LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations

摘要

此前基于图神经网络的 Text-to-SQL 模型仅考虑节点图而忽略了边特征的结构拓扑，往往这有可能导致过平滑问题，同时利用元路径描述节点关系不仅构建需要大量人力且忽略了局部与非局部差异。作者提出了利用线图增强的 Text-to-SQL 模型 (LGESQL)，在原有图基础上生成以边为中心的线图作为对偶图以同时捕获节点与边的结构拓扑进行迭代计算；引入局部与非局部边特征，避免过平滑问题同时使模型更为关注局部边特征；同时引入图修剪辅助任务辅助提升主任务即 Text-to-SQL 生成性能。最终模型获得当时 Text-to-SQL 问题上的 SOTA 性能。

关键词： Text-to-SQL；语义解析；论文复现

1 引言

随着电子设备的发展与普及、互联网的更新换代，表格数据已成为存储结构化数据的数据库主流形式。用户根据自身需求存取数据库中的结构化数据，这在诸多领域都有应用。而 SQL 语言则是当前使用关系数据库的主要查询语言，当今的金融、电子商务以及医疗等专业领域都存在有大量数据存储在关系数据库中。虽然数据库专业的熟练人员能够通过手动编写 SQL 语言有效访问数据库，但是这对于普通人士而言需要具备一定专业知识，并且重复地手动编写 SQL 语句也是低效的行为。因此，Text-to-SQL 任务，旨在给定数据库模式的条件下将自然语言问题自动转换为 SQL 查询，从而有效促进更广泛的非专业用户访问数据库，引起了工业界和学术界的广泛关注。它可以使非专家用户能够毫不费力地查询表，并在智能客户服务、问答和机器人导航等各种实际应用中发挥着核心作用。

Text-to-SQL 系统框架如图 1所示。其中，给定数据库模式和用户问题 “where are the major cities in the state of Kansas”，Text-to-SQL 系统输出对应的 SQL 查询语句，利用这一语句用户可向数据库系统查询得到对应结果。 [3]

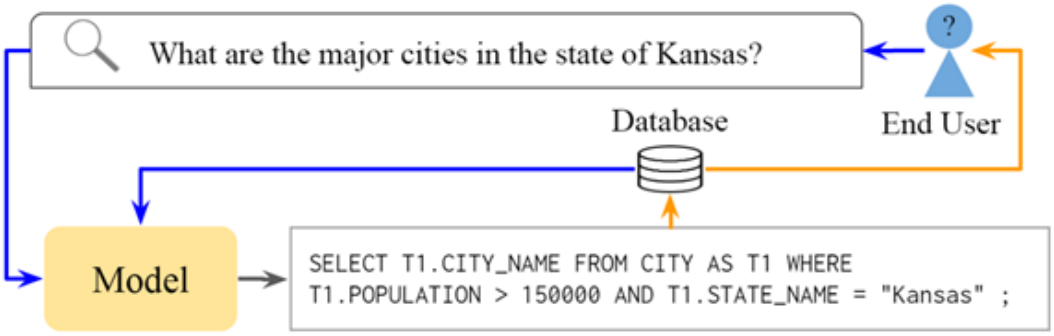

图 1. 系统示意图

早期 Text-to-SQL 实现主要基于模板与规则进行转化与生成，但此类方法需要耗费大量人力物力进行模板的设计与交互，同时为各种应用场景或领域分别设计对应的模板也是十分困难。近年来随着深度学习技术等方向的迅速发展，越来越多的研究通过利用神经生成模型提升 Text-to-SQL 问题的解析性能。基于 Seq2Seq 的方法已经成为 Text-to-SQL 解析的主流方法，主要是因为它们可以以端到端的方式进行训练，并且减少了对专门领域知识的需求。

目前更多研究尝试通过改进模型编解码器来提升性能表现。例如本文所复现的 LGESQL，便是利用图神经网络对数据库模式与给定问题之间的关系结构进行编码。这项工作旨在解决 Text-to-SQL 任务中具有挑战性的异构图编码问题，即数据库模式通常是以表、列和关系的形式表示的，而自然语言问题则是由词语和语法结构构成的，如何有效地将数据库模式和自然语言问题表示为图结构以便进行联合学习和推理，是对于该任务极为重要的问题之一。以前的方法通常以节点为中心，仅使用不同的权重矩阵来参数化边特征，这忽略了上下文信息，即嵌入在边拓扑结构中的丰富语义，同时无法区分每个节点的局部和非局部关系，每个节点将平等地关注所有其他节点，有可能导致过平滑问题。在这项工作中，作者提出了利用线图增强的 Text-to-SQL 模型（LGESQL 模型 [2]），主要特点是分别利用节点图和线图（互为对偶图）捕获节点与边的结构拓扑信息，从而改善模型对局部特征的捕获能力。LGESQL 达到了当时在基准测试 Spider 上的 SOTA 性能。

2 相关工作

2.1 基于上下文的语义解析

语义解析 (Semantic Parsing) 是自然语言处理 (NLP) 领域的一个重要任务，其目标是将自然语言文本映射到形式化的表示形式，把自然语言转化为机器可读可执行的逻辑语句，通常是逻辑形式、查询语言或程序。这个任务涉及将自然语言的语义结构转换为一种可以被计算机理解和处理的形式。所以语义解析在自动问答，对话系统里面有重要的用途。而基于上下文的语义解析是语义解析的一种扩展形式，它考虑了上下文信息，以更全面、准确地理解和处理自然语言的语义。传统的语义解析通常只关注单个句子或查询的语义，而基于上下文的语义解析则通过考虑周围文本的语境来提高解析的精度和鲁棒性，使得系统能够更好地理解和处理复杂的自然语言语境。基于上下文的语义解析现受到深度学习模型的影响，例如 LSTM、Transformers 等，这些模型能够更好地捕捉长距离上下文中的语义信息。

2.2 模式链接

在 Text-to-SQL 任务中，模式链接（Schema Linking）是指将自然语言问题中的实体（例如表、列等）映射到数据库模式中相应的元素。这是 Text-to-SQL 任务中的一个关键步骤，因为它涉及将自然语言问题中的词语或短语与数据库模式中的实体建立关联，从而为后续的 SQL 查询生成提供了重要的信息。模式链接的成功与否直接影响着后续 SQL 查询生成的准确性。如果系统能够准确地将自然语言中的实体映射到数据库模式中，那么生成的 SQL 查询就更有可能正确地涵盖用户的意图。

2.3 图神经网络应用

图神经网络是一类专门处理图结构数据的神经网络模型，其旨在捕捉节点之间的关系以及整个图的结构信息，从而能够对图上的节点进行学习和预测。图神经网络模型广泛应用于一些涉及到结构化信息或依赖于上下文关系的自然语言处理任务中。因此对于 Text-to-SQL 任务较为适配，此前在 Text-to-SQL 任务中使用以节点为中心的图神经网络聚合来自相邻节点的信息。GNNSQL [1] 采用关系图卷积网络 [5] 来考虑模式项之间的不同边类型。然而，这些边特征是直接从固定大小的参数矩阵中检索，缺乏上下文信息，尤其是边的结构拓扑。而元路径是指在图结构数据中定义的一类路径，通过定义一系列的节点类型和边类型来描述节点之间的结构，有助于 RATSQL 模型更好地理解数据库模式的结构，并在生成 SQL 查询时考虑到表和列之间的关系。RATSQL [7] 引入了一些有用的元路径，但它以相同的方式 (相对位置嵌入，[6]) 处理所有关系，而不区分局部与非局部关系，这往往可能导致过平滑问题。此外构建元路径也存在困难，元路径的数量随着路径长度呈指数增长。 [4]

3 本文方法

3.1 本文方法概述

本文在此前应用在 Text-to-SQL 任务上的图神经网络研究（RATSQL 等）的基础上，提出了 Line Graph Enhanced Text-to- SQL (LGESQL) 模型。首先模型从原始的节点中心图（node-centric graph）构造了一个边中心图（edge-centric graph），即线图。这两个图分别捕获了节点和边的结构拓扑。然后进行迭代，每个图中的每个节点从其邻域收集信息，并结合对偶图的边特征来更新其表示。对于节点中心图，我们将局部和非局部边特征结合到计算中。局部边特征表示单跳关系，由线图中的节点嵌入动态提供，而非局部边的特征直接从参数矩阵中直接提取。这种区别使模型在保持多跳邻接信息的同时更多地关注局部的边特征。同时作者还引入图修剪的辅助任务，要求模型可以从自然语言问题中知道哪些模式项会出现在最终的 SQL 语句中，通过进行多任务学习以提高模型的判别能力。

模型结构如图 2所示。模型主要分为以下三个部分：图输入模块、LGE 隐藏层模块以及输出模块。

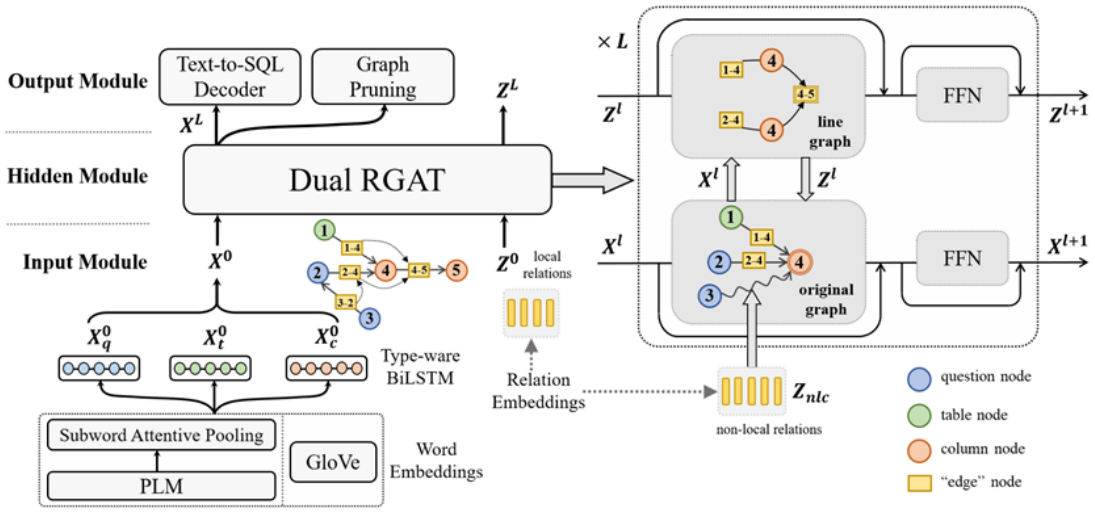

图 2. 系统框架图

3.2 图输入模块

图输入模块为节点和边提供初始的 embedding。对于边的初始特征，直接从一个参数矩阵中取出即可。对于节点的初始表示，我们分别从 Glove 和预训练模型（BERT 等）中获得。

3.3 LGE 隐藏层模块

模型的主要部分。LGE 隐藏层模块由 L 个堆叠的 dual relational graph attention network（Dual RGAT）层组成。每一层包含有两个 RGAT 模块来分别捕获原始图和 Line graph 的结构信息。每个图中的节点 embedding 都表示了对偶图中的边的特征信息。初始的线图构建只利用了 1 阶连接关系，对于节点图中的高阶邻域，论文采取静态动态混合特征或多头多视图拼接的方法，将更远的节点及其边类型引入到节点图中对节点特征的更新中，这些包含远距离信息的边特征不会迭代更新，仅仅使用静态参数初始化。

3.4 图输出模块

图输出模块主要包含两个任务：Text-to-SQL 的解码复现细节，以及图修剪任务。图修剪任务旨在通过输入的问题与数据库模式，判断哪些模式项可能出现在最终输出的 SQL 语句中，即通过对图进行剪枝操作，从而促进主任务效果。经过论文实验可以发现，此多任务学习能够有效地提升模型在 Text-to-SQL 任务上的性能。

4 复现细节

4.1 与已有开源代码对比

本文所复现的论文其代码已经全部开源。原代码链接：https://github.com/X-LANCE/text2sql-lgesql

代码结构如图 3所示：

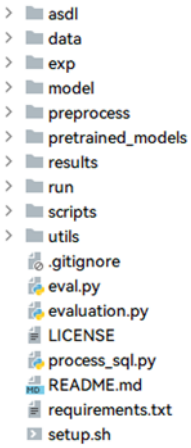

图 3. 代码结构

其中，/asdl 用于 SQL 抽象语法描述语言相关；/data 主要存放数据；/exp 用于存放实验训练模型参数以及日志；/preprocess 用于预处理；/pretrained_models 用于存储模型使用的预训练模型；/result 用于存放实验结果；/run 主要存放自动化批处理脚本；/scripts 用于 Text-to-SQL 脚本；/utils 用于一些工具类。

4.2 实验环境搭建

为实现论文所显示的实验，根据其提供的信息，本文复现环境如下：（1）Python 3.6；（2）PyTorch 1.6.0 + dgl 0.5.3 with CUDA10.1；（3）Tesla V100-PCIE-32GB；

所使用的数据集为 Spider。Spider 数据集是一个用于 Text-to-SQL 的广泛使用的数据集，该数据集规模相对较大，涉及跨多个领域数据以确保模型在不同主题下的泛化性能，并涵盖一定复杂性，包括联接多个表、使用聚合函数、存在子查询等，主要用于评估模型在理解自然语言问题并生成相应的 SQL 查询时的性能。

5 实验结果分析

论文进行的主要实验结果如图 4所示。可以看到作者所实现的 LGESQL 模型在使用不同模型配置下均取得了最先进的结果。其中使用词向量 GLOVE，性能从 57.2% 提高到 62.8%，绝对提高 5.6%。使用更大预训练模型能进一步解决上下文联合编码问题，结合 ELECTRA 达到了 72.0% 的准确率，验证了模型的优越性。 [8]

Model	Dev	Test
Without PLM		
GNN (Bogin et al., 2019a)	40.7	39.4
Global-GNN (Bogin et al., 2019b)	52.7	47.4
EditSQL (Zhang et al., 2019b)	36.4	32.9
IRNet (Guo et al., 2019)	53.2	46.7
RATSQL (Wang et al., 2020a)	62.7	57.2
LGESQL	**67.6**	**62.8**
With PLM: BERT		
IRNet (Guo et al., 2019)	53.2	46.7
GAZP (Zhong et al., 2020)	59.1	53.3
EditSQL (Zhang et al., 2019b)	57.6	53.4
BRIDGE (Lin et al., 2020)	70.0	65.0
BRIDGE + Ensemble	71.1	67.5
RATSQL (Wang et al., 2020a)	69.7	65.6
LGESQL	**74.1**	**68.3**
With Task Adaptive PLM		
ShadowGNN (Chen et al., 2021)	72.3	66.1
RATSQL+STRUG (Deng et al., 2021)	72.6	68.4
RATSQL+GRAPPA (Yu et al., 2020)	73.4	69.6
SmBoP (Rubin and Berant, 2021)	74.7	69.5
RATSQL+GAP (Shi et al., 2020)	71.8	69.7
DT-Fixup SQL-SP (Xu et al., 2021)	75.0	70.9
LGESQL+ELECTRA	**75.1**	**72.0**

图 4. 论文实验结果

复现的实验结果如图 5至图 8所示。我对模型从头开始训练并进行测试。可以清楚地看到，我复现的结果在 Spider 的不同难度的测试数据下与原论文所给出的测试结果正确数基本一致，并且最终的 exact match 也仅有微小的 0.1%0.3% 的变化，复现结果基本准确。

	easy	medium	hard	extra	all
count	248	446	174	166	1034
========== EXACT MATCHING ACCURACY ==========					
exact match	0.915	0.767	0.667	0.488	0.741

图 5. 原论文 LGESQL+BERT

	easy	medium	hard	extra	all
count	248	446	174	166	1034
========== EXACT MATCHING ACCURACY ==========					
exact match	0.915	0.762	0.667	0.482	0.738

图 6. 复现 LGESQL+BERT

	easy	medium	hard	extra	all
count	248	446	174	166	1034
========== EXACT MATCHING ACCURACY ==========					
exact match	0.919	0.783	0.649	0.524	0.751

图 7. 原论文 LGESQL+ELECTRA

	easy	medium	hard	extra	all
count	248	446	174	166	1034
========== EXACT MATCHING ACCURACY ==========					
exact match	0.919	0.778	0.649	0.524	0.750

图 8. 复现 LGESQL+ELECTRA

6 总结与展望

本文介绍了复现论文的基本情况，详细介绍了 LGESQL 的相关工作与具体实现方法。同时通过根据开源的代码进行了分析，并对其进行实验并成功有效地复现其工作。虽然本文没有对代码部分进行过多的改进，并且由于某些原因实验实验可能准备还不够充分，但是通过此次《计算机前沿技术》复现实验报告，我不仅动手实现并验证了这一论文及其有效性，并且通过这次学习更加深入地了解了整体系统的实现机理，这有助于我进一步深入学习有关研究领域及方向。

本文所复现论文在图神经网络上引入边结构拓扑，进一步加强模型对实体关系的理解，加强对模式链接的学习。所以我认为模式链接的准确性对于 Text-to-SQL 任务具有十分重要的意义，未来可以着重于有关方面对于如何加强模式链接来提升 Text-to-SQL 上的模型性能。

参考文献

[1] Ben Bogin, Jonathan Berant, and Matt Gardner. Representing schema structure with graph neural networks for text-to-sql parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Jan 2019.

[2] Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2541–2555, Online, August 2021. Association for Computational Linguistics.

[3] Naihao Deng, Yulong Chen, and Yue Zhang. Recent advances in text-to-sql: A survey of what we have and what we expect, 2022.

[4] Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, Fei Huang, and Yongbin Li. A survey on text-to-sql parsing: Concepts, methods, and future directions, 2022.

[5] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. *Modeling Relational Data with Graph Convolutional Networks*, page 593–607. Jan 2018.

[6] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Jan 2018.

[7] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jan 2020.

[8] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Jan 2018.