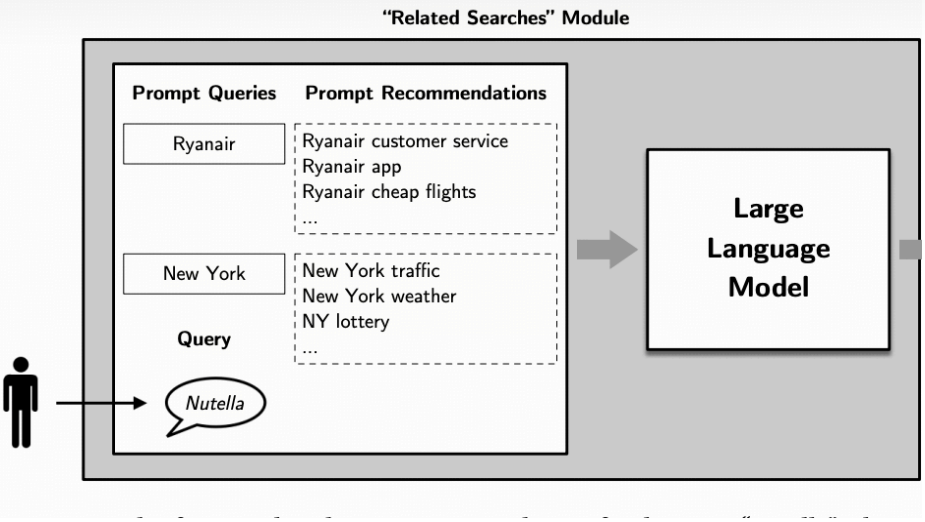
知乎



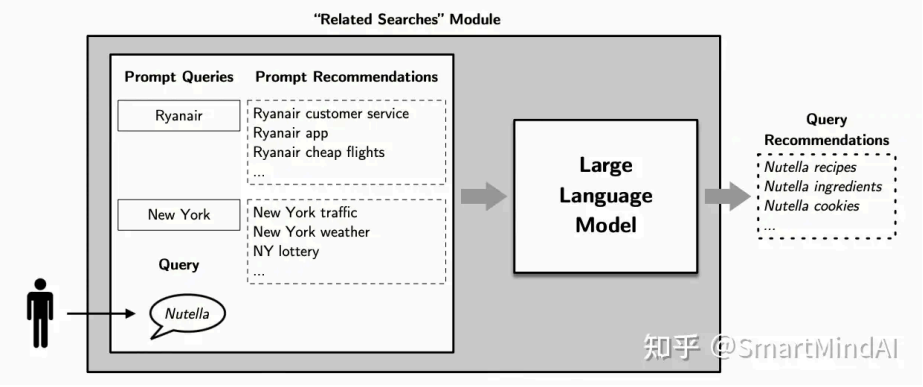
2024 Spotify：RA-GQR技术借助LLM，将相关推荐变为为创意生成模式，提升个性化音乐体验

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

Introduction

相关搜索+是搜索引擎结果页面（SERP）中的基本组件。通过向用户提交的相关推荐相关查询，相关搜索模块引导用户更准确地定位信息需求，这些被称作查询建议。通常，相关搜索功能由相关推荐系统实现。

假设用户想找Corporate Ltd.的信息，输入了Corporate。搜索引擎首个结果是Corporate Ltd.的官网，但不是用户想要的。相关搜索会给出如Corporate products和Corporate job openings等更精确的建议，用户只需点击即可看到所需信息。然而，建立或优化相关推荐系统具有挑战性，需要考虑三个关键因素：查询日志中的用户历史交互数据，基于这些数据训练的机器学习模型，以及支持实时推荐的后端服务基础设施。我们提出了一种新方法，利用大型语言模型+（如OpenAI的GPT-3+）的强大表达和生成能力，构建有效的相关搜索模块，无需依赖特定系统的机器学习模型进行训练。



本研究借鉴零样本学习中的提示驱动方法，创新性地利用OpenAI的GPT-3等大型语言模型，通过提供查询示例及其推荐，让模型生成针对新查询的建议。我们构建提示，即使目标查询与提示示例不完全相同，也能引导模型理解语义并提供相关推荐。

我们研发了Retrieval Augmented Generative Query Recommendation (RA-GQR)，这是GQR的升级版，它巧妙地利用查询日志自动生成提示，以提升推荐质量。RA-GQR首先从日志中找到类似查询，然后动态调整提示，使其更能适应当前用户的需求。实验在Robust04和ClueWeb09B等基准上证实，无论使用GPT-3作为基础，GQR均展现出显著优于竞品的性能，提升NDCG\@10指标约4%。

知乎

我们提出一种名为Generative Query Recommendation (GQR)的新方法，将相关推荐视为生成任务，利用预训练的序列到序列语言模型（LLM）进行。GQR通过构建提示，包含输入查询文本q和候选推荐列表$\mathcal{L}$，利用LLM理解和生成推荐。具体操作是，利用预训练LLM在提供示例集$\mathcal{C}$下的预测，选取前k个预测可能性最高的推荐，即

$$\hat{\ell} = \max_{\ell \in \mathcal{L}} P(\ell|q)$$

由于推荐列表可能巨大，我们不直接存储所有选项，而是利用LLM的生成能力，从候选集中生成最相关的k个推荐。这种方法避免了存储和计算整个推荐空间的挑战，从而优化了相关推荐的生成过程。

$$\hat{\ell} = k\text{-}\arg\max_{r_j \in \mathcal{L}} P_{\mathcal{M}}(r_j|q, \mathcal{C})$$

在实际应用中，我们使用预设的模板或提示来增强输入，通过$T(\cdot,\cdot)$函数进行格式化。这个函数将输入查询(q)和推荐列表(ℓ)结合，形成新的示例对(q', ℓ')。(q')是格式化后的查询，旨在帮助LLM理解推荐目标。这样，我们就通过LLM的上下文学习能力生成最相关的k个推荐建议。

$$T(q, \ell) = \textbf{query } q \textbf{ recommendations } \ell$$

其中 $\mathcal{M}(\mathcal{C}, q)$ 代表了一个模型，它利用了与相关推荐相关的广泛背景信息$\mathcal{C}$，而不仅仅是查询变量q。$\mathcal{C}$提供了必要的上下文，使模型能够理解生成的推荐应围绕的主题和内容。因此，通过这种方式，LLM能够在理解了整个环境后，不依赖于特定查询设计，直接生成适应的推荐结果。这种方法显著提高了推荐的准确性和效率。

$$T(q'_1, \ell'_1)\ T(q'_2, \ell'_2)\ \cdots\ T(q'_n, \ell'_n)\ T(q, \_)$$

在GQR方法中，我们通过向LLM提供包含查询上下文信息的$\mathcal{C}$，以及查询自身，来生成推荐。这个过程可以用公式表示为：

$$\text{GQR}(q, \mathcal{C}) = \mathcal{M}(\mathcal{C}, q)$$

例如，提示可以是："根据用户过去的搜索记录和当前兴趣，为查询'q'推荐k个相关选项。"这样，LLM就可以利用这些信息生成针对特定查询的个性化推荐，而无需额外针对每个查询进行单独的训练。

接着，LLM在接收到这样的提示后，会对输出进行动态扩展，生成针对查询q'的完整推荐。为了确保提示的广泛适用性，我们收集了涉及多种主题的多样实例，如公司、人物、产品和历史事件等。针对每个提示，我们精心设计了对应的推荐，并会在公开发布*前进行验证。这样，用户可以期待获得涵盖广泛内容的个性化建议。

GQR通过结合通用大语言模型*和提示，利用LLM在无额外训练的情况下生成相关推荐，避免了多维约束和特定"针对性"系统的复杂性。提示设计涵盖多种主题以确保广泛适用性。这种方法利用LLM的预训练知识，省去了对文档索引的依赖，减少了对用户数据的需求，具有良好的冷启动性能。由于LLM的开放和无约束性，模型能理解和适应各种查询，增强了对错误的鲁棒性*。

Datasets

我们运用了三个数据集作为研究基础：TREC Robust 2004 Disk 4-5，包含250个样本，共528,155文档；ClueWeb09B，包含200个样本，拥有50,220,423份文档；以及从AOL数据抽取*的192个样本，用于用户行为分析*。此外，我们还从查询日志的长尾部分选取了前200个仅出现一次的查询。

Tested Models

实验中，我们通过比较，将GQR与两个匿名的公开Web搜索引擎系统（称为System 1和System 2）进行竞争，通过抓取它们对特定测试查询的推荐来评估。由于ClueWeb09B数据集的规模过大，我们仅在Robust 2004数据集上应用Bhatia等人（2011年）的方法。我们采用了两种GQR模型，GPT-3（通过OpenAI's API，参数量175B）和Huggingface的Bloom模型（参数量176B），在默认设置下使用10个示例提示。

评估推荐系统的性能，我们采用了两种方法。首先，对于查询q和系统生成的推荐集合$r_1, r_2, \ldots, r_k$，我们采用*替换评估*协议。该协议通过将原查询替换成一条推荐，如r_i，并对其性能进行单独评估。这样，我们关注的是推荐的针对性和相关性，判断它们能否有效替代原查询，提供有价值的信息。

$$\text{Substitution}(q, i) \equiv r_i$$

在实验中，我们不仅关注单个推荐的效果，还通过模拟用户选择推荐作为提交查询的过程，进行更全面的评估。为此，我们采用了两个策略：*替换评估*，即将每个推荐r_i替换原始查询q，单独评估其有效性；同时，我们引入了*串联评估*，这是一种更为深入的方法，它通过逐步结合原始查询和推荐，形成序列$q_1, q_2, \ldots, q_{i+1}$，再评估最终的组合查询，以考察推荐序列对信息丰富性和整体效果的影响。这样，我们能更全面地理解推荐系统的效能。

$$\text{Concat}(q, i) \equiv q \oplus r_1 \oplus \cdots \oplus r_i$$

这个策略的主要目的是量化推荐查询集合如何通过增加搜索结果的多样性而提升用户体验，通过追踪每次结合推荐后的查询质量和信息增益，我们系统性地评估推荐序列对整体搜索效果的提升。

Performance Metrics

我们简化了清晰度分数（CS）的计算，采用简化清晰度分数（SCS）。与原始的CS相比，SCS在计算速度上更高效。通过建立针对推荐查询的关联模型，然后与整个语料库的模型对比，使用Kullback-Leibler散度来评估其相关性和一致性，SCS旨在衡量推荐查询与语料库内容的相关程度。

$$SCS(q) = \sum_{w \in q} p(w|q) \log_2 \frac{p(w|q)}{p(w|C)}$$

首先，通过统计语言模型计算词在查询中的概率$p(w|q)$，这是查询清晰性的直观度量，通过词频除以总词数。接着，计算文档集合中词的全局概率$p(w|C)$。高分的SCS表明生成的查询既明确又不含糊，能有效检索相关文档。通过NDCG\@10这一指标，评估推荐查询的检索效果，此指标用于比较与原始查询相关文档的准确性。实验中，我们利用PyTerrier库的BM25方法，并应用配对t检验和Holm-Bonferroni校正来检测统计显著性差异，阈值为$p < 0.01$。

User study

相关推荐系统的核心目标是提供真正有用的建议，而非仅限于检索效率。我们通过用户标注员的评估来确定哪个系统最能吸引人。实验邀请12位专家对三个系统（System1, System 2和GQR，包括GPT-3）的推荐进行评估，共192个随机选取的查询，分为三组，每组64个。匿名处理且推荐按随机顺序展示以避免偏见。评估标准是提供非重复且能有效满足用户信息需求的多样推荐。统计显著性差异分析使用t检验（$p < 0.01$）和Holm-Bonferroni校正进行。

Results and Analysis

我们通过**RQ1**比较无查询日志的GQR系统与现有推荐系统在相关性和实用性方面的表现。在表X和表Y中，我们展示了简化清晰度分数（SCS）和NDCG\@10。

GQR系统在Robust04数据集上，无论是最佳还是最差推荐，均在SCS上领先，仅在ClueWeb09B上稍逊于。总体上，GQR（GPT-3）系统在所有评估中表现最佳，且标准差最小，显示其生成的推荐具有高稳定性和一致性。

知乎

Robust04			
System 1 (a)	9.21	10.17	9.80 ± 0.35
System 2 (b)	9.35	10.56	9.82 ± 0.39
Bhatia [3] (c)	7.64	9.20	8.28 ± 0.53
GQR (Bloom) (d)	7.50	11.00	8.84 ± 1.66
GQR (GPT-3) (e)	10.54	10.77	10.65 ± 0.08
RA-GQR (GPT-3)	**16.71**^{abcde}	**17.10**^{abcde}	**16.98 ± 0.19**^{abcde}
ClueWeb09B			
System 1 (a)	9.80	10.80	10.37 ± 0.32
System 2 (b)	10.49	11.31	10.87 ± 0.30
Bhatia [3] (c)	-	-	-
GQR (Bloom) (d)	7.68	11.20	9.84 ± 1.19
GQR (GPT-3) (e)	10.94	11.22	11.12 ± 0.10
RA-GQR (GPT-3)	**19.42**^{acde}	**19.76**^{acde}	**19.53 ± 0.20**^{abcde}

Table 1: SCS for the *Substitution* protocol for each system on Robust04 and ClueWeb09B. The best values across all systems are boldfaced. The letters indicate a statistically significant difference w.r.t. GQR (GPT-3).

在NDCG\@10指标上，GQR（GPT-3）显著超越了，平均相对提升分别达到23.86%和26.63%，且推荐的标准差也较小，说明其推荐结果更集中且有力。

Model	Min	Max	Avg ± Std
Robust04			
System 1 (a)	0.2038	0.2638	0.2377 ± 0.0211
System 2 (b)	0.2546	0.3739	0.3102 ± 0.0478
Bhatia [3] (c)	0.2388	0.2640	0.2566 ± 0.0108
GQR (Bloom) (d)	0.1743	0.3655	0.2628 ± 0.0714
GQR (GPT-3) (e)	0.3727	0.3947	0.3842 ± 0.0092
RA-GQR (GPT-3)	**0.4197**^{abde}	**0.4476**^{abcde}	**0.4339 ± 0.008**^{abcd}
ClueWeb09B			
System 1 (a)	0.0940	0.1129	0.1056 ± 0.0082
System 2 (b)	0.0946	0.1294	0.1108 ± 0.0164
Bhatia [3] (c)	-	-	-
GQR (Bloom) (d)	0.0468	0.1144	0.0802 ± 0.0268
GQR (GPT-3) (e)	0.1280	0.1659	0.1403 ± 0.0146
RA-GQR (GPT-3)	**0.1597**^{abde}	**0.1832**^{abde}	**0.1708 ± 0.10**^{abde}

Table 2: NDCG@10 for the *Substitution* protocol for each system on Robust04 and ClueWeb09B. The best values across all systems are boldfaced. The letters indicate a statistically significant difference w.r.t. GQR (GPT-3).

原文《Generating Query Recommendations via LLMs》

发布于 2024-06-07 10:43 · IP 属地北京

Spotify LLM 相关搜索