费记录，获得用户的兴趣领域和偏好
- 上下文信息：基于用户当前的查询上下文信息（如搜索的时段、地点、会话内的前后关联查询）实现动态改写

总结

综上，Query改写链路可以看做为一个小型搜索系统，所以搜索系统中的方法或多或少可以作为Query改写实现中的方案参考。Query改写的核心目标是通过增强查询的准确性、表达的丰富性和意图的清晰度，让系统在更大范围内捕捉和满足用户的搜索意图，从而提升整体的用户体验和检索效果。所以，在设计改写方案时，需要明确改写目标和应用场景。在评估和迭代优化中通过A/B测试和用户反馈收集以优化改写规则和模型，确保改写系统不断学习和改善。

参考文献

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
2. Sentence Embeddings using Siamese BERT-Networks
3. Momentum Contrast for Unsupervised Visual Representation Learning
4. Amazon.com Recommendations Item-to-Item Collaborative Filtering
5. Large Scale Product Graph Construction for Recommendation in E-commerce
6. SimRank: A Measure of Structural-Context Similarity
7. Training language models to follow instructions with human feedback