

赞同 4



分享

Spotify 2023：解锁音乐搜索新境界，掌控受控查询生成，提升长尾内容推荐效率与精度



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

4 人赞同了该文章

Introduction

在Spotify内容平台中，用户可以搜索他们已经熟悉的实体。例如，可以查询想听的歌曲名称或想读的书名。这种基于书目数据（如标题、艺术家、作者等）查找实体的搜索行为⁺被称为“狭义”意图查询。然而，用户信息需求是多样化的，并且可能因他们当前的心态而变得更复杂。

当用户在论坛上提出广义查询以解决复杂需求时，现有搜索系统处理效果不佳。当用户带有探索性心态进行搜索时，他们会有更高的容忍度，并倾向于通过“广义”查询来探索到不同选择。在搜索系统中，我们将文档的可检索性定义为多少查询导致实体出现在顶部-k结果中，例如，如果我们假设用户只与列表中排名第一的实体进行交互，那么与，在嵌入空间中没有被查询到的其余实体相比，这些实体将具有高度集中的可检索性，即可检索性偏差。可检索性偏差限制了探索，因为当实体的可检索性分数较低时，通过搜索发现新实体变得更加困难。

本文主要研究生成查询对系统可检索性的影响。与以前的查询生成方法不同，我们提出了 CtrlQGen，它可以控制底层意图。通过为实体生成狭窄和宽泛的查询，我们能够 (I) 训练针对这两种类型意图的密集检索模型，并 (II) 向用户建议更广泛和更具探索性的查询。

此外，通过使用弱标注函数提出的弱标注，CtrlQGen 并不严格要求任何训练数据来为给定实体生成合成查询。考虑到实体的可检索性取决于两个因素：检索模型的修改和查询集。

我们提出了两个可检索性偏差假设：

H1：与使用CQG查询训练密集型检索模型相比，使用真实查询及其相应的点击实体训练，可检索性偏差会更小。点击数据容易出现不同的偏差，例如，许多查询将针对最受欢迎的实体发布，即流行性偏差，在用此类数据训练模型后，这种偏差将在与系统的后续交互中得到加强。相反，使用CQG我们可以为任何给定实体获取查询实体用来训练模型，可以从集合中随机采样得到。

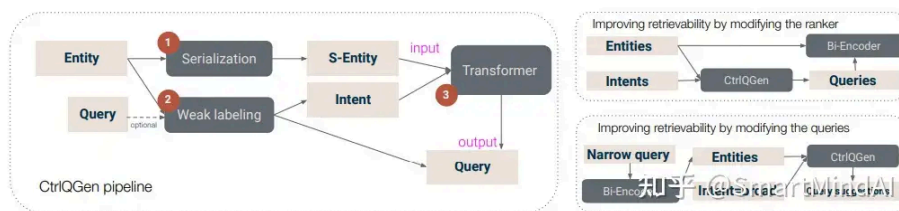
H2：使用CQG建议宽泛查询可导致更少的可检索性偏差。根据定义，相比宽泛查询，窄查询的相关实体更少。通过使用宽泛查询的建议来帮助用户构建他们的查询，我们可能会影响用户的查询行为，并因此影响查询类型的分布。

本文主要贡献是：

- 提出了CQG，它根据所需的底层意图（可以是狭窄的，也可以是广泛的）为给定实体生成查询。我们展示了如何使用生成的查询两种方式：一是作为密集检索模型的训练数据，二是作为查询建议。
- 找到了证据支持假设H1：通过在合成查询上微调的密集模型，相较于仅在点击数据上微调的模型，其可检索性偏差有显著降低。我们发现，使用所提出的查询，相较于仅依赖点击数据的模型，我们的方法在Gini分数上平均降低了10%的可检索性偏差，同时使9%的实体集合的可检索性从无变为有。
- 展示了使用cqq生成查询建议可以减少系统检索偏差高达9%，并且在由无偏查询集训练的双编码器模型时，增加非零检索量的实体数量11%（对于Tracks集合）。

Controllable Query Generation

对大量标记数据的依赖；最后是意图感知生成模块，它可以控制不同类型的意图（如宽泛和狭义）。



Model Components

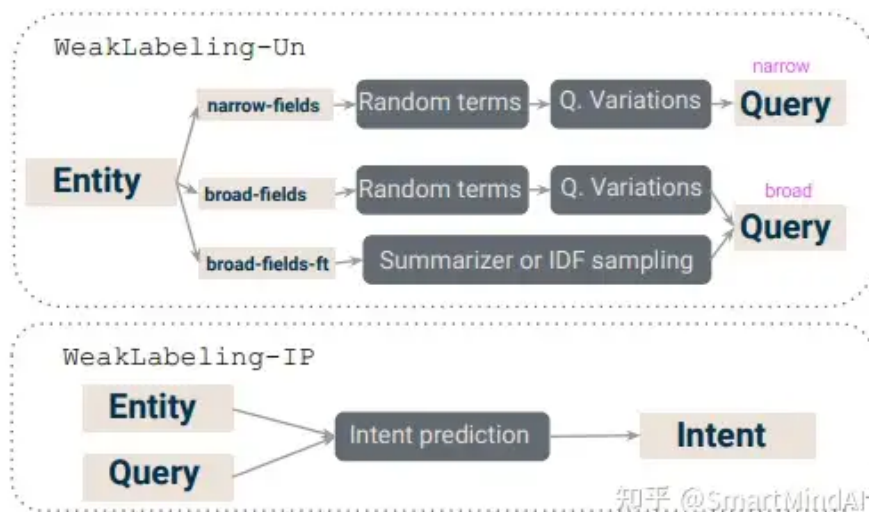
Serialization

这个模块接收一个实体 e 作为输入，并输出该实体的字符串表示形式： $e_{\text{serialized}} = s(e)$ 。序列化函数 s 将实体的每个元数据 $+$ 列与其相应的值连接起来，使用一个特殊标记：

$s(e) = col_1 : val_1 [SEP] col_2 : val_2 [SEP] \dots [SEP] col_n : val_n$ 例如，标题为“魔戒”的书变为： $s(\text{<code>标题” : </code>魔戒”})$

Weak Labeling

为了训练 CtrlQGen，我们需要一个训练数据集 $\mathcal{D} = \{(e_i, i_i, q_i)\}_{i=1}^M$ ，其中实体、意图和查询分别是输入、控制变量和输出。在每个三元组 $+$ 中，当查询 q 与实体 e 匹配时，具有潜在意图 i 。获取此类数据的选项之一是要求注释器为给定实体创建窄查询和宽查询。另一种选择是使用弱标注函数生成此类数据。这里我们介绍两种弱标注函数的变体。第一种是完全无监督的(WeakLabeling-Un)，因此能够对任何给定实体生成查询和意图。第二种需要与每个实体相关的查询，因此基于给定查询的意图预测(WeakLabeling-IP)。



Weakly Supervised Learning for Query Intent Classification

主要思想是定义两组元数据列，一组与特定查询（窄字段）相关并可识别实体，另一组捕捉实体的通用特征（宽字段）。通过从各自字段的所有组合中随机抽取术语来生成查询和意图。窄意图查询的生成需要考虑实体相关的标题、专辑和艺术家。随机函数 $+$ 可用于生成查询变体，如打乱单词、添加拼写错误和移除前缀。宽泛查询需要考虑自由文本列（宽字段-ft）与类别术语（宽字段）的区别。使用IDF高的术语抽样策略和文本摘要模型选择信息丰富的术语。

Intent-aware Generation

作为语言模型提示的一部分。我们使用以下提示来训练模型：“从 \<序列化的实体> 生成具有 narrow / broad 意图的查询”，并以其相应的查询作为输出。

Applications

Synthetic training data

我们可以从随机采样的实体集E'中应用带有所需意图的cqq生成查询，训练Bi-Encoder检索模型。通过应用G函数，可以得到狭窄查询 $q'_{narrow} = G(e, \text{narrow})$ 和宽泛查询 $q'_{broad} = G(e, \text{broad})$ 。然后，根据期望的权重比例（ P_{narrow}, P_{broad} ），从合成的生成查询集Q'中采样训练实例来训练Bi-Encoder。这为我们提供了一个由合成查询和各自相关实体对组成的数据集，可用于训练Bi-Encoder模型，控制底层意图的期望比例。

Query suggestion

我们可以利用 CtrlQGen 模型来进行查询扩展，如图所示。由于实体查询的意图往往较为单一，一种改变用户行为的做法是推荐更广泛的查询。

为了实现这一点，我们可以通过以下方式使用生成的查询。

首先，对于给定的输入查询 q，我们可以用一个排序模型得出一个由 k 个排序实体组成的列表 \mathcal{R}_q 。

然后，对于排名列表中的每个实体，我们用 CtrlQGen 生成一组推荐的广泛查询 \mathcal{Q}' ：
 $\mathcal{Q}' = \{G(e_i, \text{broad})\}$ 对于 e_i 在 \mathcal{R}_q （在我们的实验中，根据接受百分比，将这个查询集 \mathcal{Q}' 附加到日志查询集 \mathcal{Q} 中，以计算检索偏差）。

这种方法的复杂度为 $O(n^2 * d * k)$ ，其中 n 是序列长度，d 是Transformer 模型的维度数，k 是用于生成建议的列表的大小。

Experimental Setup

Datasets

Broad queries datasets

我们采用了两个具有潜在宽泛意图的较小查询和相关标签集，分别是 **Tracks_{broad}** 和 **Podcasts_{broad}**，它们包含1309和500个查询。**Tracks_{broad}** 是根据用户交互信号预测具有高宽泛性的查询样本。基于这个集合，我们获得了点击的实体，避免了查询似乎很宽泛但实际上是狭隘的交互的情况。对于 **Podcasts_{broad}**，我们使用了一组手动策划的宽泛查询和实体对，同时避免完全匹配和匹配不同的元数据字段。

Metadata	(1)	Title	Title	Title
	(2)	Album name	Show name	Series name
	(3)	Artist names	Host names	Author names
	(4)	Release year	Ingested date	Publication year
	(5)	Language	Language	Language
	(6)	Genres	Categories	Genres
	(7)	Descriptors	Episode & show description	Description
	(8)	Lyric	Transcript	User reviews
	(9)	User Playlists	Topics	User lists
# docs	682k	600k	617k	
# queries	100k	100k	100k	
Click # qrels train/val/test	75.9k/9.5k/9.5k	14.4k/1.8k/1.8k	117.5k/14.7k/14.7k	
Avg doc len	55.87	80.76	161.58	
Avg query len	1.96	3.06	4.47	

Implementation Details

Query generation models

作为生成合成查询的基线，我们使用 QGen，一种从文档生成查询的常用方法，并在 click 数据集的 10k 对查询实体子集上对 T5 (t5-base) 进行 fine-tuning。生成查询的第二个基线是 InPars，该模型使用上下文学习，并使用大型语言模型⁺。

为了公平比较，每次生成输出查询时，会随机抽取示例来使用提示，以便 InPars 可以访问与 QGen 相同的查询和实体的训练对数量。我们依靠开源的 bigscience/bloom-760m 发布版来实现。对于 CtrlQGen，我们还依靠 T5 (t5-base) 模型。使用 T5 生成查询时，对于 QGen 和 CtrlQGen，我们采用 do_sample=True 和 top_k=10。

Retrieval models

Evaluation Procedure

我们使用R@100评估检索系统⁺的有效性，旨在提高前100个选项的物品检索率。在95%的置信水平⁺上，使用Bonferroni校正进行学生t检验，比较模型之间的统计学差异。为了评估检索系统的可检索性偏差，我们首先估计实体e的可检索性分数，由以下公式定义：
 $r(e) = \sum_{q \in Q} o_q \cdot f(k_{eq}, c)$ 。其中，Q是查询集（大小为10万）， o_q 是每个查询的权重（使用发布查询的用户数量），如果实体e被搜索系统排名高于c（c设置为100），则 $f(k_{eq}, c)$ 为1，否则为0。为了得到可检索性分数的集中程度或偏差程度，我们计算基尼系数⁺：

$$G = \frac{\sum_{i=1}^N (2 \cdot i - N - 1) \cdot r(e_i)}{N \sum_{j=1}^N r(e_j)}$$

其中，G=1意味着只有一个实体集中所有可检索性，G=0意味着集合中每个实体都具有相同的可检索性分数。为了对基尼系数进行统计测试，我们遵循此文的方法和数据集。

Results

针对 H1的实验结果，即使用合成查询训练的密集检索模型产生的检索偏差小于使用真实查询和点击实体训练的检索偏差。这一结果表明，合成查询在训练模型时能够有效地模拟用户查询，从而减少了检索偏差。

随后，我们描述了针对 H2 的实验结果，即使用我们提出的方法 cqg 生成的宽泛查询相比来自日志的查询集，产生的检索偏差更小。这一结果表明，cqg 方法能够有效地生成宽泛查询，从而提高了检索系统的准确性和鲁棒性⁺。

	Tracks	Podcasts	Books	Tracks	Podcasts	Books	Tracks _{broad}	Podcasts _{broad}
QGen [40]	0.289	0.512	0.756	0.701	0.674	0.766	0.009	0.744
+ (1) Serialization	0.312[†]	0.509	0.761[†]	0.694[†]	0.666[†]	0.761[†]	0.006	0.711
+ (2a,3) Intent-aware Generation WeakLabeling-IP	0.305[†]	0.505	0.761[†]	0.688[†]	0.669[†]	0.756[†]	0.008	0.751
+ (2a,2b,3) Intent-aware Generation WeakLabeling-IP+WeakLabeling-Un	0.289	0.508	-	0.704	0.704	-	0.036[†]	0.787[†]
+ (1,2a,3) CtrlQGen WeakLabeling-IP	0.300[†]	0.522[†]	0.763[†]	0.694[†]	0.676	0.760[†]	0.007	0.713
+ (1,2a,2b,3) CtrlQGen WeakLabeling-IP+WeakLabeling-Un	0.283	0.490	-	0.701	0.674	-	0.029[†]	0.790[†]

H1: Modifying the Ranker with Generated Queries as Training Data

窄意图查询评估

表格显示了三个数据集上不同检索模型的R@100和基尼系数得分，这些数据集主要包含窄意图查询。

- * 零样本模型在训练时无法访问任何点击相关性标签，因此在性能上表现不佳。
- * 双编码器在目标领域查询和实体的性能上表现不佳，其效果比 BM 差。
- * 当使用目标训练数据微调密集检索模型（带点击的双编码器）时，其性能显著优于零样本模型。
- * 无论是点击数据训练的模型还是预训练的双编码器，其偏见都明显高于 BM。

我们发现，当将来自 CtrlQGen 的合成查询与来自 Click 数据集的查询以10% 和90% 的比例组合时（行g），我们可以实现与在 Click 数据集上训练的模型相似的有效性，同时对于 Tracks 和 Podcasts 数据集的可检索性偏差更少，且具有统计意义。因此，这种组合是Pareto最优的，同时满足了两个目标。

	R@100↑			Gini↓		
Zero-shot (no target domain Click training data)	Tracks	Podcasts	Books	Tracks	Podcasts	Books
(a) BM25	0.182 ^b	0.436 ^b	0.721^{bd}	0.752 ^{bh}	0.666^{bcd efh}	0.779^b
(b) Bi-Encoder	0.142	0.323	0.415	0.818 ^h	0.765	0.836 ^d
(c) Bi-Encoder _{WeakLabeling-Un} (Ours)	0.222^{abd}	0.465^b	-	0.748^{abh}	0.730 ^{bh}	-
Fine-tuned on synthetic data (target domain Click training data to train query generators)						
(d) Bi-Encoder _{InPars} [15]	0.202 ^{ab}	0.474 ^{ab}	0.492 ^b	0.712 ^{abh}	0.677 ^{bch}	0.842
(e) Bi-Encoder _{QGen} [40]	0.296 ^{abcd}	0.503^{abc}	0.755 ^{abd}	0.701 ^{abcdh}	0.674^{bcd fh}	0.766 ^{abgh}
(f) Bi-Encoder _{CtrlQGen} (Ours)	0.333^{abcde}	0.500 ^{abc}	0.770^{abde}	0.693^{abcdeh}	0.676 ^{bdch}	0.762^{abegh}
Fine-tuned on target data or in combination with synthetic data (access to target domain Click training data)						
(g) Bi-Encoder _{Click+CtrlQGen} (Ours)	0.361 ^{abcde f}	0.622 ^{abcde f}	0.775^{abde}	0.831^{abh}	0.741^{bh}	0.768 ^{ab}
(h) Bi-Encoder _{Click}	0.366^{abcde f}	0.634^{abcde f}	0.769 ^{abde}	0.856	0.763 ^b	0.767^{abg}

使用宽泛意图查询进行评估

为了了解模型在探索性和复杂信息需求方面的表现，我们仔细研究了模型在仅包含宽泛意图查询的集合中的有效性和可检索性。表显示，当我们在训练中包含更广泛的查询时，从 CtrlQGen 获得的合成查询训练的模型会得到显著改善。仅在合成宽泛查询上训练的模型在 Tracks 上比在 Click 数据上训练的模型高出 111% 的 R@100（行 a 与行 b 进行比较）。

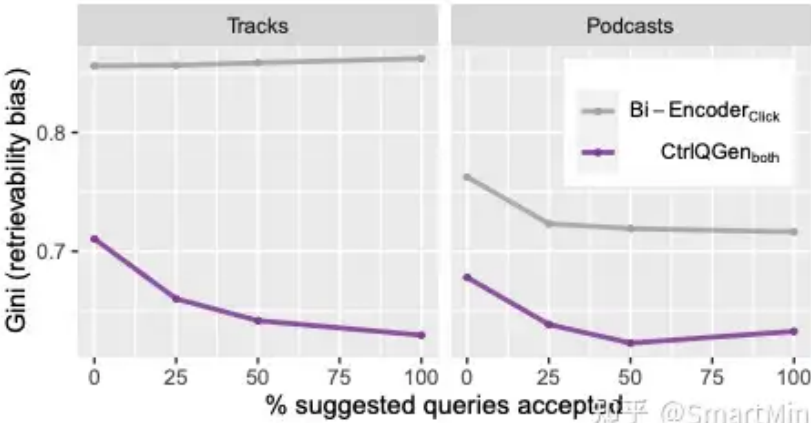
此外，当我们将 CtrlQGen 查询与使用 Click 数据训练的模型进行比较时，我们观察到可检索性偏差有显著下降，从 0.878 降至 0.596，从 0.846 降至 0.831，如表中所示（行 a 与行 b 进行比较）。生成查询的基线模型（行 d 和 e）的可检索性偏差也显著低于在点击数据上训练的模型（行 b）（再次证明我们的第一个假设，即使用合成查询训练的模型可减少可检索性偏差）。

主要的发现是： I. 序列化组件对窄查询和宽查询的R@100和基尼系数都是有益的。 II. WeakLabeling-Un 仅对宽查询有益，因为 WeakLabeling-IP 很好地覆盖了窄查询（它们是可用现有查询的大多数）。

	CTR GEN	CTR GEN	CTR GEN	CTR GEN
(a) Bi-Encoder _{CtrlQGen_{broad}}	0.074^{bcdef}	0.800 ^{ef}	0.596 ^b	0.831 ^{bf}
(b) Bi-Encoder _{Click}	0.035 ^{def}	0.756 ^f	0.878	0.846
(c) Bi-Encoder _{CtrlQGen_{both}}	0.033 ^{def}	0.780 ^f	0.492 ^{abef}	0.831 ^{bf}
(d) Bi-Encoder _{InPars [15]}	0.010	0.827^{cef}	0.489^{abcef}	0.816^{abcf}
(e) Bi-Encoder _{QGen [40]}	0.009	0.744 ^f	0.540 ^{ab}	0.820 ^{abcf}
(f) Bi-Encoder _{CtrlQGen_{narrow}}	0.003	0.609	0.511 ^{de}	0.835 ^{bf}

H2: Modifying the Set of Queries by Suggesting Generated Queries

为了验证我们的第二个假设，即 CtrlQGen 生成的宽泛查询相比日志查询具有更小的可检索性偏差，我们进行了一个模拟实验。在模拟中，一定比例的推荐查询被接受并添加到查询集中，详情如第 3.2.2 节所述。对于日志查询的前 5 个排名实体中的每个实体，我们创建了 3 个查询建议。图显示了模拟实验的结果。从图中可以看出，如果 Bi-Encoder 是在一组宽泛查询上训练的，随着 CtrlQGen 推荐的宽泛查询的百分比增加，系统的可检索性会显着下降，基尼系数下降高达 11% 和 7%，这对于 Tracks 和 Podcasts 来说都是正面的证据，支持我们的第二个假设。如果我们考虑用户接受所有查询的情况，那么与使用 CtrlQGen_{both} 和日志查询相比，Tracks 数据集将有 78k (11%) 个实体变得可以检索，即可检索性不同于零。此外，我们还发现仅修改查询集是不够的，因为使用 Click 数据训练的双编码器无法达到相同的效果，这表明还需要使用既训练过窄查询也训练过宽查询的模型。




Conclusion

我们提出了一个新方法，称为 CtrlQGen，用于生成实体的合成查询。此方法允许我们控制查询意图，并利用内容元数据在没有注释数据的情况下工作，通过使用弱标记函数。我们对生成的查询进行了研究，以了解其对减少检索偏见和有效性的影响，旨在帮助搜索引擎发现更多实体，同时避免对结果相关性的负面影响。我们在三个不同领域进行的实验结果表明，使用 CtrlQGen 生成的合成查询训练密集检索模型可以显著降低系统的检索偏见，同时保持相当的有效性。我们还展示了如何通过建议 CtrlQGen 生成的查询来减少检索偏见。未来的重要研究方向包括：(I)在衡量实体可访问性时考虑推荐和搜索之间的相互作用，(II)改进缺少大部分元数据信息的实体的表示，以及(III)研究减少重新排序场景中系统的检索方法。 (IV)研究提高内容可检索性对内容发现的影响。

欢迎点赞、收藏、关注

编辑于 2023-11-25 12:05 · IP 属地北京

Spotify 搜索引擎 推荐系统



理性发言，友善互动

2 条评论

默认 最新

2023-11-09 · 广东

回复 喜欢

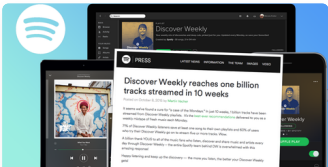
SmartMindAI 作者

关注后发私信

2023-11-09 · 北京

回复 喜欢

推荐阅读

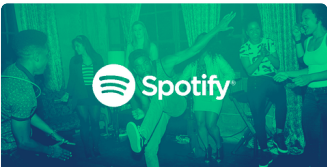


分析一下Spotify的每周推荐为何如此了解你的品位

迟疑症患者

Spotify 每周推荐功能：基于机器学习的音乐推荐

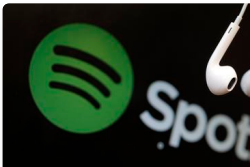
原文地址：Spotify' s Discover Weekly: How machine learning finds your new music 原作者：Sophia Ciocca 译文出自：掘金翻译计划 本文永久链接：github.com/xitu/gold-m... 译者：Isvih



Spotify的每周新发现Discover Weekly：机器学习如何为你...

RRRRL...

发表于数问Dat...



Spotify使用体验报告 (2)

夏以韜