arXiv:2603.01795v1 [cs.HC] 2 Mar 2026

PleaSQLarify：面向自然语言数据库查询的视觉语用修复

Robin Shing Moon Chan
苏黎世联邦理工学院
苏黎世，瑞士
robin.chan@inf.ethz.ch

Rita Sevastjanova
苏黎世联邦理工学院
苏黎世，瑞士
rita.sevastjanova@inf.ethz.ch

Mennatallah El-Assady
苏黎世联邦理工学院
苏黎世，瑞士
melassady@ai.ethz.ch

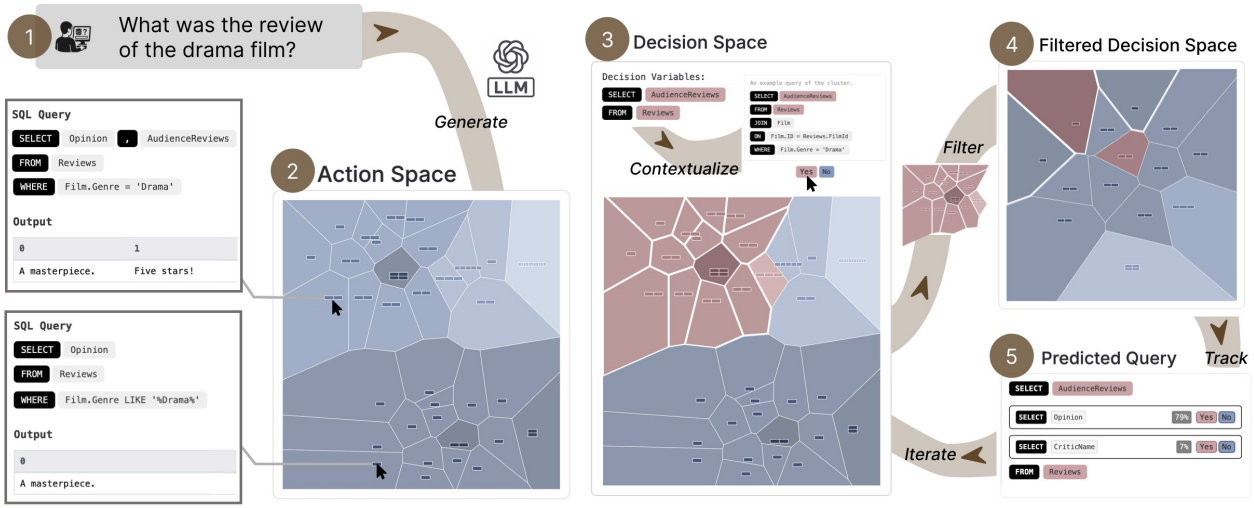

图 1：PleaSQLarify 概览，这是一种应用于文本到 SQL 的交互式语用修复方法。(1) 用户提出一个信息不充分的自然语言查询。(2) 系统生成一个初始动作空间，其中包含可能的 SQL 解释。(3) 系统向用户展示最有信息量的决策变量，即能够以有意义的方式划分动作空间的变量。(4) 用户通过参与对比替代解释的澄清，逐步筛选该空间。(5) 预测查询会持续更新，并在各轮交互中保持可追溯性，使用户能够探索、比较和完善各种解释，直至与其意图相符。

摘要

自然语言数据库接口拓展了数据访问途径，但在输入存在歧义时仍显脆弱。标准方法通常会将不确定性归结为单一查询，几乎无法处理用户意图与系统解释不匹配的情况。我们通过语用推理重新审视这一挑战：用户会精简表达，而系统则基于动作空间上的先验进行操作，这些先验可能与用户的先验并不一致。从这一视角来看，语用修复——通过最少交互逐步澄清——是解决欠指定问题的自然策略。我们提出 PleaSQLarify，通过围绕可解释的决策变量来组织交互，实现语用修复，从而高效地进行澄清[1]。可视化界面则通过呈现动作空间以供探索、请求用户消除歧义，并使跨轮次的信念更新过程清晰可追溯，为这一方法提供补充。在一项有十二位参与者参加的研究中，PleaSQLarify 帮助用户识别替代解释，并高效地消除歧义。我们的研究结果表明，语用修复是一项设计原则，有助于在自然语言界面中实现有效的用户控制。

关键字

歧义、文本/语音/语言、可视化、大语言模型、语用学

1 引言

面向数据系统的自然语言界面使用户能够使用日常语言查询数据库并与之交互 [12, 13, 37, 38]。例如，数据分析师可能会在自然语言数据库接口中输入“显示上一季度的顶级客户”，以探索不熟悉的数据集。这些交互让人感觉很自然，因为人们可以用自己的话表达意图，无需使用技术命令或形式语法。然而，这种自然性也使它们很容易出错：这类请求往往存在歧义，而系统通常会选取最可能的解释并执行相应的系统操作 [13, 23, 39]。这一挑战在自然语言编程中尤为突出，例如文本转 SQL [12, 20, 38, 44], ，在此类任务中，自然话语必须转换为精确的数据库查询。尤其是在这类任务中，看似简单明了的请求往往不止一种有效解释 [35]。CoSQL 数据集 [41] 中的证据进一步印证了这一点：在用户以对话方式请 SQL 专家编写基本 SQL 查询的 2,459 次交互中，31.9% 的交互

[1]代码：github.com/chanr0/pleasqlarify

包含一条要求澄清用户意图的指令。尽管有些系统尝试通过交互式界面 [21, 28], 来处理这类歧义，但当前系统仍无法有效地呈现歧义，也未能提供可靠的澄清引导机制。

退一步来看，这个问题反映了语言使用的一项基本属性：欠指定。语言使用的语用学理论 [11, 14, 31] 将人类描述为讲求效率的说话者，他们会有意省略细节，并依靠听者根据语境和共享知识还原预期含义。这种欠指定是人类沟通的自然结果；然而，在自然语言界面中，它会使系统面临多个看似合理的操作，而用户却不知道系统考虑了哪些操作，也不知道原因。类似于人类沟通中的澄清，我们提出，应通过语用修复直接以交互方式处理歧义。也就是说，在人类对话中，欠指定的表达通常会逐步得到澄清：说话者只补充足以将其意图与备选方案区分开来的信息，而听者则会有针对性地请求澄清。类比而言，自然语言界面可以通过呈现替代解释，并提出既高效又易于用户理解的澄清请求，更有效地消除歧义。

为将这一理念付诸实践，我们首先提出自然语言界面中语用修复的一般框架，并将其应用于文本到 SQL，提出 PleaSQLarify。如图 1 所示，该方法将一种算法与可视化界面相结合：算法负责选择信息丰富的分组决策变量；可视化界面则（1）呈现模型的动作空间，（2）通过结合上下文的澄清支持引导式缩小范围，（3）使跨轮次的更新过程可追溯。系统不再迫使用户接受单一且不透明的查询，而是鼓励他们探索、比较并逐步完善各种解释，直至得到满意的查询。在一项有十二名参与者参加的研究中，PleaSQLarify 帮助用户发现他们此前未曾想到的替代解释，并高效地引导系统得出满意结果。交互轨迹揭示了三种话语澄清工作流程——广泛探索、定向预筛选和输出驱动式组装，展现了用户如何灵活调整修复策略，以满足自身目标并应对任务复杂度。总体而言，本文作出了四项贡献：

- 面向自然语音界面的语用修复概念框架。我们将语用推理理论拓展至人机交互，展示如何将用户话语中的欠指定作为逐步澄清的一种资源。
- 一种用于自然语言界面中语用修复的算法，以及面向文本到 SQL 的实例化。该算法将原子特征分组为语义上可解释的决策变量，并依据期望信息增益在其中进行选择，从而实现既高效又有意义的澄清。在 AMBROSIA 数据集上的定量评估表明，该方法能以比基线方法更少的轮次降低语义不确定性。
- 一种可视化界面，通过使模型的动作空间可见，将语用修复落实到文本到 SQL 中，允许用户呈现信息丰富的决策变量，并在各轮次中保持更新的可追溯性。
- 来自用户研究的实证见解，展现了所提方案的有效性，并为如何更广泛地支持自然语言界面中的修复提供了设计启示。

2 相关工作

本节综述文本到 SQL 中歧义概念的研究现状，以及基于算法和用户反馈的消歧方法。

文本到 *SQL* 中的歧义。 Liu 等人的一项近期综述 [21]将歧义识别为数据库自然语言接口中一项关键且尚未解决的挑战。研究者已识别出多种歧义形式，例如范围和模糊性，并由此推动创建分类体系以对其进行系统组织（例如， [6, 10]）。若干基准测试集应运而生，以应对不同类型的歧义，其中包括 CoSQL 数据集 [41]„ 其中包含超过 10’000 条已标注的 SQL 查询，这些查询澄清了歧义并标记了无法回答的问题；SPLASH [9]„ 其中包含带有错误 SQL 解释和自然语言反馈的话语；AmbiQT [2], , 其使用一种新颖的解码算法创建，该算法将基于计划的模板生成与受约束填充相结合；以及 AMBROSIA [35], , 这一基准测试集包含三种不同的歧义类型：作用域歧义、附着歧义和模糊性，其查询旨在支持多种有效解释，以反映现实世界的复杂度。在本工作中，我们使用 AMBROSIA 数据集，既用于对我们所提出算法进行定量评估，也用于提取用户研究中使用的查询样本。

基于算法的消歧。 近年来，已有若干算法方法被提出，用于在文本到 SQL 流水线的不同阶段进行自然语言查询消歧。例如，Kim 等人 [18] 提出了 APA（与感知歧义对齐，Alignment with Perceived Ambiguity）流水线，使大语言模型在执行前处理含歧义的查询。另一个例子是“先消歧、后解析”（Disambiguate First, Parse Later）范式，它采用两阶段方法：首先通过生成所有可能的自然语言含义来消解歧义，随后对每个由此得到的无歧义解释进行解析 [36]。Wang 等人 [40] 提出了一种弱监督模型 DTE （先检测后解释，Detecting-Then-Explaining），通过将词元定位构建为序列标注任务来处理含歧义且无法回答的问题，并将每个词元标注为含歧义、无法回答或其他。歧义消解也可以在 SQL 转换过程中进行。例如，Chen 等人 [5] 将预训练分类器与保形预测方法结合使用，以评估转换步骤中错误检测的可能性。Qiu 等人 [33] 近期同期的工作提出了一种 SQL 消歧方法，该方法利用与相似度无关的子句级期望信息增益，同时忽略了查询之间的功能等价性以及特征之间的交互。

基于用户反馈的消歧。 除了纯算法方法之外，交互式可视化系统也常被用于让用户通过直接操作和反馈来解决歧义。然而，这些系统提供反馈的方式各不相同。例如，Debug-It-Yourself（DIY）为用户提供沙盒环境，让他们与问题到查询的映射和数据库子集进行交互，并在检测到错误时进行调整

[28]。一个名为 FISQL（反馈增强型 SQL 生成工具）的交互式框架，通过采用提示策略，利用用户提供的自然语言反馈来完善 SQL 生成 [26]。AmbiSQL 使用上下文学习来检测含歧义的短语、对其进行分类，并生成有针对性的澄清问题，提供多项选择以供重写查询 [8]。若干研究聚焦于特定领域内的消歧。例如，DataTone 采用混合主动式消歧，通过交互式歧义控件将自然语言查询转化为可视化结果；这些控件会呈现系统的决策，并允许用户立即解决歧义 [13]。用户的更正会作为约束存储，并影响后续查询，从而形成一个学习系统。Setlur 等人 [37] 提出了一种系统，用于解决可视化分析界面中话语的歧义。他们结合句法和语义约束以及启发式方法，限制解空间。Ma 等人 [24] 将分层澄清策略扩展到开放域问答，通过构建消歧树，将含歧义的查询系统地分解为可导航的解释空间。这与我们的方法在概念上相似：模式元素和连接路径构成自然的决策层级。Zamfirescu-Pereira 等人 [43] 开展的一项更具实证性的研究揭示了非专业用户如何迭代地完善查询；他们强烈倾向于选择低表述成本的策略，例如从给定选项中选择，而不是一开始就编写详细的规范。Poesia 和 Goodman [32] 将交互式消歧应用于编程语言，使程序员能够使用受控程度的歧义。

我们的工作建立在这些基础之上，同时关注文本转 SQL 系统中歧义消解的算法层面和界面层面。我们的目标是摆脱近期利用大语言模型进行消歧的方法所具有的黑箱特性 [8, 18, 26, 36]，转而构建以人为中心且可解释的框架，让每项用户决策及其影响都能得到解释。具体而言，我们提出一种解决方案，以语用语言理论为基础，构建一种能够选出信息丰富的分组决策变量的算法，并将该算法与可视化交互组件相结合，以引导用户做出知情的查询消歧决策。

3 背景：语用学

语用推理与 RSA 用户，令 $u \in U$ 表示用户从可能话语集合 U 中选出的一条自然语言话语，用以传达预期含义 $m^* \in \mathrm{M}$。在概率语用学的理性言语行为理论（RSA）中，交际效率 [11], 被表示为

$$p_S(u \mid m^*) \propto \exp(\alpha \cdot U(u, m^*) - C(u)), \tag{1}$$

其中， $U(u, m^*)$衡量 u将 m^*与替代项区分开来的程度，而 $C(u)$衡量 u的生成成本。我们注意到，$U(u, m)$由说话者对含义的先验 $p_S(m)$所塑造。解释者，即听者的 L，其任务是推断含义 m，而该含义来自说话者所选的话语 $u \sim p_S(u \mid m^*)$。在RSA中，含义在概念上是从依据贝叶斯规则定义的后验 $m \sim p_L(m \mid u)$中采样的，给定听者对含义的先验 $p_L(m)$，

$$p_L(m \mid u) \propto p_S(u \mid m) \cdot p_L(m). \tag{2}$$

语用修复。 在人际交流中，话语接地 指说话者与听者通过互动建立相互理解，确认话语在语境中得到了正确解读，从而使双方对 m^* 的信念趋于一致。如果说话者与听者拥有共同的共同基础，他们的先验就会高度一致。在这种情况下，即便是相对欠明确的话语，也可能产生低熵、峰值集中的后验 $p_L(m \mid u)$。维持相互理解的语用机制是语用修复，也就是在话语存在歧义或意义不确定时进行的渐进式澄清。在 Clark 和 Wilkes-Gibbs 的 [7] 协作指称模型中，语用修复是逐步进行的：说话者通常先用最简短、有时较为宽泛的描述来指称对象，再逐步加以细化，只补充足以将预期含义与其余备选方案区分开来的新信息。这种由粗到细的细化体现了最小协作努力原则 [14], ：对话者避免一开始就过度描述，而是在需要消除剩余不确定性时才补充细节。

4 自然语言界面的语用修复

当用户发出欠明确命令时，自然语言界面必须以某种方式恢复预期含义。我们认为，这一挑战可以通过语用推理的视角来理解：用户会精简其表达，而系统则依据可能与用户期望不一致的先验进行运作。在本节中，我们将这一错位形式化，并讨论语用修复作为一种有原则的澄清策略，以文本到 SQL 领域作为贯穿示例以便说明。

4.1 自然语言界面中的语用推理

在自然语言界面中，用户向系统提供一段话语 u，其交际意图 m^* 是通过系统操作 $a^* \in \mathrm{A}$.[2] 获得所期望的系统结果。例如，在图 2 所示的文本到 SQL 场景中，用户可能会向系统 u =询问“我们销售哪些产品？”，并期望系统返回销售表中的产品列表。随后，系统根据用户的话语推断用户意图 m，并将其映射到合适的、可执行的系统操作 $a \in \mathrm{A}$，以期实现用户所期望的结果。在上述示例中，这对应于找到查询 a^* = “SELECT product FROM sales”。需要注意的是，通常多个系统操作都可以满足用户意图，只要它们在功能上等价；例如，本例中的目标查询“SELECT product FROM sales”和“SELECT s.product FROM sales s”。形式化地说，系统会根据意图 $m \in \mathrm{M}$ 生成可执行操作上的分布 $p(a \mid m)$ $a \in \mathrm{A}$，。该映射的定义域为 $\mathrm{dom}(f)$ $= \{ m \in \mathrm{M} \mid \exists\, a \in \mathrm{A}\ \ p(a \mid m) > 0\}$，即系统至少支持一个可执行操作的意图集合。因此，我们可以将功能等价程序的先验概率汇总为意图上的先验 $p_S(m)$ 和 $p_L(m)$。

在自然语言界面中，一个显著的挑战是，用户往往并不完全了解可执行操作集合 A（参见执行鸿沟 [29]）。因此，用户表达的意图m^* 可能无法映射到任何可执行系统操作，或者用形式化的方式表示为，

[2]集合 $\mathcal{A}$ 表示 可执行系统操作。

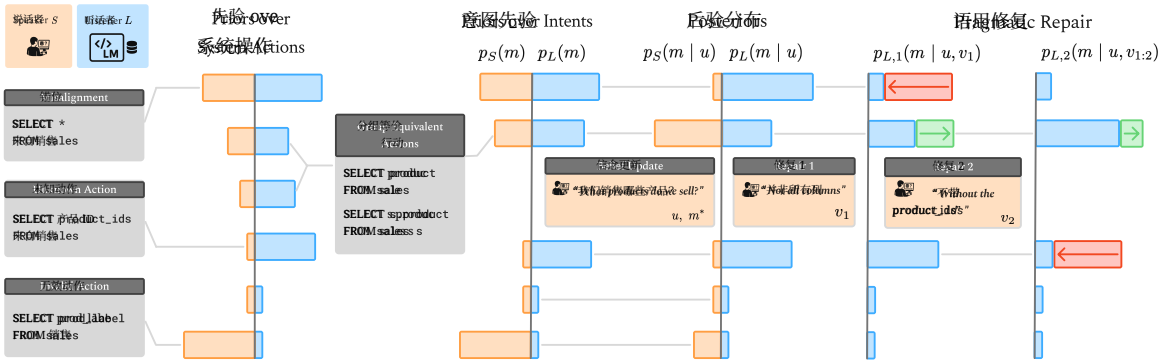

图 2：应用于文本转 SQL 任务的人机交互语用修复。对系统操作的先验存在错位，即用户不了解系统可能执行或认为可能执行哪些操作，这会加剧歧义。

$m^* \notin \mathrm{dom}(f)$。在图 2 的示例中，用户可能会询问“产品标签”，尽管数据库中并不存在“prod_label”。

更一般而言，用户与系统关于系统操作的先验之间可能存在错位。换言之，用户可能并不了解系统认为哪些系统操作是可能的，也可能无法估计所产生的话语 u 区分 m^* 其他可能意图的程度。在图 2 的示例中，系统可能总体上偏向于显示所有列，即生成 a = “SELECT * FROM sales”。用户并未意识到这是他们需要加以区分的对象，并假定其话语 u =“我们销售哪些产品？”会得到一个在 m^*处呈尖峰的后验 $p_L(m \mid u)$。然而，对系统而言，该话语是有歧义的；其所得后验 $p_L(m \mid u)$是高熵的，从中采样会导致意外行为。

因此，自然语言界面中的交际效率不仅取决于共享先验下话语的信息性，还取决于先验本身的对齐。这种对齐可以通过以下方式促进：（a）向用户呈现动作空间 A，从而降低对可能结果的不确定性；（b）调整或校准系统的先验，以更好地反映用户的期望，例如通过澄清机制。

4.2 作为语用修复的交互式话语语义落地

虽然展示可用的动作空间 A 有助于用户提出更准确的查询，但直接呈现完整的动作集合往往并不实际，而且会带来较高的认知负担，尤其是在 A 很大或技术性很强时。自然会引出这样一个问题：类似于语用学中的话语接地，是否也能通过这类澄清性修正，以交互方式让用户未充分说明的查询在含义上得到语义落地？为此，系统可以利用用户初始话语所产生的后验分布 $p_L(m \mid u)$，仅聚焦于可采取行动的候选含义。这样一来，问题便从“搜索 A”转变为“在少数几个可能的意图中消除歧义”，从而通过有针对性的渐进式澄清来解决歧义。形式化地，如图

2, 0 = m p L m | 所示，

u > 0 u

令 M { : ( ) } 为初始话语之后的候选 意图集合。在每个交互轮次 t，系统选择一个决策变量$Z_t : \mathrm{M}_{t-1} \to \mathrm{V}_t$——即候选项之间存在差异的一个显著维度——并向用户展示备选方案 $v \in \mathrm{V}_t$。关键在于，为尽量减少交互轮次，系统选择能最大化相对于意图分布的期望信息增益的变量 Z_t。用户的选择 $v_t \in \mathrm{V}t$会产生更新后的候选集 $\mathrm{M}_t = \{m \in \mathrm{M}_{t-1} \mid Z_t(m) = v_t\}$。这一细化相当于基于新增证据对后验进行渐进式条件化：

$$p_{L,t}(m \mid u, v_{1:t}) \propto p_L(m)\, p_S(u \mid m) \prod_{i=1}^{t} \mathbb{1}\{Z_i(m) = v_i\}. \tag{3}$$

随着多轮交流的进行，候选集通过有针对性且与语用相关的修复操作逐步缩小，这与人类说话者反复完善信息不足的指称表达的过程相似，直到双方都能将其含义视为彼此认可。

4.3 语用修复的设计启示

上述论述为我们提供了有关支持语用修复的系统之设计需求的启示。

R1 总体而言，系统应以渐进且易于人类理解的方式呈现可用的动作空间 A。**R2** 假设用户的目标是实现特定结果 m^*，澄清应区分不同的意图，而非系统操作，因为区分功能等效的操作与用户目标无关。**R3** 最小协作努力语用原则倡导通过优先向用户提出信息量大的澄清来尽量减少澄清交互的次数，即选择能够最大程度减少对预期含义的不确定性的澄清。

5 自然语言接口中的语用修复算法

接下来，我们描述一种为自然语言界面实现语用修复的算法。遵循第4.3节中规定的设计原则，我们通过以下方式表示动作空间：(1) 生成一组可能的系统操作，(2) 聚合功能相似组并对其进行聚类，(3) 基于这些簇中最具判别性的特征提取决策变量，以及 (4) 通过信息增益提取并呈现信息量最大的特征。

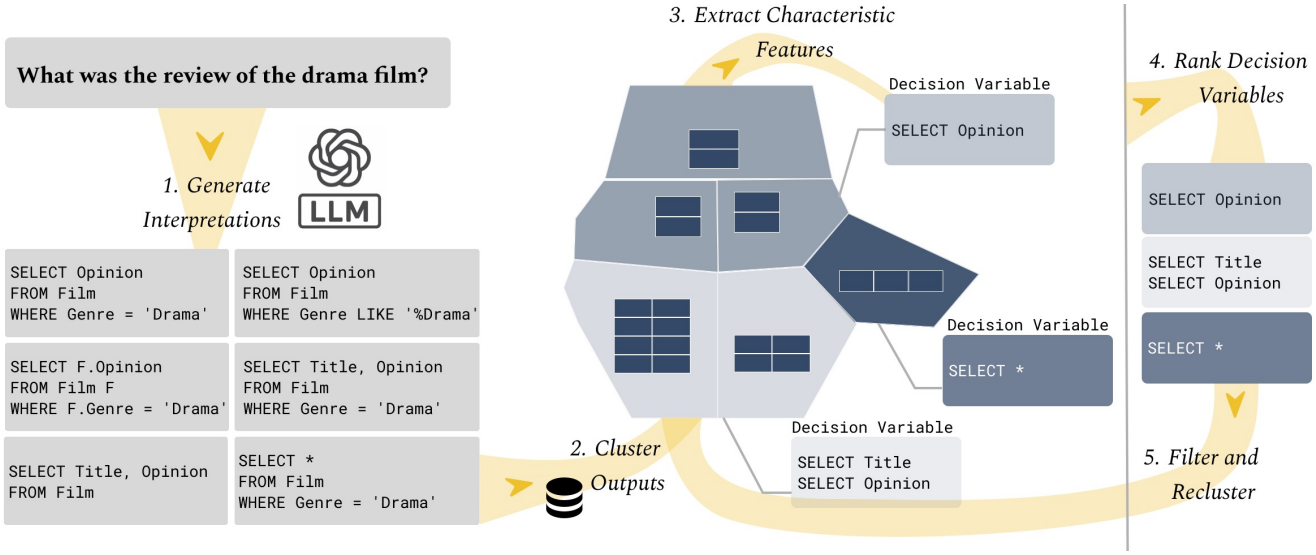

图3：将第5节所述算法应用于文本到 SQL 的可视化。首先，大语言模型生成多样化的输出，然后将这些输出嵌入并聚类。对于每个簇，我们根据提升度提取最具代表性的特征。最后，根据期望信息增益对决策变量排序。用户做出决策后，候选集会经过筛选，剩余候选项则重新聚类。

1. 生成一组可能的操作。 为了支持对输入话语 u的系统解释进行广泛探索，我们首先在该输入话语下生成一个有限候选集，其中包含 N可能的系统操作 $\mathrm{A} = \{a_1, \ldots, a_N\}$。生成 A 的机制取决于具体领域；对于基于语言模型的任务，可通过以包含任务指令的输入话语 u 为条件对语言模型进行重采样来实现，即 $\mathrm{A} = \{a_i \sim p_{\mathrm{LM}}(a \mid u)\}_{i=1}^{N}$。

2. 功能聚类。 如第3节所述，多个不同的系统操作可以实现相同的意图 m，即执行相同的功能。我们将其形式化为：如果两个系统操作产生相同的功能结果，就称它们在功能上等价。若在上下文 C 中执行操作 a 的结果为 $m(a, \mathrm{C})$，则如果 $m(a_i, \mathrm{C}) = m(a_j, \mathrm{C})$，操作 a_i 和 a_j 在上下文 C 中功能等价。我们进一步引入两个系统操作之间的功能相似度 $S(a_i, a_j)$，将其定义为 $m(a_i, \mathrm{C})$ 与 $m(a_j, \mathrm{C})$ 之间的相似度分数。如果 $S(a_i, a_j) \approx 1$，则结果几乎完全相同。利用这一相似度度量，我们对候选操作进行聚类，从而在搜索空间中形成语义连贯的区域。

3a. 提取原子决策变量。为了区分 A 中的候选操作，我们将每个操作表示为一个 d维二元原子特征向量$\mathbf{z}(a) \in \{0,1\}^d$，其中每个维度对应一个特定原子操作组件是否存在。这些原子是区分不同操作的最小单位，有助于实现增量式、可解释的决策（**R1**）。随后，如第4.2节所述，这些原子特征将组合成更高层次、便于人类理解的决策变量 Z_t。

3b. 原子决策变量分组。 虽然静态原子特征是区分系统操作的有用内部判别器，但它们有两个局限：（1）等价性掩盖——差异极大的原子特征集合可能产生功能上相似的输出；（2）交互忽略——有意义的差异只有在联合考虑特征时才会显现，即其语义含义是组合式涌现的。为解决这一问题，我们旨在构建分组后的决策变量，使其既是有用的功能性 判别器，又对用户保持语义上可解释，从而以有意义的方式揭示可用的动作空间（**R1**，**R2**）。在第2步所得簇的基础上，我们为每个簇提取最具特征性的特征。对于簇 C内的一组原子特征 g，定义

$$p_{\mathrm{in}}(g) = \frac{\#\{\, a \in C \mid g \in \mathbf{z}(a) \,\}}{|C|}, \quad p_{\mathrm{all}}(g) = \frac{\#\{\, a \in \mathcal{A} \mid g \in \mathbf{z}(a) \,\}}{|\mathcal{A}|}, \tag{4}$$

其中，A 是可能系统操作的（有限）集合。然后，我们计算提升度：

$$\mathrm{lift}(g, C) = \frac{p_{\mathrm{in}}(g)}{p_{\mathrm{all}}(g)}. \tag{5}$$

提升度大于 1 表示 g 在该簇内的出现频率高于根据其全局频率所预期的频率，因此它是一组具有代表性的特征。最后，为了捕捉特征之间的隐式依赖，我们计算经验共现

$$p(z_Z = 1 \,|\, z_g = 1) = \frac{\#\{\, a \in \mathcal{A} \mid z_g(a) = 1 \text{ and } z_Z(a) = 1 \,\}}{\#\{\, a \in \mathcal{A} \mid z_g(a) = 1 \,\}}, \tag{6}$$

其中，$z_g(a)$ 是一个二元指示变量，表示查询 a 是否包含特征组 g；$z_Z(a)$ 是一个二元指示变量，表示该查询是否包含特征 Z。该量表示在 g 出现的条件下， Z存在的概率，从而识别隐式包含的变量。

4. 决策变量排序。 令 M_t为在步骤 t 后仍与用户澄清相一致的意图集合，并令 $p_t(m)$ 为当前在 M_t 上的信念分布。根据要求

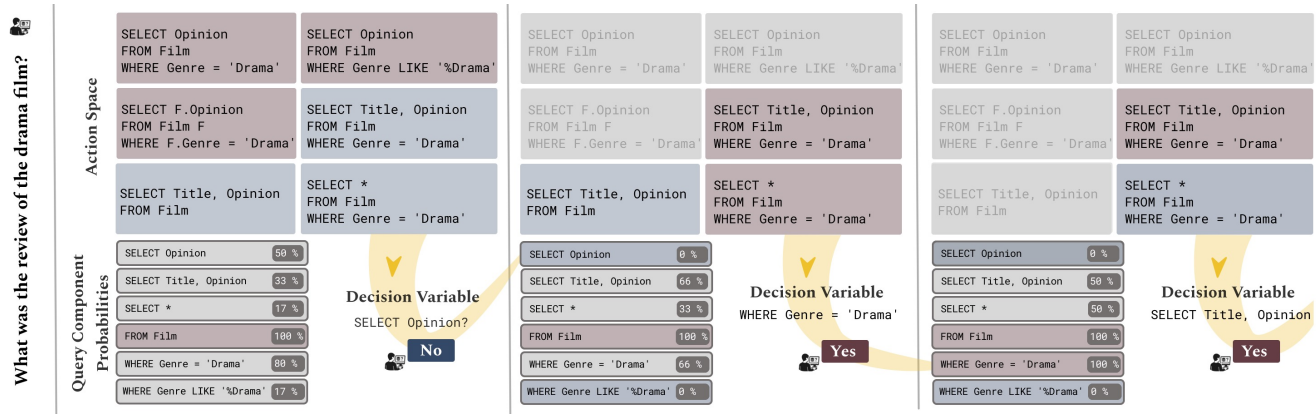

图 4：将第 5 节提出的算法应用于文本到 SQL，通过迭代候选过滤和查询填充。初始候选集通过针对算法提出的决策变量收集用户反馈来迭代过滤，直到候选集仅包含一个查询；也就是说，直到用户确定最符合其意图的查询。

R3，应寻求能够最大程度降低意义后验中预期语义不确定性的澄清。这正是信息增益 [27] 背后的原则。具体而言，对于每个候选决策变量 Z，时刻 t 的期望信息增益为：

$$\mathrm{IG}_t(Z) = H(p_t) - \sum_{v \in \mathcal{V}(Z)} P_t(Z = v)\, H(p_t(\cdot \mid Z = v)), \tag{7}$$

其中，$H(p_t) = -\sum_{m \in \mathcal{M}_t} p_t(m) \log p_t(m)$ 表示香农熵，$P_t(Z = v) = \sum_{m \in \mathcal{M}_t} p_t(m)\, \mathbb{1}\{Z(m) = v\}$. 以期望信息增益为优化准则，我们选择下一个需要澄清的决策变量如下：

$$Z_t^* = \arg\max_Z \mathrm{IG}_t(Z). \tag{8}$$

5. 过滤与重新聚类。用户针对决策变量 Z_t^* 提供用户反馈后，我们过滤 A_t，仅保留符合所述偏好的操作，并相应地更新信念分布 p_t。随后对缩减后的候选集 $\mathrm{A}_{t+1} \subset \mathrm{A}_t$ 重新聚类，并从步骤 2 重新开始。该循环持续进行，直至候选集中仅剩一个操作。

6 算法应用

文本到 SQL 部分中，我们将语用修复算法实例化用于文本到 SQL：用户提出自然语言问题，再将其转换为可执行数据库查询。我们在图3中展示算法的每个步骤，并在图4中展示更高层次的迭代式澄清过程。在这一领域，我们根据输入话语，从大语言模型中采样不同的 SQL 解释，以生成动作空间（参见步骤1，图3）。为了在功能上对系统操作进行聚类，我们使用 执行相似度作为相似性标准；如果两个 SQL 查询在某个测试数据库 D 上运行时产生相似的输出，我们就认为它们在功能上相似（参见步骤2，图3）。请注意，在这种设置中，功能等价是近似的：当且仅当两个 SQL 查询在所有可能的数据库上都产生相同的输出时，它们才严格等价 [19]。为了构建决策变量，我们将每个查询解析为抽象语法树，并对其进行编码，将子句元素（例如 GROUP BY 列_x 或 SELECT 列_y）编码为二元特征。算法其余部分与应用领域无关；对于给定的二元原子特征向量，我们完全按照第5节所述进行分组、提取和排序判别特征（参见图3中的步骤3至5）。

向其他领域的可迁移性。如上所示，提取相关特征并依据功能相似度对其进行聚类需要针对具体任务进行调整，而后续的决策变量排序与选择则以与任务无关的方式作用于生成的原子特征向量。因此，该算法也适用于更广泛的自然语言任务：在这些任务中，系统输出可分解为可比较的原子组件，并能反映某种功能解释。当输出遵循形式语法且具有简单的功能评估时，例如 SQL，这种分解很直接；但对于非结构化或自由形式的生成，分解就颇具挑战。在后一种情况下，已有研究证明，针对具体任务构建分类体系是有效的；这类分类体系能够明确并衡量那些会导致输入解释分歧的组件。例如，在开放领域问答中，[17, 24]输入歧义可以系统地分解为若干离散维度，而不同解释会在这些维度上产生分歧，例如实体、时间和地理位置的具体程度，从而即使没有形式语法，也能实现层次消歧。同样，用于处理自然语言数据可视化歧义的工具 [13, 15, 37] 也会利用此类分类体系，识别数据轴和视觉欠指定之处，因为它们可能导致非预期结果。

除了提取有意义的特征之外，我们的方法还依赖于所提取的原子组件对目标用户具有可解释性，因为用户需要直接针对所提取的决策变量提供澄清。在我们的文本到 SQL 实例化中，我们将 SQL 子句呈现为决策变量，从而使澄清主要可供熟悉 SQL 的用户进行。然而，诸如通过自然语言问题 [22, 24, 28] 来表示决策变量等策略已被证明有助于普通用户理解更为复杂的区分。

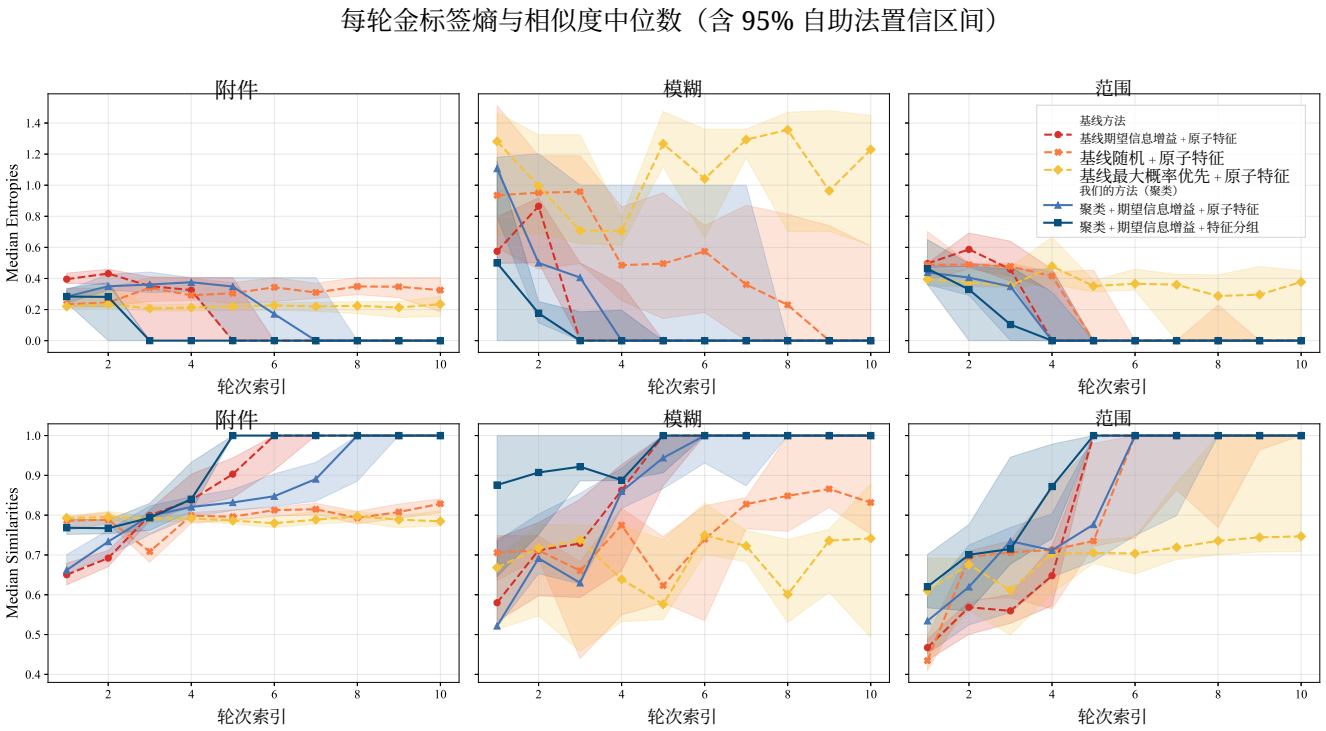

图5：AMBROSIA 数据集中三种歧义类型的每轮金标签熵中位数（上）和功能输出相似度中位数（下）。与基线方法（红色/橙色虚线）相比，基于聚类的方法（蓝色实线）能更快地消除不确定性，并在更少的轮次内收敛到功能上连贯的假设。

7 定量评估

我们首先对第5节中概述的用于文本到 SQL 的算法进行简要定量评估，以突出其在最少的用户交互（**R3**）内区分语义不同的程序（**R2**）的能力。

设置。 我们在 AMBROSIA 数据集[35]上评估算法。该文本到 SQL 数据集旨在涵盖多种类型的歧义，具体区分了范围 歧义和附件 歧义，以及模糊性。对于每个样本，我们通过在高温度下[4]使用含歧义的查询提示 GPT-4o [30] $N = 50$ 次[3]，生成一组候选查询 A。我们使用 Spider SQL 解析器 [42]从每个生成的查询中提取抽象语法树。随后，将 AST 中的每个值编码为二元特征，得到二元特征矩阵 $\{\mathbf{z}(a)\}^N$。我们通过在 AMBROSIA 提供的测试数据库上执行每个查询，并使用 all-MiniLM-L

6 2 -v

[34]计算基于嵌入的输出相似度，构建输出相似度矩阵[16], S。我们使用层次聚类对输出进行聚类，因为该算法具有确定性，可让我们精确指定簇的数量。由于数据集为每个样本提供两个（或更多）金标准解释，我们还可以手动将每个生成的候选查询分配给其中一个高层级金标准意图。

[3]我们会从候选集中剔除语法无效的候选查询，因为它们不属于动作空间 $\mathcal{A}$。因此，实际考虑的候选项数量为 ≤ 50。[4]我们采用 0.7，在生成 SQL 的语法正确性与生成程序的多样性之间取得平衡。

作为消歧基线方法，我们实现了：（1）随机选择决策变量；（2）贪心选择决策变量，即优先选择按当前后验概率最高的取值进行划分的决策变量；以及（3）不使用功能聚类的期望信息增益。我们将这些方法与基于聚类的算法的两个变体（原子决策变量与分组决策变量）进行比较：选择一个金标签作为用户意图，并选择与该意图一致的决策变量取值。交互算法会在所有剩余程序在功能上等价时终止。

结果。 图5展示了在交互式文本到 SQL 修复过程中，不同澄清策略如何影响多轮消歧的进展速度。上排报告候选含义分布的金标签熵中位数，这是衡量残余语义不确定性的直接指标。下排追踪剩余候选集中的功能相似度中位数，反映留存查询在执行空间中的聚集程度。

在三种歧义类型中，所提出的两种基于聚类的策略始终优于基线方法。它们实现了更陡峭的熵下降，在更少的轮次内达到较低的不确定性水平，并且更早地在交互过程中获得更高的分支内功能相似性。这一行为符合我们的设计需求 **R3**：优先选择能够最大程度降低语义不确定性的澄清，从而最大限度地减少所需的修复轮次。特别是，在分组特征上运行期望信息增益能够迅速压缩假设空间——通常压缩至一个

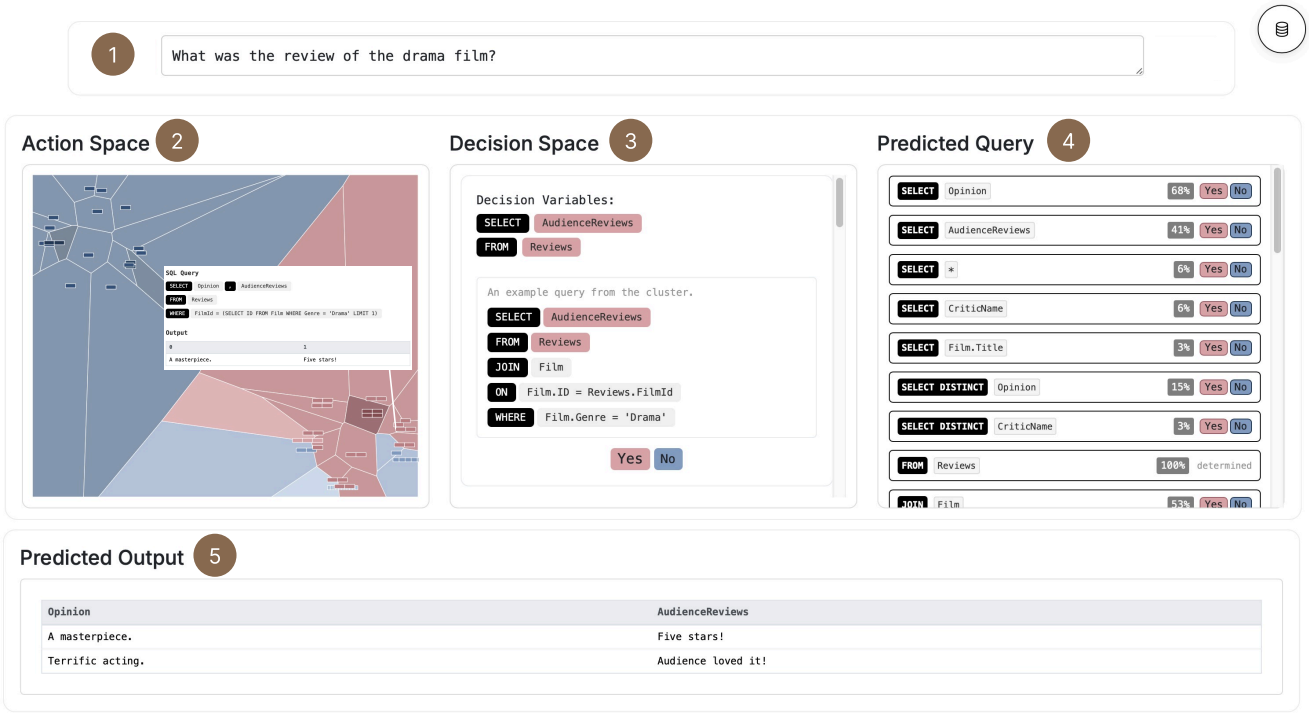

图6：可视化交互界面由用户输入/话语栏（1）、三个主要分析视图组成，即用于探索的动作空间 **（2）、用于决策的**决策空间 **（3）、用于确认任务的**预测查询 **（4），以及**预测输出 **视图（5），后者显示数据库的预测结果。**

功能上同质的集合——在三到五轮内即可形成，而基线方法可能需要更多交互才能达到相当的确定性。结果表明，呈现语义上有意义且信息量高的决策变量，不仅能加快欠明确话语的语义落地，还能生成在功能上更加一致的候选集，从而更快、更有把握地消解用户意图。

8 可视化界面

虽然原子决策变量可能具有预测性或信息量，但它们往往对应不易解释的潜在因素。此外，单独呈现决策变量并不能让用户清楚了解可用的动作空间 A。为应对这些尚存的挑战，我们构建了一个可视化界面，旨在：(1) 通过将剩余候选集 M_t 在每个决策步骤 t 中以语义聚类的形式可视化，呈现系统的可能动作空间 A（**R1**）；(2) 呈现分组后的决策变量和隐式包含变量，使用户决策更易于解释（**R2**），这些变量如第5节所述；以及 (3) 以最少的迭代支持高效的意图澄清（**R3**），同时记录决策及其对最终查询的影响。

我们指出，该界面主要是作为一种研究工具构建的，旨在了解用户在文本到 SQL 中进行话语消歧时的工作流程。虽然它并非直接面向大语言模型的一般终端用户，但我们可以设想，某些独立的界面组件或我们的见解，或许能为未来自然语言界面中的话语消歧设计提供参考。我们还在局限性部分的第11.2节对此作了进一步讨论。

8.1 设计依据

如图6所示，我们的可视化交互界面分为三个部分：用户话语位于页面顶部，中央区域展示三个查询优化组件，页面底部则显示中间预测结果表。三个用于查询优化的核心组件，即动作空间、决策空间和预测查询，旨在满足第4.3节中提出的系统需求。动作空间 的主要目的是帮助用户探索替代解释，并手动筛选出最相关的查询。决策空间 通过建议决策变量来协助用户，以最高效的方式消除话语含义的歧义。最后，预测查询 支持追溯所做决策的来源。此外，该视图还概览了根据用户决策得出的最终查询中最可能出现的原子特征。由于这三个视图彼此关联，用户在一个视图中的交互会触发其他视图的相应更新。特别是，动作空间 中的查询筛选会影响后续建议的决策变量集合。接受或拒绝某个决策变量后，系统会筛选出包含或排除特定特征的查询，并更新相应视图。下文将介绍支持界面这些核心组件易用性的设计考量。

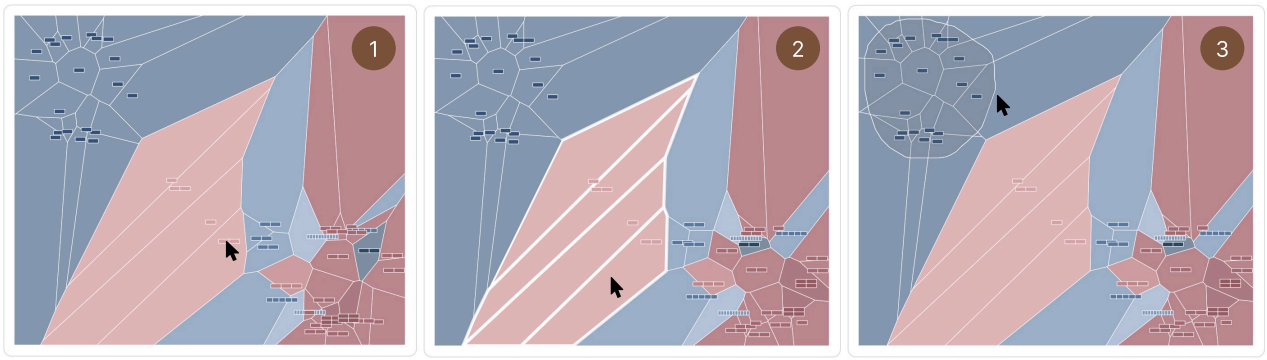

图7： 动作空间 **是对以图形符号呈现并由沃罗诺伊单元包围的 SQL 查询进行的 UMAP 投影。它们的颜色对应聚类索引，其中红色和蓝色的不同深浅分别表示包含和不包含这些决策变量的查询。此视图的主要目的是支持数据探索。用户可以通过点击单个查询图形符号（1）、一簇查询图形符号（2），或使用套索（3）选中一组查询图形符号，来筛选感兴趣的查询。**

查询可视化。 SQL 查询是界面的主要构建模块。在 PleaSQLarify 中，查询有两种表示形式。如图 1 所示，默认情况下，查询以原子特征列表的形式可视化，其中每个特征由一个关键字（黑色背景的文本片段，例如 SELECT）和一个值（浅灰色背景的文本片段，例如 Opinion）构成。除了这种原子特征可视化方式外，我们还采用一种更紧凑的表示形式，以概览可能动作空间 A。这种紧凑形式旨在突出不同查询所产生输出之间的差异。因此，查询以图形符号表示，该符号模拟从数据库检索到的输出表（行数和列数按比例缩放，最大尺寸为 50 px ）。图形符号仅用于强调输出之间存在差异；因此，我们不希望用户查看精确的行数和列数，因为对于较大的表而言，这很难做到。

全局颜色编码。 该界面的主要目的是向用户清晰地传达决策变量，并说明接受或拒绝这些变量会如何影响接下来可能的选择。因此，我们采用全局二元颜色编码，以清楚呈现决策的影响。我们用红色而非浅灰色表示充当当前决策变量的原子特征。此外，我们使用红色色阶为查询（即查询图形符号）着色，这些 SELECT AudienceReviews 义的决策变量后被筛除；不同深浅的红色代表算法确定的不同簇（见第5节）。选择红色，是因为先前研究表明，这种颜色可作为相关性的信号，传达刺激很重要、值得关注这一信息 [3]。我们使用蓝色色阶为不包含当前决策变量的查询（即查询图形符号）着色。

8.2 视觉与交互组件

现在，我们将讨论界面的三个核心视图所采用的视觉呈现和交互方式：动作空间、决策空间和预测查询。

动作空间。 动作空间 旨在概览所有可能的系统操作，即大语言模型生成的查询

这些查询是针对给定用户话语生成的，同时也使用户能够按相关性筛选查询。如第8.1节所述，每个查询都通过一个图形符号表示，该符号模仿其在数据库中的输出。图形符号在二维平面上的位置由 UMAP 投影算法 [25]应用于查询的距离矩阵而确定；距离矩阵的计算方法将在第5节中说明。如第8.1节所述，若查询图形符号包含当前显示在决策空间 中的决策变量（见下一段），则将其标为红色，否则标为蓝色。此外，我们使用不同的深浅色调来突出显示数据库结果相似的查询簇。我们将每个图形符号置于一个沃罗诺伊单元内，以提高簇的可读性。这是可视化交互式标注界面中用于展示类别边界的常见设计选择 [4], ，并将周围多边形的颜色设置为与查询图形符号相同 [1]。

我们采用多种交互方式来支持查询探索。用户将鼠标悬停在查询图形符号上时，会显示一个工具提示，其中展示该查询的原子特征和输出表（示例见图6）。用户探索动作空间 并找出可能符合其预期含义的查询后，我们允许他们将范围缩小到最相关的查询。如图7所示，用户可以通过三种不同的交互方式筛选查询。如果查询符合用户的理解，可点击相应的图形符号将其选中。将鼠标悬停在沃罗诺伊单元上时，相应的簇会以白色多边形边界高亮显示。要选择一个簇，需要在点击该簇中的某个沃罗诺伊单元时按住*Shift* 键。最后，用户还可以使用套索功能来选取所需的一组查询图形符号。

决策空间。 决策空间 的目标是引导用户逐步做出决策，以便以最高效的方式消歧其话语。为此，此视图会列出信息增益最高的决策变量，其信息增益由第5节介绍的算法确定。即使决策变量能带来较高的信息增益，它们也未必能体现与用户意图相关、最有助于消歧的特征，因为可能存在许多种意图。因此，用户每次查看一个（分组后的）决策变量，但可以通过

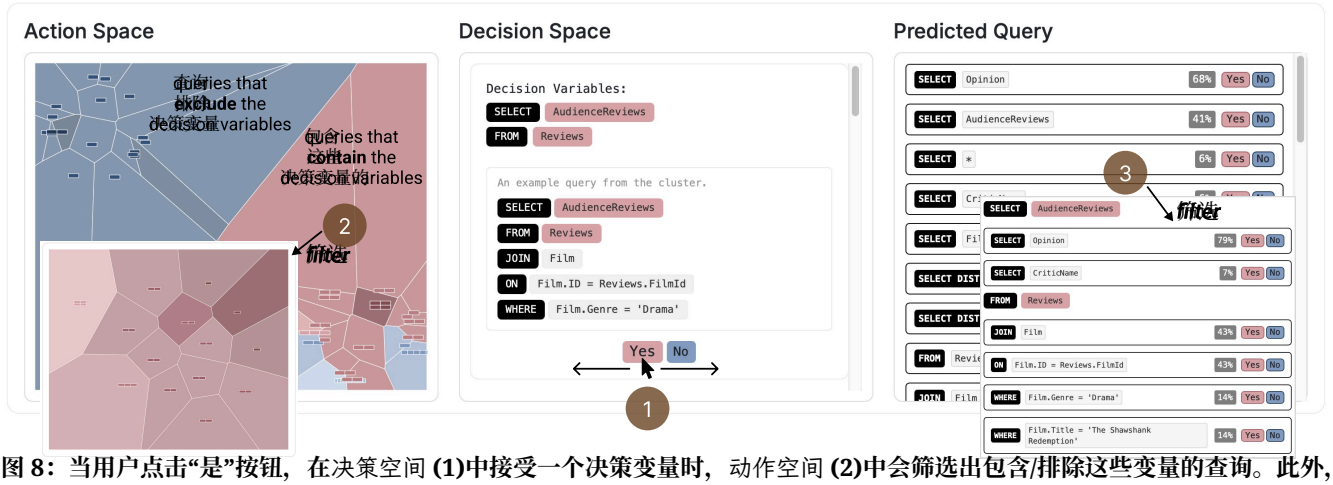

图 8：当用户点击“是”按钮，在决策空间 (1)中接受一个决策变量时，动作空间 (2)中会筛选出包含/排除这些变量的查询。此外，预测查询 面板会更新，显示可能出现在最终查询中的原子特征(3)。

点击键盘上的向上和向下箭头键，或使用相应面板的滚动功能，逐一浏览备选方案。在标红的决策变量下方，我们会显示一个包含该变量的示例查询。此示例有助于理解和诠释给定变量，尤其是在该变量由多个原子特征构成时。该示例查询采用第8.1节介绍的全局设计依据进行展示，即以原子特征列表的形式呈现，并将决策变量标为红色。此外，我们使用低不透明度的红色叠加层，标识并突出显示示例查询中若用户选择该决策变量便会隐式纳入最终查询的特征（例如，图 9中的 JOIN Film 特征，决策 x_{+1}）。

将鼠标悬停在决策变量上时，动作空间中包含这些特征的查询会高亮显示。将鼠标悬停在示例查询上时，相应的查询图形符号会高亮显示。最重要的交互是二元决策：选择某个变量，从而筛选出包含该变量的查询；或拒绝该变量，从而筛选出排除该变量的查询。用户点击查询示例下方显示的“是”或“否”按钮，即可作出二元决策。如图 8 所示，点击其中一个按钮后，动作空间中会筛选出包含/排除这些变量的查询，并根据新的查询样本计算新的决策变量。此外，预测查询 面板（将在下一段介绍）会更新，显示可能出现在最终查询中的原子特征。每项决策都可以通过点击返回 按钮撤销，如图 9 所示。或者，用户也可以跳过建议的决策变量，使用上下方向键或滚动条探索其他变量。

预测查询。 预测查询 旨在支持对所做决策的溯源。系统会向用户说明其所做的决策，以及可能会包含在最终查询中的原子特征。如图 9 所示，原子特征的可视化方式与另外两个面板相同。与其他面板不同的是，对于那些被认为很可能出现、但尚未得到用户确认的原子特征，会在其周围显示边框。原子特征的可能性根据动作空间中当前显示的查询样本计算得出。每当创建新的查询样本时，无论是通过在动作空间中选择查询，还是通过在决策空间中选择决策变量，都会筛选原子特征（概率 > 0）并更新其概率。

在此面板中，我们采用与决策空间中类似的联动与刷选方法。用户可以将鼠标悬停在某个原子特征上，所有包含该特征的查询都会在动作空间中高亮显示。此外，用户可以通过点击“是”或“否”按钮做出相同的二元决策，以筛选动作空间。系统将基于新的查询样本计算新的决策变量，并将其列在决策空间中；原子特征将被筛选，其概率也会在此面板中更新。

9 用户研究

通过我们的用户研究，我们考察人们如何使用PleaSQLarify来消歧话语，并在文本到 SQL的应用案例中评估语用修复的实际效果。如第4.3节所述，语用修复旨在通过简洁、信息丰富且清晰的交互，帮助用户理解并在可用的动作空间内开展工作。我们将这一目标围绕两个关键问题展开：

- **理解可能动作空间**：该系统是否帮助用户识别并区分大语言模型对某一话语的可能解释?
- **在动作空间中导航**：决策变量是否能让用户以既高效又可理解的方式在该空间中移动?

综合来看，这些方面决定了语用修复能否成功支持在复杂技术领域中以交互方式对交际意图进行语义落地的过程。

9.1 参与者

本研究招募了 12 名计算机科学研究生及专业人士（年龄：$M = 27.3$, $SD = 1.8$；性别：4 名女性，8 名男性）。由于预期用户需要理解并与大语言模型生成的

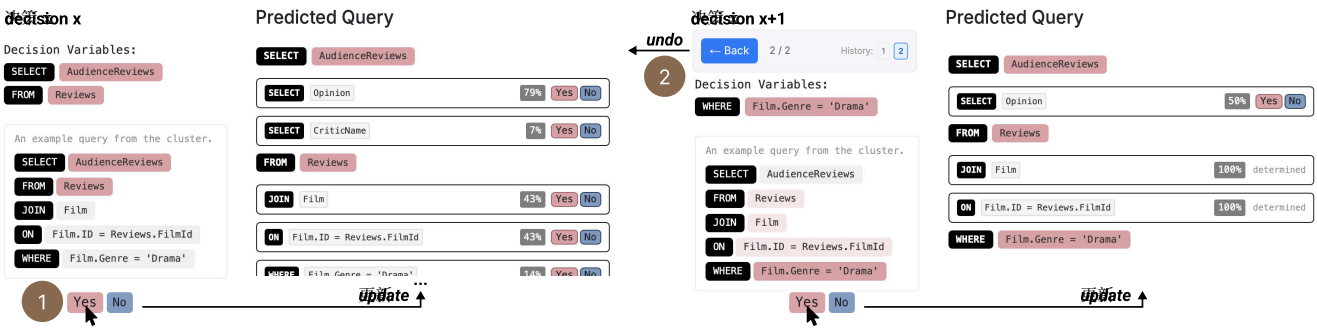

图 9：每当用户做出一项决策 (1)，预测查询 中的原子特征数量就会减少。具体而言，只有包含在筛选后的查询中的特征才会保留。我们会为选定的决策变量着色，并更新其余变量的概率。如果某项特征属于最终查询的概率为 100%，则会将其标记为已确定。在决策空间 (2)中，点击显示的返回 按钮即可撤销已做出的任何决策。

SQL 查询，因此我们要求每位参与者对 SQL 有一定的基本了解。9 名参与者表示曾接触过 SQL 查询的编写和解读（中级），另有 3 名参与者表示经常进行这类工作（高级）。所有参与者均表示曾使用大语言模型进行代码生成或调试。

9.2 流程

每次用户研究持续 45 至 60 分钟，由四个主要部分组成。八场研究在线下进行，另外四场因参与者时间有限而远程进行。远程研究中，我们请参与者在完成任务时共享屏幕。每场研究均录音并转录，以供后续分析。我们征得了每位参与者的知情同意，并向他们支付了 25 美元作为报酬。

介绍（*5* 分钟）。欢迎参与者后，我们简要介绍了研究流程，并在需要时协助他们在设备上安装 PleaSQLarify。随后，我们收集了参与者的人口统计信息，以及他们的 SQL 专业知识和使用大语言模型编写代码的经验。

系统引导（*15* 分钟）。为了让参与者熟悉系统功能，我们让他们观看一段 5 分钟的系统介绍视频，视频涵盖了所有相关系统组件。接着，我们让参与者在一个测试样例上试用这些功能；该样例并非任务中展示的样例。具体而言，我们确保参与者知道如何访问测试数据库、如何与动作空间中的数据点交互，以及如何使用决策空间和预测查询面板中的子句进行导航。

任务（*25*分钟）。参与者表示已能熟练使用系统的各项功能后，我们请他们使用 PleaSQLarify 完成 5 项难度各异的任务。为了解参与者在解题过程中的行为和推理方式，我们请他们边思考边大声说出自己的想法。参与者在解题时可随时提问。如果参与者对某项任务的回答不够明确，主持研究的作者会请其进一步说明。

反馈（*10* 分钟）。最后，参与者回答了有关系统总体效用的问题，并将其与标准语言模型查询进行比较，同时对各个组件提供反馈。值得注意的是，这包括将决策空间中分组且经过优先级排序的决策变量，与预测查询面板中按顺序排列的原子决策变量基线进行比较。此外，参与者还填写了系统可用性量表（SUS）问卷，以评估系统的可用性。在填写问卷期间和之后，参与者可以口头表达对其系统使用体验的任何其他定性反馈。

9.3 任务

数据。我们使用第7节定量评估中的 AMBROSIA 样本设计了五项任务。每条 AMBROSIA 话语至少有两种高层语义解释，并提供了对应的标准 SQL 查询。每个 AMBROSIA 样本都包含一个小型数据库，约有 5 张表，每张表有 < 10 行和若干列。为减轻认知负担，我们选择的所有样本都来自同一领域（电影制作）。我们从“模糊”和“附件”两类歧义中各选取了两个样本，并从“范围”歧义类型中选取了一个样本。“模糊”类查询主要包含由措辞不精确导致的较基本的列歧义，而“附件”和“范围”类型则更具挑战性——无论是消歧所需的推理，还是生成的 SQL 程序的长度和复杂度，都是如此。表1概述了研究任务及用户的任务完成情况。

子任务。为配合用户研究的目标，我们将任务分为三个部分：(1) 话语解释：给参与者提供一个有歧义的 AMBROSIA 话语和数据库，并请他们口头描述一种合理的解释或输出。(2) 探索：随后，参与者查看备选项及其聚类，以了解它们之间的 a) 功能差异（即输出有何不同）和 b) 结构差异（即哪些 SQL 子句造成了这些差异）。(3) 导航：最后，参与者需要将候选集缩小到最符合其原始话语解释的一项。

表1：参与者的任务完成情况。

	参与者编号			P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
	SQL 专业知识			A	I	I	I	A	A	I	I	I	I	I	I
Task	歧义类型	复杂度	子任务												
《低俗小说》的评论是什么 《低俗小说》？	列模糊性	Low	2a	■	■	■	■	■	■	■	■	■	■	■	■
			2b	■	■	■	■	■	■	■	■	■	■	■	■
			3	■	■	■	■	■	■	■	■	■	■	■	■
这部电影的评分是多少有莱昂纳多的电影迪卡普里奥?	列模糊性	媒介	2a	■	■	■	■	■	■	■	■	■	■	■	■
			2b	■	■	■	■	■	■	■	■	■	■	■	■
			3	■	■	■	■	■	■	■	■	■	■	■	■
给我看看剧本和有截止日期的编辑 **截至 21.04.03。**	附件	High	2a	■	■	■	■	■	■	■	■	■	■	■	■
			2b	■	■	■	■	■	■	■	■	■	■	■	■
			3	■	■	■	■	■	■	■	■	■	■	■	■
显示所有动作电影和浪漫喜剧片，时长 2 小时。	附件	媒介	2a	■	■	■	■	■	■	■	■	■	■	■	■
			2b	■	■	■	■	■	■	■	■	■	■	■	■
			3	■	■	■	■	■	■	■	■	■	■	■	■
提供哪些电影类别每个电影节？	范围	High	2a	■	■	■	■	■	■	■	■	■	■	■	■
			2b	■	■	■	■	■	■	■	■	■	■	■	■
			3	■	■	■	■	■	■	■	■	■	■	■	■

任务完成

■ 已解决
■ 部分匹配（找到更好的查询）
■ 部分匹配（未找到原始查询）
■ 未解决

SQL 专业知识

I 中级
A 高级

子任务

2a 识别解释簇
2b 理解语义簇之间的差异
3 前往话语解释

10 项结果

10.1 一般说明

任务完成。 表1显示，使用我们的系统后，参与者在不同 SQL 专业知识水平下的总体任务完成率达到 84.4%[5] 。重要的是，每项任务都至少完成过一次。子任务2a要求参与者描述动作空间簇中的样例。除 P3、P9 和 P12 因时间原因在完成任务4后选择退出外，大多数参与者都完成了各项任务中的这项子任务。子任务2b要求参与者解释各簇之间的语义差异。这在任务3中尤其困难，因为差异源于一个细微的筛选条件；在任务5中也很困难，因为该任务涉及许多连接和复杂的筛选要求。[6] 对于子任务3，约半数参与者（7/12）在至少一项任务中难以找到自己最初的话语解释。不过，在提示他们考虑备选方案后，他们能够得出其他令人满意的最终解释，如表1中的橙色部分所示。

[5]所有 180 项任务的平均完成率，部分解决方案也计为正确。[6]其中一种实现需要检查每个电影节是否都出现了该类别，并通过聚合逻辑统计和筛选电影节出现次数。

系统可用性。 图10中的系统可用性量表结果表明，参与者认为该系统总体上令人愉悦且相对易于使用（M = 5.08 采用 7 点李克特量表评分），各组件的整合也很好（ M = 5.83 ）。与此一致，参与者评论说，该界面“用起来很棒”（P4），工作流程也“很流畅”（P2）。大多数参与者认为，经过短暂探索后，他们就能自信地与系统交互（M = 5.17）。虽然少数参与者提到最初存在学习曲线（P6、P7、P11），但他们认为这段适应期很短，不会削弱对系统整体可用性的印象。积极的系统可用性量表评分表明，PleaSQLarify 不仅支持任务求解，还提供了一种参与者愿意采用的交互模式，例如用于识别意外行为（P6），或在增强用户控制的同时支持查询编写（P10）。

10.2 理解可能的动作空间

发现 10 话语解释 与传统语言模型查询相比，参与者认为该系统在帮助他们识别并探索话语的多种解释方面表现出色（ M = 5.92 采用 7 点李克特量表）。这表明，与传统大语言模型查询的单响应范式不同，动作空间 视图和聚类

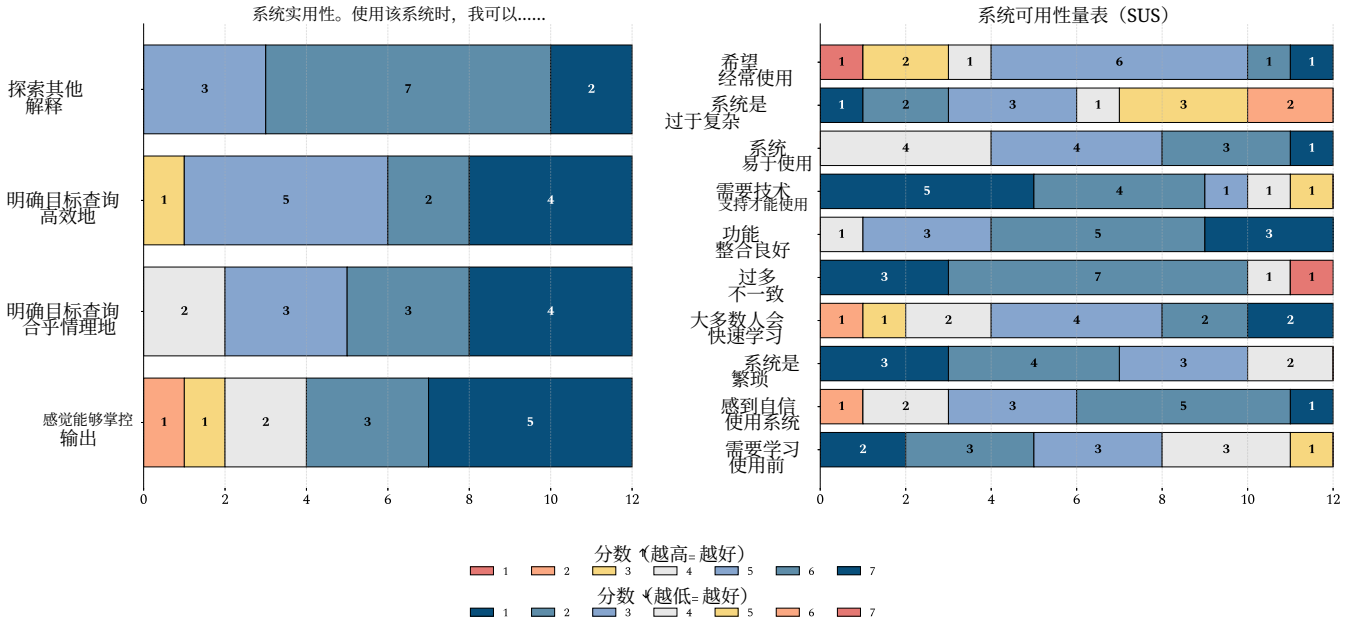

图10：系统实用性与可用性问卷回复。

有助于展现大语言模型可能解释的多样性。事实上，有几位参与者（P1、P2、P6、P10）明确表示，直到看到替代解释后，他们才意识到原始话语存在歧义。例如，当 P6 探索附着歧义样本 4 时，他们注意到查询中还存在额外的模糊性：“哦，‘持续两小时’，这意味着恰好两小时，还是超过两小时？我之前确实没想过这一点。”这表明，呈现动作空间有助于满足第一项设计需求：以拓展用户认知的方式，让用户看见大语言模型的动作空间。

发现更好的备选方案。 近一半的参与者（5/12）表示，他们发现了自己认为比最初预期查询更有用的备选方案，即使最初预期查询已包含在候选集中也是如此。我们在表1中用黄色标出了这些情况。例如，在任务2期间，P6查看当前输出后意识到，加入标题列会很有帮助。同样，在最终确定选择时，P5注意到分组会让结果看起来更美观；而在任务3的最后一步，P6发现还可以对表格排序，以提升可读性。

查询编写指导。 P2和P10表示，基于输出的方法结合决策变量，帮助他们编写了原本自己不会想到要写的查询。这可能是因为系统将复杂操作拆解为易于理解的决策，并以易于理解的层次呈现。

10.3 浏览决策空间

感知控制 通过将大语言模型在不确定性下的生成转化为明确的决策过程，系统让参与者觉得自己对生成内容有更大的掌控力（M = 5.50 在 7 点李克特量表上的评分，见图10），与传统的语言模型查询相比。大多数参与者都能借助该系统获得自己想要的输出，这也印证了这一点（见表1）。不过，有些人觉得自己有时受到决策算法过多的引导（P3、P7）。例如，P3觉得自己“更多是在确认，而不是在掌控”。还有人觉得原始候选集限制过多（P2、P5、P7、P11）。例如，P5一直想查看所有列以了解整体情况，但候选集中很少提供这一选项。同样，在任务3中，P7评论道：“我原以为会有 *[更多一些]* 列；我想展示关于 *[表]* 的全部内容。”我们将在局限性部分重新讨论这些问题（见第11.2节）。

交互效率。大多数参与者认为，系统能够支持他们高效地澄清查询，表现令人满意（M = 5.67，采用 7 点李克特量表，参见图10）。参与者认为，这主要得益于决策空间中的决策变量优先级排序算法：他们无需重写查询，而是可以逐步引导系统，通常只需几步就能得到满意的解读。与此相印证的是，相较于预测查询面板中的原子决策变量，参与者认为在决策空间面板中与经过优先级排序的分组决策变量交互所需的精力更少（分组：M = 3.25，原子：M = 4.42）。例如，P3 将基于优先级排序的决策过程比作“对查询进行二分搜索，基本上取决于簇大小”。参与者还表示，最初显示的决策变量通常被认为更相关，而之后出现的变量则被描述为“相关性较低”（P6）或“可选的”（P7）。这凸显了我们的算法在呈现语义重要特征方面的价值。

决策可理解性。 系统在使决策易于理解 方面也获得了积极评分（M = 5.75采用 7 点

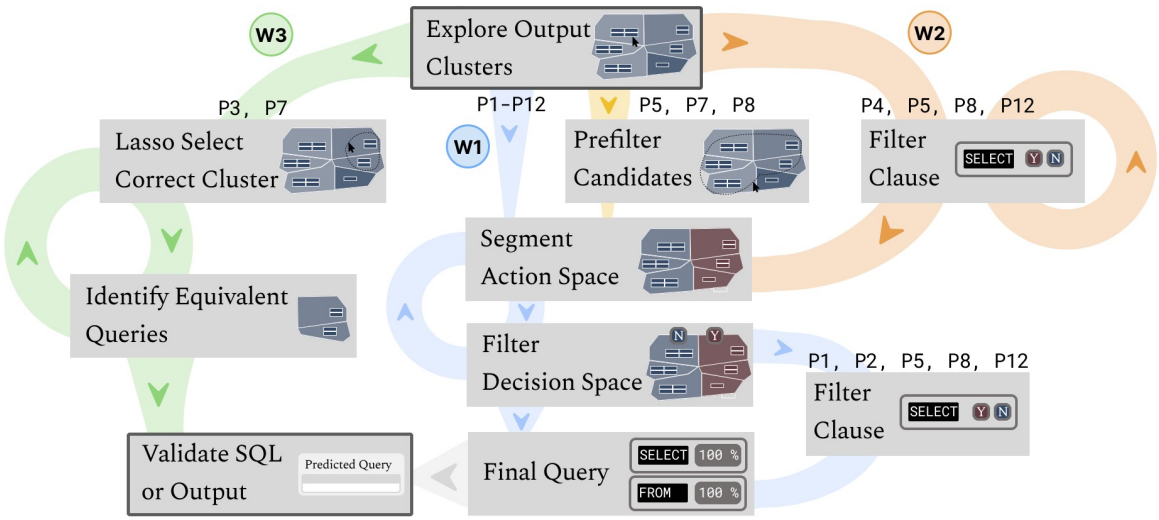

图 11：探索和意图澄清期间观察到的用户交互流程。探索初始动作空间后，用户采用了三种常见的意图澄清方法。在工作流中，用户主要借助决策空间导航至目标查询；而在另一种工作流中，用户则通过筛选预测查询 面板中的各个子句进行导航。最后，还出现了一种基于输出的方法，用户会明确选择所需的输出行为。

李克特量表，参见图10）。尽管分组决策变量和原子决策变量有时难以单独解读（分组：$M = 5.33$，原子：$M = 4.92$），参与者仍可借助示例查询以及决策空间 面板中的隐含决策来理解其上下文。一些参与者指出，优先级排序偶尔会让选择显得不那么有条理，也使他们更难跟踪决策过程中的进展（P1、P2、P5、P8、P12）。因此，在澄清过程接近尾声时，参与者会浏览或依靠预测查询 面板来获得方向。

10.4 使用模式与系统组件分析

由于系统提供了多种探索和浏览候选集的方式，参与者在完成任务时采用了不同的方法。我们在图 11 中展示了这些工作流，并将在下文进行描述。

1. 初始探索。 PleaSQLarify 中的探索始终从动作空间开始，它概览了候选查询的整体分布，并帮助参与者确定方向。他们浏览各个簇，并将鼠标悬停在图形符号上查看工具提示。一些参与者首先排除了他们认为不合理或“幻觉生成的”候选项（P5、P7、P8），另一些参与者（P3、P7）则使用套索选择聚焦相关簇，并生成决策变量划分。这一普遍的起点为三种各有侧重的工作流奠定了基础，每种工作流都强调一种界面组件。

2a. 基于决策的导航。 该工作流最为常见。在该工作流中，参与者首先考察动作空间 中的簇，然后使用决策空间，按分组且排序后的决策变量进行筛选，直至收敛到预期查询。参与者主要依赖决策变量，同时选择在预测查询 面板中的简单子句（P1、P2、P5、P8、P12）说明了动作空间 与决策空间 如何相互补充：动作空间 在对比功能性输出组方面被评为更有用（$M = 5.67$ 对 4.83，参见图12），而决策变量在理解这些差异方面被评为更有用（$M = 5.33$ 对 4.67，参见图12）。参与者通常将鼠标悬停在预测查询 面板上以跟踪进度，主要在开始时（例如固定 SELECT 子句）或接近结束、剩余决策很少时使用子句选择。尽管部分参与者提到了初期学习曲线（P6、P8），许多人最终采用了这一工作流，因为他们认为它在识别语义上有意义的候选组方面最为高效（P4、P7、P10）。

2b. 子句级导航。 这种工作流高度依赖在预测查询 面板中逐个选择子句，通常适用于参与者已经清楚自己想要的确切查询的情况（P5、P12）。这些用户不依赖决策变量，而是直接选择子句并查看工具提示来筛选候选集。正如 P4 所说：“如果你确切知道自己在找什么，工具提示和选择子句会很有用”，而 P5 则称这种工作流是获得预期结果最快的方式。许多参与者认为，与分组决策变量相比，子句级选择更容易理解，因为后者会对变量进行优先级排序（参见图12）；但他们也表示，由于需要在没有优先级排序的情况下滚动浏览长列表，这种方式需要付出多得多的努力（$M = 4.42$ 与 3.25 相比，参见图12）。因此，大多数参与者在处理复杂查询时更偏好决策空间 。例如，P4 和 P11 指出，他们一开始会使用子句筛选，但当导航变得更复杂、可能的 WHERE子句列表难以区分时，就会转而使用经过优先级排序的决策变量。

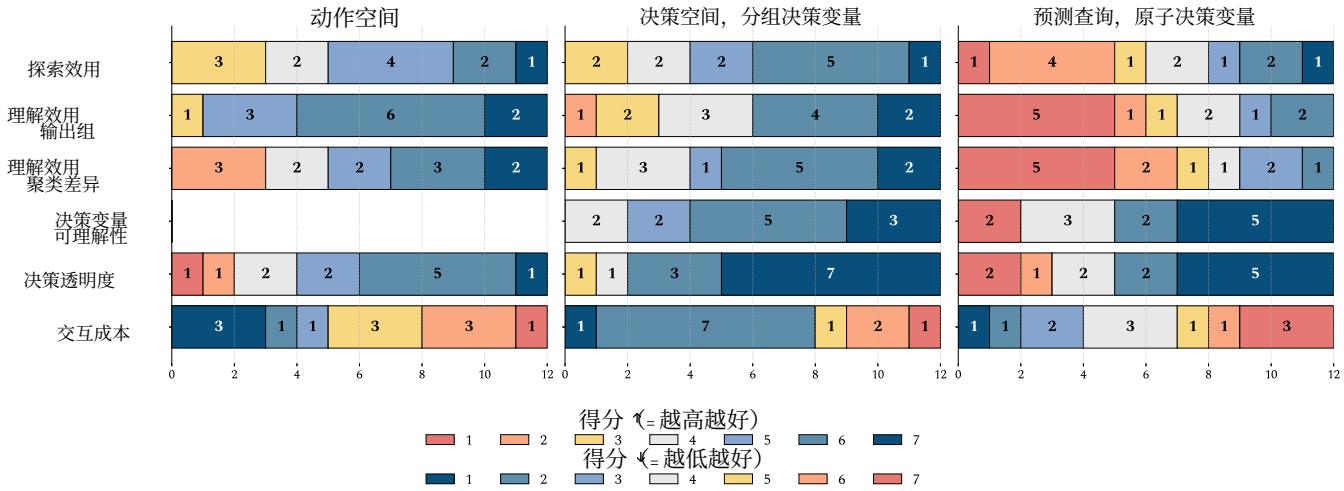

图12：组件效用问卷的回答，按组件分组。

2c. 基于输出的导航。 这种工作流是在动作空间中直接选择有希望的输出来实现的。主要采用这种方式的是对输出形态有所预期、但不太确定对应子句的参与者（P3、P7）。他们使用套索选择来圈定簇、比较等价查询，并在预测输出 视图中验证结果。尽管这种方法要求用户预先了解期望的输出结构，但它能够自然地将等价程序纳入考量；参与者认为，通过子句级导航更难区分这些程序（$M = 4.67$ 与 2.67 相比，参见图12）。随着查询复杂度增加，这种工作流变得更加常见。例如，P2 表示，在早期任务中，对于简单的列歧义，子句级筛选已经足够；但面对更复杂的查询时，由于子句变得过于复杂、难以理解，他们越来越依赖动作空间 分段和输出比较。

3. 验证。无论选择哪种工作流程，参与者都会在分析结束时，通过 预测查询和预测输出面板验证最终查询。

11 讨论

11.1 一般设计启示

我们对 PleaSQLarify 的研究不仅证明了其在文本到SQL修复方面的有效性，也揭示了交互式自然语言界面设计中更广泛的启示。我们观察到的模式表明，界面可以遵循若干原则，更好地支持用户澄清欠指定意图并消除歧义。

(1) **理解歧义**。在文本到 SQL 等结构化自然语言任务中，歧义往往具有多层次性，多个不同的解释可能同时存在，而用户却未意识到这一点。因此，消解此类歧义是一个会在多个交互轮次中逐步展开的决策过程。当界面呈现出更广泛的解释空间时，参与者便能够持续引导系统，使其符合自己的意图；他们也因此能够识别话语的其他含义，并比较这些含义对应的结果。

(2) **以务实的方式澄清**。 当澄清步骤以高效且易于理解的方式缩小候选范围时，参与者便能有效地引导系统。优先处理决策变量有助于提高效率；而将决策置于具体语境中，则能提升可理解性：如果选择以具体示例为依据、与其在查询中的作用相关联，或作为系统代为作出的隐含选择呈现，参与者就最容易理解这些选择。

(3) **适应异质的修复策略**。 第10.4节讨论的三种工作流程体现了动作空间、决策空间和预测查询各自互补的优势，以及参与者灵活运用这些方式的能力。动作空间 支持广泛探索，决策空间 提供结构化且按优先级排列的指导，而预测查询 则帮助参与者监控收敛过程。与其强制采用单一途径，未来系统不妨提供多种策略供用户选择——从开放式澄清、高效定位到输出驱动推理——具体采用哪种策略，可视用户目标和任务复杂度而定。

从更广泛的角度看，语用修复并不局限于 SQL。第3节所述的先验失配问题，以及第4.2节提出的语用修复框架，适用于大多数自然语言界面。通过结合信息性澄清的算法选择与能够揭示备选方案、使后果清晰可见的可视化技术，系统可以将歧义转化为协作式问题求解的资源——即使模型解释与用户意图不一致，也能支持用户控制。

11.2 局限性与未来工作

实验范围。 由于本研究的范围是意图澄清，实验设置（参见第5节）假设用户具有明确定义的意图，且该意图能够合理地映射为有效的 SQL 查询。实验未涵盖用户研究中观察到的其他有趣相关情形，例如用户不确定自己的意图或改变意图。

有限候选池。 为控制延迟和计算成本，候选集固定为初始生成的约 50 个语言模型结果。虽然这或许足以让用户了解模型可能的动作空间，但无法根据用户请求进行更灵活的动态调整。因此，一些参与者表示无法找到自己确切的预期查询（P5、P7、P9）。一个有趣的拓展方向是根据用户输入动态更新候选池；但在当前设置下，这并不可行。如图5所示，在 AMBROSIA 数据集上评估时，我们的算法在采用最优澄清行为[7] 和固定候选池的情况下，需要经过两到八个澄清步骤才能收敛到目标查询。因此，每一步重新采样 50 个模型生成结果，单个澄清序列最多需要 400 次 API 调用，从而带来显著的延迟和过高的计算成本。未来工作可以通过自适应重采样策略解决这一局限，例如在预期多样性降低时，于后续阶段减少采样数量；也可以采用按需方式，让用户在遇到动作空间中代表性不足的区域时选择重新生成候选集。

非技术用户的可用性。 尽管用户研究中的参与者在 SQL 专业知识方面各不相同，但系统仍要求用户具备基本的 SQL 知识才能消除查询歧义，因为用户需要解读子句并评估候选输出。虽然据报告，可视化的输出驱动工作流提升了难以编写的查询的易用性，但非 SQL 熟练用户可能仍难以理解我们的决策变量。整合代码转文本方法或许有助于进一步降低使用门槛 [22, 28]。

终端用户集成。 第8节中介绍的可视化界面主要是一种研究工具，用于跟踪和分析用户在消除文本到 SQL 话语歧义时的工作流，而非供终端用户使用。不过，某些独立元素（例如提取决策变量或简洁展示动作空间概览）可以改造成轻量级小组件，并在对话式界面中重复使用。

可扩展性。 当前设置探索的是小型测试数据库，这类数据库常见于最受欢迎的文本转 SQL 基准测试中 [35, 42]。以这种方式限制数据库规模，既能像第7节那样进行受控的定量评估（因为我们可以轻松枚举可能的金标准意图标签），也让用户研究中的参与者能够在合理的时间内熟悉数据库。对于更大的数据库，我们的方法可能需要更多澄清步骤才能收敛，因为表和列数量增加会扩大合理查询解释的范围。此外，我们提出的基于输出的分析可能会变得成本高昂，也更难以可视化。未来值得探索的方向包括针对大型数据库中较小且信息丰富的子集生成并开展分析，以便在更贴近现实的设置中验证基于输出的查询区分方法的效用。

12 结论

我们介绍了 PleaSQLarify，这是一个将语用修复应用于交互式文本转 SQL 消歧的系统。该系统结合了一种算法，用于选择信息丰富的决策变量以及一个可视化界面，用于展示模型的动作空间、逐步请求用户澄清，并使各轮交互中的更新过程可追溯。在一项有十二位参与者参与的研究中，该系统通过高效且易于理解的引导，帮助用户发现此前未曾预料到的解释，并让他们更能掌控生成输出，从而获得满意的结果。多样化的使用工作流程进一步凸显了支持异质修复策略的重要性。更广泛地说，这项工作展示了如何将自然语言界面中的歧义转化为协作式问题求解的资源，即使模型解释与用户意图不一致，也能增强用户控制。

[7]也就是说，不选错决策变量的取值。

致谢

作者感谢匿名审稿人提出全面且富有建设性的意见。作者还要感谢 Ben Lipkin、Emily Cheng、Raphaël Baur 和 Yanis Schmit 提供的宝贵讨论和反馈。最后，作者感谢 SNF 基金（编号 10003068）和 foryouandyourcustomers 提供的资助。

参考文献

[1] Franz Aurenhammer。1991。Voronoi 图——基础几何数据结构综述。*ACM*计算调查（*CSUR*）23，3（1991），345–405。[2] Adithya Bhaskar、Tushar Tomar、Ashutosh Sathe 和 Sunita Sarawagi。2023。歧义条件下文本到 SQL 生成的基准测试与改进。*arXiv* 预印本 *arXiv:2310.13659*（2023）。[3] Vanessa L Buechner、Markus A Maier、Stephanie Lichtenfeld 和 Sascha Schwarz。2014。红色——细看一番。*PloS one* 9，9（2014），e108111。[4] Jie Chen、Honghui Du、Dairui Liu、Siteng Ma、Brian Mac Namee 和 Ruihai Dong。2025。ActiViz：通过边界可视化理解主动学习中的样本选择。载于第*34*届*ACM*国际信息与知识管理会议论文集。270–279。[5] Kaiwen Chen、Yueting Chen、Nick Koudas 和 Xiaohui Yu。2025。通过自适应弃答实现可靠的文本到 SQL。*ACM*数据管理论文集3，1（2025），1–30。[6] Ziru Chen、Shijie Chen、Michael White、Raymond Mooney、Ali Payani、Jayanth Srinivasa、Yu Su 和 Huan Sun。2023。利用代码语言模型纠正文本到 SQL 错误。载于第*61*届计算语言学协会年会论文集（第*2*卷：短文），Anna Rogers、Jordan Boyd-Graber 和 Naoaki Okazaki（编）。计算语言学协会，加拿大多伦多，1359 1372 10 18653 1 2023 117 –。doi:。/v / . acl-short。[7] Herbert H. Clark 和 Deanna Wilkes-Gibbs。1986。指称是一种协作过程。认知 22, 1 1986 1 39 10 1016 0010 0277 86 90010 7 （），–。doi:。/ [8] - ( ) - Zhongjun Ding 、 Yin Lin 和 Tianjing Zeng。2025。AmbiSQL：用于文本到 SQL 的交互式歧义检测与消解。*arXiv* 预印本 *arXiv:2508.15276*（2025）。[9] Ahmed Elgohary、Saghar Hosseini 和 Ahmed Hassan Awadallah。2020。与解析器对话：通过自然语言反馈进行交互式文本到 SQL。载于第*58*届计算语言学协会年会论文集，Dan Jurafsky、Joyce Chai、Natalie Schluter 和 Joel Tetreault（编）。计算语言学协会，线上，2065–2077。doi:10.18653/v1/ 2020.acl-main.187[10] Avrilia Floratou、Fotis Psallidas、Fuheng Zhao、Shaleen Deep、Gunther Hagleither、Wangda Tan、Joyce Cahoon、Rana Alotaibi、Jordan Henkel、Abhik Singla 等。2024。NL2SQL 已是一个解决了的问题……并非如此！载于*CIDR*。[11] Michael C. Frank 和 Noah D. Goodman。2012。预测语言博弈中的语用推理。科学336，6084（2012），998–998。doi:10.1126/science.1218633[12] Dawei Gao、Haibin Wang、Yaliang Li、Xiuyu Sun、Yichen Qian、Bolin Ding 和 Jingren Zhou。2024。大语言模型赋能的文本到 SQL：基准评测。*VLDB*论文集17，5（2024年1月），1132–1145。doi:10.14778/3641204.3641221[13] Tong Gao、Mira Dontcheva、Eytan Adar、Zhicheng Liu 和 Karrie G. Karahalios。2015。DataTone：管理数据可视化自然语言界面中的歧义。载于第*28*届*ACM*用户界面软件与技术年会论文集（美国北卡罗来纳州夏洛特）*(UIST '15)*。美国计算机协会，美国纽约州纽约市，489 500 10 1145 2807442 2807478 –。doi:。/ 。[14] H. P. Grice。1975。逻辑与会话。载于句法与语义：第 卷：言语行为，Peter Cole 和 Jerry L. Morgan（编）。学术出版社，纽约，41–58。[15] Mert Inan、Anthony Sicilia、Alex Xie、Saujas Vaduguru、Daniel Fried 和 Malihe Alikhani。2025。识别并交互式地细化有歧义的用户目标，以 3

生成数据可视化代码。载于2025年自然语言处理实证方法会议论文集，Christos Christodoulopoulos、Tanmoy Chakraborty、Carolyn Rose 和 Violet Peng（编）。计算语言学协会，中国苏州，25257–25274。doi:10.18653/v1/2025. emnlp-main。[16] Leonard Kaufman 和 Peter J. Rousseeuw。1990。《数据中的群组发现：聚类分析导论》。约翰·威立父子出版公司，纽约。doi:10.1002/9780470316801[17] Gangwoo Kim、Sungdong Kim、Byeongguk Jeon、Joonsuk Park 和 Jaewoo Kang。2023。澄清树：利用检索增强大语言模型回答歧义问题。载于2023年自然语言处理实证方法会议论文集，Houda Bouamor、Juan Pino 和 Kalika Bali（编）。计算语言学协会，新加坡，996–1009。doi:10.18653/v1/2023.emnlp-main.63[18] Hyuhng Joon Kim、Youna Kim、Cheonbok Park、Junyeob Kim、Choonghyun Park、Kang Min Yoo、Sang-goo Lee 和 Taeuk Kim。2024。使语言模型能够明确处理歧义。载于2024年自然语言处理实证方法会议论文集，Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen（编）。计算语言学协会，美国佛罗里达州迈阿密，[19] emnlp-main。Leo Köberlein、Dominik Probst 和 Richard Lenz。 基于图的QSS：一种用于自动线性SQL评分的语义相似度量化图方法。载于《商业、技术与网络数据库系统》（BTW 2025）。德国信息学会，波恩，[20] W - Carolin Lawrence 和 Stefan Riezler。 通过人类老虎机反馈的反事实学习改进神经语义解析器。载于第56届计算语言学协会年会论文集（第1卷：长文），Iryna Gurevych 和 Yusuke Miyao（编）。计算语言学协会，澳大利亚墨尔本，[21] Mengyi Liu 和 Jianqiu Xu。 NLI DB：数据库自然语言界面的系统综述。arXiv 预印本 arXiv:2503.02435 （ 2025 ）。[22] Michael Xieyang Liu 、 Advait Sarkar 、 Carina Negreanu、Benjamin Zorn、Jack Williams、Neil Toronto 和 Andrew D. Gordon。2023。“它想让我说什么”：弥合终端用户程序员与代码生成大语言模型之间的抽象差距。载于2023年计算机系统人因国际会议论文集（德国汉堡）（CHI '23）。美国计算机协会，美国纽约州纽约市，文章598，共31页。doi:10.1145/3544548.3580817[23] Ewa Luger 和 Abigail Sellen。2016。“就像有一个特别糟糕的私人助理”：用户对对话代理的期望与体验之间的落差。载于2016年计算机系统人因国际会议论文集（美国加利福尼亚州圣何塞）（CHI '16）。美国计算机协会，美国纽约州纽约市，[24] Kenneth Aleksander Robertsen 和 Peggy Chi。2025。AmbigChat：用于歧义开放域问答的交互式层次澄清。载于第38届ACM用户界面软件与技术年会论文集（UIST '25）。美国计算机协会，美国纽约州纽约市，文章[25] 。Leland McInnes 、 John Healy 、 Nathaniel Saul 和 Lukas Grossberger。UMAP：均匀流形近似与投影。开放源代码软件期刊3，29（2018），861。[26] Rakesh R Menon、Kun Qian、Liqun Chen、Ishika Joshi、Daniel Pandyan、Jordyn Harrison、Shashank Srivastava 和 Yunyao Li。2025。Fisql：通过丰富的交互式反馈增强文本到 SQL 系统。载于第28届扩展数据库技术国际会议论文集（EDBT 2025）—产业专题。Open- Proceedings.org。[27] Tom M Mitchell。1997。机器学习。第1卷。麦格劳-希尔，纽约。[28] Arpit Narechania、Adam Fourney、Bongshin Lee 和 Gonzalo Ramos。2021。DIY：评估自然语言到 SQL 系统的正确性。载于第26届智能用户界面国际会议论文集（美国得克萨斯州大学城）（IUI '21）。美国计算机协会，美国纽约州纽约市，597–607。doi:10.1145/3397481.3450667[29] Donald A. Norman 和 Stephen W. Draper。1986。《以用户为中心的系统设计：人机交互新视角》。劳伦斯·厄尔鲍姆联合出版公司，美国新泽西州希尔斯代尔。[30] OpenAI。2024。GPT-4技术报告。arXiv:2303.08774 [cs.CL] https://arxiv. org/abs/2303 08774 。Steven T. Piantadosi、Harry Tily 和 Edward Gibson。[31] 2012。歧义在语言中的交际功能。认知122，3（2012），280–291。doi:10.1016 2011 10 004 /j.cognition. . 。Gabriel Poesia 和 Noah Goodman。[32] 2021。语用代码自动补全。AAAI人工智能会议论文集35，1（2021年5月），445–452。doi:10.1609/aaai.v35i1.16121[33] Luyu Qiu、Jianing Li、Chi Su 和 Lei Chen。2025。通过期望信息增益进行交互式文本到 SQL 消歧。arXiv 预印本 arXiv:2507.06467（2025）。[34] Nils Reimers 和 Iryna Gurevych。2019。Sentence-BERT：使用孪生BERT网络进行句子嵌入。载于*2019*年自然语言处理经验方法会议论文集。计算语言学协会。https://arxiv.org/abs/1908 10084 。Irina Saparina 和 Mirella Lapata。[35] . AMBROSIA：将歧义问题解析为数据库查询的基准测试集。载于第三十八届神经信息处理系统会议数据集与基准赛道。https: // openreview.net/forum?id=llFk5U9cEg[36] Irina Saparina 和 Mirella Lapata。2025。先消歧后解析：为语义解析中的歧义消解生成解释。*arXiv* 预印本 *arXiv:2502.18448* (2025)。[37] Vidya Setlur、Melanie Tory 和 Alex Djalali。2019。在可视化分析中推断欠指定自然语言语句。载于第*24*届智能用户界面国际会议论文集(加利福尼亚州 Marina del Ray) *(IUI '19)*。美国计算机协会，纽约州纽约市，40–51。doi:10.1145/3301275.3302270[38] Peter Shaw、Ming-Wei Chang、Panupong Pasupat 和 Kristina Toutanova。2021。组合泛化与自然语言变异：语义解析方法能否兼顾二者？载于计算语言学协会第*59*届年会暨自然语言处理国际联合会议第*11*届会议论文集（第一卷：长篇论文），Chengqing Zong、Fei Xia、Wenjie Li 和 Roberto Navigli（编）。计算语言学协会，线上，[39] acl-long。Dustin A. Smith 和 Henry Lieberman。. 帮助用户理解并从自然语言界面的解释失败中恢复。载于智能与自主系统终端用户交互研讨会，计算机与人类交互会议（*CHI*）*2012*。ACM，美国得克萨斯州奥斯汀。CHI 2012 研讨会。[40] Bing Wang、Yan Gao、Zhoujun Li 和 Jian-Guang Lou。2023。了解自己的未知：处理文本到 SQL 中的歧义问题和未知问题。载于计算语言学协会会议论文集：*ACL 2023*。5701–5714。[41] Tao Yu、Rui Zhang、Heyang Er、Suyi Li、Eric Xue、Bo Pang、Xi Victoria Lin、Yi Chern Tan、Tianze Shi、Zihan Li、Youxuan Jiang、Michihiro Yasunaga、Sungrok Shim、Tao Chen、Alexander Fabbri、Zifan Li、Luyao Chen、Yuwen Zhang、Shreya Dixit、Vincent Zhang、Caiming Xiong、Richard Socher、Walter Lasecki 和 Dragomir Radev。2019。CoSQL：面向跨领域数据库自然语言界面的对话式文本到 SQL 挑战。载于*2019*年自然语言处理经验方法会议暨第九届自然语言处理国际联合会议论文集（*EMNLP-IJCNLP*），Kentaro Inui、Jing Jiang、Vincent Ng 和 Xiaojun Wan （编）。计算语言学协会，中国香港，1962–1979。doi:10.18653/v1/D19- 1204[42] Tao Yu、Rui Zhang、Kai Yang、Michihiro Yasunaga、Dongxu Wang、Zifan Li、James Ma、Irene Li、Qingning Yao、Shanelle Roman、Zilin Zhang 和 Dragomir Radev。2018。Spider：用于复杂跨领域语义解析和文本转 SQL 任务的大规模人工标注数据集。载于*2018*年自然语言处理经验方法会议论文集，Ellen Riloff、David Chiang、Julia Hockenmaier 和 Jun'ichi Tsujii（编）。计算语言学协会，比利时布鲁塞尔，3911–3921。doi:10.18653/v1/D18- 1425[43] J.D. Zamfirescu-Pereira、Richmond Y. Wong、Bjoern Hartmann 和 Qian Yang。2023。为什么 Johnny 不会提示：非人工智能专家如何尝试（并未成功）设计大语言模型提示。载于*2023*年计算机系统人因国际会议(德国汉堡) *(CHI '23)*。美国计算机协会，纽约州纽约市，文章 437，21 页。doi:10.1145/3544548. 3581388[44] Victor Zhong、Caiming Xiong 和 Richard Socher。2017。Seq2SQL：使用强化学习从自然语言生成结构化查询。arXiv:1709.00103 [cs.CL] https://arxiv.org/abs/1709.00103