无名稿件编号（将由编辑稍后填写）

使用反事实文本解释说明推荐系统

Niloofar Ranjbar · Saeedeh Momtazi* · MohammadMehdi Homayoonpour

收稿和录用日期稍后填写

arXiv:2303.11160v2 [cs.IR] 1 Jun 2023

摘要 目前，人工智能领域正在开展大量研究，以提升深度学习模型的可解释性和可理解性。研究发现，如果终端用户理解系统产生某项输出的原因，就更容易信任该系统。推荐系统是人们致力于提升其输出可解释性的系统之一。生成更具可解释性输出的一种方法是采用反事实推理，即通过改变尽可能少的特征来生成一个反事实项目，从而改变系统的输出。这个过程有助于识别对期望输出有显著影响的输入特征，进而生成有效的解释。本文提出一种方法，用于为表格特征和文本特征生成反事实解释。我们在三个真实世界数据集上评估了所提方法的性能，结果表明，与基线方法相比，该方法在寻找有效特征方面提升了 +5%（依据基于模型的度量）。

关键词 可解释推荐 · 反事实解释 ·机器学习 · 可解释人工智能 · 推荐系统

Niloofar Ranjbar 计算机工程系 阿米尔卡比尔理工大学（德黑兰理工学院）电子邮箱：nranjbar@aut.ac.ir Saeedeh Momtazi 计算机工程系 阿米尔卡比尔理工大学（德黑兰理工学院）电子邮箱：momtazi@aut.ac.ir
MohammadMehdi Homayounpour 计算机工程系 阿米尔卡比尔理工大学（德黑兰理工学院）电子邮箱：homayoun@aut.ac.ir

1 引言

深度神经模型广泛应用于各种任务，例如医疗保健、决策支持系统和信用风险评估。然而，为了做出公平且稳健的决策，确保这些模型值得信赖且可靠的需求日益增长。

推荐系统旨在预测用户会给某件物品打出的分数，然后向用户推荐排名靠前的物品。用户很难理解系统为何推荐某件特定物品，这可能使他们难以信任系统的推荐。此外，了解推荐背后的推理过程有助于开发人员调试系统。因此，研究人员开始关注推荐系统输出的可解释性。

本文提出一种基于特征的方法，利用特征矩阵预测物品分数。对于用户，该矩阵基于他们互动过的物品。随后，我们使用深度神经模型，根据这些特征预测新用户会给某件物品打出的分数。

一些方法会从用户评论中提取物品的不同方面，并将这些方面用作特征。然而，用户评论中可能不会立即提及许多特征，因此我们还利用物品描述中的元数据来提取更多特征。这样便可以结合连续型、类别型和文本特征来训练模型。

受反事实解释生成方法的启发，我们提出了使用成本函数来识别并突出显示对预测商品分数影响最大的特征的方法。例如，如果预测某款手机会获得高分，这些方法就能识别出促成高分的特征，例如型号、电池、RAM 和摄像头。在其中一种方法中，我们采用 Gumbel Softmax 技巧，从而能够同时考虑所有类型的重要特征（连续型、类别型和文本型）。据我们所知，这是首次在反事实解释中同时纳入所有类型的特征。

值得注意的是，我们选择使用反事实解释生成方法，因为 Yang 等人（2020年）的研究表明，这类方法更容易为人理解，也更符合人类的思维过程。

我们的论文为推荐系统领域做出了多项贡献：–我们提出了一种新颖的基于特征的方法，用于预测商品分数，该方法能够结合连续、类别和文本特征进行训练。– 我们修改了 Tan 等人（2021）提出的反事实解释生成方法，使其能够在推荐系统将原始文本作为特征时生成解释。– 我们提出了一种新的反事实解释生成方法，该方法利用遗传算法，在推荐系统将原始文本作为特征时生成解释。

– 我们提出了一种基于 Gumbel-softmax 方法的反事实解释生成方法，能够在推荐系统中使用所有类型的特征（连续、类别和文本）时生成解释。

这些贡献可在多个领域中得到实际应用。

本文结构如下：第2节回顾反事实解释和可解释推荐方面的相关研究。第3节介绍我们提出的三种解释生成方法，包括 CountER、遗传算法和基于 Gumbel-Softmax 的方法。第4节介绍我们为评估不同方法而开展的实验，并展示在亚马逊和 Yelp 数据集上进行实验的结果。最后，第5节总结研究发现，并探讨未来工作的方向。

2 相关研究

由于我们提出了一种反事实解释方法，并在推荐系统中对其进行测试，因此将相关研究分为两类：应用于文本的反事实解释方法，以及可解释推荐系统。我们的方法将反事实解释与推荐系统创新性地结合起来，预计会引起这两个领域的研究人员和从业者的兴趣。

2.1 反事实解释相关研究

简单来说，预测的反事实解释会找出对输入特征所做的最小改动，使结果变为期望或预定义的结果，而非预测结果（Molnar，2022年）。许多研究都尝试将反事实解释应用于文本数据。在这类情况下，解释应当符合人类的自然理解。仅仅从文本中删除词语来生成反事实解释并不可行。Yang 等人（2020年）通过确保替换后的词语符合语法，解决了这一问题。他们在一项情感分析任务中展示了这种方法，并引入了两份词语列表：一份列出根据语法适合替换的词语，另一份列出与情感词典中的词语含义相反的词语。随后，他们找出这两份列表的交集，并将主文本中的词语替换为交集中的词语，直到预测类别发生变化。这种方法有助于生成更易于人类理解的反事实解释。

许多研究都使用 BERT（Devlin 等人，2018年）等语言表示模型，探讨如何在文本中生成听起来自然的反事实解释。Fern 和 Pope（2021年）提出了这类方法的一个实例。他们首先生成一组候选词语，用于替换文本中的

每个词语。随后，他们将 BERT 用作语言模型，以确定给定位置上每个候选词元的概率。在第二步中，他们利用 Shapley 值（Kalai 和 Samet，1987年）找出最佳的修改组合，并通过束搜索生成解释。这种方法能够在文本中生成更加连贯、易于理解的反事实解释。

Madaan 等人（2021年）和 Wu 等人（2021年）提出的模型都使用 GPT2 以条件方式生成反事实解释。Madaan 等人（2021年）将命名实体标签、语义角色标签或情感定义为词语的条件，而 Wu 等人（2021年）则控制文本中扰动的类型和位置。这两种方法都能更有针对性、更可控地生成文本中的反事实解释。

近年来，使用预训练语言模型生成替代文本，作为文本的对抗样本，已成为一种流行方法。由于生成对抗样本的目标与生成反事实解释的目标相似（即只对输入文本进行最小程度的修改，以改变预测类别），因此这一方向的研究可视为与我们的方法相关。Guo 等人（2021年）尝试使用 BERT 作为语言模型生成概率最高的句子，而 Garg 和 Ramakrishnan（2020年）以及 Li 等人（2020年）则使用 BERT 建议替换词语。这些研究展示了语言模型在生成能够欺骗预测模型的替代文本方面的效用。

许多任务，例如推荐和医疗保健，都涉及由文本、连续特征和类别特征组成的混合数据。要为这类数据生成反事实解释，需要一种能够同时处理所有数据类型的方法。前面提到的方法都没有解决这一问题。本文提出一种方法来解决这一问题，并为混合数据生成反事实解释。

2.2 可解释推荐的相关工作

为提升推荐系统的可解释性，人们进行了大量探索。一种方法是从用户评论中提取物品的各种方面，并将其作为黑盒模型的输入特征。训练好黑盒模型后，目标是找出一组最小特征，这些特征对于特定用户的物品排序最为重要。Zhang 等人（2014）采用矩阵分解来实现这一目标，而 Chen 等人（2016）和 Wang 等人（2018a）则采用了张量分解。其他研究则利用反事实推理生成解释。例如，Zhou 等人（2021）采用了三种策略，通过注意力权重、对抗扰动和反事实扰动，以及从用户评论中提取方面作为特征，使白盒、灰盒和黑盒模型具备可解释性。Ghazimatin 等人（2020）试图找出一组最小的用户行为（如评分），移除这些行为后，推荐物品便会发生变化。Tan 等人（2021）提出了一种基于方面的推荐系统，并尝试

通过反事实推理使其具备可解释性，具体做法是尝试求解一个联合优化问题，以最小幅度地改变物品方面，使得新物品不再被推荐。Pan 等人（2021）尝试通过同时最小化预测损失和解释损失，将不可解释特征映射为可解释特征。

另一方面，Wang 等人（2018b）认为，基于树的模型具有可解释性，而基于神经网络的模型在推荐系统中也有可接受的表现。因此，他们提出将这两类模型结合起来，构建一个可解释的推荐系统。

其他研究利用知识图谱提高推荐系统的可解释性。例如，Wang 等人（2020）尝试利用实体及其关系的语义信息来表示知识图谱路径，从而使知识图谱生成的推荐更具可解释性。Syed 等人（2022）使用一阶逻辑从用户的复杂查询中生成三元组，随后尝试利用知识图谱找出满足这些逻辑查询的实体。这些实体根据其从上下文中获取的信息进行排序，并利用这些三元组生成解释。Shimizu 等人（2022）提出了一种知识图谱注意力网络，利用物品的辅助信息进行推荐并生成解释。Geng 等人（2022）在由实体和边构成的知识图谱路径上训练语言模型，这些路径基于用户行为、物品特征及其间的关系。随后利用生成的图谱生成解释。

尽管通过这些方法生成的许多解释有助于用户理解为什么某件物品会被推荐给他们，但这些解释未必能帮助卖家和管理者更好地满足用户的需求。另一方面，反事实解释可以向卖家和管理者提供信息，指出需要更改物品的哪些特征，才能更有效地向用户推荐这些物品。本文利用反事实推理来生成准确、可信且全面的解释。

3 解释生成方法

本节介绍三种生成解释的方法。

3.1 CountER

受 Tan 等人（2021年）工作的启发，我们尝试找出一个微小的变化向量，在对文本特征向量取平均时将其加到权重上。本节首先讨论基础反事实方法，然后提出我们自己的方法。

3.1.1 CountER 基础模型

假设我们有一组m用户， $\mathcal{U} = \{u_1, u_2, ..., u_m\}$，以及一组$n$物品，$\mathcal{V} = \{v_1, v_2, ..., v_n\}$。对于每种类型的物品（例如手机），我们提取一个包含r个方面的列表， $\mathcal{A} = \{a_1, a_2, ..., a_r\}$。然后，我们使用按以下方式计算的分数构建用户-方面偏好矩阵， $\mathcal{X} \in \mathbb{R}^{m \times r}$，以及物品-方面质量矩阵， $\mathcal{Y} \in \mathbb{R}^{n \times r}$:

$$\begin{aligned} \mathcal{X}_{i,k} &= \begin{cases} 0, \text{ if user } u_i \text{ did not mentioned aspect } a_k \\ 1 + (N-1)\left(\frac{2}{1+exp(-t_{i,k})} - 1\right), \text{ otherwise} \end{cases} \\ \mathcal{Y}_{j,k} &= \begin{cases} 0, \text{ if item } v_j \text{ is not reviewed on aspect } a_k \\ 1 + \left(\frac{N-1}{1+exp(-t_{j,k}.s_{j,k})}\right), \text{ otherwise} \end{cases} \end{aligned} \tag{1}$$

在此上下文中，N 表示评分尺度，此处设为 5。$t_{i,\ k}$ 表示用户 u_i 提及方面 a_k的频率， $t_{j,\ k}$ 表示方面 a_k 在商品 v_j 评论中被提及的频率，而 $s_{j,\ k}$表示这些提及内容的平均情感。

训练黑盒模型后，研究人员通过求解以下优化问题来确定最有效的方面:

$$\begin{aligned} \text{minimize } \mathrm{C}(\Delta) &= \|\Delta\|_2^2 + \gamma\|\Delta\|_0 \\ \text{s.t. } \mathrm{S}(\Delta) &= s_{i,j\Delta} \leq s_{i,jK+1} \end{aligned} \tag{2}$$

$\Delta = \{\delta_0, \delta_1, ..., \delta_r\}$ 是一个由零值或负值组成的向量。在此优化问题中，针对商品 v_j 学习得到 Δ ，使其应用于 $\mathcal{Y}_j$ 向量（$\mathcal{Y}_{j+\Delta}$）后，该商品不再出现在向用户 u_i推荐的前K 名物品列表中。术语$\|\Delta\|_2^2$ 控制 Δ 值的变化量，而 $\|\Delta\|_0$ 控制 Δ向量中发生变化的值的数量。$s_{i,\ jK+1}$是边界物品（前 K 名推荐物品列表中的最后一件物品）的排名分数，而 $s_{i,\ j\Delta}$ 是将Δ 应用于商品 v_j 的方面向量后的排名分数。

通过优化该方程， Δ 会被学习，使商品的方面向量在被移出用户 u_i 的前K 个推荐物品之前，变化幅度最小。

最后， Δ 中取值为负的方面很重要，将它们从方面列表中移除会改变系统的推荐决策。

由于术语 $\|\Delta\|_0$ 和 $s_{i,\ j\Delta} \leq s_{i,\ jK+1}$ 不可微，他们选择$\|\Delta\|_1$ 而非 $\|\Delta\|_0$，并将 $s_{i,\ j\Delta} \leq s_{i,\ jK+1}$ 放宽为铰链损失。因此，最终的优化方程如下:

$$\underset{\Delta}{minimize} \|\Delta\|_2^2 + \gamma\|\Delta\|_1 + \lambda max(0, \alpha + s_{i,j\Delta} - s_{i,jK+1}) \tag{3}$$

其中 $a = 0.2$、 $\lambda = 100$ 和 $\gamma = 1$ 基于 Tan 等人（2021年）提出的模型。

3.1.2 用于词向量的 *countER*

为了确定文本中哪些词语对模型更重要、更关键，我们尝试像基础 CountER 模型那样学习一个向量 $\Delta = \{\delta_0, \delta_1, ..., \delta_z\}$，并将其加到基础权重向量 $\Theta = \{1, 1, ..., 1\}$ 上，以得到商品 v_j 的词特征列表 $W = \{w_0, w_1, ..., w_z\}$。基础权重向量为列表中的每个词语 w_t 赋予权重 1，这些词语属于商品 v_j 的特征。最后，权重低于阈值 t 的词语被视为重要特征；移除这些词语会导致该商品不再位于前K 名列表中。

由于特征中词语的存在与否是一个二元决策，我们从公式3中移除术语 $\|\Delta\|_2^2$，因为在这种情况下，变化量并不重要。

3.2 遗传算法

如前所述，特征中是否包含词语是一个二元决策，因此我们引入另一种基于遗传算法的算法。传统形式的遗传算法使用一组合可能的解，这些解作为解决待求优化问题的方案表示（Kramer，2017）。对于商品 v_j 的词特征列表 $W = \{w_0, w_1, ..., w_z\}$，我们将染色体定义为大小为 z 的二进制向量。算法步骤如下：

– **随机生成种群：** 我们考虑的种群规模为 $\in\{100, 200, 400\}$。初始种群是随机生成的，每条染色体中 1 的数量所占概率为 0.9，0 的数量所占概率为 0.1，因为我们的目标是尽量减少被删除的词语数量。

– **选择：** 为了从染色体中选出下一世代的个体，我们首先计算每条染色体的适应度分数。染色体被选中的概率取决于其适应度分数，适应度分数越高的染色体越有可能被选中。对于大小为z的染色体c，适应度计算如下：

$$
\begin{aligned}
fitness_c &= \frac{1}{\lambda \times (\alpha + s_{i,jc} - s_{i,jK+1})} + countScore_c \\
countScore_c &= \begin{cases} 0.5 \times (1 - \frac{\sum_{r:c_r \neq 0} 1}{z}) & s_{i,jc} > s_{i,jK+1} \\ \beta \times (\frac{\sum_{r:c_r \neq 0} 1}{z}) & \text{otherwise} \end{cases}
\end{aligned} \tag{4}
$$

其中，$s_{i,jc}$ 表示将染色体 c 的值作为权重应用于商品 v_j 的主向量后得到的排名分数，而$s_{i,jK+1}$ 表示边界物品的排名分数。可以看出，适应度函数由两部分组成。第一部分会随着商品分数的降低而增加。 $\alpha = 1$会加到分数差上，以防止出现负值。适应度函数的第二部分（$countScore_c$）是一个惩罚项，旨在尽量减少被移除特征的数量。当商品的排名分数大于边界物品的边际排名分数时，需要移除一些特征来降低排名分数，因此

移除的特征越多越好，但我们为此采用较低的系数（0.5）。另一方面，当商品的排名分数低于边界物品的边际排名分数时，意味着该商品已不再位于前 K 个物品之中，目标也就达成了。因此，对于该模型，应尽可能减少被移除特征的数量。 λ和 β 是需要调优的超参数。– **交叉：** 选择阶段结束后，便开始对染色体配对进行交叉操作。交叉率为 99%。– **变异：** 变异率为 10 %，最多作用于 50% 的种群。对于每条染色体，最多改变其 10% 的基因。

如果最好的适应度分数超过阈值（根据实验选择 1），则将步骤 2 至 4 重复 10 次迭代；否则，算法最多继续迭代 50 次，以便有机会找到更好的解。

3.3 基于 Gumbel-Softmax 的方法

在反事实解释中，我们需要对特征进行最小幅度的更改，使推荐物品不再出现在前K 名列表中。对于连续特征，这很简单；但对于文本特征，更改它们就相当于删除它们或用其他词语替换它们。从文本中删除词语可能会产生无意义的句子。因此，用其他词语替换它们是更好的做法。类别特征也是如此，因为它们必须取自预先定义的取值。由于无法从特征中删除这些取值，更改类别特征就相当于从预定义集合中选择一个新值。

对于参数为离散值的优化问题（例如本问题），Jang 等人（2016年）提出了一种名为 Gumbel-Softmax 的新型梯度估计器。它基于 Gumbel-Softmax 分布，是一种强大的梯度估计器，可将类别分布中不可微的样本替换为 Gumbel-Softmax 分布中的可微样本。该分布的关键特性是能够平滑地退火至类别分布。Guo 等人（2021年）利用这一技巧生成新的、有意义的文本；我们则用它来寻找最好的替代词语。接下来，我们将介绍用于文本特征的方法。由于文本特征与类别特征相似，因此这一方法也可用于类别特征。

假设我们有一个词语列表（作为文本特征） $w = \{w_0, w_1, ..., w_z\}$，用于商品 v_j。更改这些词语，相当于将它们替换为其他可能的词语，同时尽量不改变整个句子的含义。我们使用 BERT 作为语言模型，为每个词语找出五个最可能的替代词。我们将一个词语屏蔽起来，让 BERT 预测替代词。

考虑一个由矩阵 $\Theta \in \mathbb{R}^{z\times(5\times z)}$ 参数化的分布 P_Θ。对于每个词语 w_r，我们在名为 π_r 的 Θ 矩阵中有一个词元概率向量，其中 $\pi_r = Softmax(\Theta_r)$。首先，我们选择概率，使每个词语都将自身选为替代词。随后，我们对参数矩阵 Θ 进行优化，使商品 ˜ 的分数在以最少词语替换的情况下下降。由于 Softmax 函数不可微，因此使用 Gumbel-Softmax 近似。从 Gumbel-Softmax 分布 P_Θ 中抽取样本的方式如下：

$$(\tilde{\pi}_r)_k := \frac{exp((\Theta_{r,k} + g_{r,k})/T)}{\sum_{v=1}^{V} exp((\Theta_{r,v} + g_{r,v})/T)} \tag{5}$$

其中 $(\tilde{\pi}_r)_k$ 是向量 $\tilde{\pi}_r$ 的 k^{th} 值，$g_{r,k} \sim Gumbel(0,1)$，$V = 5 \times z$是备选词列表的大小，而 $T > 0$ 是控制 Gumbel-softmax 分布平滑度的温度参数。当 $T \to 0$ 时，该分布会收敛为类别分布。

为了找到每个备选词的上下文向量，我们将句子中的主词替换为备选词，然后提取上下文词向量，具体内容将在第 4.2.1 节中讨论。最后，我们得到一个矩阵 $C \in \mathbb{R}^{V\times 768}$，其中每个备选词对应一个大小为 768 的向量。将 $\pi \in \mathbb{R}^{z\times V}$ 矩阵与 $C \in \mathbb{R}^{V\times 768}$ 矩阵相乘后，我们得到一个维度为 $z \times 768$ 的矩阵。因此，对于 $w = \{w_0, w_1, ..., w_z\}$ 中的每个词 w_r，我们都有一个大小为 768 的向量。公式3中定义的优化方程变为：

$$\underset{\Theta}{minimize}\ \lambda max(0, \alpha + s_{i,j\Theta} - s_{i,jK+1}) \tag{6}$$

其中，$s_{i,\ j,\ \Theta}$ 表示使用 Θ 计算 π 值后，再对词向量应用 π 得到的用户 u_j 对应商品 v_i 的预测分数，如前文所述。为了用更可能出现的词替换每个词，我们定义一个矩阵 $L \in \mathbb{R}^{z\times V}$。对于每个 $l_{i,\ k} \in L$，如果词语 a_k 在词语 w_i的备选词列表中，那么 $l_{i,\ k}$ 等于 BERT 输出层中词语 a_k对应的 logit；否则等于 -1。请注意，logit 值越高，表示该词在前 5 个备选词中的排名越高。此外，我们将使用原词（不替换任何词）计算得到的主 π 与优化 Θ 矩阵时计算得到的 π 之间的差的 l_1 范数，作为衡量替换词数量的惩罚项。因此，最终的优化方程如下：

$$\underset{\Theta}{minimize}\ \lambda max(0, \alpha + s_{i,j\Theta} - s_{i,jK+1}) + \beta\frac{1.0}{\tilde{\pi}\cdot L} + \gamma\|\tilde{\pi}_{main} - \tilde{\pi}\|_1 \tag{7}$$

其中，$\alpha = 0.2$、$\lambda = 100$（如第 3.1 节所述）、β 和 γ 是超参数，$\tilde{\pi}\cdot L$ 是 $\tilde{\pi}$ 与 L的点积。值得注意的是，随着模型选择更可能出现的词，该点积会增大。

对于类别特征，我们采用相同的表述，只是每个类别特征的备选值不再是词语。此外，无需计算公式7中的第二项。最后，对于

连续特征、类别特征和文本特征的组合，优化问题如下：

$$
\begin{aligned}
&\underset{\Delta,\Theta 1,\Theta 2,\Theta 3}{minimize}\ l_{score} + l_{textual} + l_{cat} + l_{continuous} \\
&l_{score} = \lambda max(0, \alpha + s_{i,j_{\Delta,\Theta 1,\Theta 2,\Theta 3}} - s_{i,jK+1}) \\
&l_{textual} = \beta \frac{1.0}{\tilde{\pi_1} \cdot L} + \gamma \|\tilde{\pi}_{main_1} - \tilde{\pi}_1\|_1 \\
&l_{cat} = \gamma \|\tilde{\pi}_{main_2} - \tilde{\pi}_2\|_1 \\
&l_{continuous} = \|\Delta\|_2
\end{aligned}
\tag{8}
$$

这里， l_{score} 与公式6中所述的损失相同，主要区别在于：计算新分数时会使用所有类型的特征。因此， Δ、 $\Theta 1$、 $\Theta 2$、 $\Theta 3$ 的所有取值都会用于计算该分数。 $l_{textual}$ 与公式7中定义的相同，其中使用 $\Theta 1$ 来计算 $\tilde{\pi}_1$。对于所有类别特征，都会计算 l_{cat}，其中使用 $\Theta 2$ 来计算 $\tilde{\pi}_2$。最后，使用 $l_{continuous}$ 确保用于连续特征的 Δ 变化尽可能小。

4 实验

本节首先介绍实验所需的数据集和设置。接着，我们从定量和定性两个方面评估所提出的方法。最后，我们讨论实验结果。请注意，所有实验均使用 Colab 在 NVIDIA T4 Tensor Core GPU 上进行。

4.1 数据集

我们在亚马逊和 Yelp 数据集上测试了所提出的方法。亚马逊数据集包含 29 个子数据集，每个子数据集对应一个特定的商品类别。我们使用了两个规模不同的子数据集：“手机及配件”和“CD 和黑胶唱片”。每个数据集都包含商品的用户评论、商品描述，以及一些作为文本数据的附加特征。另一方面，Yelp 数据集包含各类商家的信息，例如地理位置（经度和纬度）、营业时间以及其他特殊特征（例如餐厅的停车情况、食物类型等）。它还包含用户对商家的评论和提示。我们将经度和纬度视为连续特征，将提示视为文本特征，并将其他一些特征视为类别特征。

表 1 展示了各数据集的统计信息。对于所有数据集，我们分别剔除评论数少于 5 条的用户和评论数少于 10 条的物品。为准备训练集、验证集和测试集，我们只使用交互过 15 个以上物品的用户。对于每位用户，这些物品均视为正样本。我们留出最后 10 个物品用于验证集和测试集（各 5 个），其余物品用于训练集。我们还从未交互过的物品中随机采样负实例，采样比例为 1:5，即每个正实例对应采样五个负实例。

表 1：数据集统计信息

数据集	用户	物品	评论	Tips	训练集	Test
手机及配件	6794	1945	36762	N/A	2871	93
CD 和黑胶唱片	11467	11677	224090	N/A	686172	3631
Yelp	243531	48089	1426442	626069	501606	3529

4.2 数据准备

4.2.1 准备亚马逊数据集

如图1所示，对于亚马逊数据集，我们使用 BERT 从描述、标题和物品特征中提取文本特征。提取文本特征的步骤如下：

– **清理文本：** 我们移除网址和一些 XML 标签，例如“"，以及它们之间的文本。我们移除由数字和字符组合而成的词语，因为文本中有许多这类无意义的词语。– **使用 BERT 提取上下文向量：** 我们使用“bert_base_uncased

"

该模型会为每个隐藏层中的每个词语生成一个大小为 768 的向量。为了提取句子中词语的上下文化向量，我们使用 BERT 的最后一个隐藏层。– **预处理：** 为了从句子中提取有价值的词语，我们会从文本中移除停用词和标点符号。– **生成最终的物品向量：**我们计算剩余词向量的平均值，然后拼接描述、标题和特征的向量，得到一个大小为 2304 的向量，作为最终的物品向量。请注意，对于不在 BERT词汇表中的词语，如果 BERT 将其标记化为多个部分，我们会计算所有部分向量的平均值，作为该词语的词向量。

4.2.2 准备 *Yelp* 数据集

所提出的方法在 Yelp 数据集上进行了测试，该数据集包含文本、连续和类别特征的组合。商家的用户提示被用于提取文本特征，并按照前文所述的方法生成向量，亚马逊数据集也采用了相同的方法。连续特征包括每个商家的纬度和经度，类别特征则包括商家的营业时间段、是否提供停车位以及供应的食物类型等信息。我们创建了一个二进制向量来表示各个时间段，其中值为 1 表示商家在该小时营业，值为 0 则表示不营业。87 个类别特征采用三个值进行编码：-1 表示商家不具备该特征，0 表示商家是否具备该特征尚未注明，

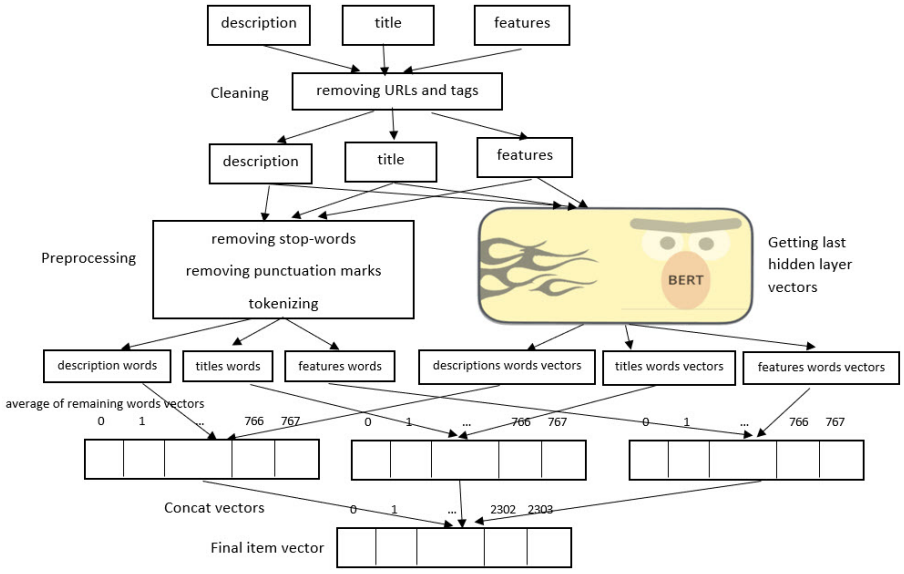

图 1：为黑盒模型准备亚马逊数据

1 表示商家具备该特征。随后，将所有连续特征和类别特征缩放为平均值为零、方差为一。每个商家的最终向量由文本特征向量与缩放后的连续特征和类别特征拼接而成，得到大小为$168+87+768=1023$的向量。

4.3 实验设置

4.3.1 黑盒模型

黑盒模型是一个包含两个隐藏层的前馈网络，两个隐藏层分别有 512 个和 256 个神经元。模型的输入由两个向量拼接而成。第一个向量对应商品，第二个向量对应用户。根据数据可用性，该模型可以使用任何类型的文本特征，以及连续特征和类别特征。请注意，我们通过对用户交互过的物品向量取平均值来生成用户向量。

除最后一层外，每一层后都应用 ReLU 激活函数；最后一层则使用 Sigmoid 激活函数。输出层将排名分数 $s_{i,\ j}$ 映射到 (0,1) 范围内的值，使我们能够根据预测的排名分数为用户推荐前 K 个物品。模型使用交叉熵损失进行训练，并采用学习率为 0.01 的随机梯度下降（SGD）优化器。

4.3.2 超参数

除遗传方法外，所有方法均使用学习率为 (*lr*) 的随机梯度下降（SGD）优化器进行优化。对于 countER 方法，取值为 $lr = 0.01$；对于基于 Gumbel-softmax 的方法，则对其进行调优。

如第 3.1 节所述，对于主要的 countER 方法，我们将公式3中的超参数设为与 Tan 等人（2021年）所用相同的值。然而，对于用于词向量的 countER，我们在验证集上测试了 γ 取值为 0.2，0.3，0.4，...1.0，以及阈值 t 取值为 0.01，0.1，0.2，0.3，0.4，0.5 的不同设置，发现最好的取值分别为 $\gamma = 0.7$ 和 $t = 0.3$。
对于遗传方法，我们在验证集上测试了 λ 取值为 2，5，10，以及 β 取值为 2， 10，50 的不同设置，发现最好的取值分别为$\lambda = 10$ 和 $\beta = 10$。

基于 Gumbel-softmax 的方法将温度 T、学习率 lr、 β 和 γ 作为超参数。我们针对每个数据集分别测试不同的取值。在 $\{0.5, 1.0, 1.2, 1.4, 1.5, 2.0, 2.2, 2.4\}$ 中测试了 T（最好的 $= 2.0$ 和 2.2），在 $\{0.1, 0.2, ..., 0.7\}$ 中测试了 lr（最好的 $= 0.5$ 和 0.7），在 $\{1000, 2000, 10000\}$ 中测试了 β（最好的 $= 1000$），并在 $\{1, 2, 5\}$ 中测试了 γ（最好的 $= 1$）。

4.3.3 评估指标

我们使用 NDCG（归一化折损累计增益）作为衡量黑盒模型排名质量的指标。

受 Tan 等人（2021年）的启发，我们从两个角度评估解释生成方法：面向用户的评估和面向模型的评估。

– **面向用户的评估：**在面向用户的评估中，我们考察解释生成方法提取的商品特征在多大程度上出现在用户对该商品的评论中。

因此，对于商品 v_j 和用户 u_i，如果解释生成方法输出 $E = e_1$， e_2，.....， e_N 作为重要特征，而用户 u_i 在其对商品v_j的评论中提到了词语 $G = g_1$， g_2，.....， g_M，则面向用户评估的精确率、召回率和 F1 值指标计算如下：

$$
\begin{aligned}
Precision = \frac{\sum_{k=1}^{r} p_{i,j}^{k}}{N}, Recall = \frac{\sum_{k=1}^{r} p_{i,j}^{k}}{M} \\
F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \\
p_{i,j}^{k} = \begin{cases} 1 & e_{i,j}^{k} \in G_{i,j} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{9}
$$

其中，N 是解释生成方法选取的特征数量，M 是用户在其商品评论中使用的词语数量。最后，我们对所有配对的分数取平均，得到

最终的精确率、召回率和 F1值。需要注意的是，在将用户评论用作真实特征之前，我们会对其进行预处理，去除标点符号和停用词。随后，将评论中的词语进行词元化，并将其用作真实特征。

– **面向模型的评估：**在面向用户的评估中，我们确定生成的解释特征是否与用户偏好一致。为了检验生成的解释特征是否真正捕捉了模型行为，我们使用另外两个指标：必要性概率（PN）和充分性概率（PS）。

PN 指标回答的问题是："生成的特征是否为模型正确预测排名所必需？"为了回答这个问题，我们从特征列表中移除指定特征，并检查该商品是否仍会被推荐给用户。

PS 指标回答的问题是："生成的特征是否足以让模型正确预测排名？"为了回答这个问题，我们仅使用指定特征，并从特征列表中移除其他特征，然后检查该商品是否仍会被推荐给用户。

我们计算 PN 和 PS 的调和平均数，作为第三个指标 FNS，其计算方式与 F1度量类似。

评估解释生成方法的另一个重要指标是其稳定性。该指标衡量模型在不同运行次数中生成的解释特征有多一致。我们将模型对商品 v_j 和用户 u_i 的稳定性定义如下：

$$Stability_{i,j} = \frac{1}{N(N-1)} \sum_{k=1}^{N} \sum_{l=1,k\neq l}^{N} \frac{|r_k \cap r_l|}{|r_k \cup r_l|} \tag{10}$$

其中，r_k 是该方法在第k^{th} 次运行中生成的解释特征集合，N 是运行次数，对于所有数据集，N 均设为 10。模型的最终稳定性通过对所有配对的分数取平均值来确定。我们使用测试集中的 10 个随机选取的示例，计算所有数据集的稳定性。

4.3.4 可比基线

在实验中，我们采用 Tan 等人（2021）提出的基线方法作为唯一的对比方法。该方法在其论文中与其他最先进的方法进行了比较，结果表明其表现显著优于这些方法。因此，我们选择将我们的方法与该基线方法进行比较，而该方法本身已优于其他方法。

尽管我们承认可能还有其他方法可用作竞争方法，但鉴于基线方法表现更优，且已与该领域的其他竞争方法进行过比较，我们决定将重点放在基线方法上。此外，考虑到数据集发生了变化，如果不重新运行其他论文中的代码，直接比较我们的结果并不合适，

表 2：亚马逊数据集解释结果，基于用户和基于模型

数据集		基于用户			基于模型		
		Pre	Rec	F1	PN	PS	FNS
手机	基线	**0.273**	**0.298**	**0.250**	0.950	0.918	0.934
	countERText	0.1075	0.0497	0.057	0.915	0.971	0.9426
	遗传	0.085	0.092	0.072	0.890	**0.995**	0.940
	Gumbel	0.110	0.068	0.073	**0.970**	0.990	**0.980**
CDs	基线	**0.223**	**0.329**	**0.228**	0.778	0.679	0.725
	countERText	0.177	0.016	0.025	0.820	0.997	0.90
	遗传	0.121	0.034	0.038	**0.957**	0.997	**0.97**
	Gumbel	0.117	0.024	0.028	0.796	**0.998**	0.886

因此，我们选择重新运行 Tan 等人（2021）的代码。该研究取得了该领域最好的结果，且已与其他竞争方法进行过比较。

4.4 亚马逊数据集结果

由于亚马逊数据集和 Yelp 数据集使用了不同类型的特征，我们分别呈现它们的结果。亚马逊数据集的实验结果见表 2 和表 3。需要注意的是，这些表格中的基线方法指的是 Tan 等人（2021年）提出的主要 CountER 方法。

4.4.1 生成的特征是否基于用户偏好?

如表 2 所示，与基线相比，所有引入的方法在基于用户的度量上的精确率、召回率和 F1 分数均有所下降。原因如下：基线方法使用一组固定的方面，在手机数据集和 CD 数据集中分别包含 88 个和 230 个方面，模型则从中输出一部分作为重要特征。然而，我们的方法会从商品的描述、标题和特征中提取重要词语，而这些词语并非来自一个特定的词集。此外，这些词语在用户做决定时可能确实非常重要，但用户未必会在其评论中直接使用。因此，使用这一度量将我们的方法与基线方法进行比较并不公平。

通过使用这一指标比较我们的方法，我们可以发现它们之间没有显著差异。

4.4.2 生成的特征是否规定了模型行为?

表 2 所示结果表明，我们提出的基于模型的度量方法显著优于基线。仔细分析数据后可以发现，遗传算法和 Gumbel-softmax 算法在这两种

表3：亚马逊解释结果，其他指标

数据集		NDCG	特征平均值	解释发现率	稳定性	时间（秒）
手机	基线	0.3561	2.97	78%	**0.862**	1.28
	countERText	**0.4047**	6.05	98.9%	**0.862**	1.69
	遗传	**0.4047**	15.94	98%	0.175	4
	Gumbel	**0.4047**	9.61	99%	0.51	7.4
CDs	基线	0.481	5.09	72%	**0.68**	1.35
	countERText	**0.712**	2.567	98%	0.65	0.43
	遗传	**0.712**	7.32	95%	0.47	1.76
	Gumbel	**0.712**	6.97	74%	0.54	4.8

数据集上都取得了最好的结果。具体而言，在手机数据集上，Gumbel-softmax 算法在 PN 和 FNS 度量方面表现更优；与遗传算法相比，其 PS 度量的性能差异可以忽略不计。另一方面，在 CD 数据集上，与所有其他方法相比，遗传算法在 PN 和 FNS 度量方面有显著提升，而其 PS 度量的表现则与其他方法相当。

4.4.3 更改特征对排名的影响

如表3所示，衡量黑盒模型排名质量的归一化折损累计增益（NDCG）值，在两个数据集上均较基线显著提高。这一提升源于使用了不同的特征输入。不过，需要注意的是，三个模型 counterText、遗传和 Gumbel 使用了相同的推荐方法，但受益于不同的解释生成模型。因此，三个模型的 NDCG 结果相同。这是因为 NDCG 只评估推荐的准确性和质量。

总体而言，使用这些特征似乎能够更可靠地向用户推荐物品。下一步是确定所提出的方法是否足够可靠，能够根据用户偏好生成准确的解释。

4.4.4 哪种方法平均生成的解释所包含的特征数量更少？

如表3所示，基线（countER）和 countERText方法分别在手机和 CD 数据集上生成特征数量最少的解释。Gumbel-softmax 方法在这一排名中位居其次，最后是遗传算法，其找到的特征数量最多。不过，值得注意的是，遗传算法也能够找到更多解释，这可能表明它能够通过使用更多特征，为其他算法无法处理的配对找到解释。另一个原因可能是差异

在遗传算法的表述和优化方面，与其他方法有所不同。

4.4.5 哪种方法能为更多配对找到解释?

评估指标中一个值得注意的差异是解释找到率，它衡量能够找到解释的用户-物品对数量。如表3所示，总体而言，countERText 技术表现最好，在手机数据集上的解释找到率为 98.9%，在 CD 数据集上为 98%。遗传和 Gumbel 技术的表现也不错，但解释找到率较低。这种差异可以理解，因为 Gumbel-softmax 算法需要将词语替换为其他语义恰当的词语，这可能会增加解释生成方法找到解释的难度。相比之下，遗传和 countERText 方法只需从文本中删除一些词语，任务相对简单。不过，值得注意的是，Gumbel-softmax 方法旨在适用于包含多种特征类型的数据集，例如文本、类别和数值特征。因此，对于仅包含文本特征的数据集，不应期待它有特别出色的表现。尽管如此，在两个数据集中，Gumbel-softmax 方法仍优于基线。

4.4.6 稳定性

表3显示，基线和 countERText 方法在两个数据集上的稳定性都最高。遗传方法在两个数据集上的稳定性都较低，在手机数据集上的稳定性得分为 0.175，在 CD 数据集上的稳定性得分为0.47。Gumbel 方法的稳定性也较低，在手机数据集上的稳定性得分为0.51，在 CD 数据集上的稳定性得分为0.54。

遗传方法和 Gumbel 方法的稳定性得分较低，可能是因为这些方法在算法中使用了随机性。例如，遗传方法采用遗传算法，其中包含随机变异和交叉操作，因此不同运行次数可能会产生不同结果。同样，Gumbel 方法涉及生成随机变量并用它们计算概率分布，这也可能导致不同运行次数产生不同结果。此外，不同运行之间结果的方差可能归因于各方法所使用的特征类型。基线方法采用的特征集有限，因此不同运行之间的方差较小。相反，其他方法将词语用作特征，因此结果可能具有更大的变异性。然而，在推荐系统的上下文中，不同运行时获得不同的特征集或许有益，因为这样可以从多个角度为用户提供多样化的解释。因此，在这种情况下，判定稳定性较高或较低哪种更好并不合适。

4.4.7 时间复杂度

我们通过测量为一个用户-物品对生成解释所花费的平均时间来评估所提出的方法。表3以秒为单位报告了所用时间。对于手机和 CD数据集，由于它们仅包含文本特征，使用我们提出的方法生成解释所花费的时间完全取决于每件商品的文本描述、标题和特征的长度。从我们提出的三种方法可以看出，Gumbel-Softmax 耗时最长，而 countERText 耗时最短。不过，基线方法所需时间与 countERText 和遗传算法几乎相同。这是因为，我们所提出的算法中用作特征的词语数量多于基线方法所用的方面数量，而且搜索空间也比基线方法更大。因此，这种差异是合乎情理的。

4.5 Yelp 数据集结果

对于 Yelp 数据集，我们进行了三项实验，以研究不同类型的特征对黑盒模型准确率的影响。此外，我们还考察了多种特征组合对模型生成的解释的影响。这些实验旨在确定哪些特征及特征组合最有效，既能实现准确预测，又能清晰解释模型的决策。

在第一项实验中，我们使用所有特征（包括文本特征〔从用户提示中提取〕、类别特征和连续特征）来训练黑盒模型。为了寻找解释，我们结合公式3和公式7定义了一个新的损失函数，可用于寻找前述各类特征的解释。具体而言，公式3用于连续特征，公式7用于文本特征和类别特征。第二项实验仅关注文本特征，并比较了多种寻找解释的方法，与此前在第4.4节亚马逊数据集实验中的做法相同。最后一项实验仅使用类别特征和连续特征来训练黑盒模型，并采用与第一项实验相同的方法生成解释。Yelp 数据集实验结果见表4和表5。

4.5.1 生成的特征是否基于用户偏好 ？

如表4所示，基于用户偏好评估特征所得的结果与亚马逊数据集上的结果一致。由于用户偏好的真实特征是从文本特征中提取的，因此无法评估不使用文本特征的方法。通过比较文本特征的结果可以推断，在这方面，基线方法取得了更好的结果，

表4：Yelp 数据集结果，基于用户和基于模型

特征类型		基于用户			基于模型		
		Pre	Rec	F1	PN	PS	FNS
All	Gumbel	**0.101**	0.059	0.044	0.607	0.609	0.608
文本	基线	0.080	**0.1661**	**0.0909**	0.9544	**0.9619**	0.9582
	countERText	0.092	0.058	0.051	0.8966	0.9385	0.9171
	遗传	0.099	0.101	0.072	**0.9712**	0.956	**0.9638**
	Gumbel	0.072	0.040	0.041	0.75	0.918	0.826
非文本	Gumbel	N/A	N/A	N/A	0.647	0.495	0.555

表5：Yelp 数据集结果，其他指标

特征类型		NDCG	特征平均值	解释发现率	稳定性	时间（秒）
All	Gumbel	**0.8499**	**7.6**	**98.3%**	0.59	5.4
文本	基线	0.515	**7.51**	68%	**0.74**	1.36
	countERText	**0.6172**	13.819	**93.4%**	0.58	0.78
	遗传	**0.6172**	17.24	73.76%	0.21	4
	Gumbel	**0.6172**	11.39	59.2%	0.51	2.8
非文本	Gumbel	**0.8434**	**2.73**	89.9%	**0.77**	2

因为该方法需要识别的特征较少。本节的结果分析与第4.4节所呈现的发现一致。

4.5.2 生成的特征是否明确了模型行为？

如表4所示，仅使用文本特征时，生成的解释能更好地帮助理解模型行为。然而，使用非文本特征时，生成的解释在明确模型行为方面效果不佳。这可能是因为许多非文本特征与某些商家无关，因此将它们从特征列表中移除，并不会显著影响这些商家的排名分数。因此，PN分数低于其他方法。类似地，对于PS分数，也会出现同样的现象：某些特征与特定商家无关，因而不足以使其进入前 K 名列表，导致分数低于其他方法。解决这一问题的一种潜在方法是，分别考虑与每类商家相关的类别特征。

此外，使用所有特征时，结果优于仅使用非文本特征，但不如仅使用文本特征时的效果。这可能是因为非文本特征被包含在文本特征中。

此外，使用文本特征时，遗传算法表现最好，但找到的特征数量也更多。其次表现较好的是基线和 countERText 方法，而 Gumbel 方法表现最差。

总之，我们的贡献在于能够通过 Gumbel-softmax 同时找到包含多种类型特征的解释

方法。通过改进类别特征和连续特征的表述，希望能够进一步提升结果。

4.5.3 特征变化对排名的影响

如表5所示，使用所有特征或仅使用类别特征和连续特征时，归一化折损累计增益值更高。相反，使用文本特征会导致黑盒模型的归一化折损累计增益值较低。这可能是因为本研究所用的文本特征是从用户对物品的提示中提取的，而不是直接从物品特征中提取的。在使用用户评论提取亚马逊数据集的文本特征时，也观察到了相同趋势。在这种情况下，为了避免这个问题，我们直接使用商品的描述和特征，而不是用户评论。尽管如此，非文本特征仍表现出值得关注的归一化折损累计增益分数。通过比较提取文本特征的方法，可以推断，特征提取方法对黑盒模型的归一化折损累计增益分数起着至关重要的作用。

4.5.4 哪种方法能为更多配对找到解释，且平均使用的特征数量更少？

如表5所示，使用所有特征时，解释生成算法能够为98%的配对生成解释，平均使用7.6个特征。尽管需要搜索大量特征，该算法仍取得了如此高的成功率，这表明它能够为相当大比例的配对生成解释，且所需特征相对较少。另一方面，仅使用文本特征的基线方法只能为68%的配对生成解释，平均使用7.51个特征。countERText方法表现良好，解释找到率达到93.4%，但平均使用的特征数量较多（13.81个）。遗传算法的表现也不错，但平均使用的特征更多，达到17.24个。然而，Gumbel softmax方法的表现相对较差，只为59.2%的配对生成了解释，平均使用11.39个特征。仅使用非文本特征时，可为89.9%的配对生成解释，平均使用2.73个特征。总体来看，使用所有特征时，Gumbel方法能够为很大比例的配对生成解释，而且平均使用的特征数量相对较少。

4.5.5 稳定性

如表5所示，当仅使用文本特征时，基线方法的稳定性为0.74，明显高于其他方法，这与亚马逊数据集的发现一致。另一方面，使用全部类型

表6：手机数据集示例 1

user_id	A3VVMIMMTYQV5
item_id	B00U7YKO78
基线	手机、电池、灯光
countERText	移除词语：’版本’,’国际’ ,’保修’,’三星’,’无线’,’us’
遗传	移除词语：’qi’,’galaxy’,’note’, ’go’,’select’, ’daydream’,’more’,’corner’,’三星’,’国际’, ’美国’,设备’,’ 微型’,’USB’,’充电’,’WPC’
Gumbel	词语：[’封面’, ’三星’, ’无线’, ’国际’, ’白色’], 更改为: [’白日梦’, ’与’, ’银河’, ’主板’, ’;’]
用户评论 关于商品	[’塑料’, ’背板’, ’我有一个 Spigen 手机壳，带有透明塑料背板’], [’充电’,’快速’,"充电速度太快了，我还没来得及真正 弄清楚它什么时候充满电"], [’手机’, ’马上’, ’我马上给手机充了电，试试看’], [’手机壳’, ’透明’, ’我有一个 Spigen 手机壳，带有透明塑料背板’], [’手机壳’, ’透明’, ’我的 Spigen 手机壳是透明塑料的’], [’灯光’、’蓝色’、’蓝色灯光将开始闪烁’], [’灯光’、’蓝色’、’它会沿着正面发出两道深蓝色的管状灯光 让你知道

特征的Gumbel方法稳定性为0.59，高于仅使用文本特征时的稳定性。这表明，使用全部类型的特征时，Gumbel方法生成的特征更加一致。使用非文本特征时，其稳定性高于其他方法。这可能是因为非文本特征的数量远少于文本特征。这些结果凸显了为特定任务选择适当特征并采用稳健方法生成稳定特征的重要性。

4.5.6 时间复杂度

Yelp 数据集生成解释所花费的时间如表5所示。这一时间不仅取决于作为文本特征使用的提示长度，也取决于我们拥有的类别特征数量。不过，由于 Yelp 数据集中的提示通常比亚马逊数据集中的描述、特征和标题更短，因此仅针对文本特征生成解释所花费的平均时间少于亚马逊数据集。然而，随着添加其他类别特征并使用所有类型的特征，生成解释所花费的时间会增加。尽管如此，与亚马逊数据集一样，我们的方法用于搜索并生成反事实解释的特征空间比基线方法更大，因此生成解释所花费时间的差异是合乎情理的。

表7：手机数据集示例 2

user_id	A1F7YU6O5RU432
item_id	B00Z7RQ0NC
基线	手机、质量、按键、设备、手机壳、保护、保护壳、屏幕 ，颜色，握持感
countERText	移除词语：’使用寿命’、’日期’、’保护”、’图形’、’Plus’、’iPhone’ 、’吸收’、’无缝’、’夜晚’、’耐受’、’边缘’、’白色’
遗传	移除词语：’手机’、’iPhone’、’放大器’、’iPhone’、’仅’、’日期’ ，’iPhone’，’详情’
Gumbel	未找到解释
用户评论 关于商品	[’手机壳’，’漂亮’，’喜欢它的外观，而且当 太阳照在手机壳上的银色斑点上，真漂亮’], [’手机壳’,’透明’,’手机壳是透明的，所以无论你的 iPhone 是什么颜色 你都能透过闪光隐约看到它’], [’照片’,’在线’,’从在线照片中一开始可能很难看出来的是 这个手机壳的透明塑料里面有许多漂亮的闪光 透明塑料\’

表8：CD 数据集示例1

user_id	A3LEN0P07MGJE2
item_id	B001TRDPB4
基线	未找到解释
countERText	移除词语：’圣诞节’，’干杯
遗传	移除词语：’成员’，’生活’，’出售’，’感觉’，’2008’ ，’全国范围’，’巡演’，’碗’，’带来’，’另一个’，’追饮’，’铃声’， ’圣诞节’ ，’干杯’
Gumbel	词语：[’欢呼’,’圣诞节’,’驯鹿’] 更改为: [’”’, ’拍摄’, ’今晚’, ’！ ’]
用户评论 关于商品	[[’转折’,’新的’,’有些歌曲对我来说是新的 以及对经典最喜欢作品的新演绎——引入了 《We Three Kings》让人想到其中暗示的 《碟中谍》主题曲’], [’演绎’, ’在’,’《红鼻子驯鹿鲁道夫》中的演绎’]]

4.6 定性评估

本节将展示不同方法针对不同数据集找到的一些特征示例。

手机数据集的示例见表6和表7。由于用户评论篇幅较长，这里只列出了真实方面及包含这些方面的句子。需要注意的是，如果无法查看完整评论，可能就无法全面评估各方法的有效性。不过，正如第一个示例所示，遗传算法识别出的方面词“charge”也出现在用户评论中。此外，词语“case”和“plastic”也出现在用户评论中，且与 Gumbel 方法识别出的方面“cover”相似。基线方法也将方面“light”识别为用户评论中的重要方面。然而，为了进行公平比较，有必要查看商品的完整描述

表 9：CD 数据集示例 2

user_id	A1SCJWCMQ3W3KK
item_id	B00006879E
基线	歌曲，摇滚乐、布鲁斯、曲调、发行、唱片、合集、热门歌曲
countERText	移除词语：’简’
遗传	移除词语：’first’、’歌曲’
Gumbel	词语：[’史蒂维’、’简’、’歌曲’], 改为：[’他们’、’节奏’、’那些’]
用户评论 关于商品	[’发行’、’在’，”而《魔发奇缘》更让人想起某种东西 摩城唱片会在20世纪70年代发行的”], [’发行’、’回到’、1，“而《魔发奇缘》则更让人想起 摩城唱片在20世纪70年代发行的作品。”], [’音乐’，‘新鲜’，‘这张唱片至今听起来仍像新鲜的音乐’], [’乐队’，‘在’，‘这让乐队的感觉稍微柔和了一些’] [’布鲁斯’，‘一点’，‘一点布鲁斯’] [’摇滚乐’，‘在’，‘以及一种前所未有的摇滚乐方式 在美国广播电台中鲜为人知（至少在大范围内）] [’摇滚乐’、’少量’、’少量摇滚乐’], [’曲目’、’第一’、’前八首曲目精彩绝伦’] [’放克’、’少量’、’一点放克\’]

和特征，以及全部用户评论，以理解这些词语的使用上下文及其表达的具体含义。由于这些文本篇幅较长，因此本文未予展示。

CD 数据集的示例见表 8 和表 9。如表中所示，基线方法在第一个示例中未能识别出任何解释，而其他方法则找到了一些解释。不过，与手机数据集一样，如果无法查看完整的评论和描述，就无法判断这些已识别方面的相关性。在第二个示例中，可以看出用户表达了对前几首曲目的偏好，而遗传算法将“第一”和“歌曲”识别为重要词语。此外，用户在评论中提到了“摇滚乐”“蓝色”和“发行”这些词语，基线方法也识别出了它们。Gumbel方法和 CountER 都将一个名字识别为重要方面，这可能与用户相关，但也可能出现在评论中未收录于此表的其他部分。

Yelp 数据集的示例见表 10 和表 11。如示例所示，利用全部类型的特征时，Gumbel方法能够识别连续特征、类别特征和文本特征的变化。例如，在第一个示例中，提示的措辞以及商家的纬度和经度都略有变化，该商家便不再被推荐给用户。

此外，第二个示例表明，仅使用非文本特征时，如果将清真选项从“未提及”更改为“False”，表示该餐厅不供应清真食品，那么该餐厅便不再推荐给用户。

表10：Yelp 数据集示例 1

user_id		nlReKgQoRz6uPfVaEG93mw
item_id		tU692E8N0xBQ7Ogc78gN2g
所有特征		纬度从 36.177038 变为 36.17646826 经度从 -86.749691 变为 -86.75116574 词语: ['前往', '任何事', '雨', '土豆', '房子'] 变为: ['关于', '地点', '很多', '!', '黑色']
文本特征	基线	员工，味道，内部
	CountERText	移除词语: ['奶味的', '前往', '方式', '桌子', '咖啡', '地方', '很棒的', '摩卡', '尝试', '美味的', '麻木的', '最好的', '纳什维尔', '早餐', '令人惊叹的', '聚会场所', '得到']
	遗传	移除词语: ['顺序', '节庆', '令人上瘾的', '绿色', '爱', '氛围', '奶味的', '最好的', '城镇', '乔', '一切', '错误', '巴西的', '模糊', '炸弹', '可疑', '曾经', '烘烤', '很棒的', '早餐', '简单', '糖浆', '鸢尾花', '花', '奶油', '切片','桌子','聚会场所','绽放','冰镇','美式咖啡', '葡萄柚','茶','黄瓜','甜','超级','拿铁', '办公室','酒吧','披萨','苹果','酱', '不错','咖啡']
	Gumbel	未找到解释
非文本特征		纬度从 36.177038 变为 36.17698086 经度从 -86.749691 变为 -86.7495435
商家上的用户评论		这是个很棒的地方，可以进来用餐，也可以买些东西带走

对该商家的用户评论进行分析后发现，不同的解释方法都将“鸡肉”“沙拉”和“食物”识别为重要词语，这表明这些方法的结果是可以接受的。

值得注意的是，对于每个商家，都可以分别识别类别特征的变化，并建议商家考虑这些特征。例如，告知经理提供清真食品或自行车停车位可能会吸引更多顾客。

5 结论与未来工作

在本文中，我们介绍了三种用于生成文本解释的方法，并在三个推荐系统任务的真实世界数据集上对其进行了评估。CountERText 和遗传方法只能找到作为解释的文本特征，而采用了 Gumbel Softmax 的 Gumbel 方法则可应用于所有类型的特征，包括文本、类别和连续特征。我们通过实验评估了这些方法，发现使用商品文本特征时，我们的方法在基于模型的度量方面优于基线，这意味着找到的解释特征对于模型做出准确预测既是必要的，也是充分的。尽管使用商品上的用户提示作为文本特征时，模型表现不佳，但基于 Gumbel Softmax 的方法有

表11：Yelp 数据集示例 2

user_id		0du93EkEwKuxRG _x6hqVUg
item_id		KnsY8rh5tigp5t6WpilGdA
所有特征		纬度：从 36.103133 变为 36.10256344 经度：从 -86.8185 变为 -86.81997484 全天营业：从未提及变为 True 词语： ['酒吧','加仑','星期日','袋子','女孩', '香肠', '早餐','更好','棒极了'] 更改为： ['厨师', '?', '红色', '男人', '沙拉', '食品', '你', '开始', '黎明']
文本特征	基线	咖啡，最喜欢的，地点，奶酪，品尝， 价格，进食，小时，食材，茶， 巧克力、街区、商店
	countER Text	移除词语：['平均值'、'类别'、'包含'、'日'、'酒吧'、'跑步'、 '纳什维尔','自助餐','小时','山丘','黄油' ， '柴茶','伙计们','高峰','来','食物','啤酒', '感恩节','看见','规则','挑选','今天','沙拉]
	遗传	移除词语：['袋子','果汁', '午餐','跑步','多样性', '果汁', '传统的','感恩节','感觉','购物','女孩','感谢','不错的','黄油', '肉','观看','曲棍球','猪肉','精彩的','行动','点了', '拿了','来','沙拉','得到','泰国菜','泵','星期六','思考' ,'来','隐藏的','酒吧','杏仁','包括','啤酒','Narragansett', '热的','想明白了','美味的','沙漠','早餐','一致的','选项', '选择' ,'选择' ,'早餐' ,'鸡肉' ,'热的','规则', '寻找中','史诗般的','烤制的','不错的','胡椒','伙计们','很棒的','有机的', '披萨','点单','桌子','棒极了','橙子','食材','免费]
	Gumbel	词语:['南部的','星期六','最好的','山丘','地点','规则', '酒吧','伙计们', '今天','精彩的','棒极了'] 改为: ['中午','是','任何','村庄',';', '做了','也','香肠', '喜欢','得到','.','去','吃','少量']
非文本特征		纬度：从 36.103133 变为 36.10318994 经度：从 -86.8185 变为 -86.81864752 清真：从未提及变为否
用户评论关于商家		[['奶酪', "你会发现有趣的奶酪"], ['鸡肉沙拉', "还有美味的鸡肉沙拉"], ['食品', '他们准备的食品也相当不错']]

望生成基于多种特征的解释，这对涉及多模态特征的任务（如医疗保健）可能有所帮助。未来的工作可以改进模型，针对不同类型的商家分别生成解释，避免将其类别特征合并。此外，还可以仅将名词作为文本特征，并基于这些特征生成解释。进一步而言，可以通过考察用户评论与所有已找到特征之间的语义相似度，改进基于用户偏好评估解释的方式。

参考文献

Chen X, Qin Z, Zhang Y, Xu T (2016) 学习对多个类别推荐特征进行排序。载于：第39届国际ACM SIGIR信息检索研究与发展会议论文集，第305–314页 Devlin J, Chang MW, Lee K, Toutanova K (2018) BERT：用于语言理解的深度双向Transformer预训练。arXiv预印本 arXiv:181004805 Fern X, Pope Q (2021) 通过潜在优化和Shapley引导搜索生成文本反事实。载于：2021年自然语言处理实证方法会议论文集，第5578–5593页 Garg S, Ramakrishnan G (2020) BAE：基于BERT的文本分类对抗样本。arXiv预印本 arXiv:200401970 Geng S,FuZ, Tan J, Ge Y, De Melo G, Zhang Y(2022) 基于知识图谱的路径语言建模，用于可解释推荐。载于：2022年ACM Web会议论文集，第946–955页 Ghazimatin A, Balalau O, Saha Roy R, Weikum G (2020) PRINCE：在推荐系统中通过反事实解释实现提供方可解释性。载于：第13届国际Web搜索与数据挖掘会议论文集，第196–204页 Guo C, Sablayrolles A, Jégou H, Kiela D (2021) 针对文本Transformer的基于梯度的对抗攻击。载于：2021年自然语言处理实证方法会议论文集，第5747–5757页 Jang E, Gu S, Poole B (2016) 使用Gumbel-Softmax进行分类重参数化。arXiv预印本 arXiv:161101144 Kalai E, Samet D (1987年) 关于加权Shapley值。《国际博弈论杂志》16(3):205–222 Kramer O (2017) 遗传算法。载于：《遗传算法要义》，施普林格，第11–19页 Li L, Ma R, Guo Q, Xue X, Qiu X (2020) BERT-Attack：使用BERT攻击BERT。载于：2020年自然语言处理实证方法会议（EMNLP）论文集，第6193–6202页 Madaan N, Padhi I, Panwar N, Saha D (2021) 生成你的反事实：迈向文本反事实生成的可控化。载于：AAAI人工智能会议论文集，第35卷，第13516–13524页 Molnar C (2022)《可解释机器学习》，第2版。网址：https://christophm.github.io/interpretable-ml-book Pan D, Li X, Li X, Zhu D (2021) 通过可解释特征映射和可解释性评估实现可解释推荐。载于：第二十九届国际人工智能联合会议论文集，第2690–2696页 Shimizu R, Matsutani M, Goto M (2022) 一种基于改进的知识图谱注意力网络的可解释推荐框架

，具有海量辅助信息。《基于知识的系统》239:107970 Syed MH, Huy TQB, Chung ST (2022) 基于领域知识图谱的上下文感知可解释推荐。《大数据与认知计算》6(1):11 Tan J, Xu S, Ge Y, Li Y, Chen X, Zhang Y (2021) 反事实可解释推荐。载于：第30届ACM信息与知识管理国际会议论文集，第1784–1793页 Wang N, Wang H, Jia Y, Yin Y (2018a) 通过观点文本数据中的多任务学习实现可解释推荐。载于：第41届国际ACM SIGIR信息检索研究与发展会议论文集，第165–174页 Wang T, ZhengX, He S, ZhangZ, Wu DD(2020) 学习用户-物品路径以实现可解释推荐。《IFAC-PapersOnLine》53(5):436–440 Wang X, He X, Feng F, Nie L, Chua TS (2018b) Tem：用于可解释推荐的树增强嵌入模型。载于：2018年万维网大会论文集，第1543–1552页 Wu T, RibeiroMT, Heer J, Weld DS (2021) Polyjuice：生成反事实以解释、评估和改进模型。载于：计算语言学协会第59届年会暨第11届国际自然语言处理联合会议（第1卷：长篇论文）论文集，第6707–6723页 Yang L, Kenny E, Ng TLJ, Yang Y, Smyth B, Dong R (2020) 为金融文本分类中的深度Transformer生成合理的反事实解释。载于：第28届国际计算语言学会议论文集，第6150–6160页 Zhang Y, Lai G, Zhang M, Zhang Y, Liu Y, Ma S (2014年) 基于短语级情感分析的显式因子模型，用于可解释推荐。载于：第37届国际ACM SIGIR信息检索研究与发展会议论文集，第83–92页 Zhou Y, Wang H, He J, Wang H (2021) 从内在解释到反事实解释：论上下文化推荐系统的可解释性。arXiv预印本 arXiv:211014844