rag相关数学模型（你们看着用）

主要包括基于语义熵的不确定性度量方法、自反思机制的概率模型以及动态检索优化的算法策略。

传统的大语言模型不确定性计算往往基于词元层面的对数概率，但这无法区分词汇层面的不确定性与语义层面的不确定性。为此，我们引入 Kuhn 等人提出的语义熵概念。

语义等价类的定义与聚类算法

假设对于输入 x，模型生成一系列候选回答序列 $S = \{s_1, s_2, ..., s_M\}$。我们定义一个语义等价关系 $E(s_i, s_j)$，当且仅当 s_i 与 s_j 表达相同的语义真值时，E 为真。

1. 双向蕴含判定
   在算法实现中，我们利用自然语言推理模型来近似判断双向蕴含关系：

$$E(s_i, s_j) \iff (s_i \Rightarrow s_j) \land (s_j \Rightarrow s_i)$$

2. 贪婪语义聚类算法
   基于上述关系，我们将生成空间划分为 K 个语义等价类 $C = \{c_1, c_2, ..., c_K\}$。具体的聚类过程定义如下：

- 初始化：$C \leftarrow \emptyset$;
- 对于每一个采样序列 $s_i \in S$,遍历现有的每个类 $c_k \in C$ 的代表元素 r_k
- 判定：若 $E(s_i, r_k)$ 为真，则将 s_i 加入 c_k,若 s_i 与所有现有类的代表元素均不构成蕴含关系，则创建一个新类 $c_{new} = \{s_i\}$ 并加入 C。

最终得到的等价类集合满足：

$$c_k = \{s \in S \mid \forall s' \in c_k, E(s, s') \text{ is True}\}$$

基于香农熵的语义不确定性计算公式

每个语义类 c_k 的后验概率近似为该类中所有采样序列的概率之和：

$$p(c_k|x) \approx \sum_{s \in c_k} p(s|x)$$

其中 $p(s|x)$ 是序列 s 的联合生成概率，通常由模型输出的 Token 概率连乘得到。为了数值稳定性，实际计算中使用归一化后的概率。

系统的语义熵定义为语义类分布的香农熵：

$$SE(x) = -\sum_{k=1}^{K} p(c_k|x) \log p(c_k|x)$$

当 $SE(x)$ 超过预设阈值 τ 时，我们判定模型处于高幻觉风险状态，从而触发 RAG 检索机制。

自反思检索增强机制

参考 Asai 等人的 Self-RAG 框架，我们将生成过程建模为一个序列决策问题。系统不仅生成文本标记 y_t，还通过预测特殊的反思令牌来评估生成质量。

检索-生成-批评 的概率模型

我们将任务定义为在给定输入 x 和检索文档集合 D 的条件下，生成目标序列 y。该过程被建模为以下的条件概率分布：

$$P(y|x) = \mathbb{E}_{d \sim P(d|x)}[P(y|x, d)]$$

在引入自反思机制后，生成模型 M 被扩充为能够生成批评令牌集合 R。对于每个生成步 t，模型不仅输出单词 y_t，还并行输出批评令牌 r_t：

$$P(y_t, r_t|x, d, y_{<t}) = P(y_t|x, d, y_{<t}) \cdot P(r_t|x, d, y_{<t}, y_t)$$

反思令牌的数学定义与权重分配

反思令牌用于量化生成的质量，我们主要关注以下三类指标：

相关性 : $r_{rel} \in \{\text{Rel}, \text{Irrel}\}$，衡量文档 d 是否支持问题 x。

支持度 : $r_{sup} \in \{\text{Fully}, \text{Partial}, \text{No}\}$，衡量生成内容 y_t 是否由文档 d 充分支撑。

有用性 : $r_{use} \in \{1, 2, 3, 4, 5\}$，衡量回答是否解决了用户需求。

综合评分函数：
为了在推理阶段选择最优的生成片段，我们定义如下线性加权评分函数 $Score(y_t)$：

$$Score(y_t) = P_\theta(y_t|x, d) + \sum_{k \in \{rel, sup, use\}} w_k \cdot P_\theta(r_k = \text{Positive}|x, d, y_t)$$

其中：

$P_\theta(y_t|x,d)$ 是基础生成概率。

$P_\theta(r_k = \text{Positive})$ 是 Critic 模型预测该片段为高质量的概率。

w_k 是超参数权重，用于调节不同反思维度的重要性（例如在医疗场景中，我们赋予 w_{sup} 更高的权重以确保事实性）。

动态检索策略优化

为了解决单一检索方式在不同场景下的局限性，我提出了一种结合稀疏检索与稠密检索的混合策略，并引入重排序模型以提升检索精度。

混合检索的加权公式

传统的关键词检索，如BM25在精确匹配上表现优异，而向量检索擅长捕捉语义相关性。我们采用倒数排名融合思想的加权变体，定义混合检索得分 S_{hybrid}：

$$S_{hybrid}(q,d) = \alpha \cdot \mathcal{N}(S_{dense}(q,d)) + (1-\alpha) \cdot \mathcal{N}(S_{sparse}(q,d))$$

其中：

$S_{dense}(q,d) = \cos(\mathbf{v}_q, \mathbf{v}_d) = \frac{\mathbf{v}_q \cdot \mathbf{v}_d}{\|\mathbf{v}_q\|\|\mathbf{v}_d\|}$ 为基于 embedding 的余弦相似度。

$S_{sparse}(q,d)$ 为 BM25 算法计算的关键词匹配得分。

$\mathcal{N}(\cdot)$ 为归一化函数，将两种得分映射到 $[0,1]$ 区间以消除量纲差异。

α 为平衡因子，根据查询的实体密度动态调整。

基于 Cross-Encoder 的重排序模型

混合检索通常作为初筛阶段，召回 Top-K个文档。为了进一步提升精度，我们引入基于 Cross-Encoder 的重排序模型。

与双塔模型独立编码不同，Cross-Encoder 将查询 q 和文档 d 拼接后输入 BERT 模型，通过全注意力机制捕获细粒度的语义交互：

$$S_{rerank}(q,d) = \sigma\left(\mathbf{W} \cdot \text{BERT}_{\text{CLS}}([CLS]; q; [SEP]; d)\right)$$

其中：

$[CLS]$ 和 $[SEP]$ 为特殊分隔符。

$\mathrm{BERT}_{\mathrm{CLS}}$ 表示取最后一层 CLS 位置的输出向量。

$\mathbf{W}$ 为线性投影层权重。

$\sigma(\cdot)$ 为 Sigmoid 激活函数，输出文档 d 与查询 q 相关的概率值。

最终，系统根据 S_{rerank} 对 Top-K 文档进行降序排列，并截取前 N 个（例如 $N = 3$）作为大模型的上下文输入。