“What Do You Mean by That?” A Parser-Independent Interactive Approach for Enhancing Text-to-SQL

ByteDance

01 Introduction

02 Methodology

03 summary

04 takeaway

01

Introduction

针对落地过程中遇到的困难，提出了一种与具体parser相独立的可交互方法，这种方法可针对任意解析器产生多个可选问题

Introduction

挑战

- 虽然text-2-sql这个领域最近几年取得了很大的进展，但这些学术界验证比较好的方法，在实际落地过程中遇到了很多困难，一个核心挑战是parser还是很难完整理解用户的问句
- 提问的用户是最理解自己问句的人，所以与用户交互是最有希望解决上面挑战的方法。

相关研究

- 2017年，提出的方法是让用户检查sql，但显然行不通，只有sql专家类型的用户能够用起来。
- 2018年，首次提出一种方法：通过返回多个相似问句与非sql专家交互，但局限性是设计的策略只能适应简单的场景，不能覆盖复杂的场景
- 2019年，取得了一个重要的进展，提出了一种衡量nl解析器parser不确定性的方法，可以在不同不确定性时表现不同的行为，但是目前为止，各家的实现系统解决这个问题用的还都是强规则而没用完全nn的方法，而往往nl2sql是为第三方服务，假设解析器是个黑盒的话，则无法设计出可交互的方法。

Introduction

提出

- a Parser-IndependentInteractiveApproach (PIIA) to interact with hu-man users and help parsers better understand NLquestions.

- 一个与parser独立无关的方法
- 这个方法可以与人类进行交互
- 帮助解析器parser更好的理解用户的自然语言问题

02

Methodolog

详细介绍论文中提出的三个模块：Error Locator、Question Generator、NL Modifier

ByteDance

Methodolog overview

Error Locator

employs an alignmentmethod to help parsers locate uncertain tokens in the NL questions

能够指出哪里解析器识别不清楚

Question Generator

de-signs multi-choice questions in natural language for users, which offers a pleasant interactive experience.

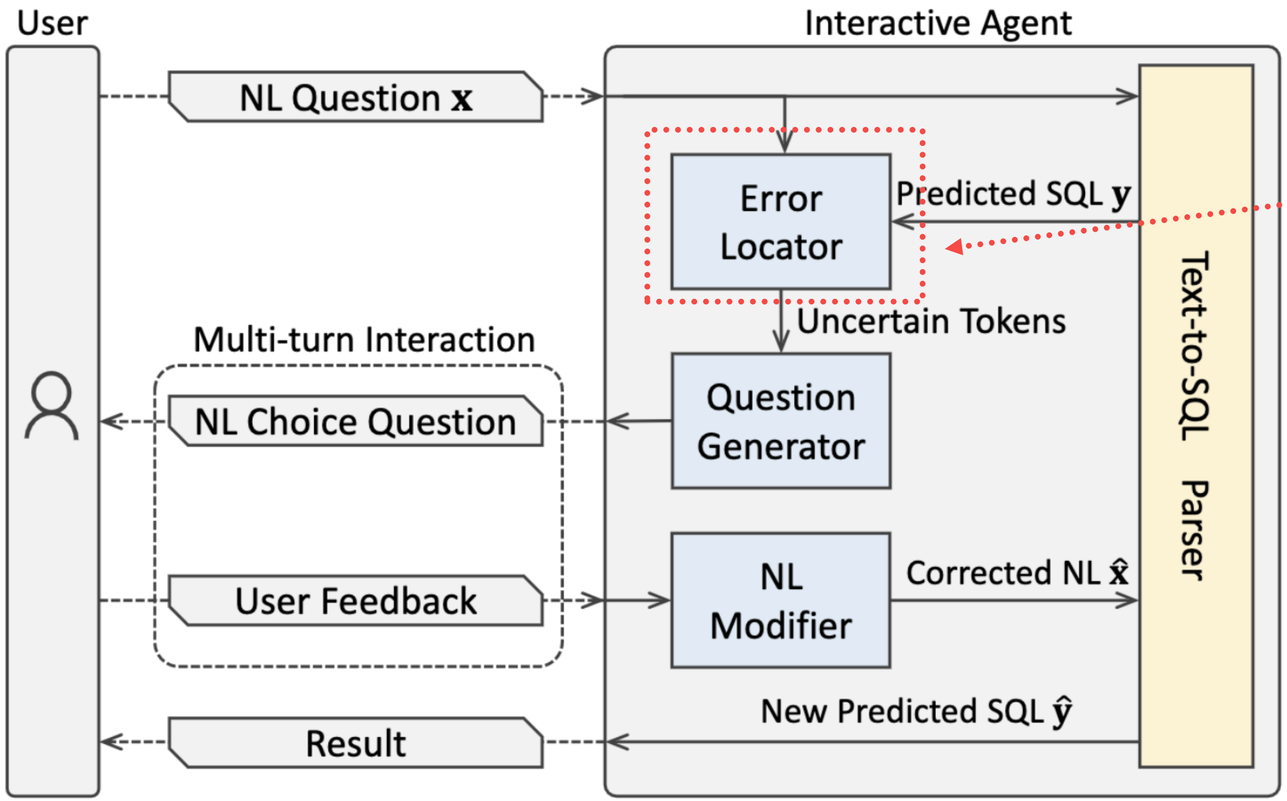
给用户选择的问题尽可能的少，且贴近用户意图

NL Modifier

rewrites the NL questions according to the users' feedback and produces more legible questions to facilitate downstream parsing

把用户的问题转换成对解析器来说更清晰易懂

流程图



与我们askdata相比，多了Error Locator，可以将解析器识别不准的tokens传递出，Question Generator可以针对性生成

用户因为不熟悉系统规则关键字，可能会问出莫名的表达，系统很难理解

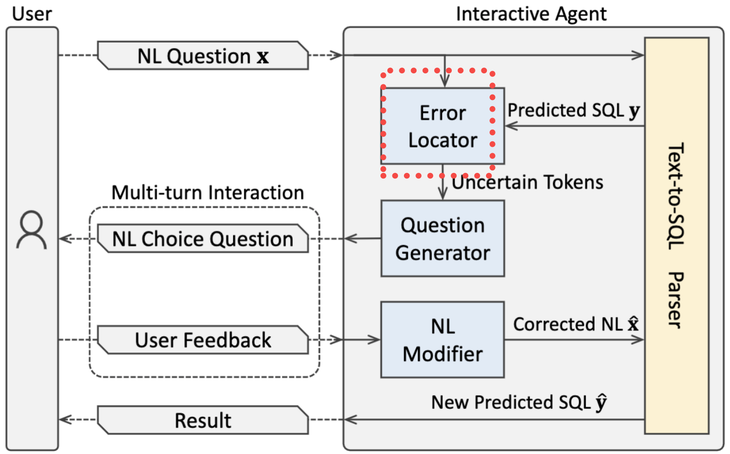
作者分析了开源数据集及内部系统大量的badcases，发现47.3%的badcase产生的原因是column_name、table_name与values相近

Thus, we build PIIA upon parsers that can interactively revise inexplicit expressions in x with the help of users' feedback, thus enhancing the performance of text-to-SQL.

修改用户不清楚的表达

ByteDance

Error locator



Error locator认为列名和值重合的token片段为易混淆不确定的token片段

因为PIIA要做与parser相独立，即认为parser是一个黑盒，所以设计了一种方法，可以自动对比x和y，x中所有的片段都会与y进行对齐（asign），未对齐的token作为uncertain tokens.

具体的方法是：先把y转换成x2，然后对比x2与x进行对齐

ByteDance

Error locator

SQL-TO-TEXT Restatement

作者设计了一个中间语言，先把sql转成中间语言，
再通过一个模板把中间语言转成NL

他这个中间语言类似于我们askdata的QueryModel
，
这个模板类似于我们nlg模块的模板

$$
\begin{aligned}
Z &::= intersect\ R\ R \mid union\ R\ R \mid except\ R\ R \mid R \\
R &::= Select\ Filter\ Order \mid Select\ Filter \\
&\quad \mid Select\ Order \mid Select \\
Select &::= A \mid A\ A \mid A\ A\ A \mid A\ A\ A\ A \mid A\ A\ A\ A\ A \\
Filter &::= Filter\ and\ Filter \mid Filter\ or\ Filter \mid = A\ V \mid > A\ V \\
&\quad \mid < A\ V \mid \geq A\ V \mid \leq A\ V \mid \neq A\ V \mid between\ A\ V\ V \\
Order &::= asc\ A \mid asc\ A\ limit\ number \\
&\quad \mid dec\ A \mid dec\ A\ limit\ number \\
A &::= none\ C\ T \mid \max C\ T \mid \min C\ T \\
&\quad \mid count\ C\ T \mid sum\ C\ T \mid avg\ C\ T \\
C &::= column \qquad T ::= table \qquad V ::= value \mid R
\end{aligned}
$$

Figure 2: The grammar of the intermediate language. In a specific database, *column* refers to distinct column names while *table* comprises several table names and *value* indicates the value tokens expressed by the user.

Error locator

作者设计一个基于bert的模型做x和x2的对齐

对于askdata来说，不用这么复杂，askdata有解析中间产物的evidence，只要在x2中remove掉evidence即可达到类似效果

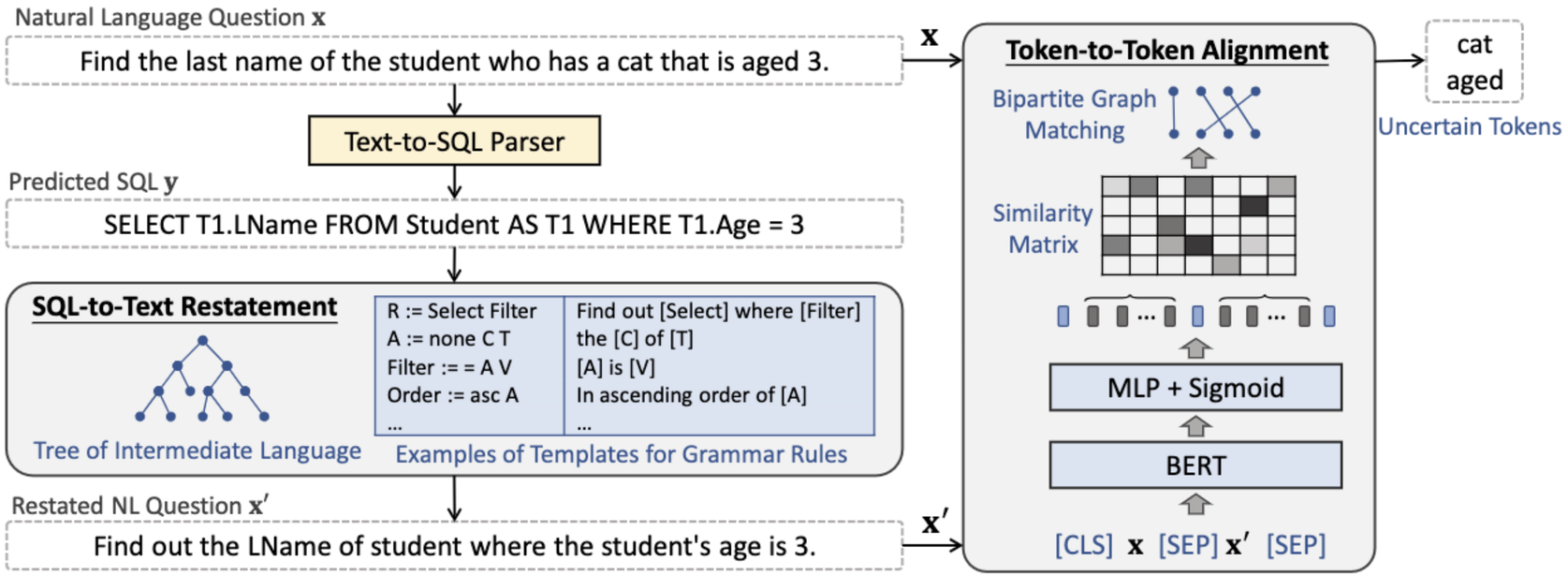


Figure 3: Illustration of Error Locator with a real case from IRNet on the Spider dataset. The NL question $\mathbf{x}$ is parsed to SQL $\mathbf{y}$, and then converted to restated NL question $\mathbf{x}'$ via SQL-to-text restatement. Then, a token-to-token alignment similarity matrix between $\mathbf{x}$ and $\mathbf{x}'$ is computed to detect uncertain tokens (i.e., cat and aged).

Question Generator

如果把所有的易混淆片段转成问题，问题会非常多。作者统计对于复杂的数据库表，问题数可能超过40个

比较选项option与混淆片段的相似性（编辑距离或语义相似性）

比askdata的消歧多提供了可忽略、value两个选项

本质上是对短语进行消歧，即这个短语究竟是一个列名还是一个值，而askdata只能对值进行消歧，这个值属于哪一列

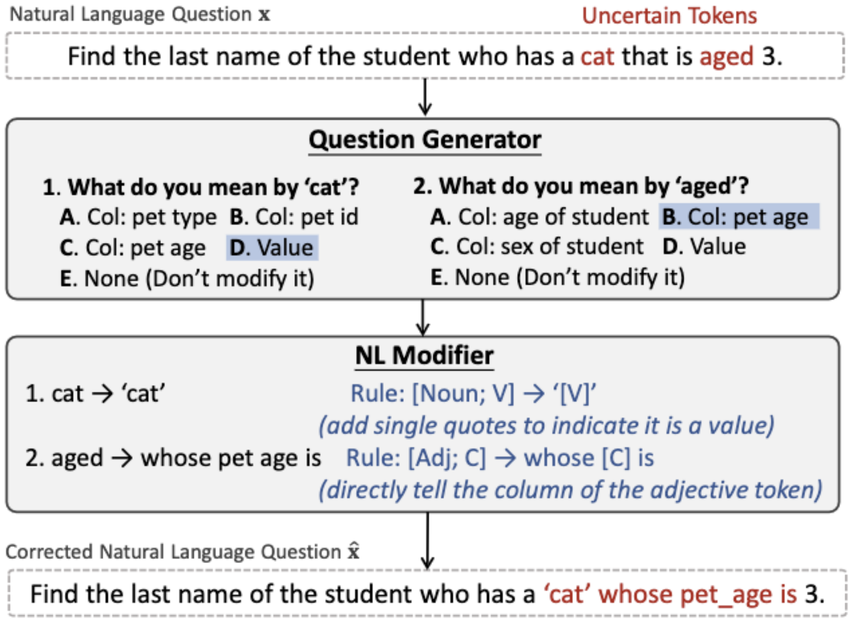


Figure 4: Question Generator and NL Modifier: an example. Shaded options in the multi-choice questions are selected by the user.

NL Modifier

提供了一套modifier规则来替换NL，不直接替换的原因是因为替换的短语可能是形容词、动词等，直接替换会看起来很怪。

通过以上提供的这套规则替换之后，可以很明确的知道哪些是值，哪些是列。

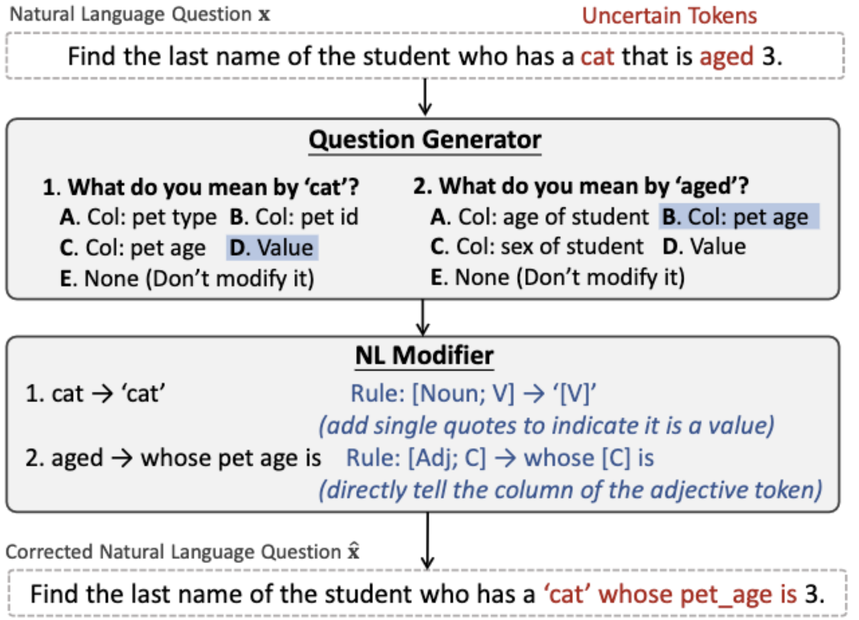


Figure 4: Question Generator and NL Modifier: an example. Shaded options in the multi-choice questions are selected by the user.

03

summary

论文中的方法对我们有哪些借鉴意义的思考

ByteDance

Error Locator

employs an alignmentmethod to help parsers locate uncertain tokens in the NL questions

能够指出哪里解析器识别不清楚

Question Generator

de-signs multi-choice questions in natural language for users, which offers a pleasant interactive experience.

给用户选择的问题尽可能的少，且贴近用户意图

NL Modifier

rewrites the NL questions according to the users' feedback and produces more legible questions to facilitate downstream parsing

把用户的问题转换成对解析器来说更清晰易懂

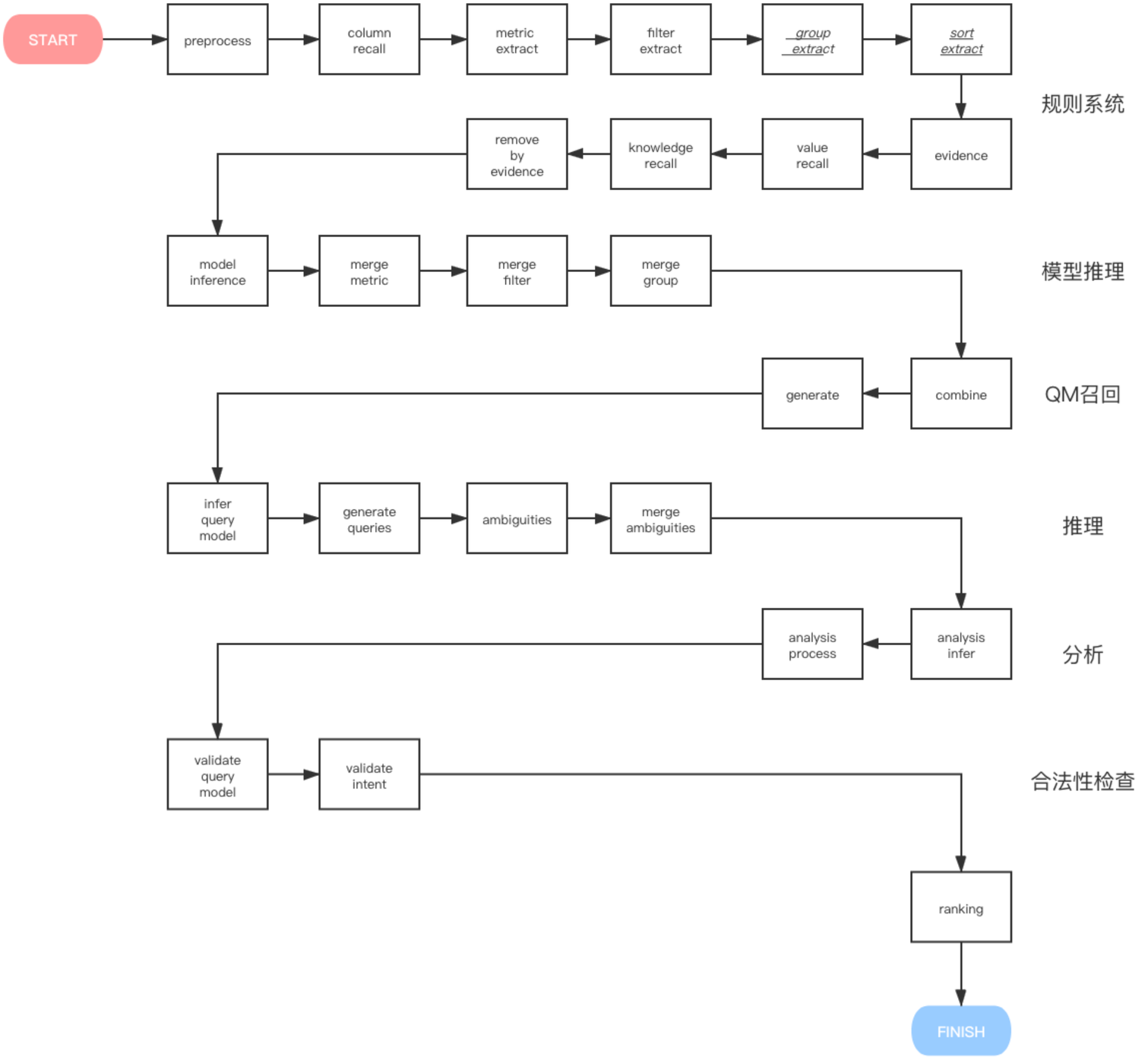
- The interaction process in PIIA is user-friendly that asks multi-choice questions and reduces the number of questions as much as possible.

给用户选择的问题
尽可能的少，
且贴近用户意图

提出有一种方法：可以将text-to-sql的不确定性，转换成具体的问题，与用户进行交互来消除不确定性

而且这种方法是针对用户NL token短语，即帮助明确词短语是一个哪一列，还是哪个值

ask的流程



https://feishu.processon.com/feishu/diagraming/5fba27e2e4b0b3f260412b72

04

takeaway

ByteDance

代码

论文

谢谢

演讲人