软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(3):1194−1206 [doi: 10.13328/j.cnki.jos.007074]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

Navi: 基于自然语言交互的数据分析系统*

谢宇鹏[1], 骆昱宇[2], 冯建华[3]

[1](青海大学 计算机技术与应用系, 青海 西宁 810016)

[2](香港科技大学(广州), 广东 广州 511400)

[3](清华大学 计算机科学与技术系, 北京 100084)

通信作者: 冯建华, E-mail: fengjh@tsinghua.edu.cn

摘 要: 随着大数据时代的到来, 数据分析的作用日益显著. 它能够从海量数据中发现有价值的信息, 从而更有效地指导用户决策. 然而, 数据分析流程中存在三大挑战: 分析流程高耦合、交互接口种类多和探索分析高耗时. 为了应对上述挑战, 提出了基于自然语言交互的数据分析系统 Navi. 该系统采用模块化的设计原则, 抽象出主流数据分析流程的 3 个核心功能模块: 数据查询、可视化生成和可视化探索模块, 从而降低系统设计的耦合度. 同时, Navi 以自然语言作为统一的交互接口, 并通过一个任务调度器实现了各功能模块的有效协同. 此外, 为了解决可视化探索中搜索空间指数级和用户意图不明确的问题, 提出了一种基于蒙特卡洛树搜索的可视化自动探索方法, 并设计了基于可视化领域知识的剪枝算法和复合奖励函数, 提高了搜索效率和结果质量. 最后, 通过量化实验和用户实验验证了 Navi 的有效性.

关键词: 数据分析; 数据查询; 可视化; 自然语言; 蒙特卡洛树搜索

中图法分类号: TP311

中文引用格式: 谢宇鹏, 骆昱宇, 冯建华. Navi: 基于自然语言交互的数据分析系统. 软件学报, 2024, 35(3): 1194–1206. http://www.jos.org.cn/1000-9825/7074.htm

英文引用格式: Xie YP, Luo YY, Feng JH. Navi: Data Analysis System Powered by Natural Language Interaction. Ruan Jian Xue Bao/Journal of Software, 2024, 35(3): 1194–1206 (in Chinese). http://www.jos.org.cn/1000-9825/7074.htm

Navi: Data Analysis System Powered by Natural Language Interaction

XIE Yu-Peng[1], LUO Yu-Yu[2], FENG Jian-Hua[3]

[1](Department of Computer Technology and Applications, Qinghai University, Xining 810016, China)

[2](The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China)

[3](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: With the advent of the big data era, the significance of data analysis has increasingly come to the forefront, showcasing its ability to uncover valuable insights from vast datasets, thereby enhancing the decision-making process for users. Nonetheless, the data analysis workflow faces three dominant challenges: high coupling in the analysis workflow, a plethora of interactive interfaces, and a time-intensive exploratory analysis process. To address these challenges, this study introduces Navi, a data analysis system powered by natural language interaction. Navi embraces a modular design philosophy that abstracts three core functional modules from mainstream data analysis workflows: data querying, visualization generation, and visualization exploration. This approach effectively reduces the coupling of the system. Meanwhile, Navi leverages natural language as a unified interactive interface to seamlessly integrate various functional modules through a task scheduler, ensuring their effective collaboration. Moreover, in order to address the challenges of exponential search space and ambiguous user intent in visualization exploration, this study proposes an automated approach for

* 基金项目: 国家自然科学基金(61925205, 62232009, 62102215); 中国铁路总公司科技研究开发计划(K2022S005)
本文由“面向多模态数据的新型数据库技术”专题特约编辑彭智勇教授、高云君教授、李国良教授、许建秋教授推荐.
收稿时间: 2023-07-17; 修改时间: 2023-09-05; 采用时间: 2023-10-24; jos 在线出版时间: 2023-11-08
CNKI 网络首发时间: 2023-12-22

visualization exploration based on Monte Carlo tree search. In addition, a pruning algorithm and a composite reward function, both incorporating visualization domain knowledge, are devised to enhance the search efficiency and result quality. Finally, this study validates the effectiveness of Navi through both quantitative experiments and user studies.

Key words: data analysis; data query; visualization; natural language; Monte Carlo tree search

步入大数据时代, 数据分析在商业智能、科学研究等诸多领域发挥着重要的作用[1]. 然而, 用户在进行数据分析过程中仍面临着数据分析流程高耦合、交互接口种类多、探索分析高耗时等挑战.

如图 1 所示, 主流的数据分析流程包括以下关键步骤: (1) 用户通过 SQL 查询或 Python 等方式从数据库中查询用于分析的数据子集; (2) 用户可以通过表格或者可视化的方式理解数据集蕴含的数据规律, 并使用可视化结果进行分析结果的呈现与交流; (3) 在可视化阶段, 用户通常使用可视化查询语言如 Vega-Lite[2]或交互式可视化工具如 Tableau[3]创建相应的可视化结果. 如果可视化结果不能满足用户的数据分析需求, 则用户可能会重复上述若干步骤.

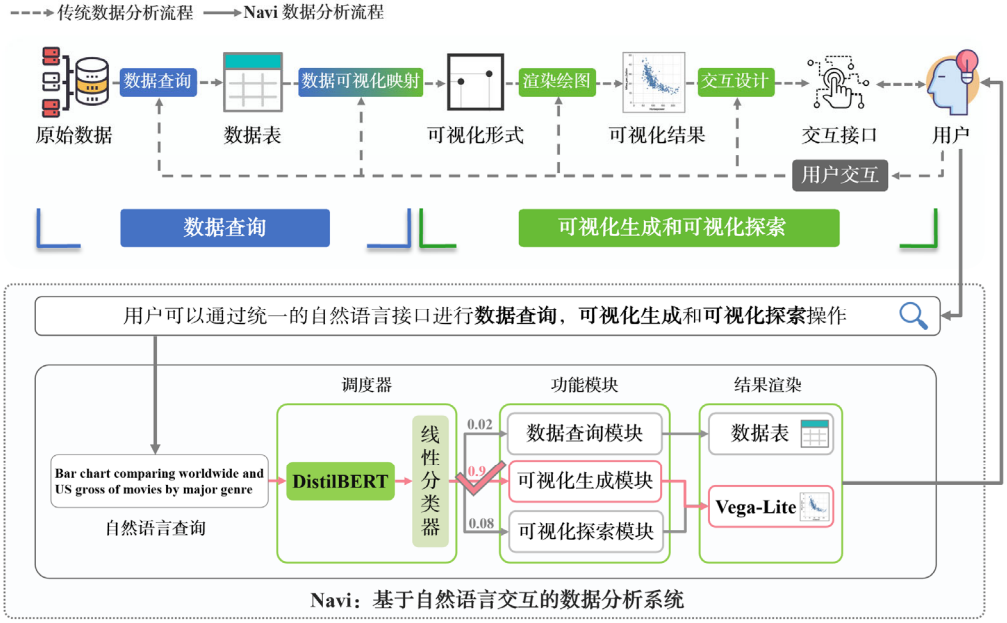


图 1 基于自然语言交互的数据分析系统 Navi 框架图

由此可见, 数据分析过程中的各个步骤相互依赖, 通常需要反复迭代以进行探索式分析, 分析流程耦合且耗时. 此外, 数据分析作为一项专业且复杂的任务, 对用户的专业技能和可视化分析能力的要求较高.

为降低用户进行数据分析的门槛和耗时并提高数据分析的质量和效率, 研究人员对数据分析的各流程进行了优化, 以提高人机协作的效率. 如表 1 所示, 现有工作主要优化数据分析流程中的某一环节, 难以同时应对分析流程高耦合、交互接口种类多和探索分析高耗时这三大挑战.

表 1 与现有工作比较

系统	数据查询	可视化生成	可视化探索
RATSQL[4]	✓	✗	✗
SEQ2VIS[5]	✗	✓	✗
DeepEye[6]	✗	✗	✓
KG4VIS[7]	✗	✗	✓
Navi	✓	✓	✓

为应对上述挑战, 如图 2 所示, 本文提出了以下研究目标.

(1) 分析流程模块化. 为提高数据分析流程的灵活性和可复用性, 本文将数据分析过程中的各个环节进行了抽象和模块化设计, 降低各环节之间的耦合度. 通过模块化设计, 能够灵活地利用不同的数据查询、可视化和交互方法, 实现多样化和定制化的数据分析需求.

(2) 交互接口统一化. 数据分析是一个涉及多种不同任务类型的复杂过程, 每个任务都具有特定的交互接口, 用户需要在不同任务之间进行转换. 为了应对交互接口种类多的挑战, 本文旨在设计一个统一的自然语言查询交互接口, 该接口能够有效地整合各类数据分析任务(如数据查询、可视化生成、可视化探索等), 以提高协作效率.

(3) 探索分析自动化. 探索式分析是数据分析的重要步骤, 传统的方式通常需要用户主动生成和选择可视化图表, 并通过不断迭代来优化分析结果. 上述过程繁琐、复杂、耗时且对用户专业技能要求高. 为了应对这些问题, 本文提出一种探索分析自动化的方法, 基于蒙特卡洛树搜索(Monte Carlo tree search, MCTS)[8]算法, 结合库内数据特征、可视化领域知识和用户分析偏好, 以自动地探索数据集并生成有价值的可视化图表, 而无须用户过度的干预和反馈, 有效地提高了探索分析的效率和质量.

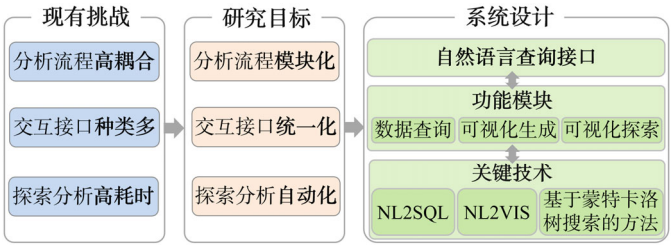


图 2 本文的研究目标和系统设计

为了实现上述研究目标, 本文设计并实现了一个基于自然语言交互的数据分析系统 Navi, 如图 1 所示. Navi 采用模块化的设计原则, 将主流的数据分析流程[9]抽象为 3 个功能模块: 数据查询模块、可视化生成模块和可视化探索模块(实现目标(1)). 并以自然语言查询作为统一的交互接口, 设计了一个任务调度器, 负责解析用户输入的自然语言查询, 并将其分发给相应的功能模块(实现目标(2)). 如图 2 的系统设计所示: Navi 系统的各模块均基于深度学习实现, 数据查询模块的关键技术是将自然语言查询转换为对应的 SQL 查询(natural language to SQL, NL2SQL), 可视化生成模块的关键技术是将自然语言查询转换为对应的可视化查询(natural language to visualization, NL2VIS). 对于可视化探索模块, 本文提出了基于蒙特卡洛树搜索的方法, 以高效地探索可视化空间(实现目标(3)).

因此, Navi 可为用户提供一个基于自然语言查询交互的数据分析系统, 实现数据查询、可视化生成和可视化探索功能. 综上, 本文的主要贡献如下.

(1) 采用模块化的数据分析系统设计原则, 将主流的数据分析流程抽象为 3 个常见的模块, 降低了系统设计的耦合度(第 1 节).

(2) 提出使用自然语言查询接口作为用户统一的交互接口, 并实现了基于自然语言的数据查询、可视化生成和探索模块(第 2 节).

(3) 提出一种基于 MCTS 的可视化自动探索方法, 实现对数据集的自动探索和可视化, 并设计基于可视化领域知识的剪枝算法和复合奖励函数, 从而提高探索效率和分析质量(第 3 节).

(4) 基于上述技术, 本文实现了端到端的基于自然语言交互的数据分析系统 Navi, 并通过实验验证了其有效性(第 4 节).

1 Navi 系统概述

1.1 模块化设计

Navi 采用模块化的设计原则, 包括数据查询、可视化生成和可视化探索模块这 3 个功能模块, 如图 3 所示. 模块化设计降低了系统设计的耦合度. 以下是各个模块的功能介绍.

- 数据查询模块. 该模块能够将用户输入的自然语言查询转换为 SQL 查询, 并返回查询结果, 如图 3③所示. 该模块支持对数据表进行选择和连接以及对数据进行筛选、排序和聚集等操作, 旨在返回用户感兴趣的数据子集, 提供了便捷的数据查询体验. 用户还可以对数据子集进行收藏或切换操作, 以便基于所需分析的数据子集进行进一步的探索, 如图 3④所示.
- 可视化生成模块. 该模块根据用户输入的自然语言查询分析用户意图, 并根据数据特征和分析目标生成相应的可视化结果, 如图 3⑤所示.
- 可视化探索模块. 该模块主要协助用户理解数据集. 通过结合数据特征生成多种可视化结果, 展示数据规律, 如图 3⑥所示. 该模块适用于用户分析意图不明确的场景, 能够为用户提供一个分析的起点.

图 3 Navi 系统概览

这种模块化的设计原则使 Navi 具备灵活性、可扩展性和兼容性: 灵活性体现在用户可以根据自己的需求, 选用不同的功能模块, 并且这些模块之间可以无缝切换, 通过协作的方式完成整个数据分析任务; 可扩展性体现在 Navi 可以随时替换或更新某个模块, 以适应技术的发展; 而兼容性体现在 Navi 能够与其他遵循相同交互接口的系统进行模块复用, 实现功能的互通.

1.2 任务调度器

任务调度器将各功能模块有效地组合起来, 其作用是解析用户输入的自然语言查询, 并将其分发给相应的功能模块. 本节把任务调度器视为一个多分类器, 并介绍其实现步骤.

- 数据集构建. 本节构建了一个包含大量自然语言查询及其对应模块标签的数据集, 用于训练和验证任务调度器. 该数据集涵盖了各类查询, 包括 NL2SQL、NL2VIS 和可视化探索. 本节从现有的 NL2SQL 基准 Spider[10]和 NL2VIS 基准 nvBench[5]中提取自然语言查询, 并为其打上相应的标签. 同时, 还将 nvBench 中的自然语言查询转化为适用于可视化探索模块的查询, 这些查询更简洁且模糊, 不包含明确的可视化类型、排序要求等详细信息.
- 模型训练. 任务调度器采用预训练的 DistilBERT[11]模型作为基础模型, 该模型基于 Transformer[12]架

构, 具有较少的参数和较低的计算复杂度, 但在性能上仍然优异. 本节用 Hugging Face Transformers 库中的 AutoTokenizer 将查询文本转换为 DistilBERT 输入格式, 并将数据集划分为训练集、验证集和测试集. 为构建任务特定的调度器, 本节在 DistilBERT 的基础上添加了一个具有 3 个输出单元的线性分类器, 使用交叉熵损失作为损失函数, 并采用了带权重衰减的 AdamW[13]优化器进行权重更新.

引入任务调度器, 使得 Navi 具有全面性和易用性. Navi 通过灵活组合功能模块, 以满足用户对数据查询、可视化生成和探索的需求. 同时, 它采用自然语言查询作为交互接口, 简化用户的操作, 实现与数据的“对话”式探索.

2 基于自然语言的数据查询和可视化生成

数据查询和可视化生成本质上同属一类任务, 它们都涉及将输入序列(例如自然语言查询)转换成输出序列(例如 SQL 或 Vega-Lite[2]查询). 为了解决此类问题, 本节采用序列到序列(Sequence-to-Sequence, Seq2Seq)[14]模型进行具体实现.

Seq2Seq 模型一般由编码器和解码器两部分组成, 可以采用不同的神经网络实现. 如图 4 所示, 编码器的任务是理解输入序列并生成一个较小的向量 h 来表示输入, 而解码器的任务是根据 h 生成一系列输出. 在本节中, 自然语言查询作为输入是由令牌(或单词)组成的序列$[q_1,q_2,\ldots,q_l]\in V_{in}$, 而 SQL 查询或 Vega-Lite[2]查询作为输出也是由令牌组成的序列$[y_1,y_2,\ldots,y_k]\in V_{out}$. 这里, V_{in} 和 V_{out} 分别代表输入和输出的词汇表. 接下来, 本节将基于 Transformers[12]介绍编码器和解码器网络的实现.

- 基于 Transformer 的编码器. 给定输入序列 $n_V=[q_1,q_2,\ldots,q_l]$, 本节将 n_V 与数据库表结构信息 $A=[a_1,a_2,\ldots,a_m]$连接起来. 连接后的输入序列为

$$X=[x_1,x_2,\ldots,x_n]=[q_1,q_2,\ldots,q_l,a_1,a_2,\ldots,a_m] \tag{1}$$

其中, $n=l+m$. 然后, 本节使用预训练的全局词嵌入(GloVe)[15]将每个令牌 x_i 映射为其向量表示. 完成令牌嵌入后, 本节将嵌入的令牌输入到双向 Transformer 网络中, 输出一系列编码向量:

$$h=[h_1,h_2,\ldots,h_n] \tag{2}$$

- 基于 Transformer 的解码器. 解码器同样采用基于 Transformer 的架构, 并引入了注意力机制[16]. 根据隐藏状态 h, 解码器生成输出序列:

$$Y=[y_1,y_2,\ldots,y_k] \tag{3}$$

在每个时间步 t, 解码器根据当前状态 s_i、先前的令牌和注意力向量 c_i 来预测 SQL 查询令牌 y_t.

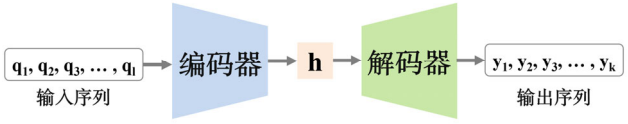


图 4 序列到序列模型

基于该模型, 本章只需对参数进行适当调整, 然后利用大量与任务相关的输入序列和输出序列样本对作为训练数据, 即可分别训练得出数据查询和可视化生成模型. 具体而言, 数据查询模块的训练采用了跨领域 NL2SQL 基准 Spider[10]中的自然语言查询和 SQL 查询(NL,SQL)样本对. 例如: 对于自然语言查询输入: “fetch all COVID-19 cases”, 相应的 SQL 查询输出为: “select * from COVID-19”. 而可视化生成模块的训练则使用 NL2VIS 基准 nvBench[5]中的自然语言查询和可视化查询(NL,VIS)样本对. 例如: 一个示例的自然语言查询输入是: “draw a line chart to show the trend of numbers of cases by each case type in Utah”, 其对应的输出类似于 Vega-Lite[2], 如: “mark line encoding x date y aggregate none number color cases transform filter states=‘Utah’”.

3 基于 MCTS 的可视化自动探索方法

可视化探索的目标是根据用户需求和数据特征, 自动生成并展示合适的可视化结果. 实现该目标面临两

大挑战: 一是如何在庞大的可视化设计空间中搜索和评估候选的可视化结果; 二是如何在用户意图不明确的情况下, 根据用户提供的自然语言查询推测出用户偏好, 从而生成出最符合用户意图的可视化结果.

为应对上述挑战, 本节提出了一种基于蒙特卡洛树搜索的可视化自动探索方法. 该算法利用 MCTS 的高效搜索策略, 能够有效地生成并评估不同的可视化查询, 从而推荐出高质量的可视化结果. 可视化查询是一种用于描述可视化图表的语言, 它由一系列可视化子句组成, 每个子句描述了一个可视化图表的某个方面, 如数据列、图表类型、聚集操作等. 在 MCTS 树 T 中, 每个节点 v 对应一个可视化子句 c. 例如: 在图 5 中, 第二层的节点 *bar* 就对应了图表类型子句. 因此, 可视化查询 Q 可以被表示为从根节点到叶节点的一条路径 $Q=[v_1,v_2,\ldots,v_n]$, 其中, n 是查询中子句的数量.

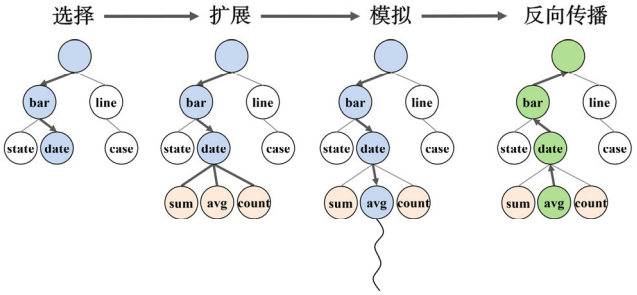


图 5 蒙特卡洛树搜索的流程

针对第 1 个挑战, 本节采用了数据集解析和逐步搜索的方法构建了一个包含多个可视化子句的搜索树.在搜索过程中, 本节引入了基于可视化领域知识的剪枝算法和复合奖励函数, 旨在引导系统以更高效的方式选择最佳的可视化结果.

为应对第 2 个挑战, 本节对用户输入的自然语言查询进行意图提取, 以获取与分析任务(如分布、趋势、对比)和可视化子句(如数据列名、数据聚集方式)相关的关键词. 基于这些关键词, 本节能够在搜索树节点的扩展阶段根据用户偏好限定节点的扩展方向, 使得搜索树向用户偏好的方向进行搜索, 并进一步对搜索空间进行剪枝操作. 如图 5 所示, 该模块具体包括以下 4 个步骤.

(1) 选择. 在选择阶段, 算法会从根节点出发, 逐步递归地寻找最优的子节点, 一直持续到抵达一个尚未完全展开的节点为止. 如图 5 所示, 蓝色节点代表本次选择阶段所选择的节点. 为了更好地利用奖励函数提供的反馈信息, 并平衡探索和利用的关系, 以避免陷入局部最优解, 本节采用了上置信界(upper confidence bound, UCB)[17]算法. 该算法的核心思想是: 在每次迭代中, 选择具有最高置信上界的子节点进行下一步的探索. UCB 算法的具体选择策略可以使用以下公式表示:

$$UCB(i)=\bar{X}_i+C\sqrt{\frac{\ln N}{n_i}} \tag{4}$$

其中, $UCB(i)$是第 i 个子节点的上置信界, $\bar{X}_i$ 是第 i 个子节点的平均奖励值, N 是当前节点的访问次数, n_i 是第 i 个子节点的访问次数, C 是一个控制探索与利用平衡的参数. 通过这种选择策略, UCB 算法可以确保在有限的模拟次数内, 既能充分探索节点空间, 又能有效利用已知信息.

(2) 扩展. 在扩展阶段, 为了高效地缩减搜索范围, 本节引入了基于可视化领域知识的剪枝算法. 此算法综合考虑了数据类型、图表类型以及编码规则等多个维度的信息, 以筛选出具有较高潜在价值的候选操作.

(3) 模拟. 在模拟阶段, 算法选取当前节点作为模拟的起点, 并在时间限制的条件下, 依据预设的约束规则对其进行扩展. 在更新模拟节点的过程中, 算法将优先选取潜在价值最高的操作, 并计算模拟结果的奖励值以优化决策过程.

(4) 反向传播. 在反向传播阶段, 算法会沿着已确定的路径, 更新节点的访问次数和累积奖励, 如图 5

所示. 其中, 每个可视化结果的得分都将通过奖励函数进行计算.

- 奖励函数设计

由于可视化没有像围棋那样清晰的奖惩规则, 且单一评判标准得出的奖励可能存在偏差, 因此, 本节考虑了数据特征、可视化领域知识和用户偏好这 3 个方面, 综合评价一个可视化结果的质量.

(1) 在数据特征方面, 本节通过 LambdaMART[18]实现评分函数. 这种方法可以从数据中提取有用的特征(例如最大值、最小值、唯一值的比率等等), 并利用机器学习为可视化图表评定分数.

(2) 在可视化领域知识方面, 本节采用了基于规则的评分策略. 这些领域知识规则旨在确保所生成的可视化结果能够与人类的视觉感知相适应, 同时, 真实地展现数据的核心属性. 以柱状图为例, 一个含有超过 50 根柱子的图表可能会导致用户难以捕捉重要信息. 因此, 通过整合领域知识作为约束, 本章可以为不同类型的可视化设置不同的规则, 从而更有效地指导探索的方向.

(3) 在用户偏好方面, 本节将其划分为两个部分: 起始阶段的可视化初始评分和交互阶段的当前用户评分. 在进行可视化探索的起始阶段, 由于缺乏当前用户的偏好信息, 本章采用生成对抗网络[19]训练了一个用户模型并进行评估. 该模型的训练数据来源于 Plotly 社区的可视化语料库[20], 其中包含了真实用户的历史交互记录. 通过这些记录, 模型能够学习到用户偏好和行为特征的共性, 进而对可视化结果给出更为准确的初始评价. 在起始阶段, 该模型的评价结果被视为主要的评判标准.

在交互阶段, 系统会实时记录用户的交互行为, 并引入推荐系统[21]中的实时反馈机制[22]来动态调整用户评分. 针对不同的用户操作行为, 本节设计了一套奖励机制. 当用户对某个可视化结果执行点击、保存或编辑等操作时, 系统会根据预设的规则, 将这些操作转换为具体的数值, 并动态地调整相应可视化结果和子句的得分. 通过这种方式, 系统能够为用户感兴趣的部分赋予更高的奖励分数. 进入下一轮蒙特卡洛树搜索时, 算法会倾向于探索用户感兴趣的方向, 从而生成更符合用户实际需求的可视化结果[23,24]. 在整合用户偏好评分方面, 系统采用加权组合的方式, 将初始评分和当前用户评分相结合, 并为当前用户评分赋予更高的权重. 这样做旨在更好地满足用户当前的偏好, 并鼓励他们进行更深入的数据探索.

接下来, 本节将介绍如何利用这3个部分来综合评判一个可视化结果的好坏. 对于一个可视化结果, 本节首先根据领域知识进行初步筛选: 若不符合领域知识, 则直接赋予较低的分数; 若符合, 则进一步根据数据特征进行评分, 以得到数据特征分. 随后, 根据用户模型和当前用户偏好计算得出用户偏好分. 最后, 通过对这两个分数进行加权平均, 计算出该可视化结果的最终分数, 其中, 权重值是基于经验进行设定的.

综上所述, 本节提出了一种基于 MCTS 的可视化自动探索方法, 该方法结合了基于可视化领域知识的剪枝算法和复合奖励函数, 提高了搜索的准确性和效率.

4 实验评测

4.1 量化实验

4.1.1 数据查询模块的有效性评测

本实验采用 Spider[10]基准数据集来评测数据查询模块的有效性. Spider 是一个大规模、复杂和跨领域的自然语言查询到 SQL 查询的数据集, 它包含了 10 181 个(NL,SQL)样本对, 这些样本对分布在 200 个数据库中, 覆盖了 138 个不同的领域. 本实验在其测试集上进行评估, 并确保训练集和测试集之间没有自然语言查询和数据库的重复.

本实验通过两个指标来量化模型的性能[25]: 精确集合匹配准确率和执行准确率. 精确集合匹配准确率是指生成的 SQL 查询与真实的 SQL 查询在标准化后的数据结构上是否完全一致, 执行准确率是指生成的 SQL 查询在数据库上执行后得到的结果是否与真实的 SQL 查询得到的结果相同.

实验结果见表 2: 数据查询模块在两个指标精确集合匹配准确率和执行准确率上都取得了较高的分数, 分别为 71.08%和 74.37%. 与现有工作 BRIDGE[26]和 RATSQL[4]相比, 数据查询模块有显著改进, 在精确集合匹配准确率方面分别提升了 5.06%和 2.95%, 在执行准确率方面分别提升了 7.19%和 5.16%.

表 2 数据查询模块的实验结果(%)

系统	精确集合匹配准确率	执行准确率
BRIDGE[26]	66.02	67.18
RATSQL[4]	68.13	69.21
数据查询模块	**71.08**	**74.37**

这表明数据查询模块可以更有效地将自然语言转换为正确且可执行的 SQL 查询, 并更好地适应不同领域和复杂度的数据库.

4.1.2 可视化生成模块的有效性评测

本实验采用 nvBench[5]基准数据集来评测可视化生成模块的有效性. nvBench 是一个大规模、复杂和跨领域的 NL2VIS 任务的数据集, 由 750 张关系表和 25 750 个(NL,VIS)样本对组成, 涵盖了 105 个不同的领域. 本实验在其测试集上进行评测, 并使用准确率作为评测指标. 准确率是指生成的可视化查询与真实的可视化查询是否完全匹配, 即 $Accuracy=N/M$, 其中, N 是匹配的结果数量, M 是测试样本的数量.

实验结果见表 3: 可视化生成模块的准确率达到了 76.78%, 显著优于的现有工作 NL2Viz[27]和 SEQ2VIS[5], 在准确率方面分别提升了 18.13%和 12.66%. 这说明可视化生成模块可以灵活地处理不同类型和复杂度的自然语言查询, 并准确地生成符合用户意图和数据特征的可视化图表. 这证明可视化生成模块能够很好地实现可视化生成的目标.

表 3 可视化生成模块的实验结果(%)

系统	准确率
NL2Viz[27]	58.65
SEQ2VIS[5]	64.12
可视化生成模块	**76.78**

4.1.3 可视化探索模块的有效性评测

本实验采用 KG4VIS[7]的数据集来评测可视化探索模块的有效性. KG4VIS 是一个基于知识图谱的可视化探索方法, 该工作从 VizML[28]中筛选出 88 548 个(数据集,可视化集)样本对, 并将其按 7:3 的比例分为训练集和测试集. 本实验使用其测试集进行评测, 并沿用了 KG4VIS 的评价指标, 包括:

- 平均排名: 表示正确的可视化设计选择在所有生成结果中的平均位置, 越低越好;
- *Hits*@2: 表示正确的可视化设计选择出现在前两个生成结果中的概率, 越高越好;
- 轴准确率: 表示生成结果中轴属性(即 x 轴或 y 轴)与真实结果一致的概率, 越高越好.

本次实验着重评估系统首次生成的可视化结果的质量. 因此, 奖励函数的用户偏好部分主要以可视化的初始评分作为评判标准. 实验结果见表 4: 本文的可视化探索模块在 3 个指标上都取得了较好的成绩, 分别为 1.726 8, 87.22%和 98.12%. 与现有工作 KG4VIS[7]和 DeepEye[6]相比, 本文的可视化探索模块有明显的优势, 在平均排名方面分别降低了 0.341 4 和 0.277 6, 在 *Hits*@2 方面分别提高了 12.15%和 9.94%, 在轴准确率方面分别提高了 38.11%和 4.93%. 这说明可视化探索模块可以更智能地根据数据特征和用户意图生成合适的可视化图表, 并更精确地确定轴属性和数据映射. 这证明了可视化探索模块能够较好地实现可视化探索的目标.

表 4 可视化探索模块的实验结果

系统	平均排名	*Hits*@2 (%)	轴准确率(%)
KG4VIS[7]	2.068 2	75.07	60.01
DeepEye[6]	2.004 4	77.28	93.19
可视化探索模块	**1.726 8**	**87.22**	**98.12**

- 消融实验

可视化探索模块中的复合奖励函数综合了数据特征、可视化领域知识和用户偏好这 3 个方面的评分. 其中, 用户偏好部分包括了起始阶段的可视化初始评分和交互阶段的当前用户评分两部分. 为验证本文提出的复合奖励函数的有效性, 本节设计了 5 种 Navi 的变种实现, 如下所示.

(1) Navi: 评测 Navi 在不考虑当前用户评分的情况下, 首次生成的可视化结果的质量.

(2) Navi 除去可视化领域知识的评分.

(3) Navi 除去数据特征的评分.

(4) Navi 除去用户偏好的初始评分.

(5) Navi+: 评测当用户进行多轮交互后, 结合当前用户评分情况下, 所生成的可视化结果的质量.

表 5 展示了消融实验的结果, 从表中可以看出: 当用户参与交互后, 考虑当前用户的评分时所得到的可视化结果质量是最高的, *Hits*@2 和轴准确率分别为 90.36%和 98.83%; 相比之下, 如果不考虑当前用户评分, 只评估 Navi 首次生成的可视化结果质量, 则 *Hits*@2 和轴准确率分别降低了 2.64%和 1.02%; 此外, 如果只评估 Navi 首次生成的可视化结果的质量, 并分别去掉可视化领域知识评分、数据特征评分或用户偏好的初始评分中的任意一个, 则 *Hits*@2 和轴准确率都会有所下降. 这些实验结果表明: 复合奖励函数中考虑的 3 个方面都对可视化探索模块的性能有正向影响; 而且, 通过记录用户的交互行为, 并根据其调整奖励函数的评分, 可以进一步提升生成的可视化结果的质量. 因此, 本文设计的复合奖励函数能够更全面地评价不同可视化结果的优劣, 并更有效地指导可视化探索模块生成高质量的可视化结果.

表 5 复合奖励函数的消融实验结果(%)

方法	*Hits*@2	轴准确率
Navi 除去可视化领域知识的评分	78.56	92.15
Navi 除去数据特征的评分	77.82	89.31
Navi 除去用户偏好的初始评分	76.78	91.22
Navi(无用户交互)	87.72	97.81
Navi+(有用户交互)	**90.36**	**98.83**

4.1.4 任务调度器的有效性评测

本实验在第 1.2 节任务调度器所述的测试集上进行评测, 该测试集是基于 Spider[10]和 nvBench[5]数据集构造的. 评测的目的是: 检验任务调度器能否准确地识别用户输入的自然语言查询, 并将其分配到合适的功能模块. 本实验采用 3 个评价指标, 即准确率、召回率和 *F*1 值: 准确率表示任务调度器正确分配查询到相应模块的比例; 召回率表示正确分配给相应模块的查询数占总应分配数的比例; *F*1 值表示准确率和召回率的调和平均值, 反映了两者之间的平衡.

表 6 展示了任务调度器的实验结果, 任务分类模型在测试集上达到了 98.22%的准确率、98.23%的召回率和 98.55%的 *F*1 值. 这些结果表明: 本文的任务调度器能够有效地识别不同类型的分析任务, 并将其准确地分配到相应的模块中.

表 6 任务调度器的实验结果(%)

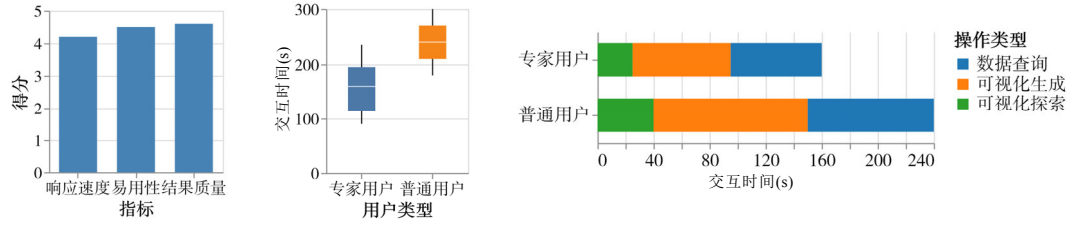
模块	准确率	召回率	*F*1 值
任务调度器	**98.22**	**98.23**	**98.55**

4.2 用户实验

本节邀请了 5 名专家用户和 15 名普通用户参与用户实验, 评测了 Navi 端到端的效果.

- 用户任务. 本实验为每位用户设计了涉及数据查询、可视化生成和可视化探索这 3 个方面的不同数据分析任务. 用户通过自然语言查询与系统交互, 并根据系统返回的结果完成相应的目标.
- 用户评价. 在完成每个任务后, 本实验邀请用户填写一个问卷, 对系统在易用性、响应速度、结果质量方面进行评分. 同时, 本节也记录了用户完成每个任务所花费的时间和操作日志.
- 实验结果. 图 6(a)展示了 Navi 在各个评价维度上的用户满意度, 其平均得分均超过了 4 分(满分为 5 分). 这一结果充分体现了其在数据分析支持上的出色表现, 为用户带来了便捷且高效的使用体验. 图 6(b)显示了不同类型的用户在使用 Navi 时的交互时间, 可以看出, 部分普通用户的交互时间已经接近专家用户的水平, 说明 Navi 能够有效地提升用户的数据分析能力和效率. 而从图 6(c)中可以观察到: 在进行可视化探索任务时, 用户的平均交互时间最短, 这不仅凸显了可视化探索模块的高效

性, 同时也说明了 Navi 在帮助用户快速揭示数据规律方面的有效性.



(a) 用户评价 (b) 用户交互时间 (c) 不同操作类型的交互时间

图 6 用户实验结果

5 相关工作

- NL2SQL

在当今大数据时代, 从庞大的数据集中高效地检索出有用的信息显得尤为重要[29,30]. NL2SQL 技术应运而生, 它允许用户通过自然语言查询的交互方式, 更方便且高效地从结构化数据中检索相关信息. 近年来, 基于深度学习的 NL2SQL 模型在数据库和自然语言处理领域取得了显著的研究进展[31−36]. 典型的 NL2SQL 模型如 RESDSQL[32]、RASAT[33]以及 Graphix-T5[34], 通常都采用基于微调(fine-tuning)的方式进行训练. 这些模型需要大量的(NL,SQL)样本对来进行训练, 并在诸如 Spider[10]等基准数据集上进行效果评估. 近期, 随着大语言模型(large language model, LLM)的兴起, 一些研究者开始探索其在 NL2SQL 任务上的应用潜力. 通过零样本[35]和少样本提示, 模型如 C3[35]和 DIN-SQL[36]在 NL2SQL 任务上均展现出较好的效果. 然而, 当前大语言模型在处理数据库中多表和多列间关系时仍然面临挑战, 这导致它在处理复杂查询任务时的性能不如传统的基于微调的方法[32].

- NL2VIS

NL2VIS 的实现方法主要分为基于规则的方法和基于深度学习的方法[37,38], 其中, 基于规则方法的代表性研究工作有 NL4DV[39]、QRec-NLI[40]和 NL2Viz[27]. 此类方法主要基于语义解析器(如 NLTK[41]、Stanford CoreNLP[42]和 NER[43])对自然语言查询进行解析, 进而提取其中的词性、命名实体等语言特征, 最终通过启发式规则转换为对应的可视化查询. 但这类方法在处理自然语言的模糊性上存在局限性, 鲁棒性有待加强.

近年来, 深度学习在自然语言处理领域的突破性进展[37]为 NL2VIS 提供了新的技术路径. 研究者开始将深度学习引入 NL2VIS 任务, 其中具有代表性的研究工作有 ADVISor[44]、ncNet[45]和 Chat2vis[46]. ADVISor 采用了基于 BERT[47]的框架, 先完成 NL2SQL 的映射, 再基于规则生成可视化查询. ncNet 则是一个基于 Transformer[12]实现的端到端模型, 它使用了 NL2VIS 的首个公共基准数据集 nvBench[5]进行训练, 该数据集汇集了各种领域与场景下的(NL,VIS)样本对, 为深度学习模型的训练与评估提供了标准. Chat2vis 则巧妙地将大语言模型(如 Codex[48]和 GPT-3[49])与提示工程[50]相结合, 实现了自然语言查询到可视化查询的转换, 推进了 NL2VIS 研究的发展.

- 可视化探索

创建符合用户分析意图的可视化结果是一项充满挑战的任务, 它需要用户了解数据、明确分析意图并熟悉可视化技术. 因此, 研发能够自动进行可视化探索的数据分析系统变得尤为重要. 例如, Draco[51]和 DeepEye[6,52,53]都能为特定数据集提供排序后的可视化结果. 其中, Draco 采用硬约束与软约束相结合的方法, 以识别并排序出优质的可视化结果; 而 DeepEye 则采用了决策树模型, 评估可视化图表的质量, 并利用 Learning-to-rank[54]技术进行图表排序. 为了使可视化探索过程更加贴切用户的分析意图, 一些研究工作尝试引入更先进的技术. 例如: LineNet[55]采用基于 Vision Transformer[56]的 Triplet Autoencoder 结构, 能够根据用户提供的折线图图像, 自动推荐相似的折线图结果; Lux[57]是一个可集成在 Jupyter Notebook 中的工具, 能够根据用户的实际分析意图, 实时地提供数据分析支持; Sevi[58]则尝试采用语音交互方式来获取用户的分析意图,

使用户能够通过语音交互方式来便捷地进行可视化的创建和探索.

6 结论与未来的工作

本文设计并实现了一个基于自然语言交互的数据分析系统 Navi, 其采用模块化的设计原则, 集成了数据查询、可视化生成和可视化探索这 3 个功能模块, 并由任务调度器统一管理. 量化实验和用户实验表明, Navi 能够有效地支持用户进行数据分析操作. 未来的工作主要包括: 首先, 支持多轮对话, 融合上下文信息, 使用户能够持续地进行数据分析操作; 其次是探索与大语言模型的结合, 利用大语言模型在自然语言处理方面的优势, 提高 Navi 在各个任务上的泛化能力和效果.

References:

[1] Ward MO, Grinstein G, Keim D. Interactive Data Visualization: Foundations, Techniques, and Applications. CRC Press, 2010.

[2] Satyanarayan A, Moritz D, Wongsuphasawat K, *et al*. Vega-Lite: A grammar of interactive graphics. IEEE Trans. on Visualization and Computer Graphics, 2016, 23(1): 341–350.

[3] Tableau. 2023. https://www.tableau.com/

[4] Shi P, Ng P, Wang Z, *et al*. Learning contextual representations for semantic parsing with generation-augmented pre-training. Proc. of the AAAI Conf. on Artificial Intelligence, 2021, 35(15): 13806–13814.

[5] Luo Y, Tang N, Li G, *et al*. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In: Proc. of the 2021 Int'l Conf. on Management of Data. 2021. 1235–1247.

[6] Luo Y, Qin X, Chai C, *et al*. Steerable self-driving data visualization. IEEE Trans. on Knowledge and Data Engineering, 2020, 34(1): 475–490.

[7] Li H, Wang Y, Zhang S, *et al*. KG4Vis: A knowledge graph-based approach for visualization recommendation. IEEE Trans. on Visualization and Computer Graphics, 2021, 28(1): 195–205.

[8] Browne CB, Powley E, Whitehouse D, *et al*. A survey of Monte Carlo tree search methods. IEEE Trans. on Computational Intelligence and AI in Games, 2012, 4(1): 1–43.

[9] Munzner T. Visualization Analysis and Design. CRC Press, 2014.

[10] 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. arXiv:1809.08887, 2018.

[11] Sanh V, Debut L, Chaumond J, *et al*. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108, 2019.

[12] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: Advances in Neural Information Processing Systems, Vol. 30. 2017. 5998–6008.

[13] Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv:1711.05101, 2017.

[14] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, Vol. 27. 2014. 3104–3112.

[15] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). 2014. 1532–1543.

[16] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014.

[17] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 2002, 47: 235–256.

[18] Wu Q, Burges CJC, Svore KM, *et al*. Ranking, boosting, and model adaptation. Technical Report, MSR-TR-2008-109, Microsoft Research, 2008.

[19] Goodfellow I, Pouget-Abadie J, Mirza M, *et al*. Generative adversarial networks. Communications of the ACM, 2020, 63(11): 139–144.

[20] Qian X, Rossi RA, Du F, *et al*. Personalized visualization recommendation. ACM Trans. on the Web (TWEB), 2022, 16(3): 1–47.

[21] Afsar MM, Crump T, Far B. Reinforcement learning based recommender systems: A survey. ACM Computing Surveys, 2022, 55(7): 1–38.

[22] Chen M, Chang B, Xu C, *et al*. User response models to improve a reinforce recommender system. In: Proc. of the 14th ACM Int'l Conf. on Web Search and Data Mining. 2021. 121−129.

[23] Baghi V, Motehayeri SMS, Moeini A, *et al*. Improving ranking function and diversification in interactive recommendation systems based on deep reinforcement learning. In: Proc. of the 26th Int'l Computer Conf., Computer Society of Iran (CSICC). IEEE, 2021. 1−7.

[24] Xiao T, Wang D. A general offline reinforcement learning framework for interactive recommendation. Proc. of the AAAI Conf. on Artificial Intelligence, 2021, 35(5): 4512−4520.

[25] Zhong RQ, Yu T, Klein D. Semantic evaluation for text-to-SQL with distilled test suites. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). 2020. 396−411.

[26] Lin XV, Socher R, Xiong C. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. arXiv:2012.12627, 2020.

[27] Wu Z, Le V, Tiwari A, *et al*. NL2Viz: Natural language to visualization via constrained syntax-guided synthesis. In: Proc. of the 30th ACM Joint European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. 2022. 972–983.

[28] Hu K, Bakker MA, Li S, *et al*. Vizml: A machine learning approach to visualization recommendation. In: Proc. of the 2019 CHI Conf. on Human Factors in Computing Systems. 2019. 1−12.

[29] Qin X, Chai C, Luo Y, *et al*. Interactively discovering and ranking desired tuples by data exploration. The VLDB Journal, 2022, 31(4): 753−777.

[30] Chai C, Liu J, Tang N, *et al*. GoodCore: Data-effective and data-efficient machine learning through coreset selection over incomplete data. Proc. of the ACM on Management of Data, 2023, 1(2): 1−27.

[31] Kim H, So BH, Han WS, *et al*. Natural language to SQL: Where are we today? Proc. of the VLDB Endowment, 2020, 13(10): 1737−1750.

[32] Li HY, Zhang J, Li CP, *et al*. RESDSQL: Decoupling schema linking and skeleton parsing for text-to-SQL. Proc. of the AAAI Conf. on Artificial Intelligence, 2023, 37(11): 13067−13075.

[33] Qi J, Tang J, He Z, *et al*. RASAT: Integrating relational structures into pretrained Seq2Seq model for text-to-SQL. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. 2022. 3215−3229.

[34] Li J, Hui B, Cheng R, *et al*. Graphix-T5: Mixing pre-trained transformers with graph-aware layers for text-to-SQL parsing. In: Proc. of the AAAI. 2023. 13076−13084.

[35] Dong X, Zhang C, Ge Y, *et al*. C3: Zero-shot text-to-SQL with ChatGPT. arXiv:2307.07306, 2023.

[36] Pourreza M, Rafiei D. Din-SQL: Decomposed in-context learning of text-to-SQL with self-correction. arXiv:2304.11015, 2023.

[37] Shen L, Shen E, Luo Y, *et al*. Towards natural language interfaces for data visualization: A survey. IEEE Trans. on Visualization and Computer Graphics, 2023, 29(6): 3121−3144.

[38] Luo YY, Qin XD, Xie YP, Li GL. Intelligent data visualization analysis techniques: A survey. Ruan Jian Xue Bao/Journal of Software, 2024, 35(1): 356−404 (in Chinese with English abstract). https://www.jos.org.cn/1000-9825/6911.htm [doi: 10.13328/j.cnki.jos.006911]

[39] Narechania A, Srinivasan A, Stasko J. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. IEEE Trans. on Visualization and Computer Graphics, 2020, 27(2): 369−379.

[40] Wang X, Cheng F, Wang Y, *et al*. Interactive data analysis with next-step natural language query recommendation. arXiv:2201.04868, 2022.

[41] Bird S. NLTK: The natural language toolkit. In: Proc. of the COLING/ACL 2006 Interactive Presentation Sessions. 2006. 69−72.

[42] Manning CD, Surdeanu M, Bauer J, *et al*. The Stanford CoreNLP natural language processing toolkit. In: Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014. 55−60.

[43] Finkel JR, Grenager T, Manning CD. Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005). 2005. 363−370.

[44] Liu C, Han Y, Jiang R, *et al*. Advisor: Automatic visualization answer for natural-language question on tabular data. In: Proc. of the 2021 IEEE 14th Pacific Visualization Symp. (PacificVis). IEEE, 2021. 11−20.

[45] Luo Y, Tang N, Li G, *et al*. Natural language to visualization by neural machine translation. IEEE Trans. on Visualization and Computer Graphics, 2021, 28(1): 217−226.

[46] Maddigan P, Susnjak T. Chat2vis: Generating data visualisations via natural language using ChatGPT, codex and GPT-3 large language models. arXiv:2302.02094v2, 2023.

[47] Kenton JDMWC, Toutanova LK. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the NAACL-HLT. 2019. 4171−4186.

[48] Chen M, Tworek J, Jun H, *et al*. Evaluating large language models trained on code. arXiv:2107.03374, 2021.

[49] Brown T, Mann B, Ryder N, *et al*. Language models are few-shot learners. In: Advances in Neural Information Processing Systems, Vol. 33. 2020. 1877−1901.

[50] Zhou Y, Muresanu AI, Han Z, *et al*. Large language models are human-level prompt engineers. arXiv:2211.01910, 2022.

[51] Moritz D, Wang C, Nelson GL, *et al*. Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. IEEE Trans. on Visualization and Computer Graphics, 2019, 25(1): 438−448.

[52] Luo Y, Qin X, Tang N, *et al*. DeepEye: Towards automatic data visualization. In: Proc. of the 2018 IEEE 34th Int'l Conf. on Data Engineering (ICDE). IEEE, 2018. 101−112.

[53] Luo Y, Qin X, Tang N, *et al*. DeepEye: Creating good data visualizations by keyword search. In: Proc. of the 2018 Int'l Conf. on Management of Data. 2018. 1733−1736.

[54] Burges C, Shaked T, Renshaw E, *et al*. Learning to rank using gradient descent. In: Proc. of the 22nd Int'l Conf. on Machine Learning. 2005. 89−96.

[55] Luo Y, Zhou Y, Tang N, *et al*. Learned data-aware image representations of line charts for similarity search. Proc. of the ACM on Management of Data, 2023, 1(1): 1−29.

[56] Liu Z, Lin Y, Cao Y, *et al*. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. of the IEEE/CVF Int'l Conf. on Computer Vision. 2021. 10012−10022.

[57] Lee DJL, Tang DX, Agarwal K, *et al*. Lux: Always-on visualization recommendations for exploratory dataframe workflows. Proc. of the VLDB Endowment, 2021, 15(3): 727−738.

[58] Tang J, Luo Y, Ouzzani M, *et al*. Sevi: Speech-to-visualization through neural machine translation. In: Proc. of the 2022 Int'l Conf. on Management of Data. 2022. 2353−2356.

附中文参考文献:

[38] 骆昱宇, 秦雪迪, 谢宇鹏, 李国良. 智能数据可视分析技术综述. 软件学报, 2024, 35(1): 356−404. https://www.jos.org.cn/1000-9825/6911.htm [doi: 10.13328/j.cnki.jos.006911]

谢宇鹏(1996－), 男, 硕士生, CCF 学生会员, 主要研究领域为数据可视化.

骆昱宇(1996－), 男, 博士, 副研究员, CCF 专业会员, 主要研究领域为智能数据管理与可视分析.

冯建华(1967－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据库, 数据安全与隐私保护, 信息检索.