SEED：通过自动证据生成提升文本到 SQL 的性能和实用性

Janghyeon Yun
Seoul National University , Korea
janghyeon@europa.snu.ac.kr

Sang-goo Lee*†
* Seoul National University , Korea
† IntelliSys , Korea
sglee@europa.snu.ac.kr

(a) 蜘蛛数据集中文本到 SQL 问题的假设

(b) 在 BIRD 数据集中关于 text-to-SQL 问题的假设

(c) 使用 SEED 的文本到 SQL 问题的假设

Fig. 1: 文本到 SQL 问题的假设图示。

***Abstract*—Text-to-SQL 通过将自然语言查询转换为 SQL，使非专家能够从数据库中检索数据。然而，最先进的 text-to-SQL 研究依赖于 BIRD 数据集，该数据集假设证据与问题一起提供。尽管 BIRD 促进了研究进展，但它假设用户具备专业技能和领域知识，这与 text-to-SQL 的基本目标相悖。此外，BIRD 中人为生成的证据存在缺陷，包括证据缺失或错误，这影响了模型的性能。为了解决这个问题，我们提出了 SEED（证据提取和领域知识生成系统），一种自动生成证据的方法，以改善性能并在现实世界场景中提高实用性。SEED 系统地分析数据库模式、描述文件和值以提取相关信息。我们在 BIRD 和 Spider 上评估了 SEED，结果表明在无证据场景下，它显著提高了 SQL 生成的准确性，并且在某些情况下，甚至超越了提供 BIRD 证据的设置。我们的结果表明，SEED 生成的证据不仅弥合了研究与现实世界部署之间的差距，还提高了 text-to-SQL 模型的适应性和鲁棒性。我们的代码可在 https://github.com/felix01189/SEED 找到。**

***Index Terms*—TEXT-to-SQL, TEXT2SQL, NL2SQL, SQL, LLM.**

I. 介绍

从数据库中检索特定数据需要领域知识和 SQL 专业技能。然而，文本到 SQL 通过将用户的自然语言请求翻译为 SQL 查询来缓解这一需求，从而使非专业人员可以轻松地从数据库中检索数据 [1]–[3] 。鉴于其潜力，关于文本到 SQL 的广泛研究已经进行，并且随着大型语言模型（LLMs）的迅速发展，该领域正以前所未有的速度进步。

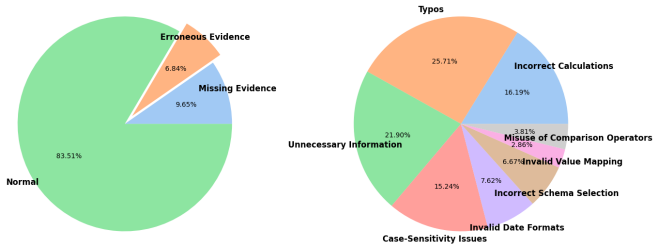


Fig. 2: BIRD 开发集证据错误率（左）和类型（右）。

为了验证这些快速发展的技术的有效性，众多研究人员还开发了基准数据集。早期的数据集如 WikiSQL [4] 和 Spider [5] 为文本到 SQL 的研究铺平了道路，目前 BIRD [6] 数据集在该领域中作为一个重要的基准。

与早期的数据集不同的是，BIRD [6] 数据集通过描述文件提供关于模式和数值的信息，并提供辅助 SQL 生成的证据。每个问题附带的证据包括模式到数值的映射和 SQL 生成所需计算的数学公式。早期的数据集，如 Spider [5] ，假设只提供问题（如图 1 a 所示），而使用 BIRD 证据则假设用户也提供证据（如图 1 b 所示）。然而，这一假设与文本到 SQL 的基本目标相矛盾。

尽管这种假设不现实，大多数基于 BIRD [6] 开发的文本到 SQL 方法，如 CHASE-SQL [7] 、CHESS [8] 和 RSL-SQL [9] ，都使用这些手动提供的证据。事实上，在 BIRD 排行榜上的前 30 项中，除了一项未发表的研究外，其余都使用了证据 [7]–[17] 。因此，将这些最新的文本到 SQL 模型应用于没有此类证据的现实世界场景中，会在学术研究和实际部署之间造成差距，导致性能显著下降。我们的实验证实，现有的文本到 SQL 模型在省略证据时性能会大幅退化。

此外，我们的分析揭示了人类生成的 BIRD 证据中的一些缺陷。如图 2 所示，对开发集（1534 个问题-SQL-证据对）的彻底审查发现，9.65%（148 对）完全缺乏证据，而 6.84%（105 对）包含不正确的证据。这 105 对中的错误包括错误的计算、拼写错误、不必要的信息、大小写敏感性问题、无效日期格式、不正确的模式选择、无效的值映射以及比较运算符的误用。表 I 提供了这些问题的例子。鉴于开发集中大约 7% 包含有缺陷的证据，这些错误为文本到 SQL 模型引入了噪声，可能限制其性能。表 II 比较了手动校正证据前后 CodeS [15] 在 105 个错误对上的性能。这表明错误的证据能显著降低性能。

为了解决这些挑战，我们提出了 SEED（S ystem for E

www.xueshuxiangzi.com

TABLE I: BIRD 开发集的错误样本证明

error type	unnecessary information
question	列出所有元素 with double bond, consisted in molecule TR024.
evidence	double bond refers to bond_type = '='; element = 'cl' 表示氯；element = 'c' 表示碳；element = 'h' 表示氢； 元素 = 'o' 表示氧，元素 = 's' 表示硫；元素 = 'n' 表示氮， element = 'p' 意味着磷，element = 'na' 意味着钠，element = 'br' 意味着溴， element = 'f' 表示氟；element = 'i' 表示碘；element = 'sn' 表示锡； element = 'pb' 意味着铅；element = 'te' 意味着碲；element = 'ca' 意味着钙
revised evidence	double bond refers to bond_type = '=';
error type	case-sensitivity issues
question	How many cards of legalities whose status is 受限的 have text boxes?
evidence	restricted refers to status = "受限" ; have text boxes refers to is Textless = 0;
revised evidence	restricted refers to status = "受限" ; have text boxes refers to is Textless = 0;
error type	incorrect schema selection
question	List down at least five 全名 of superheroes with blue eyes.
evidence	blue eyes refers to colour.colour = 'Blue' WHERE eye_colour_id = colour.id; 名称 of superheroes refers to 超级英雄名;
revised evidence	blue eyes refers to colour.colour = 'Blue' WHERE eye_colour_id = colour.id; 全名 of superheroes refers to 完整名称;

TABLE II: 对 105 个错误配对进行证据纠正前后的性能比较。

	EX %	
	defective evidence	corrected evidence
SFT CodeS-15B	44.76	54.29 (↑ 9.53)
SFT CodeS-7B	44.76	55.24 (↑ 10.48)
SFT CodeS-3B	43.81	51.43 (↑ 7.62)
SFT CodeS-1B	37.14	46.67 (↑ 9.53)

vidence E xtraction and D omain knowledge generation)。SEED 通过分析数据库的架构、描述文件和采样值自动生成证据。通过消除对人工生成证据的依赖，SEED 符合文本到 SQL 的初衷，并弥合了学术研究与实际应用之间的差距。

为了验证 SEED，我们在三种不同的条件下使用多种 text-to-SQL 模型进行了实验：(1) 使用 BIRD 证据，(2) 没有证据，以及 (3) 使用 SEED 生成的证据。我们的研究结果证实，与没有证据的情况相比，SEED 生成的证据提高了 text-to-SQL 的性能，证明了其实际效用。

我们的贡献如下。

- 开发一个自动证据生成系统：我们介绍了 SEED，一个自动生成证据以改进文本到 SQL 的系统。SEED 在手动生成证据不可用的实际环境中增强了文本到 SQL 的适用性。
- 弥合研究与实际部署之间的差距：通过自动化生成证据，SEED 缩小了学术研究与实际应用之间的差距，使先进的文本到 SQL 模型更易于在现实世界中采用。
- 识别使用 BIRD 证据的基本问题：我们强调关于使用 BIRD 证据的不现实假设，揭示其与文本到 SQL 目的的不一致。我们的分析证实，现有的文本到 SQL 模型在没有证据的情况下性能显著下降，突显当前研究与实际可用性之间的差距。此外，我们发现手动生成的证据中存在错误，强调其对模型性能的负面影响。

通过 SEED，我们的目标是克服使用 BIRD 证据的限制，并通过使文本到 SQL 模型在实际场景中更加健壮、实用和有效，促进其广泛采用。

II. 相关工作

A. *BIRD* 数据集和证据

研究 BIRD [6] 数据集的学者强调，理解数据库内容至关重要，因为数据库具有噪声、多样和大规模的特性。他们认为，需要外部知识来改善文本到 SQL 模型在理解数据库值方面的能力。他们将证据分为四种类型：(1) 数值推理知识：执行数学计算所需的专业知识。(2) 领域知识：特定领域的知识。(3) 同义词知识：关于同义词的信息，包括它们的含义和替代表达。(4) 值说明：对数据库值的详细描述。

然而，除了数值推理知识之外，其余三个类别——领域知识、同义词知识和数值说明——可以通过对数据库模式、描述文件和数值样本的详细分析来获得。表 III 提供了这三种证据类型的示例，说明这些知识可以直接从给定的数据库信息中推断出来。

因此，BIRD 提供的证据主要包括用于数学推理的 SQL 相关知识和通过数据库分析提取的领域知识。这表明，许多证据不是外部知识，而是数据库本身固有的信息。

www.xueshuxiangzi.com

TABLE III: 鸟类证据的类别和样本，以及每个证据的信息来源

knowledge type	Domain Knowledge
question	Name the ID and age of patient with two or more laboratory examinations which show their 血细胞比容水平 超过正常范围.
evidence	hematoclit level exceeded the normal range refers to HCT > = 52;
information source	database description file: Laboratory.csv
information	Normal range: 29 < N < 52
knowledge type	Synonym Knowledge
question	How many clients opened their accounts in Jesenik branch were 女性?
evidence	female refers to gender = 'F'
information source	database description file: client.csv or database value: select distinct gender from client
information	F：女性 M:male
knowledge type	Value Illustration
question	Among the 每周发行 accounts, how many have a loan of under 200000?
evidence	frequency = 'POPLATEK TYDNE' stands for weekly issuance
information source	database description file: account.csv
information	"POPLATEK MESICNE" stands for monthly issuance "POPLATEK TYDNE" 代表每周发行 "POPLATEK PO OBRATU" stands for issuance after transaction

B. 文本到 SQL

早期的文本到 SQL 方法是基于规则的，依赖于预定义的模式 [2], [3] 。然而，这些方法对特定数据库高度专业化，限制了它们的泛化能力。一个有代表性的例子是 NaLIR [18] ，它允许用户通过交互式的基于 UI 的修改来改进他们的查询。

随着神经网络的进步，深度学习技术开始被纳入文本到 SQL 模型中 [1]–[3] 。其中最早的方法之一，Seq2SQL [4] ，利用了 Seq2Seq [19] 框架来基于预定义的 SQL 草图预测合适的列和操作符。

Transformer [20] 架构的引入促使了预训练语言模型 (PLMs) 的出现，如 BERT [21] 和 T5 [22] ，这些模型随后被应用于文本到 SQL 任务 [1]–[3] 。一个显著的例子是 BRIDGE [23] ，它采用了基于 BERT 的编码结合指针生成网络 [24] 对 SQL 查询进行解码。

最近，随着闭源大型语言模型 (LLMs) 的兴起，如 GPT-4 [25] 和 Gemini [26] ，以及开源 LLMs 如 LLaMA [27] 和 StarCoder [28] ，大多数最新的文本到 SQL 方法已经采用了基于 LLM 的方法。C3 [29] 是一个基于 ChatGPT 构建的零样本文本到 SQL 方法。DIN-SQL [30] 将 SQL 生成分解为子任务。DAIL-SQL [31] 通过系统实验得出并利用了适合文本到 SQL 的有效提示。MCS-SQL [14] 生成多个候选 SQL 查询，并通过自我一致性 [32] 机制选择最佳的一个。XiYan-SQL [10] 、CHASE-SQL [7] 和 MSc-SQL [33] 使用训练过的选择器从多个候选中选择最优的 SQL 查询。CHESS [8] 将单元测试器整合以验证 SQL 预测。SENSE [16] 和 CodeS [15] 专注于微调较小的模型以在文本到 SQL 任务中实现竞争性表现。E-SQL [13] 通过对给定问题进行自我打磨 [34] 机制来增强 SQL 生成能力。

截至本文撰写时，在 BIRD 排行榜中排名前 30 的模型 [7]–[17] 中，除了一个未发表的方法外，所有方法都依赖于证据，这突出了其在最近文本到 SQL 方法中的影响。

III. 方法论

SEED 根据上下文输入的长度采用两种不同的架构: SEED $_{\text{gpt}}$ 和 SEED $_{\text{deepseek}}$ 。当支持较长的上下文输入时，我们使用图 3 a 中所示的 SEED $_{\text{gpt}}$ 架构，该架构处理整个模式作为输入。相反，在上下文输入容量有限的情况下，我们采用图 3 b 中所示的 SEED $_{\text{deepseek}}$ 架构，其中模式被概括以仅保留与问题相关的信息。

当使用 GPT-4o 作为基础模型时，采用图 3 a 中的架构。然而，对于通过其 API 支持最大 8,192 个标记的 DeepSeek-R1 [35] ，更适合使用图 3 b 中的架构。SEED 框架由三个关键组件组成: Schema Summarization、Sample SQL Execution 和 Evidence Generation。

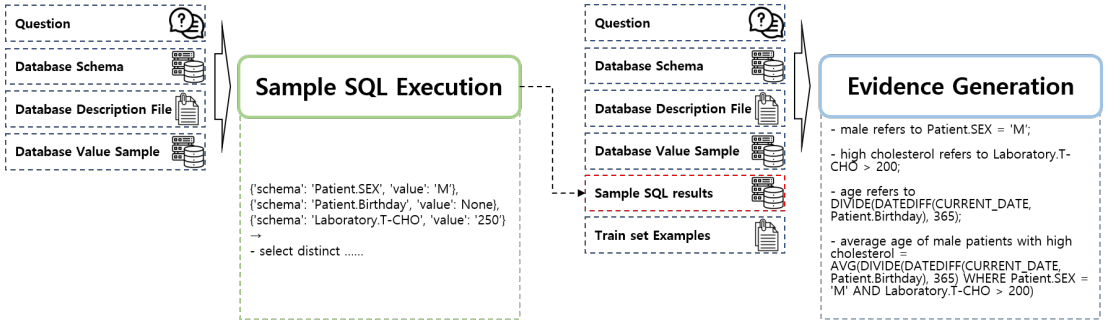
A. 模式总结

最近的研究 [11] 指出，当利用具有强大推理能力的 LLMs 进行文本到 SQL 任务时，将模式修剪作为预处理步骤实际上可能会降低 SQL 生成性能。基于这一见解，SEED 在生成证据时不修剪模式，而是将整个模式作为 LLM 的输入。

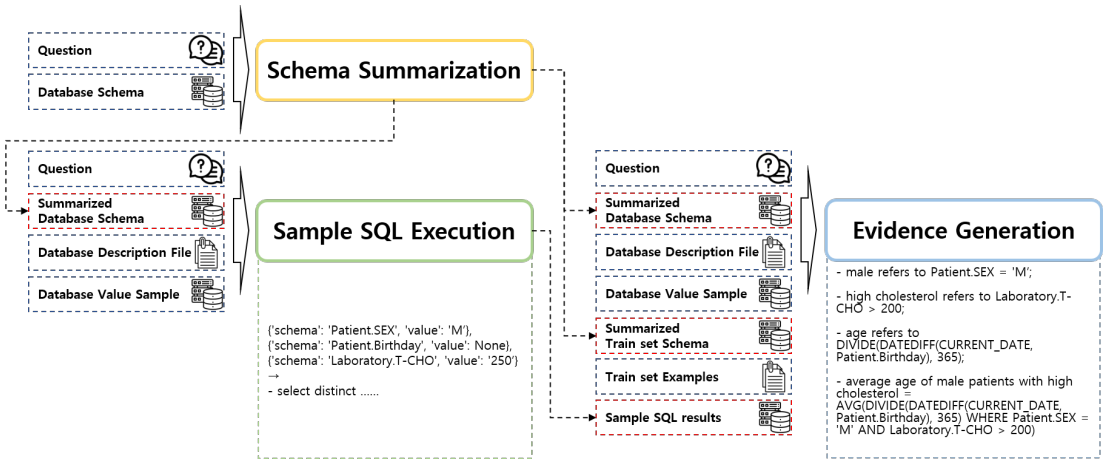
然而，可能会出现输入标记数量超过大语言模型允许的限制的情况。例如，DeepSeek-R1 [35] API 设置的最大标记限制为 8,192。为了解决此类情况，我们在方法中引入了模式摘要。

在生成证据之前，SEED 会将问题与模式进行比较，并从模式中去除无关的信息。这个预处理步骤确保了具有令

www.xueshuxiangzi.com



(a) SEED GPT 的结构



(b) SEED DeepSeek 的结构

Fig. 3: SEED 的结构

牌限制的模型，例如 DeepSeek-R1，仍然可以作为 SEED 的基础模型。

B. SQL 执行示例

考虑在没有领域知识的情况下，由人类生成 SQL 的过程。当在没有领域知识的情况下生成 SQL 时，他们有时可以通过将问题中的关键字与数据库模式进行比较来推断其含义。然而，在许多情况下，他们需要执行示例 SQL 查询来检查数据库中的值，以充分理解问题的意图。

例如，考虑一个包含术语“Fremont”的问题。在没有执行查询的情况下，我们无法明确“Fremont”是指一个县、地区还是城市。用户通常会运行一些示例查询以检查数据库，从而确定正确的列。同样，SEED 通过系统地执行示例 SQL 查询来生成领域知识，从而模拟这一人类过程。

首先，SEED 从问题中提取代表数据库列和值的关键词。然后，它将提取的列与其对应的值进行配对，并为每对生成和执行样例 SQL 查询。

提取的样本数据如下：无论数据类型如何，都提取唯一值；在字符串类型的情况下，还使用 LIKE 操作符和编辑距离提取相似值。提取的样本数据用于构建下一步的提示。

一旦获得样本 SQL 结果，SEED 会生成证据来辅助 SQL 生成。证据生成提示的结构如下：指令、训练集示例、样本 SQL 结果、数据库模式和问题。

为了构建有效的少样本示例，SEED 使用基于相似度选择的方法从训练集中识别出相似的问题。首先，SEED 从训练集中识别出与给定查询最相似的问题，然后从同一数据库中检索另外四个相关的问题。我们使用 all-mpnet-base-v2 [36] 作为嵌入模型来比较相似性，并使用余弦相似度作为相似性度量。

IV. 实验

A. 数据集

BIRD [6] 数据集通过引入噪声和大规模数据弥合了文本到 SQL 研究与实际应用之间的差距。它由 95 个数据库（33.4 GB）和 37 个领域的 12,751 个文本到 SQL 对组成，独特地提供了数据库描述文件和证据。

Spider [5] 数据集包含 200 个数据库、10,181 个问题和 5,693 个复杂的 SQL 查询，特点是包含了 JOIN、GROUP BY 和 EXISTS 等高级模式。由 11 名学生经过 1,000 多小时创建，确保了多样化和多表查询以实现更好的泛化。

B. 评估指标

基于精确 SQL 匹配来评估文本到 SQL 模型可能导致误判，因为不同的查询在语义上可能是等价的。为了解决这个问题，BIRD 和 Spider 使用执行准确度（EX），它比较的是执行结果而不是语法。此外，BIRD 引入了有效效率

www.xueshuxiangzi.com

TABLE IV: 在没有证据的情况下，文本到 SQL 模型在 BIRD 数据集上的性能下降以及通过 SEED 改进。

	dev EX %				dev VES %		
	w/o evidence	w/ evidence	w/ SEED $_{gpt}$	w/ SEED $_{deepseek}$	w/o evidence	w/ evidence	w/ SEED $_{gpt}$
CHESS $_{IR+CG+UT}$ (GPT-4o-mini)	54.69	63.04 (↑ 8.35)	56.26 (↑ 1.57)	54.11 (↓ 0.58)	56.40	66.64 (↑ 10.24)	58.34 (↑ 1.94)
CHESS $_{IR+SS+CG}$ (GPT-4o-mini)	49.61	60.43 (↑ 10.82)	54.82 (↑ 5.21)	53.65 (↑ 4.04)	51.41	64.67 (↑ 13.26)	56.75 (↑ 5.34)
RSL-SQL (GPT-4o)	54.50	65.78 (↑ 11.28)	58.28 (↑ 3.78)	58.15 (↑ 3.65)	56.02	68.31 (↑ 12.29)	60.32 (↑ 4.3)
SFT CodeS-15B	44.39	55.35 (↑ 10.96)	56.78 (↑ 12.39)	57.69 (↑ 13.3)	47.22	56.84 (↑ 9.62)	58.95 (↑ 11.73)
SFT CodeS-7B	41.92	54.76 (↑ 12.84)	56.52 (↑ 14.60)	56.58 (↑ 14.66)	46.42	57.50 (↑ 11.08)	59.65 (↑ 13.23)
DAIL-SQL (GPT-4)	35.46	56.32 (↑ 20.86)	51.63 (↑ 16.17)	53.19 (↑ 17.73)	36.68	57.70 (↑ 21.02)	53.58 (↑ 16.90)

TABLE V: 在 Spider 数据集上使用 SEED 提高文本到 SQL 模型的性能

	dev EX %		test EX %	
	w/o SEED	w/ SEED $_{gpt}$	w/o SEED	w/ SEED $_{gpt}$
SFT CodeS-15B	85.6	87.3 (↑ 1.7)	85.0	86.4 (↑ 1.4)
SFT CodeS-7B	86.4	86.8 (↑ 0.4)	84.7	86.1 (↑ 1.4)
C3 (ChatGPT)	82.0	86.6 (↑ 4.6)	80.1	84.0 (↑ 3.9)

评分（VES），它通过考虑执行时间来扩展 EX，奖励更高效的查询以获得更高的分数。

C. 基线方法

为了评估 SEED 生成证据的有效性，我们从 BIRD 和 Spider 排行榜中选择了最先进的 text-to-SQL 模型，这些模型在撰写本文时具有公开的实现。选定的模型包括 CHESS [8] 和 RSL-SQL [9]，它们代表了排行榜上的最新方法；CodeS [15]，一个经过微调的 text-to-SQL 模型；DAIL-SQL [31] 和 C3 [29]，这些模型体现了上下文学习（ICL）的方法。

1) 国际象棋： CHESS [8] 框架将文本到 SQL 的过程视为一个由四个关键组成部分组成的多代理系统：信息检索器（IR）——检索相关的数据库值和描述，模式选择器（SS）——过滤掉不必要的模式元素，候选生成器（CG）——生成多个 SQL 候选，单元测试器（UT）——执行单元测试以评估候选的 SQL 查询。此外，CHESS 根据计算预算等约束条件提供配置这些代理的指导，使其成为文本到 SQL 任务的多功能框架。

2) RSL-SQL： 最近的研究 [11] 表明，尽管模式链接通常用于减少噪音和计算开销，但它可能引入潜在风险。为了减轻这些缺点，RSL-SQL [9] 的研究人员提出了一种双向模式链接方法。首先，链接完整模式并用于生成初步的 SQL 查询。然后，提取查询中引用的模式元素。通过结合前向和后向模式链接，RSL-SQL 实现了一个稳健且有效的模式链接过程。

3) 代码 S： CodeS [15] 解决了文本到 SQL 研究中的关键挑战，例如对闭源大型语言模型（如 GPT-4 和 Gemini）的依赖，这些模型在隐私方面存在问题且 API 成本高。为了解决这些问题，CodeS 微调了 StarCoder [28] 以更好地适应文本到 SQL 任务。该模型结合了 RESDSQL [37] 的模式链接方法，并通过 BM25 索引和最长公共子串方法的结合增强了数据库值引用。尽管它的规模相对较小，仅有多达 150 亿个参数，但 CodeS 在效率和效果上优于使用 GPT-4 的 DIN-SQL [30]。

随着文本到 SQL 的上下文学习（ICL）的兴起，DAIL-SQL [31] 研究团队强调了系统提示工程的重要性。他们的研究探讨了几个关键方面，包括如何在提示中格式化数据库架构、检索有效的少样例案例和在提示中呈现案例。通过优化这些因素，DAIL-SQL 通过精心设计的提示策略实现了出色的 SQL 生成性能。

4) C3： C3 [29] 是一种零样本的文本到 SQL 方法，旨在解决少样本方法的限制，这些方法通常需要超过 10,000 个标记，以及零样本模型相比微调替代方案较低的性能。该模型包括三个阶段：清晰提示（CP），通过零样本提示指令建立模式链接；带提示校准（CH），通过错误分析识别 ChatGPT 中的偏差（如过度选择列或检索过多值）并通过提供具体提示（例如 “仅在必要时使用 COUNT(*)，LEFT JOIN 或 OR”）来减轻这些偏差；以及一致输出（CO），通过执行多次运行和应用投票机制来减少 LLM 的固有随机性。

D. 实现细节

SEED $_{gpt}$ 包含两个阶段：样本 SQL 执行和证据生成，未进行模式总结。样本 SQL 执行阶段使用 gpt-4o-mini，而证据生成阶段使用 gpt-4o。

另一方面，SEED $_{deepseek}$ 执行两次模式总结：一次是针对与问题对应的数据库，另一次是针对训练集示例。SEED $_{deepseek}$ 使用 DeepSeek-R1 [35] 作为所有阶段的基础模型。

E. 结果

1) 鸟： 表格 IV 展示了在 BIRD 中 SQL 生成性能在三种不同设置下的表现：无证据提供、有人工标注证据（来自 BIRD）提供以及由 SEED 生成的证据提供。第一个关键的观察是，没有人工生成的证据会显著降低所有模型的性能。特别是，DAIL-SQL [31] 在 EX 中的表现差距最大，差异为 20.86 %。即使是差距最小的模型，在缺少证据时仍显示出 8.35 % 的性能下降。这突出了研究设置与现实世界应用之间的巨大差距，在现实中没有证据的情况下会严重降低 SQL 生成的性能。

接下来，在比较 SEED 生成的证据与无证据条件时，我们确认在大多数模型中 EX 和 VES 都有所改善，有些情况下甚至超过了 BIRD 证据设置下的性能。我们观察到 EX 最多提高了 17.73 %，而 VES 最多提高了 17.69 %，

www.xueshuxiangzi.com

这表明 SEED 不仅提高了准确性，还生成了更高效的查询。这些结果确认了 SEED 生成的证据有助于弥合研究与实际应用的差距，并提高 text-to-SQL 模型的实际可用性。

2) 性能退化分析： 然而，我们注意到，对于 CHESS $_{\text{IR+CG+UT}}$ [8]，SEED $_{\text{deepseek}}$ 的表现略逊于无证据条件。对于 CodeS [15] 模型，SEED $_{\text{deepseek}}$ 生成的证据表现优于 SEED $_{\text{gpt}}$。然而，对于 CHESS $_{\text{IR+CG+UT}}$ 模型，使用 SEED $_{\text{deepseek}}$ 证据使性能下降，相较于完全不使用证据。为了调查这种差异的原因，我们分析了 CHESS 模型和 SEED 生成的证据，得出以下观察结果：(1) SEED 使用人工生成的 BIRD 证据作为小样本示例。然而，它生成了在示例中不存在的信息或略微改变了证据的格式。特别是，如表 **??** 所示，最显著的区别是 SEED 提供了关于连接的额外信息。(2) 早期的研究，如 CodeS 和 DAIL-SQL [31]，采用了一种简单的方法，即将证据与问题直接连接起来。相比之下，更近期的模型如 CHESS 在每个代理中多次主动结合证据，且 CHESS 的提示不仅包含如何利用证据的直接指导，还明确指出证据中包含的信息类型。

基于这些观察，我们假设像 CHESS 这样的最新模型通过提示工程被优化为适应人类生成的 BIRD 证据的格式。为了验证这个假设，我们通过 DeepSeek-V3 [38] 删除与连接相关的信息，其最显著的差异，修订了 SEED 证据。然后，我们使用这个修订过的 SEED（SEED $_{\text{revised}}$）证据评估了 CHESS 和 CodeS 的性能。如表 VI 所示，在使用 SEED $_{\text{revised}}$ 相比于 SEED $_{\text{deepseek}}$ 时，CHESS $_{\text{IR+CG+UT}}$ 在 EX 上显示了 1.37 % 的提升，而 CodeS 的性能下降了 1.3 %。这证实了我们的假设，即修改 SEED 证据使其类似于人类生成的证据会提高 CHESS $_{\text{IR+CG+UT}}$ 的性能，同时降低 CodeS 的性能。这些发现突显了未来在优化证据格式方面进行研究的必要性，基于模型如何利用证据。

3) 蜘蛛： 为了进一步验证 SEED 的稳健性，我们在 Spider 数据集上进行了实验。我们比较了两种情境：未提供证据和提供 SEED 生成的证据。为了评估，我们选择了一个微调模型（CodeS [15]）和一个基于 ICL 的模型（C3 [29]）。由于 Spider 没有数据库描述文件，我们使用 DeepSeek-V3 [38] 生成了这些文件。表格 V 的结果显示，所有模型的性能都有所提高，这证实了 SEED 生成的证据可以提升不同数据集和方法论中的 SQL 生成能力。

V. 结论

在这项研究中，我们确认了依赖人工整理的证据在学术研究与实际应用之间产生了差距。为了解决这一限制，我们提出了 SEED，一个能够在无需人工干预的情况下自主生成证据的系统。通过实验，我们证明了在缺乏证据的实际场景中，SEED 显著提高了 text-to-SQL 方法的表现。

通过弥合学术研究与实际应用在文本到 SQL 系统中的差距，我们的研究有助于使现有和未来的方法更适用于现实世界的场景。我们相信，我们的发现将引导未来的研究朝着更实用和可扩展的解决方案发展，最终促进文本到 SQL 在实际应用中的更广泛采用。

References

[1] B. Qin, B. Hui, L. Wang, M. Yang, J. Li, B. Li, R. Geng, R. Cao, J. Sun, L. Si, F. Huang, and Y. Li, "A survey on text-to-sql parsing: Concepts, methods, and future directions," 2022. [Online]. Available: https://arxiv.org/abs/2208.13629

[2] L. Shi, Z. Tang, N. Zhang, X. Zhang, and Z. Yang, "A survey on employing large language models for text-to-sql tasks," 2024. [Online]. Available: https://arxiv.org/abs/2407.15186

[3] Z. Hong, Z. Yuan, Q. Zhang, H. Chen, J. Dong, F. Huang, and X. Huang, "Next-generation database interfaces: A survey of llm-based text-to-sql," 2024. [Online]. Available: https://arxiv.org/abs/2406.08426

[4] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," 2017. [Online]. Available: https://arxiv.org/abs/1709.00103

[5] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. R. Radev, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *2018 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP 2018)*. Google; Facebook; Bloomberg; Salesforce; Apple; Amazon; Baidu; Grammarly; Naver Labs Europe; FWO; KU Leuven, Dept Comp Sci; CVTE; Ebay; Microsoft; Naver Line; Oracle; Polya; Huawei; Duolingo; Figure Eight; Nuance, 2018, pp. 3911–3921, conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, BELGIUM, OCT 31-NOV 04, 2018.

[6] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo, X. Zhou, M. Chenhao, G. Li, K. Chang, F. Huang, R. Cheng, and Y. Li, "Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 42 330–42 357. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/83fc8fab1710363050bbd1d4b8cc0021-Paper-Datasets_and_Benchmarks.pdf

[7] M. Pourreza, H. Li, R. Sun, Y. Chung, S. Talaei, G. T. Kakkar, Y. Gan, A. Saberi, F. Ozcan, and S. O. Arik, "Chase-sql: Multi-path reasoning and preference optimized candidate selection in text-to-sql," 2024. [Online]. Available: https://arxiv.org/abs/2410.01943

[8] S. Talaei, M. Pourreza, Y.-C. Chang, A. Mirhoseini, and A. Saberi, "Chess: Contextual harnessing for efficient sql synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2405.16755

[9] Z. Cao, Y. Zheng, Z. Fan, X. Zhang, W. Chen, and X. Bai, "Rsl-sql: Robust schema linking in text-to-sql generation," 2024. [Online]. Available: https://arxiv.org/abs/2411.00073

[10] Y. Gao, Y. Liu, X. Li, X. Shi, Y. Zhu, Y. Wang, S. Li, W. Li, Y. Hong, Z. Luo, J. Gao, L. Mou, and Y. Li, "Xiyan-sql: A multi-generator ensemble framework for text-to-sql," 2024. [Online]. Available: https://arxiv.org/abs/2411.08599

[11] K. Maamari, F. Abubaker, D. Jaroslawicz, and A. Mhedhbi, "The death of schema linking? text-to-sql in the age of well-reasoned language models," 2024. [Online]. Available: https://arxiv.org/abs/2408.07702

[12] T. Ren, Y. Fan, Z. He, R. Huang, J. Dai, C. Huang, Y. Jing, K. Zhang, Y. Yang, and X. S. Wang, "Purple: Making a large language model a better sql writer," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2024, pp. 15–28.

[13] H. A. Caferoğlu and Özgür Ulusoy, "E-sql: Direct schema linking via question enrichment in text-to-sql," 2024. [Online]. Available: https://arxiv.org/abs/2409.16751

[14] D. Lee, C. Park, J. Kim, and H. Park, "Mcs-sql: Leveraging multiple prompts and multiple-choice selection for text-to-sql generation," 2024. [Online]. Available: https://arxiv.org/abs/2405.07467

[15] H. Li, J. Zhang, H. Liu, J. Fan, X. Zhang, J. Zhu, R. Wei, H. Pan, C. Li, and H. Chen, "Codes: Towards building open-source language models for text-to-sql," *Proc. ACM Manag. Data*, vol. 2, no. 3, May 2024. [Online]. Available: https://doi.org/10.1145/3654930

[16] J. Yang, B. Hui, M. Yang, J. Yang, J. Lin, and C. Zhou, "Synthesizing text-to-SQL data from weak and strong LLMs," in *Proceedings of the 62nd Annual Meeting of the Association*

www.xueshuxiangzi.com

TABLE VI: 在 Spider 数据集上使用 SEED 提升文本到 SQL 模型的性能。

	dev EX %			dev VES %		
	w/o SEED	w/ SEED $_{\text{deepseek}}$	w/ SEED $_{\text{revised}}$	w/o SEED	w/ SEED $_{\text{deepseek}}$	w/ SEED $_{\text{revised}}$
CHESS $_{\text{IR+CG+UT}}$	54.69	54.11 (↓ 0.58)	55.48 (↑ 0.79)	56.40	55.82 (↓ 0.58)	57.39 (↑ 0.99)
SFT CodeS-15B	44.39	57.69 (↑ 13.30)	56.39 (↑ 12.00)	47.22	59.33 (↑ 12.11)	58.44 (↑ 11.22)
SFT CodeS-7B	41.92	56.58 (↑ 14.66)	55.80 (↑ 13.88)	46.42	59.42 (↑ 13.00)	58.42 (↑ 12.00)

for Computational Linguistics (Volume 1: Long Papers), L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 7864–7875. [Online]. Available: https://aclanthology.org/2024.acl-long.425/

[17] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang, "The dawn of natural language to sql: Are we fully ready?" *Proc. VLDB Endow.*, vol. 17, no. 11, p. 3318–3331, Aug. 2024. [Online]. Available: https://doi.org/10.14778/3681954.3682003

[18] F. Li and H. V. Jagadish, "Nalir: an interactive natural language interface for querying relational databases," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 709–712. [Online]. Available: https://doi.org/10.1145/2588555.2594519

[19] I. Sutskever, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[20] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[21] Kenton, J. Devlin, M.-W. Chang, Toutanova, and L. Kristina, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, no. 2. Minneapolis, Minnesota, 2019.

[22] Raffel, Colin, Shazeer, Noam, Roberts, and e. a. Adam, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[23] X. V. Lin, R. Socher, and C. Xiong, "Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 4870–4888. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.438/

[24] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1073–1083. [Online]. Available: https://aclanthology.org/P17-1099/

[25] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, and e. a. Ilge Akkaya, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[26] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, and e. a. Radu Soricut, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[27] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, and e. a. Yasmine Babaei, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[28] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, and e. a. Chenghao Mou, "Starcoder: may the source be with you!" 2023. [Online]. Available: https://arxiv.org/abs/2305.06161

[29] X. Dong, C. Zhang, Y. Ge, Y. Mao, Y. Gao, lu Chen, J. Lin, and D. Lou, "C3: Zero-shot text-to-sql with chatgpt," 2023. [Online]. Available: https://arxiv.org/abs/2307.07306

[30] M. Pourreza and D. Rafiei, "Din-sql: Decomposed in-context learning of text-to-sql with self-correction," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 36 (NEURIPS 2023)*, ser. Advances in Neural Information Processing Systems, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023, 37th Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, DEC 10-16, 2023.

[31] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou, "Text-to-sql empowered by large language models: A benchmark evaluation," *PROCEEDINGS OF THE VLDB ENDOWMENT*, vol. 17, no. 5, pp. 1132–1145, JAN 2024.

[32] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *ICLR 2023*, 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[33] S. K. Gorti, I. Gofman, Z. Liu, J. Wu, N. Vouitsis, G. Yu, J. C. Cresswell, and R. Hosseinzadeh, "Msc-sql: Multi-sample critiquing small language models for text-to-sql translation," 2024. [Online]. Available: https://arxiv.org/abs/2410.12916

[34] Z. Xi, S. Jin, Y. Zhou, R. Zheng, S. Gao, J. Liu, T. Gui, Q. Zhang, and X. Huang, "Self-Polish: Enhance reasoning in large language models via problem refinement," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 11 383–11 406. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.762/

[35] Guo, Daya, Yang, Dejian, Zhang, and e. a. Haowei, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[36] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 16 857–16 867. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf

[37] H. Li, J. Zhang, C. Li, and H. Chen, "Resdsql: decoupling schema linking and skeleton parsing for text-to-sql," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. [Online]. Available: https://doi.org/10.1609/aaai.v37i11.26535

[38] DeepSeek-AI, A. Liu, B. Feng, B. Xue, B. Wang, and e. a. Bochao Wu, "Deepseek-v3 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2412.19437

www.xueshuxiangzi.com