结构化数据库查询语言智能合成技术研究进展

刘雨蒙，赵怡婧，王碧聪，王潮，张宝民

引用本文

刘雨蒙，赵怡婧，王碧聪，王潮，张宝民. 结构化数据库查询语言智能合成技术研究进展[J]. 计算机科学, 2024, 51(7): 40-48.

LIU Yumeng, ZHAO Yijing, WANG Bicong, WANG Chao, ZHANG Baomin. Advances in SQL Intelligent Synthesis Technology [J]. Computer Science, 2024, 51(7): 40-48.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000143

结构化数据库查询语言智能合成技术研究进展

刘雨蒙[1,2]　赵怡婧[1,2]　王碧聪[1]　王　潮[1]　张宝民[1]
1 中国科学院软件研究所　北京 100190
2 中国科学院大学　北京 100049
(yumeng@iscas.ac.cn)

摘　要　近年来,随着大数据、云计算等技术的飞速发展,大规模数据的产生使得各类应用对于数据库技术的依赖日益加深。然而,传统的数据库一般采用形式化的数据库查询语言SQL进行操作,对无编程经验或数据库使用经验的用户来说,复杂SQL语法难度较高,降低了各个领域数据库应用者的便捷程度。近年来,机器学习、深度神经网络等人工智能技术的飞速发展,尤其是ChatGPT横空出世引发的大语言模型技术热潮,驱动了数据库与人工智能的深度结合与技术变革。通过智能方法将用户输入语言自动化合成SQL语言,以满足不同程度数据库使用者的操作需求,提升数据库的智能性、环境适应性及用户友好性。为全面聚焦数据库查询语言智能合成技术的最新研究进展,从范例输入、文本输入及语音输入这3类用户输入切入,详细阐述各类智能合成模型的研究脉络、代表性工作及最新进展,同时对各类方法的技术框架进行归纳与对比,最后对全文进行全面性的总结,并针对现有方法存在的问题和挑战展望未来发展方向。

关键词:数据库技术;SQL智能合成;语义解析;SQL语法;大语言模型
中图分类号　TP315

Advances in SQL Intelligent Synthesis Technology

LIU Yumeng[1,2],ZHAO Yijing[1,2],WANG Bicong[1],WANG Chao[1] and ZHANG Baomin[1]
1 Institute of Software,Chinese Academy of Sciences,Beijing 100190,China
2 University of Chinese Academy of Sciences,Beijing 100049,China

Abstract　In recent years,with the rapid development of technologies such as big data and cloud computing,large-scale data generation has deepened the dependence of various applications on database technology. However,traditional databases typically operate through the formalized database query language SQL,which poses a significant difficulty for users without programming or database usage experience,reducing the accessibility of databases across various fields. With the rapid advancement of artificial intelligence technologies like machine learning and deep neural networks,especially the surge of large language model technology sparked by the emergence of ChatGPT,there has been a profound synthesis and technological transformation of databases and intelligent technology. Intelligent methods are employed to automatically translate user input language into SQL,meeting the operational needs of database users of varying levels of expertise and enhancing databases' intelligence,environmental adaptability,and user-friendliness. To comprehensively focus on the latest research developments in intelligent SQL generation technology,this paper delves into three types of user inputs-example-based,text-based,and voice-based-and provides a detailed exposition of the research trajectory,representative works,and the latest advancements of various intelligent synthesis models. Additionally,this paper categorizes and compares the technical frameworks of these methods and provides an overall summary. Finally,it paper looks forward to future development directions in light of existing problems and challenges with current methods.

Keywords　Database technology,Intelligent SQL synthesis,Semantic parsing,SQL syntax,Large language models

1　引言

随着近十年来大数据技术的不断发展和数据管理需求的增加,关系型数据库已经成为一种主流的结构化数据维护、存储和访问方式。这种方式的数据操作依赖于一种专用的数据库查询语言,即结构化查询语言(SQL)。然而,SQL语言的设计初衷主要是针对深入了解数据库结构和内容的专业人员,而对于数据库经验有限的用户来说,编写合适的SQL请求具有一定的难度。特别是在当前以数据为中心的数据库架构下,用户需要付出一定的学习和实践成本才能掌握和熟练运用SQL。即便是对SQL有深入理解的专业人员,也会发现在处理日益复杂的查询需求时,依赖于手动编写SQL代码的传统方式已经越来越难以继续。因此,如何将数据库的使用方式从以数据为中心转变为以用户为中心,成为目前学术界和

到稿日期:2023-10-19　返修日期:2024-03-29
通信作者:赵怡婧(yijing@iscas.ac.cn)

工业界广泛关注的问题。

解决上述问题的关键在于建立一种能将自然语言输入转换为 SQL 语句的自然语言交互接口（NLI）。由于对自然语言的处理能力有限，早期的尝试常常采用格式化的输入方式，以限定输出端 SQL 语句的语法结构。这种方法的主要思路是在用户指定了一些关键词后，系统地生成一个预设的输出框架，然后填充必要的数据库表、字段等信息，以生成对应的 SQL 语句。这种方法简单、有效、直观，但其输入和输出形式的固定化，使得它能解决的问题范围有限。

近年来，随着深度学习模型以及大语言模型的出现，自然语言学习领域取得了飞速发展。这些进展使得输入端的自然语言表达的限制被打破，同时深度学习模型能够将数据库的相关信息和用户的需求等更丰富的信息映射到特征空间中。此外，深度学习模型的参数规模大幅度提升，使其能够处理更复杂的逻辑表达和数据组织任务，提供更多样化的 SQL 关键词支持、聚合函数、SQL 嵌套操作等，从而满足更高级别的 SQL 代码生成需求。

在现有的 SQL 语言智能合成研究中，用户提供的输入信息主要包括范例型、文本型、语音型等，这些不同类型的输入方式催生了各类不同的智能处理方法。根据用户输入类型的不同，本文将现有 SQL 语言智能合成方法划分为三大类：基于范例合成 SQL 语句方法（input-output example to SQL）、基于文本输入合成 SQL 语句方法（text to SQL），以及基于语音输入合成 SQL 语句方法（speech to SQL）。

本文后续内容主要整理和归纳了该领域中 3 类 SQL 语言合成方法，对比了目前 SQL 语言智能合成方法各自模型的特点、参数规模及其局限性等，总结了目前该领域的发展历程及最新的研究进展，并提出了一些目前仍然存在的研究瓶颈。

2 基本概念

数据库查询语言智能合成技术指，通过智能算法自动化地实现将用户输入转化为数据库查询语句的过程。这类技术的核心目标是解决用户在编写数据库查询语句时所面临的复杂性和繁琐性，不再需要手动编写复杂的 SQL 语句。相关概念的介绍如下。

数据库查询语言（Structure Query Language，SQL）：一种能够实现数据定义和数据操作的编程语言，它为用户提供了一种有效的与数据库管理系统进行交互的方式。然而，由于 SQL 语句的编写需要具备一定的编程知识和数据库知识，对于非专业的用户来说，编写 SQL 语句可能是一项具有挑战性的任务。

语义解析（Semantic Parsing）：将自然语言形式的输入映射为一种形式化的、结构化的表示，如 SQL、Python 等，这种结构化的表示可以更精确地捕获输入语句的含义。本文研究的方向属于语义解析的子方向，特指映射到 SQL 的语义解析。

SQL 语法：SQL 语法通常由关键词、比较运算符、逻辑运算符和聚合函数等组成，这些元素结合数据库中的表和字段信息，构成了 SQL 语句。理解 SQL 语法的结构和特点是生成准确的 SQL 语句的关键。

大语言模型（Large Language Models，LLM）：通过在大量文本数据上进行训练，能够提取自然语言的特征，并生成用户指定的文本输出。这种模型的出现极大地提高了自然语言处理的能力，并扩展了模型的应用范围。在数据库查询语言智能合成技术中，大语言模型可以用于提取输入的特征，并生成对应的 SQL 语句。

3 基于范例的 SQL 语句合成

自然语言具有歧义性和冗余性，而数据库 SQL 语言具有准确性和完备性，鉴于两者的特性不同，难以直接转化，因此考虑将自然语言进行一定的规范，建立某种范例或者输入模式来限制查询。如图 1 所示，基于范例的 SQL 语句合成方法需要用户提供输入和输出的范例（Input-Output Example）以限制查询范围。这类方法的核心目标是建立一种 SQL 语句生成器，通过对已提供的范例进行填充，同时建立一种 SQL 语句的排序算法，将推荐排名较高的 SQL 语句提供给用户，再由用户决定继续提供新的输入输出范例或者结束 SQL 语句的生成任务。

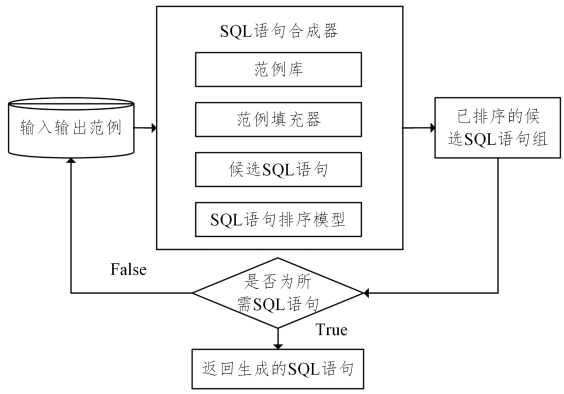

图 1 基于范例输入的 SQL 语句合成

Fig. 1 SQL synthesis based on input-output examples

使用范例化的文本进行交互的经典工作较多，代表性的包括 LUNAR[1]，RENDEZVOUS[2]，LADDER[3]，Chat-80[4] 等，这些研究成果大多使用范围较窄，只能支持 select-from 形式的功能，无法完成数据库表连接等复杂操作，因此本文不过多描述，而重点关注该方向近 10 年的研究成果。

3.1 简单合成方法

使用范例的方法进行查询最初依赖于用户提供输入输出示例，以明确地演示如何查询数据库。2013 年 Zhang 等[5] 设计了基于范例的编程技术及其实现工具，帮助终端用户实现查询任务的自动化，称为 SQLSynthesizer。SQLSynthesizer 需要用户提供输入和输出的一个例子来说明如何查询数据库，然后合成一个与输出输入示例具有相同形式的 SQL 查询。如果将合成 SQL 查询的方法应用于另一个潜在的数据库与一个类似的数据库模式，该方法可以生成一个与示例输出类似的 SQL 合成结果。这种直观性使得非专业的终端用户也能利用它生成复杂的 SQL 查询，但在处理特别复杂的查询或者在示例有限的情况下可能存在挑战。

Li 等[6] 提出了 QFE，可以利用用户提供的示例数据库以及该数据库上的一个目标查询所对应的输出表所组成的输入

对(Database-result pairs),生成一系列的候选查询语句并完成初步筛选。由用户在初步筛选后的几个候选查询中选择最终结果,并与示例数据库形成新的输入对。QFE 在一个旨在帮助研究者利用 RDBMS 进行数据分析的一个云平台 SQLShare 上完成了算法有效性的验证。

3.2 复杂合成方法

近年来,为了优化处理大型数据库和复杂查询时 SQL 合成的效率和质量,Wang 等[7]提出了 SCYTHE 系统,能够有效地根据输入输出范例(I/O Example)形成 SQL 语句。其核心思路是设计一种抽象化查询语言,将原本复杂的生成问题转化为一系列更易解决的子问题。SCYTHE 使用了 193 个实际数据标准进行性能验证,证明了其 SQL 生成的有效性及正确性。

随后,Thakkar 等[8]提出了基于范例的 EGS 算法,通过在称为"恒定共发生图"(Constant Co-occurrence Graph)的数据结构中利用模式来合成关系查询,并使用此结构有效地枚举候选程序。最后通过实验表明,EGS 在运行时间、合成程序的质量以及证明不可解性等方面优于基于枚举搜索的 SCHYTHE、基于约束解决方案的 ILASP[9]和跨多个维度的混合技术的 ProSynth[10]等 SQL 合成工具。

针对日益复杂的分析场景,Zhou 等[11]提出了 SICKLE,可高效生成拥有复杂逻辑的 SQL 语句。该方法提出了一种用户规范和计算演示,简化了分析任务的复杂性,同时提出了一种新的语言抽象方法,可结合更细粒度的计算信息来剪除不可行的过程。SICKLE 可在 10 s 左右完成大多数的复杂 SQL 语句的生成,效率大幅提高。

表 1 基于范例的 SQL 语句智能合成技术方法的对比

Table 1 Comparison of SQL intelligent synthesis techniques based on input-output examples

名称	模型组成	验证基准	模型特点	局限性
SQL-Synthesizer[5]	PART	Textbook& Online Forum Questions	支持在线交互	抗干扰能力弱 不支持未定义语法
QFE[6]	Query Generator+ Database Generator+ Result Feedback	Stack Overflow & SQLShare	利用用户反馈进行结果迭代优化	迭代时间过长 SQL 的可解释性差
SCYTHE[7]	Synthesizer+DFS	Stack Overflow & ASE'13 Benchmarks	建立抽象语言表示 支持 Union、Outer 等复杂算子	对嵌套查询支持弱
EGS[8]	Enumerator+ Evaluator+ Priority queue	Stack Overflow & Textbook	利用常量共现图生成候选示例	处理效率低 无法处理含噪声输入
SICKLE[11]	Enumerative Search Abstract data provenance	Analytical SQL online forums & TPC-DC	更细粒度的抽象语言表示	处理效率低 用户交互形式差

3.3 小结

基于范例的 SQL 语句合成方法需要用户对现有数据库模式具备清晰的认知。并且,由于实际数据库体量的不断增长,使用输入-输出范例的方式来表达使用者意图的形式变得愈发笨重且低效。

总体而言,基于范例的 SQL 语句合成方法固定性强。虽然这类方法相对易用,但其适用范围通常仅限于解决较为简单的 SQL 查询需求。在应对更复杂的查询需求时,其效果往往不尽如人意。当出现新的需求类型时,需要通过添加新的范例进行扩展。同时,此类方法对用户的基础要求相对较高,例如用户需要具备一定的数据库结构知识和对 SQL 关键词的理解。另一方面,由于受到参数规模的限制以及输入输出形式的固定化,这些模型在支持的 SQL 语法能力和模型的泛用性等方面均存在改进的空间。

4 基于文本输入的 SQL 语句合成

自然语言文本输入主要指由自然语言形成的无范式的文本输入。这种无范式导致的不确定性使得算法在提取文本中的关键信息时,流程更加复杂。如图 2 所示,使用文本输入转换 SQL 语句时,需要进行自然语言编码和数据库模式编码。自然语言编码将预处理后的文本完成实体识别和关系抽取,提取用户的查询意图。数据库模式编码将文本输入中涉及的数据库基础信息(数据库、表、字段)进行组合,并完成模式连接。然后,组合自然语言编码结果与数据库模式编码结果为 SQL 抽象语法树,得到待生成语句的结构化表示。最后通过 SQL 解码器,生成 SQL 语句。本研究方向先后出现了两个重要的公开数据集 WiKiSQL[12]和 Spider[13],围绕两个数据集先后涌现了大量的研究和验证工作,本章对其中有代表性的工作进行了总结分析。

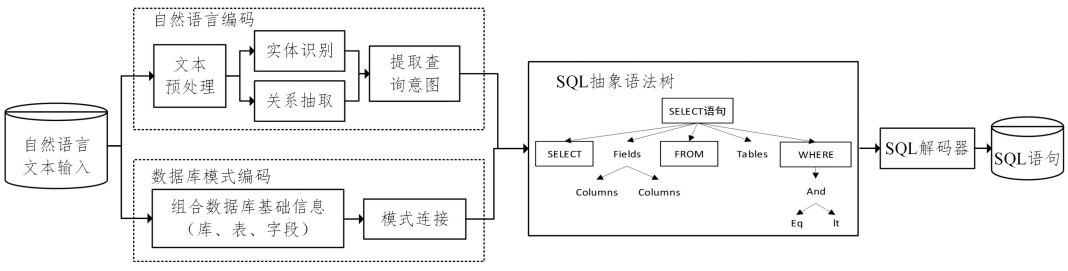

图 2 基于文本输入的 SQL 语句合成

Fig. 2 SQL synthesis based on text-to-SQL techniques

4.1 单轮跨域单表查询

基于文本生成 SQL 算法在其发展前期受限于人工获取 SQL 语句数据集标注结果的规模，使得深度学习模型的泛化性受限。2017 年 Zhong 等[12]发布了 WikiSQL，成为了第一个大规模的语义解析数据集。该数据集包括由手工注释的 80654个自然语言问题、SQL 查询以及从 Wikipedia 中提取的 24241 HTML 网页表中提取的 SQL 表。WikiSQL 具有比以往提供逻辑形式以及自然语言的语义解析数据集更庞大的数据量级。

同时，Zhong 等还发布了基于强化学习的数据库查询执行引擎 Seq2SQL，证实了在 WikiSQL 上，Seq2SQL[12]优于 2016 年提出的面向生成逻辑化输出的一种基于注意力机制的编码器解码器的泛化模型 Seq2Seq。Seq2SQL 模型在后续研究中常被用做 WiKiSQL 数据集的基准模型。

与之相反，Xu 等[14]认为使用 Seq2SQL 模型中的强化学习算法形成的算法提升效果有限，并证明该方法难以处理现实中能够产生等价结果的 SQL 语句生成问题。为了解决上述问题，Xu 等提出了一种非强化学习的 SQL-Net 模型。SQL-Net 模型主要包含两个部分：1）一种基于包含依赖图的草图方法，使得一个预测只能使用与其相关的以往预测进行计算；2）使用了一个包含列注意力机制的 Seq2Set 模型，以生成草图结构中的具体查询内容。SQL-Net 模型还增加了权重方法来强调句子中重要词或者短语的重要性。该模型在当时的 WikiSQL 数据中达到了超过同期模型 9%～13%的效果，但在复杂 SQL 语句的合成方面存在困难。

4.2 单轮跨域多表查询

为推动复杂 SQL 查询技术的发展，Yu 等[13]于 2019 年公布了包含大量复杂 SQL 语句的经典数据集 Spider。该数据集包含 10181 个问题和 5693 个不同的复杂 SQL 查询，涵盖了 200 个具有多个表的数据库，覆盖了 138 个不同的领域。研究人员在 Spider 上实验了当时较为先进的模型，包括 Seq2SQL 和 SQL-Net 等，但都没有取得理想的效果。Spider 数据集的出现也为该领域带来了一个新的挑战。

为了进一步解决复杂 SQL 语句的合成问题，Yu 等[15]提出了 SyntaxSQLNet 模型，主要目标是实现具有多个子句的复杂 SQL 查询以及跨域 SQL 查询语句的生成。SyntaxSQL-Net 使用了一个基于 SQL 语法树的解码器以及一个能够感知 SQL 生成路径历史以及数据库表的列注意力编码器。该模型在 Spider 包含多表结构、多 SQL 子句及嵌套查询的文本转 SQL 测试任务中比同期最优算法提高了 7.3%。另外，SyntaxSQLNet 首次在文本转 SQL 领域增加了交叉域扩展方法来优化模型，优化后可以在 Spider 数据集达到 14.8%的提升效果。

由于 Spider 数据集中自然语言文本的复杂性和逻辑性更高，仅使用符号级别（Token-Level）的信息编码已无法完全理解用户意图。2019 年 5 月，Lin 等[16]针对上述问题提出了 Grammar-SQL，旨在进行语法级别（Grammar-Level）的自然语言解析。Grammar-SQL 使用了一个基于动态框架的 SQL 语法来指导解码器以及一个确定性实体连接模块。他们的工作证实了类别信息可以使用连接嵌入或者标识符匿名化，并且证实了类别信息对上下文环境敏感性建模的重要性。

Spider 数据集的另一个主要挑战是其中包含了大量的跨域（cross-domain）数据表操作问题，涉及大量外键连接以及嵌套查询等复杂 SQL 语句的生成，一些研究工作开始着手通过语义级别解析来解决这类复杂 SQL 的合成。Guo 等[17]针对上述问题提出了 IR-Net。该模型主要用于解决自然语言表达与 SQL 转换之间的不匹配问题以及 SQL 中的细节补充问题。IR-Net 没有使用端到端的方式，而是将 SQL 生成方式分为了 3 个阶段。第一阶段，IR-Net 在问题和数据库框架上建立框架连接。第二阶段，使用一种基于语法的神经网络模型生成一种称作 SemQL 查询的中间表达状态，并将其作为自然语言和 SQL 语言之间的桥梁。第三阶段，IR-Net 使用域知识从生成的 SemQL 查询中推理出一个最终的 SQL 查询语句。最终在 Spider 数据集上达到 46.7%的准确率，相比 SyntaxSQLNet 算法提高了 19.5%，列于当时 Spider 数据集排行榜的首位。但是，IR-Net 仍存在无法支持 from 子句中的 self join 功能以及不能完全消除自然语言到 SQL 语言的不匹配性的问题。

为充分保留用户自然语言的原始输入信息，2020 年后一些研究者开始转向端到端语义解析模型的研究。其中面临的重要挑战之一是，语义解析器如何将数据库间的关系编码成一个可访问的方式，以及在给定查询中如何对数据库列与列之间的对齐方式建模。Wang 等[18]针对此问题提出了 RAT-SQL 模型，通过建立一个统一的框架，来应对模式编码以及模式连接的挑战。同时，RAT-SQL 基于彼此的一致性和模式相关关系，引入可感知关系的自注意力机制，学习了数据库定义模式和问题表示。在 Spider 排行榜上实现了 65.6%的最新性能，同年 RAT-SQL 成为了该领域较优秀的开源项目之一。

4.3 多轮跨域多表查询

随着计算机硬件的发展带来的计算能力大幅提升，语言模型的参数不断增加，朝着大规模神经网络发展。大规模神经网络通常采用超大规模语料数据进行预训练，并根据具体场景进行微调，这一通用范式在大部分测试项目中取得了与人类水平持平的惊人效果。2018 年开始，大规模神经网络被广泛应用于自然语言处理，也包括 SQL 语句智能合成领域。经典大规模神经网络模型包括 Google 的 BERT 模型[19]、T5 模型[20]以及 OpenAI 的 GPT 系列模型[21]、CodeX 模型[22]。

Cheung 等首先基于 BERT 提出了 BRIDGE 模型[23]。该模型使用 BERT 进行混合序列编码，结合指针生成解码器，经过训练微调后在 Spider 和 WikiSQL 数据集上取得了先进的效果。T5 模型在 BERT 模型的基础上探索了包括 fine-tune、多任务学习、多任务 fine-tune 和 Scaling 等多种训练策略。Scholak 等提出了 PICARD 方法[24]，通过增量解析约束语言模型的自回归解码器，该方法将微调的 3B 参数 T5 模型转化为在 Spider 和 CoSQL 数据集中表现优越的解决方案。同年，Shaw 等提出了混合模型 NQG-T5 模型[25]，将高精度的基于语法方法和预先训练的序列模型进行结合，在一定程度上解决了非合成数据场景下模型的泛化问题，也为自然语言合成 SQL 语句领域提出了新的研究方向。

表 2 基于文本输入的 SQL 语句智能合成技术方法的对比

Table 2 Comparison of SQL synthesis based on text-to-SQL techniques

名称	模型组成	预训练网络	验证基准	模型特点	局限性
Seq2SQL[12]	Stanford CoreNLP+ GloVe word Embeddings+ policy based RL	—	WiKiSQL	基于策略强化学习生成查询条件	模型训练效率不高；生成语句准确度一般
SQL-Net[14]	Stanford CoreNLP+ GloVe word Embeddings+ Bi-LSTM	—	WiKiSQL	使用 Seq2set 解决顺序问题；引入注意力机制	复杂查询准确度不高
SyntaxSQLNet[15]	Pre-trained GloVe+ Bi-LSTM	—	WikiSQL & Spider	支持多表、嵌套、未知数据库等复杂查询	WHERE 子句、GROUP BY 的处理精度不高
Grammar-SQL[16]	Encoder(Bi-LSTM)+ Decoder(LSTM with Attention)	—	ATIS & Spider	使用上下文相关的语法约束	语法解析能力弱；日期识别精度差
IR-Net[17]	GloVeWord Embedding+ Encoder(BERT)+Decoder	BERT	Spider	使用领域知识；使用 BERT 作为编码器	尚未完全消除自然语言与 SQL 不匹配问题
X-SQL[26]	Type Embedding+ Encoder(MT-DNN)	MT-DNN	WiKiSQL	使用基于上下文预训练模型处理自然语言多变性	语法支持能力有限，缺乏复杂数据验证
RAT-SQL[18]	Bi-LSTM+ Relation-Aware Self Attention	BERT	WiKiSQL & Spider	使用关系感知自注意力机制对推理能力进行优化	可能生成包含错误语法的 SQL 语句
BRIDGE[23]	Encoder(BERT & Bi-LSTM)+ Decoder(Sequential pointer-generator)	BERT	WikiSQL & Spider	将数据库模式表示为连接到问题的标记序列	涉及数据库模式语法支持能力有限；存在虚假语义信息获取
PICARD[24]	T5 Auto-regressive	T5	Spider & CoSQL	泛用性强；可以检测词法级和语法级错误	预测语句结束能力弱
NQG-T5[25]	T5 NQG	T5	SCAN & GeoQUERY & Spider	利用两类模型不同优势互为补充	语义解析能力不足，生成语句准确度一般
CodeX[22]	GPT	GPT-3	APPS	基于 GPT 语言模型的生成方法	可能生成包含错误语法的 SQL 语句
DIN-SQL[27]	Schema linking Query classification	GPT-4	Spider	将 text2SQL 分解为更小的子任务	验证数据集单一，仍需人工提供任务相关信息，无法完全自动化

随后，He 等[26]引入了迁移学习思想，进行了大规模神经网络在 SQL 语句生成领域的早期尝试，建立了一种基于 BERT 风格预训练神经网络 MT-DNN 的 X-SQL 模型。预训练模型使 X-SQL 可以将非结构化的上下文信息对应到与问题相关的结构中，可以更好地表征结构化信息。此外，X-SQL 使用基于 KL-散度的目标函数实现数据库各列的全局排序，相比之前其他算法使用的多二分类器方法，可更高效地提取数据库中各列的相关性信息。X-SQL 在 WiKiSQL 数据集中的性能超过了同时期的其他方法，最好的实验结果准确率达到了 90%以上。该模型分为 3 个部分：序列编码器、上下文增强的模式编码器以及输出层。在 2019 年阿里天池首届 NL2SQL 竞赛复杂的中文语言生成 SQL 的任务中，X-SQL 受到了广泛的重视并取得了理想的效果。

OpenAI 的 GPT 系列模型[21]和 CodeX 模型[22]不同于 BERT 等通过掩码机制预训练的模型，其沿用了单向语言模型的训练方式。同时，这类大模型利用更大的数据量来解决大模型对某单独领域中数据的过分依赖以及对类似领域数据的过拟合问题，不再采用微调方式完成预定任务，在翻译、问答以及文本生成等领域达到了较好的效果。Rajkumar 等[27]对 T5 系列模型、BRIDGE 模型、GPT-3 系列模型及 CodeX 模型进行了 SQL 合成能力的评估，通过实验得出 CodeX 模型在 Spider 数据集上的基准地位。同时，通过在 GeoQuery 和 Scholar 数据集上对上述模型进行基准测试证明，使用少量示例训练 CodeX 模型比在相同示例上微调规模较小的语言模型具有更好的效果。

2022 年 11 月，OpenAI 在 GPT-3.5(InstructGPT)模型的基础上发布了 ChatGPT。该模型在发布后被广泛应用于文本生成的各个领域，尤其在代码自动生成以及辅助调优方面表现出色，可实现自然语言输入合成 SQL 语句功能，有效缩短编程 SQL 语句所需的时间。

2023 年 2 月发布的 GPT-4 版本，进一步提升了自然语义理解以及特定提示下文本生成的能力，其基础框架中的基于人类反馈的强化学习(RLHF)微调机制使得模型能够判断生成结果的质量，并产出和人类认知一致的结果。同年 5 月，Pourrez 等[28]提出了 DIN-SQL 模型，并在 Spider 数据集上取得了最新的 SOTA 效果。这项最新的研究成果将 SQL 的生成过程分解成不同的子问题，并将各子问题的输出结果输入到大语言模型中进行校正。基于 GPT-4 强大的调优能力，该模型能够修复生成的 SQL 查询语句中丢失或冗余的关键词，如 DESC、DISTINCT 以及聚合函数等。在进行 DIN-SQL 结合 CodeX 以及 DIN-SQL 结合 GPT-4 两种模型效果的比较后，结果显示使用 GPT-4 的模型结果明显优于使用 CodeX 的模型。

4.4 小结

从发展历程上看，基于文本输入的 SQL 语句合成早期多集中在单领域的多表查询问题上。在 WikiSQL 数据集发布后，研究集中在单轮、多领域的单表查询，而在 Spider 数据集发布之后，转向研究多领域、多表的单轮复杂查询问题。最新的研究中开始有工作转向多伦多领域的复杂查询问题。

从模型发展角度来看，从专有模型到泛用性更高的模型，从单体模型到组合模型，从以数据库为中心到兼顾复杂查询及多种类数据库表需求，SQL 生成技术不断迭代更新。特别是在大模型出现之后，算法对于多轮的复杂查询的回复精度大幅提升，突破了原来的精准度要求。

总体来说，大规模模型范式已经在 NLP 领域的各种应用领域获得较为满意的效果，是包括 SQL 语句智能合成在内的代码合成领域未来重要的发展趋势之一。

5 基于语音输入的 SQL 语句合成

随着智能手机以及平板的流行，过去的 10 年市场上出现了大量的基于语音的应用，例如语音搜索引擎、语音智能助手、交互机器人（如 Siri、小冰、Cortana 等）等。相比文本交互模式，语音交互模式的诞生大大提高了交互的效率，由此也催生出了在数据库领域中合成 SQL 语句的实际需求。基于语音输入合成 SQL 相比文本方式的难度更高：一方面，语音信息的种类更繁杂，目前仍存在大量只有口语而没有文字表达的语言；另一方面，语音信息存在文字信息中较少出现的大量噪声，进一步增加了合成 SQL 语句时的选择难度。如图 3 所示，当前针对基于语音输入的 SQL 语句合成问题，主要存在两种解决方案。

第一种是首先采用语音识别模型将语音信息转换为文字信息，然后结合已有的基于文本的 SQL 语句合成方法（text to SQL）生成 SQL 语句，这类方法将问题分解为两个独立的部分进行分别处理，使得问题更易于解决。另外一种是直接建立端到端的模型，将原始语音信息作为输入，通过分析语音的频率、幅度、波形等特征，直接生成 SQL 语句。这种方法虽然在理论上更具挑战性，但其可能在保持输入信息完整性和减小转换误差方面具有优势。

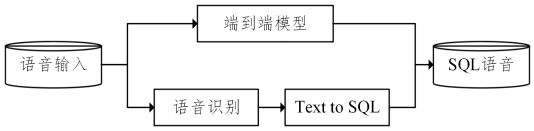

图 3 基于语音输入的 SQL 语句合成

Fig. 3 SQL synthesis based on speech-to-SQL techniques

5.1 语音转文本结合基于文本生成 SQL

基于语音输入合成 SQL 最初的探索是需要用户语音输入严格遵循特定的格式，典型的方法包括在 2017 年 Utama 等[29]设计出的 EchoQuery。EchoQuery 要求用户的语音输入要采用形如“what is the {Aggregation}{Column} of {Table}?”的形式，这种方式相当于将语音直接转化为特定格式的 SQL 语句，它简化了解析过程，但也限制了其对各种自然语言表述和复杂查询的处理能力。

为实现更友好的交互方式，Shah 等[30]提出了 SpeakQL 系统，它可以为语音输入提供更为人性化的交互界面，让用户通过语音形成 SQL 语句。例如，可以支持常规的 SQL 子集并允许用户在任何域中基于触控或语音交互的方式进行人为修正。SpeakQL 需要查询具有类似 SQL 的表述形式“Select {} from {} Where”。在其人机交互中的用户研究中，语音输入的交互界面相比文本输入的交互界面可以让使用者形成 SQL 查询的速度加快 6.7 倍。

然而，由于 SpeakQL 与 EchoQuery 仍需要限制用户输入的语音成为一种基于自然语言版本的 SQL 或者将原语音转换成 SQL 语法的一个子集，因此它们依然需要用户对数据库以及 SQL 语言具备一定的背景知识。

5.2 端到端基于语音生成 SQL

近期，更灵活智能的自然语言语音输入 SQL 合成技术随着语音识别深度学习技术的发展取得了飞跃性突破。Song 等[31]设计了名为 Speech-to-SQL 的新任务，旨在将人类语言传达的信息直接翻译成结构化查询语言（SQL）语句。该文献首次提出了一种端到端的不需要额外语音识别（ASR）系统的新型神经网络框架 SpeechSQLNet，无缝地集成了语音编码器、图形神经网络（GNN）和转换器（Transformer）作为其骨干，以对无标记的语音数据进行语音解析，而无需 ASR 步骤作为前提，并且整个网络以端到端的方式进行了优化。同时，基于 Text2SQL 数据集 WikiSQL 和 Spider，Song 等[31]提出了新的 SpeechQL 数据集，并进行了效果验证；设计了语音句子和语音项目两种预训练机制，以提高算法的实现效果。

5.3 小结

现有方法大多先利用 ASR 将语音转为文本，再利用传统 text2SQL 方法生成 SQL 语句，而目前端到端的模型仍然较少。端到端模型由于输入信息更加丰富，便于模型学习具备更加丰富信息的特征。

整体来看，基于语音输入的 SQL 语句合成技术方兴未艾。随着语音智能助手的不断发展，语音交互形式、基于语音输入的 SQL 语句合成均有着广阔的前景。尽管该领域相比基于文本输入的 SQL 语句合成技术起步较晚，但现有的基于文本输入的 SQL 语句合成技术的丰富研究和应用实践为其提供了坚实的基础，使得我们可以先利用语音转文本技术，再结合现有的文本转 SQL 方法来进行实现。另一方面，由于语音输入具有其独特的复杂性和挑战性，如处理口音、噪音、语速、语调等多种因素，仅仅将其转化为文本可能会丧失一部分信息，因此一个更理想的解决方案可能是开发直接将语音输入转换为 SQL 语句的端到端模型。未来，该方向的研究将会是该领域的重要发展方向。

6 数据集及评价方法

SQL 语句智能生成方法的发展，特别是文本转 SQL 领域的发展是由几个关键的公开数据集推动的，本章将对其进行更详细的介绍，并介绍相关的评价方法。

6.1 常用数据集

WikiSQL 和 Spider 是本领域的两个核心数据集，后续的众多数据集都是在这两个数据集的基础上建立的。

WikiSQL 由 Zhong 等[12]于 2017 年提出，包含手工注释的 80 654 个自然语言表述的问题及其对应的 SQL 查询示例，其源数据涵盖来自 WikiPedia 的 24 241 个数据表，是第一个大规模的语义解析数据集。但是，WikiSQL 中的 SQL 查询结构相对简单，不包含排序 ORDER BY、分组 GROUP BY、HAVING、子查询等复杂操作。WikiSQL 是单轮跨域单表查询的典型数据集。

Spider 由 Yu 等[13]于 2019 年提出，包含 10 181 条查询问题和 5 693 个独立分布的复杂 SQL 查询，涉及 138 个不同的领域。Spider 数据集在体量上虽不及 WikiSQL，但其中的每条查询的复杂度更高，包含 ORDER BY，GROUP BY，HAV-

ING 等高阶操作，为 SQL 智能生成领域带来了较大挑战。Spider 是单轮跨域多表查询的典型代表数据集。

由于 Spider 中的查询相对独立，因此其仅适用于单轮查询。后续出现了面向多轮查询问题的 SparC[32]，以及更进一步面向会话场景的 CoSQL[33]。

另外，基于中文与英文在自然语言解析上存在较大差异的问题，出现了翻译 Spider 中的自然语言查询问题为中文的 CSpider[34]，以及在 NL2SQL 第一届天池大赛中公开的 TableQA[35]。

公开数据集对领域的发展起到了至关重要的作用。但是，目前中文面向 SQL 生成的数据集仍十分有限，特别是在大模型蓬勃发展的时期，训练样本的质量往往直接决定所训练出模型的适应性和鲁棒性，因此更应大力加强公开数据集的建立和发展。

6.2 评价方法

SQL 智能合成目前有两种常见的评价方法，即精准匹配率(Exact Matching Accuracy/Logical Form Accuracy)和执行准确率(Execution Accuracy)。两种方法各具优劣，使用范围尚有差异，如 WikiSQL 支持精准匹配率和执行准确率，但 Spider 仅支持精准匹配率。

精准匹配率指将查询按关键字(SELECT、GROUP BY 等)拆解成不同的子句并使用集合的形式表示，再将算法生成查询的集合表示与标准查询的集合表示进行比较，当两者所有部件均一致时认为算法生成的查询为正确结果。

执行准确率指将算法生成查询的查询结果与标准查询的查询结果进行比较，当两者查询结果一致时认为算法生成的查询为正确结果。

上述两种方法应用最为广泛，但也存在一定的问题，如精准匹配率误将正确生成结果判断为错误结果的问题以及执行准确率将过多生成结果判断为正确结果的问题等。更加完善的评价方案仍待进一步探索。

7 总结与展望

目前，SQL 查询语句智能生成方法及其技术特点、相关语料库和局限性的对比如表 3 所列。基于目前各种方法在交互逻辑方面的区别，本文主要将其分为 3 种类型。

表 3 基于语音输入的 SQL 语句智能合成技术方法的对比

Table 3 Comparison of SQL synthesis based on speech-to-SQL techniques

名称	模型组成	验证基准	模型特点	局限性
EchoQuery[29]	Alexa Voice Service+Sketch-based model	Star schema Benchmark & MIMICII	基于 Alexa 完成语音交互；基于对话查询方式	没有充分利用语音原始特征
SpeakQL[30]	SQL-specified ASR+SQL Grammar	SpeechQL	基于 ASR 实现；首个基于端到端的语音合成 SQL 模型	未针对 SQL 语法生成结构进行进一步优化；生成精度受限于 ASR 精度
SpeechSQLNet[31]	Speech & Schema Encoder+Speech-Schema Relational-aware Encoder+SQL-aware Decoder	SpeechQL	端到端语音合成 SQL 模型；将语音与文本映射到相同的隐藏空间	模型参数量庞大，训练所需资源较多

第一类是用户使用输入与输出范例的形式与生成模型进行交互，生成模型主要是以输入中的主要结构作为框架，提取关键词并生成少数与输出范例结构一致的结果并进行排序。

第二类是用户使用文本输入的方式向生成模型表述需求。该类方法大部分使用编码器-解码器结构进行组合处理，首先利用编码器对输入信息进行词法解析与语法解析，得到携带任务目标信息的中间状态。然后，将该中间状态传入生成 SQL 的解码器中，完成查询生成任务。

第三类是用户使用语音信息输入查询信息，主要流程与第二类大致类似，但需要增加对语音信息的单独处理。可使用 ASR 先将语音信息转换为文本信息之后再进行查询生成，也存在直接使用端到端的方案。

近年来，在 BERT 模型得到广泛应用之后，SQL 语句智能生成领域也迎来了大规模预训练模型的大发展时期，可据此分为前 BERT 时代与后 BERT 时代。在前 BERT 时代，研究者们探索了大量的学习模型方案，包括决策树、强化学习、长短期记忆网络、注意力机制等，试图寻找能够生成具有更多子句形式、更贴合 SQL 语法逻辑的查询语句，甚至包括一些更复杂的嵌套结构。与此同时，出现了两个较为重要的数据集 WikiSQL 与 Spider，为查询生成任务提出了更高的要求。研究者开始从简单的词法解析转向更复杂的语法解析，探索了 SQL 自身与各种自然语言的语法逻辑。SQL 语句生成的本质是对用户输入信息的精准把控与 SQL 编写能力的规范性和准确性的需求。因此，随着自然语言处理中语义解析领域的不断发展，BERT 展现了强大的泛用性和高效性，这也为 SQL 语句生成领域带来了新的生机。在后 BERT 时代，前期的语义解析工作的模式基本固定，主要使用大规模模型或者其微调的版本进行词法与句法的嵌入，研究开始侧重于这些嵌入信息之间的连接方法，并且根据大规模模型的不同迭代版本进行改进。

随着 2023 年以 GPT-4 为核心模型的 ChatGPT 发布，引发了大语言模型的技术热潮，GPT-4 在各种大模型中表现出众，被业内认为是深度学习新的里程碑。GPT-4 具备强大的代码生成能力，同样在 SQL 语句智能生成领域也取得了显著效果。在拥有库表结构的基础信息时，GPT-4 能够支持排序、聚合计算、嵌套查询、联合查询等复杂功能，并且可以提供 SQL 代码的纠错、优化功能以及相关的使用建议等。此外，GPT-4 还能根据用户提供的数据信息或需求等进一步完善生成的结果。然而，需要注意的是，在面临较为复杂的问题时，GPT-4 仍存在一定的错误率，且这种错误缺乏标注信息，需要用户进行二次校对。另外，用户需要提供相对精确的任务描述和需求表达。尽管 GPT-4 在 SQL 语句智能生成方面取得了显著进展，但离实现完全自动化的 SQL 语句智能生成仍有一定距离。

结束语 本文按照用户进行交互的方式，将现有方法划分为3个类别：基于范例的SQL语句生成、基于文本输入的SQL语句生成，以及基于语音输入的SQL语句合成。此种变迁，体现了随着科技进步以及交互方式的发展，用户所需的上手时间逐渐缩短，使用难度逐步降低，操作直观性逐步上升。

SQL智能生成领域作为自然语言处理中的子应用领域，其发展历程与自然语言处理的整体发展趋势是密切相关的。这一领域从最初基于输入-输出范例的模式起步，然后通过应用各种机器学习模型，去除中间复杂的交互和选择步骤，最终朝向完全端对端的解决方案发展。随着计算硬件的进步，数据存储和计算能力得到了极大的提升，使得模型的通用性和适应性得到了提高。值得注意的是，由于SQL语言的结构相对简单，在SQL智能生成领域的研究成果将对其他编程语言的自动化生成工作提供重要的借鉴。

然而，尽管SQL智能生成领域取得了显著的进步，仍存在许多挑战和局限。当前的测试数据集在数量和质量上还难以满足实际应用的需求，这导致许多研究成果难以直接应用到实践中。此外，在预训练模型的使用上，大多是使用通用的大规模预训练模型进行微调，往往忽视了针对SQL生成特点的优化，这使得预训练模型处理数据库相关信息的潜力尚未充分发挥。同时，在SQL生成质量的评估方面也面临着一些挑战，对于非专业人员来说，使用现有的评估指标往往难以准确判断生成模型的结果是否达到了预期的效果。随着研究的深入推进，未来更先进的技术将推动SQL智能生成技术进一步的发展，为实践应用带来更大的价值。

参 考 文 献

[1] WOODS W A. Progress in natural language understanding: an application to lunar geology[C]// National Computer Conference and Exposition. Association for Computing Machinery, 1973: 441-450.

[2] CODD E F. Seven Steps to Rendezvous with the Casual User [C]// IFIP TC-2 Working Conference Data Base Management Systems. 1974.

[3] SACERDOTI E D. Language Access to Distributed Data with Error Recovery[C]// International Joint Conference on Artificial Intelligence. 1977: 196-202.

[4] WARREN D H D, PEREIRA F C. An Efficient Easily Adaptable System for Interpreting Natural Language Queries [J]. American Journal of Computational Linguistics, 1982, 8: 110-122.

[5] ZHANG S, SUN Y. Automatically synthesizing SQL queries from input-output examples[C]// 2013 IEEE/ACM 28th International Conference on Automated Software Engineering (ASE). IEEE, 2013: 224-234.

[6] LI H, CHAN C Y, MAIER D. Query from examples: an iterative, data-driven approach to query construction[J]. Proceedings of the VLDB Endowment, 2015, 8(13): 2158-2169.

[7] WANG C, CHEUNG A, BODIK R. Synthesizing highly expressive SQL queries from input-output examples[C]// ACM SIGPLAN Conference on Programming Language Design and Implementation. ACM, 2017: 452-466.

[8] THAKKAR A, NAIK A, SANDS N, et al. Example-Guided Synthesis of Relational Queries[C]// Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation. Association for Computing Machinery, 2021: 1110-1125.

[9] LAW M, RUSSO A, BRODA K. Inductive Learning of Answer Set Programs[C]// European Workshop on Logics in Artificial Intelligence. 2014: 311-325.

[10] RAGHOTHAMAN M, MENDELSON J, ZHAO D, et al. Provenance-Guided Synthesis of Datalog Programs[J]. Proceedings of the ACM on Programming Languages, 2019, 4(POPL): 1-27.

[11] ZHOU X, BODIK R, CHEUNG A, et al. Synthesizing analytical SQL queries from computation demonstration[C]// Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation. ACM, 2022: 168-182.

[12] ZHONG V, XIONG C, SOCHER R. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning[J]. arXiv: 1709.00103, 2017.

[13] YU T, ZHANG R, YANG K, et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2018: 3911-3921.

[14] XU X, LIU C, SONG D. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning [J]. arXiv: 1711.04436, 2017.

[15] YU T, YASUNAGA M, YANG K, et al. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2018: 1653-1663.

[16] LIN K, BOGIN B, NEUMANN M, et al. Grammar-based Neural Text-to-SQL Generation[J]. arXiv: 1905.13326, 2019.

[17] GUO J, ZHAN Z, GAO Y, et al. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation [C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2019: 4524-4535.

[18] WANG B, SHIN R, LIU X, et al. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020: 7567-7578.

[19] KENTON J D M W C, TOUTANOVA L K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [C]// Proceedings of NAACL-HLT. Association for Computational Linguistics, 2019: 4171-4186.

[20] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer [J]. The Journal of Machine Learning Research, 2020, 21(1): 140: 5485-140: 5551.

[21] BROWN T, MANN B, RYDER N, et al. Language Models are Few-Shot Learners[C]// Advances in Neural Information Processing System. Curran Associates, Inc., 2020: 1877-1901.

[22] CHEN M, TWOREK J, JUN H, et al. Evaluating Large Language Models Trained on Code[J]. arXiv: 2107.03374, 2021.

[23] CHEUNG A, KAMIL S, SOLAR-LEZAMA A. Bridging the Gap Between General-Purpose and Domain-Specific Compilers with Synthesis[C]// 1st Summit on Advances in Programing Languages. 2015: 51-62.

[24] SCHOLAK T, SCHUCHER N, BAHDANAU D. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models[C]// Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021: 9895-9901.

[25] SHAW P, CHANG M W, PASUPAT P, et al. Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? [C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2021: 922-938.

[26] HE P, MAO Y, CHAKRABARTI K, et al. X-SQL: reinforce schema representation with context[J]. arXiv: 1908.08113, 2019.

[27] RAJKUMAR N, LI R, BAHDANAU D. Evaluating the Text-to-SQL Capabilities of Large Language Models[J]. arXiv: 2204.00498, 2022.

[28] POURREZA M, RAFIEI D. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction[J]. arXiv: 2304.11015, 2023.

[29] UTAMA P, WEIR N, BINNIG C, et al. Voice-based data exploration: Chatting with your database[C]// Proceedings of the Workshop on Search-Oriented Conversational AI. 2017.

[30] SHAH V, LI S, KUMAR A, et al. SpeakQL: Towards Speech-driven Multimodal Querying of Structured Data[C]// Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. ACM, 2020: 2363-2374.

[31] SONG Y, WONG R C W, ZHAO X, et al. Speech-to-SQL: Towards Speech-driven SQL Query Generation From Natural Language Question[J]. arXiv: 2201.01209, 2022.

[32] YU T, ZHANG R, YASUNAGA M, et al. SParC: Cross-Domain Semantic Parsing in Context[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2019: 4511-4523.

[33] BERANT J, CHOU A, FROSTIG R, et al. Semantic Parsing on Freebase from Question-Answer Pairs[C]// Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2013: 1533-1544.

[34] MIN Q, SHI Y, ZHANG Y. A Pilot Study for Chinese SQL Semantic Parsing[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Association for Computational Linguistics, 2019: 3652-3658.

[35] SUN N, YANG X, LIU Y. TableQA: a Large-Scale Chinese Text-to-SQL Dataset for Table-Aware SQL Generation[J]. arXiv: 2006.06434, 2020.

LIU Yumeng, born in 1989, Ph.D. His main research interests include database technology, time series data analysis and data mining.

ZHAO Yijing, born in 1994, Ph.D candidate. Her main research interests include database technology and data mining.

（责任编辑：刘亚辉）