Text-to-SQL 调研

日期：2023 年 5 月 23 日

摘　要

本次调研，我从解决 text2sql 所需要的知识出发，分“理解自然语言问题”、“理解数据库 Schema”与“生成 SQL 回答”三方面阐述。后续就最近在 text2sql 领域调研到的一些有趣的工作，如新的数据集以及评价指标被提出、大语言模型在 text2sql 的应用等。最后就自己在科研习惯上的不足方面做了一些反思总结。

1　解决 text2sql 需要的知识

对于 text2sql 问题的解决，我认为可以拆分为：**理解自然语言问题**、**理解数据库 Schema**、**生成 SQL 回答**三方面，如图 1 所示。之前我提到过的“对值的类型的判断”、“解释与列名之间的关系”、“对解释的理解”都属于理解数据库 Schema 方面的知识，而“对领域词汇的理解”可以归为理解自然语言问题的范畴。除此之外为了生成有效的 SQL 回答，首先是要求模型有回答问题的基本思考能力[1]；再者就是基于对数据库 Schema 的理解与自然语言问题的驱动，从数据库 Schema 中抽取正确的列名、表名以及一些值；最后就是根据回答自然语言问题的思路，以及从数据库 Schema 中抽取到的有用信息，整合成 SQL 语句。当然后续如果还使用 PICARD[12] 规范生成的 SQL 语句则属于另外的知识，在此我将其归类为附加“插件”类型，并不作为 text2sql 模型本身的能力。

图 1: text2sql 问题需要的知识

[1]即自然语言生成类模型所具有的“接话，回复”能力，该部分并不要求模型能直接生成 SQL 语句，只需知道要生成 SQL 语句需要使用哪些信息，以及如何输出规范的 SQL 语句即可

这些知识中，对于“理解自然语言”以及“生成 SQL 回答”中基本“回复接话”能力，常用的是在使用自回归的无监督框架中学习得到，如预训练模型 GPT3[1]、T5[11] 等。而其他如“理解数据库 Schema”、以及“生成 SQL 回答”其余需要的的能力，则可以通过有监督的框架学习得到[2]。对于一些未使用预训练模型的工作如 RAT-SQL[15]，则没有使用自回归的无监督训练部分，只使用基于最大似然函数的自回归训练做有监督学习，且同样可以达到较优的训练效果。总得来说，解决 text2sql 问题所需要的知识，基本可以在一个有监督训练框架中学习得到。但是对于如果使用预训练模型，则在该部分需要一些额外的文本数据资源；而如果加入文本部分和数据库 Schema 部分的对齐，则可能需要一些额外的匹配的自然语言问题和数据库 Schema；如果使用将数据库 Schema 使用自然语言描述的方法，则基本可以不使用其他资源辅助。整体来说，我认为这些所需的知识可以在一个有监督训练框架中学习得到。

2 text2sql 领域最近一些有趣的工作

2.1 新的数据集、新的评价指标：

首先是一篇可能较有潜力的一篇文章——Can LLM Already Serve as A Database Interface[7]，为缓解 text2sql 领域在学术研究和实际应用程序之间的差距，该文章提出了 BIRD 这个更大，跨越领域更多的数据集，并提出了 text2sql 领域一个新模型评价角度——推理效率 (text-to-efficient-SQLs)，以及其评价指标 VES(Valid Efficiency Score)，来评价生成的 sql 语句是否可执行以及是否能更快完成任务。BIRD 数据集与 Spider[17] 和 WiKiSQL[18] 不同，对于每一个问题项多了 evidence(推理需要的知识，一共四类)，除了便于模型学习领域知识以及解决泛化性需要的一些知识推理能力外，有的 evidence 还提供对数据库中多元变量取值的知识，弥补了之前我在调研中提到 Spider 数据库 [17] 中 T4[4] 知识存在一些不足的问题。我认为使用该数据集训练模型可以在一定程度上提高模型的泛化能力。

2.2 大语言模型在 text2sql 的应用：

上面那篇文章 [7] 还使用预训练的 T5 模型、CodeX 和 ChatGPT 三类模型在 BIRD 数据集上评测，检验大语言模型在更具挑战的 text2sql 问题上的表现，如图 2 所示，结果上取得 SOTA 的 ChatGPT+COT(使用思维链提示的 ChatGPT) 与人类的表现仍有几乎 40，50 点的差距。

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
FT-based				
T5-Base	6.32	11.54 (+5.22)	7.06	12.89 (+5.83)
T5-Large	9.71	19.75 (+10.04)	10.38	20.94 (+10.56)
T5-3B	10.37	23.34 (+12.97)	11.17	24.05 (+12.88)
ICL-based				
Codex	25.42	34.35 (+8.93)	24.86	36.47 (+11.61)
ChatGPT	24.05	**37.22** (+13.17)	26.77	39.30 (+12.53)
ChatGPT + COT	**25.88**	36.64 (+10.76)	28.95	40.08 (+11.24)
Human Performance	-	-	**72.37**	**92.96** (+20.59)

图 2: 模型在 BIRD 数据集上的执行准确率 (Execution Accuracy)，其中 w/o knowledge 表示不提供推理所需知识，w/ knowledge 则表示提供

除此之外，还有 Divide and Prompt[10] 集中于使用思维链 (CoT) 提示与大模型组合，尝试解决 text2sql 领域问题。文章提出三种提示，均指导模型将 text2sql 问题划分为子任务，然后求解。该工作在 Spider 数

[2]该部分许多模型如 MIGA[3]，Graphix-T5[8] 等都是在有 SQL 生成结果作为标签的基础上，使用基于最大似然函数的自回归训练得到，属于有监督学习

据集 [17] 上验证，从结果上来看，即使是表现最佳的一组提示在执行准确率上表现甚至略低于 T5-3B + PICARD 模型的表现，且相对使用 zero-shot 以及 few-shot 的 GPT-3.5 模型实验中仅有 2-4 点的提升。

虽然大语言模型在许多自然语言任务上都表现出远超 SOTA 的效果，但是在 text2sql 领域，仍然有较大的进步空间。未来或许会有工作致力于研发专门面向 text2sql 领域的大模型，相信这项工作会在 Spider 数据集 [17] 以及 BIRD 数据集 [7] 上取得接近人类甚至超过人类的效果。

2.3 先前未调研到的一个有趣方向：

这次在翻看arxiv网站 text2sql 领域近期的论文时，发现了篇关于使用交互式反馈进行生成的 sql 完善的工作——Interactive Text-to-SQL Generation via Editable Step-by-Step Explanations[13]。text2sql 问题最初的目的就是降低访问数据库的门槛。而使用单轮或者多轮 text2sql 的方法，对于一些复杂的查询，用户往往会犯一些错误，文中提到的交互式反馈生成如图 3 所示，是在直接根据问题生成的基础上，同时返回对 SQL 语句的自然语言解释，并允许用户对该解释提出修改，并反馈给模型，模型会决定是重写或者简单修改，并再次输出 SQL，再重复上述操作，直至生成正确不再有反馈。

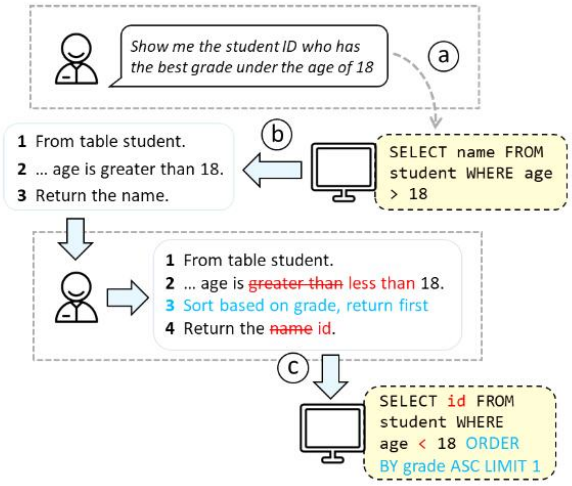

图 3: text2sql 交互式生成

关于使用交互式反馈提升 text2sql 问题的解决效果，在近些年有一些支持有一定约束形式的反馈的工作 (MISP[16]、PIIA[9]、DialSQL[5] 等)，以及一些使用较为开放地提供对话式反馈模型 (如 NL-EDIT[2])。本文提出的 STEPS[13]，在受限反馈和开放式反馈之间取得平衡，是通过允许使用者编辑生成 sql 的自然语言解释，提出的新互动机制。从结果来看，STEPS 几乎可以解决 Spider 数据集 [17] 中所有的简单和中等任务，而且对于困难和额外的困难任务也达到了 90% 以上的准确性。相比当前的 SOTA 模型，可以取得 20% 左右的准确率提升。

我认为这种交互式的 text2sql 解决方法是从该领域真正想解决的问题 (降低访问数据库的门槛) 的角度出发的思考。在内部组成方面，STEPS[13] 模型会首先调用一个 text2sql 模型生成初始 SQL 查询，后续再在该 SQL 查询的基础上执行生成解释、接收反馈、修改 SQL 查询等操作，可以看成一个类似 PICARD[12] 的“插件”，但是不同的是，该“插件”使模型的执行逻辑有了一些改变。

3 心得体会

关于老师您之前的指点，最令我受益匪浅的是“对问题的本质思考”。先前的调研中，“加入 Gate&Add 模块”和“加入 CNN，CBAM 提取”两点都属于是对一些试探性的尝试；提出尝试双塔结构实现文本模

态与数据库 Schema 模态的融合对齐，尝试使用自然语言的形式描述数据库 Schema 将两个模态转变为单模态任务以及本次调研到的使用交互式的反馈生成 sql 查询或许算是实际有效思考，想清楚的点。

以之前上手写过的项目为例，我似乎更多的把心思放在如何调一下参数，或者试着加入这个模块，那个模块试图提升模型的最终效果，虽然有一些模块的加入是出于有效的思考，但也有不少是单纯的随意尝试。现在看来，我更应该去思考模型、或者损失函数在该问题上实现的不足，或本身能力上的缺陷，并从该方向思考解决问题。想来一些学术界有巨大影响力的工作，如提出残差网络的 ResNet[6]、提出自注意机制的 Transformer[14] 等，都是从问题的本质出发，前者思考能否在网络加深的基础上保持模型能力不退步，后者思考如何让模型能看到输入中所有元素之间的相互关系，尽可能多的利用上下文信息。

参考文献

[1] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[2] Ahmed Elgohary et al. "NL-EDIT: Correcting Semantic Parse Errors through Natural Language Interaction". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021, pp. 5599–5610. DOI: 10.18653/v1/2021.naacl-main.444.

[3] Yingwen Fu et al. *MIGA: A Unified Multi-task Generation Framework for Conversational Text-to-SQL*. 2022. arXiv: 2212.09278 [cs.CL].

[4] Yujian Gan, Xinyun Chen, and Matthew Purver. *Exploring Underexplored Limitations of Cross-Domain Text-to-SQL Generalization*. 2021. arXiv: 2109.05157 [cs.CL].

[5] Izzeddin Gur et al. "DialSQL: Dialogue Based Structured Query Generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1339–1349. DOI: 10.18653/v1/p18-1124.

[6] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[7] Jinyang Li et al. *Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs*. 2023. arXiv: 2305.03111 [cs.CL].

[8] Jinyang Li et al. *Graphix-T5: Mixing Pre-Trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing*. 2023. arXiv: 2301.07507 [cs.CL].

[9] Yuntao Li et al. " "What Do You Mean by That?" A Parser-Independent Interactive Approach for Enhancing Text-to-SQL". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 6913–6922. DOI: 10.18653/v1/2020.emnlp-main.561.

[10] Xiping Liu and Zhao Tan. *Divide and Prompt: Chain of Thought Prompting for Text-to-SQL*. 2023. arXiv: 2304.11556 [cs.CL].

[11] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG].

[12] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. *PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models*. 2021. arXiv: 2109.05093 [cs.CL].

[13] Yuan Tian et al. *Interactive Text-to-SQL Generation via Editable Step-by-Step Explanations*. 2023. arXiv: 2305.07372 [cs.DB].

[14] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[15] Bailin Wang et al. *RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers*. 2021. arXiv: 1911.04942 [cs.CL].

[16] Ziyu Yao et al. "Model-based Interactive Semantic Parsing: A Unified Framework and A Text-to-SQL Case Study". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5447–5458. DOI: 10.18653/v1/d19-1547.

[17] Tao Yu et al. *Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task*. 2019. arXiv: 1809.08887 [cs.CL].

[18] Victor Zhong, Caiming Xiong, and Richard Socher. *Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning*. 2017. arXiv: 1709.00103 [cs.CL].