

We propose clause-level edit methods with Python-style representations to use language models of code for SQL correction

Text-to-SQL Error Correction with Language Models of Code

Ziru Chen, Shijie Chen, Michael White, Raymond Mooney, Ali Payani, Jayanth Srinivasa, Yu Su, Huan Sun



Our code and data:

<https://github.com/OSU-NLP-Group/Auto-SQL-Correction>



Text-to-SQL Error Correction

- Recent text-to-SQL parsers can reach decent accuracy, but they are still **not accurate enough in practice**
- We study how to **correct errors in parser-generated SQL queries** with language models of code

Show the name, country, age for all singers ordered from the oldest to youngest.

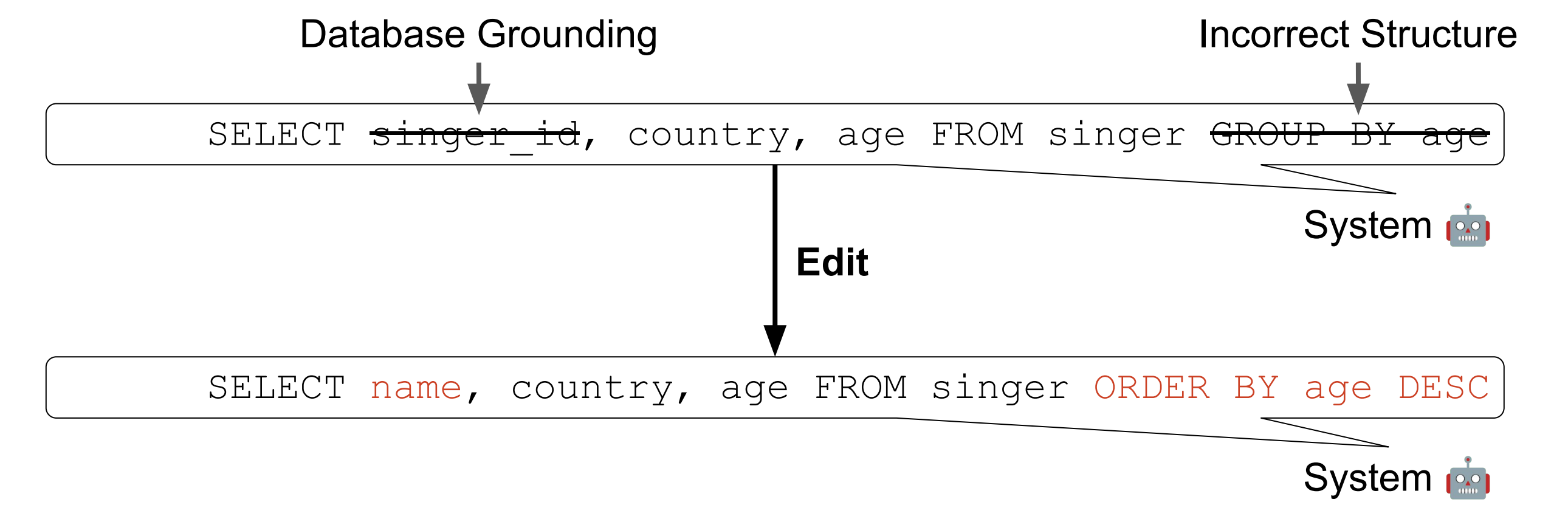
User

That's not right. Can you suggest some corrections?

User

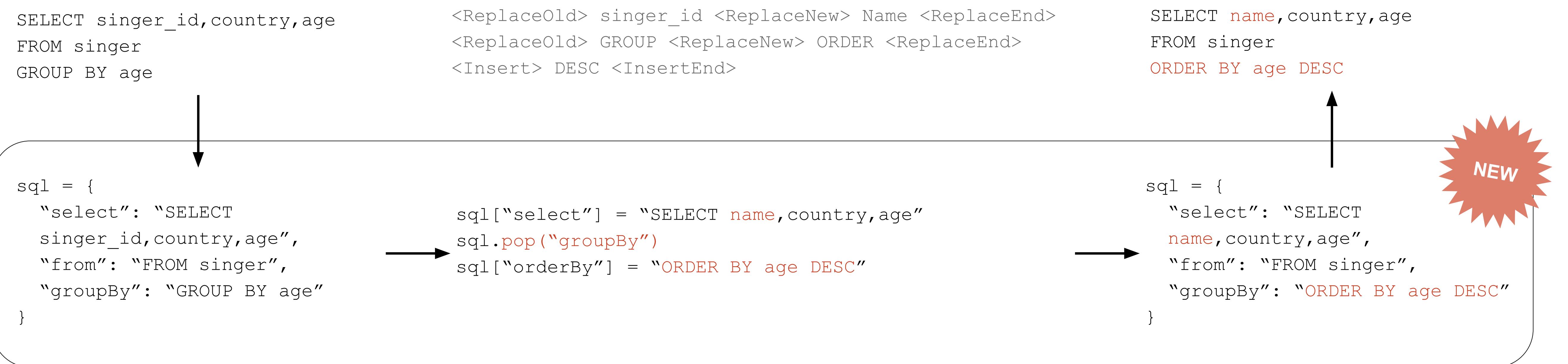
Great! That's what I want.

User



Our Method: Clause-Level Editing and Python Representation

- Token-level edits represented with special tokens can be ambiguous
 - Use **clause-level edits** to mitigate ambiguity issues
- Most language models of code (e.g. CodeT5) are not pre-trained on SQL
 - Propose **Python-style representations** to better use language models of code



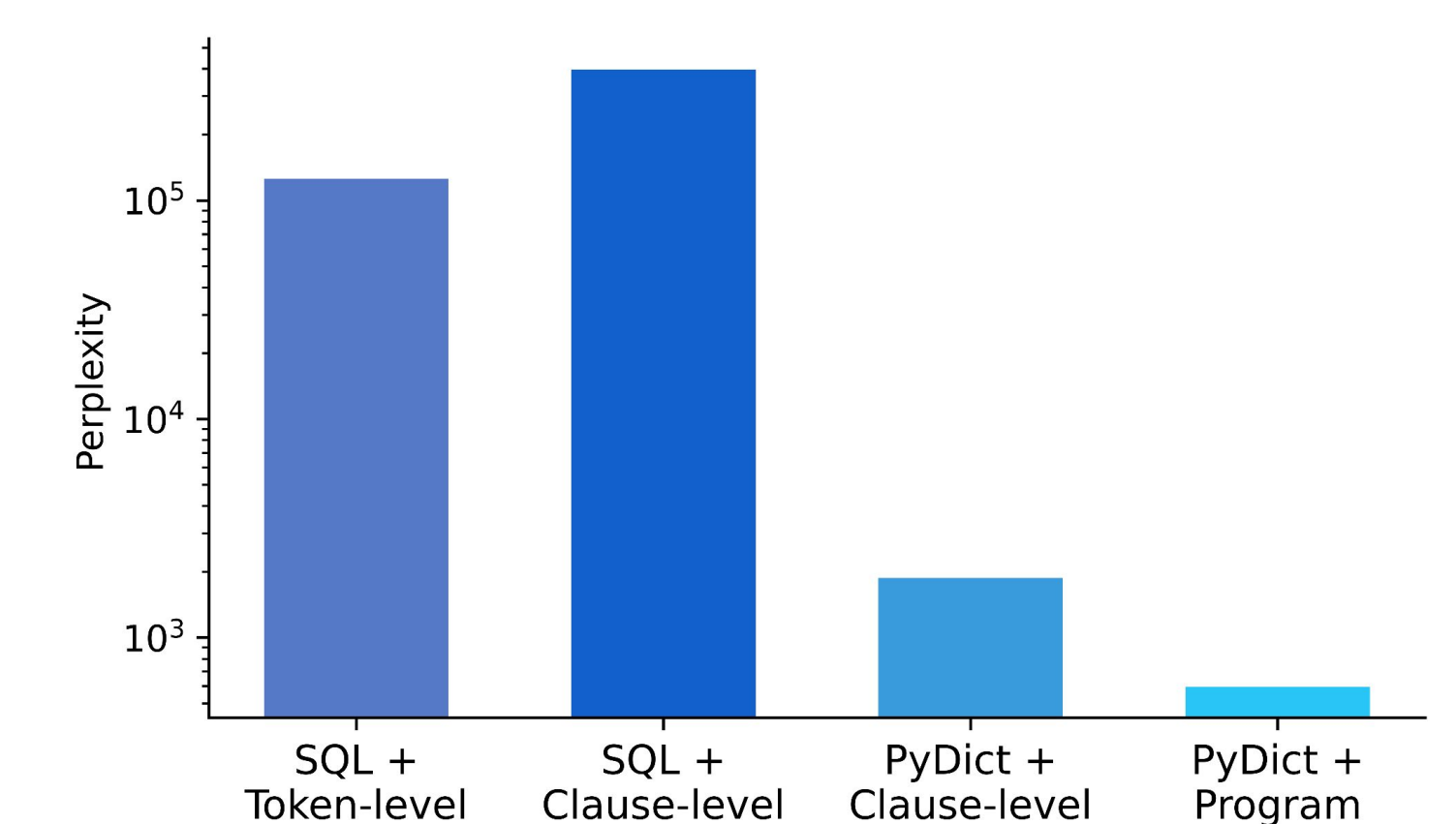
Experiments and Results

Experimental Setup

- Data synthesis: 5-fold cross-validation on Spider for each parser
- Fine-tune CodeT5 (and CoditT5) on our synthesized data with different representations

Results

- CodeT5 shows considerably **lower zero-shot perplexity** on our proposed representation
- CodeT5 consistently achieves **statistically significant improvement** using our proposed representation
- Simulating user interactions with CodeT5 can **further improve text-to-SQL parser's accuracy**



Show the name, country, age for all singers ordered from the oldest to youngest. | singer : singer_id , name , country , age | concert : ...

```
sql = {
  "select": "SELECT
singer_id, country, age",
  "from": "FROM singer",
  "groupBy": "GROUP BY
age"
}
```

x → Language Model of Code → y

```
sql["select"] = "SELECT
name, country, age"
sql.pop("groupBy")
sql["orderBy"] = "ORDER BY
age DESC"

sql = {
  "select": "SELECT
name, country, age",
  "from": "FROM singer",
  "groupBy": "ORDER BY age DESC"
}
```

