Speak to your Parser: Interactive Text-to-SQL with Natural Language Feedback

Authors: Ahmed Elgohary, Saghar Hosseini, Ahmed Hassan Awadallah
Presenter: Lixian Liu
Background

- **Text-to-SQL semantic parsing** system: Focusing on parsing natural language utterances into an executable SQL queries
Background - Interactive Semantic Parsing

Find all the locations whose names contain the word "film"

Semantic Parsing

SELECT Address FROM LOCATIONS WHERE Location_Name LIKE '%film%'

Address is wrong. I want the name of the locations

Correction

SELECT Location_Name FROM LOCATIONS WHERE Location_Name LIKE '%film%'
Background - Interactive Semantic Parsing

To enable this form of interaction, the system must:

(1) explain the produced SQL,
(2) allow for human response, and
(3) utilize the feedback and original question to come up with a more correct interpretation.
Contributions

1) define the **task** of SQL parse correction with natural language feedback
2) create a framework for **explaining** SQL parse in natural language
3) construct **SPLASH** (Semantic Parsing with Language Assistance from Humans): a new dataset
4) establish several **baseline** models
Task

- SQL parse correction with natural language feedback

Question:
Find all the locations whose names contain the word "film"

Predicted Parse:
```
SELECT Address FROM LOCATIONS WHERE Location_Name LIKE '%film%'
```

Feedback:
Address is wrong. I want the name of the locations.

Gold Parse:
```
SELECT Location_Name FROM LOCATIONS WHERE Location_Name LIKE '%film%'
```

Schema:
| Location_ID | Location_Name | Address | Other_Details |
SPLASH Dataset Construction

Pipeline:

1) (Utterance, Incorrect SQL)
2) Explaining SQL
3) Crowdsourcing feedback
SPLASH Dataset Construction

1) Utterance and Incorrect SQL

**Spider** Dataset (Questions, Gold Parse)
1. Larger in scale
2. Requires inducing parses of complex query structures

**Seq2Struct**: Parser (Incorrect SQL)
1. Neural parser with grammar-based decoder
2. Train on 80% of Spider's training set and apply it to the remaining 20% → collect the incorrect parse
3. For each question, generate 3 incorrect SQL → 3183 pairs of questions and incorrect SQL
SPLASH Dataset Construction

1) Utterance and Incorrect SQL

**Seq2Struct**: Parser (Incorrect SQL)

Use 2nd top prediction (difference in probability between the top and 2nd top is below 0.2) to add additional 1192 pairs to the dataset
SPLASH Dataset Construction

2) Explaining SQL

- Explain the incorrect generated SQL in a way that humans who are not proficient in SQL can understand
SPLASH Dataset Construction

2) Explaining SQL

- Template-based approach
- 57 templates cover 85% of Spider queries

SQL:
```
SELECT id, name from browser GROUP BY id ORDER BY COUNT(*) DESC
```

Template:
```
SELECT _cols_ from _table_ Group BY_col_ ORDER BY _aggr_ _col_
```

Explanation:

Step 1: Find the number of rows of each value of id in browser table.

Step 2: Find id, name of browser table with largest value in the results of step 1.
SPLASH Dataset Construction

3) Crowdsourcing Feedback

- Internal crowdsourcing platform
- 10 annotators participated
- Limit the maximum feedback length to 15 tokens
SPLASH Dataset Construction

3) Dataset Summary

- 9,314 questions-feedback paris
- 962 from Spider development set as the test set
- Hold 10% of the remaining set as the dev set

<table>
<thead>
<tr>
<th>Number of</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>7,481</td>
<td>871</td>
<td>962</td>
</tr>
<tr>
<td>Databases</td>
<td>111</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Uniq. Questions</td>
<td>2,775</td>
<td>290</td>
<td>506</td>
</tr>
<tr>
<td>Uniq. Wrong Parses</td>
<td>2,840</td>
<td>383</td>
<td>325</td>
</tr>
<tr>
<td>Uniq. Gold Parses</td>
<td>1,781</td>
<td>305</td>
<td>194</td>
</tr>
<tr>
<td>Uniq. Feedbacks</td>
<td>7,350</td>
<td>860</td>
<td>948</td>
</tr>
<tr>
<td>Feedback tokens (Avg.)</td>
<td>13.9</td>
<td>13.8</td>
<td>13.1</td>
</tr>
</tbody>
</table>
SPLASH Dataset Analysis

- Study the characteristics of
  1) The mistakes made by the parser
  2) The natural language feedback from annotators
SPLASH Dataset Analysis

- Error Characteristics

Figure 4: A histogram of the distance between the gold and the predicted SQL.

78%+ within a distance of 3 or less

Questions that require a join is harder and more error prone

Figure 5: A histogram of different SQL keywords appearing in edits (between the gold and predicted SQL) and their distribution across edit types (replace, insert or delete).
SPLASH Dataset Analysis

- **Feedback Characteristics (sample 200 examples)**

<table>
<thead>
<tr>
<th>Type of Feedback</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Complete Feedback:</strong></td>
<td>[81.5%]</td>
<td>Show the types of schools that have two schools.</td>
</tr>
<tr>
<td>Question:</td>
<td>Show the types of schools that have two schools.</td>
<td></td>
</tr>
<tr>
<td>Pred. SQL:</td>
<td>SELECT TYPE FROM school GROUP BY TYPE HAVING count(*) &gt;= 2</td>
<td></td>
</tr>
<tr>
<td>Feedback:</td>
<td>You should not use greater than.</td>
<td></td>
</tr>
<tr>
<td><strong>Partial Feedback:</strong></td>
<td>[13.5%]</td>
<td>What are the names of all races held between 2009 and 2011?</td>
</tr>
<tr>
<td>Question:</td>
<td>What are the names of all races held between 2009 and 2011?</td>
<td></td>
</tr>
<tr>
<td>Pred. SQL:</td>
<td>SELECT country FROM circuits WHERE lat BETWEEN 2009 AND 2011</td>
<td></td>
</tr>
<tr>
<td>Feedback:</td>
<td>You should use races table.</td>
<td></td>
</tr>
<tr>
<td><strong>Paraphrase Feedback:</strong></td>
<td>[5.0%]</td>
<td>What zip codes have a station with a max temperature greater than or equal to 80 and when did it reach that temperature?</td>
</tr>
<tr>
<td>Question:</td>
<td>What zip codes have a station with a max temperature greater than or equal to 80 and when did it reach that temperature?</td>
<td></td>
</tr>
<tr>
<td>Pred. SQL:</td>
<td>SELECT zip_code FROM weather WHERE min.temperature_f &gt; 80 OR min.sea_level_pressure_inches &gt; 80</td>
<td></td>
</tr>
<tr>
<td>Feedback:</td>
<td>Find date, zip code whose max temperature f greater than or equals 80.</td>
<td></td>
</tr>
</tbody>
</table>
## SPLASH Dataset Analysis

- **Feedback Characteristics (sample 200 examples)**

<table>
<thead>
<tr>
<th>Feedback Type</th>
<th>%</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Missing</td>
<td>13%</td>
<td>I also need the number of different services</td>
</tr>
<tr>
<td>- Wrong</td>
<td>36%</td>
<td>Return capacity in place of height</td>
</tr>
<tr>
<td>- Unnecessary</td>
<td>4%</td>
<td>No need to return email address</td>
</tr>
<tr>
<td>Conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Missing</td>
<td>10%</td>
<td>ensure they are FDA approved</td>
</tr>
<tr>
<td>- Wrong</td>
<td>19%</td>
<td>need to filter on open year not register year</td>
</tr>
<tr>
<td>- Unnecessary</td>
<td>7%</td>
<td>return results for all majors</td>
</tr>
<tr>
<td>Aggregation</td>
<td>6%</td>
<td>I wanted the smallest ones not the largest</td>
</tr>
<tr>
<td>Order/Uniq</td>
<td>5%</td>
<td>only return unique values</td>
</tr>
</tbody>
</table>
Baselines

- Handcrafted re-ranking with feedback

Initial parse: select first_name, last_name from students

Candidate parse: select first_name from teachers

Diff: {last_name, students, teachers}

Feedback: use teachers instead of students

Assign score 2 / 3 to this candidate parse
Baselines

- Seq2Struct + Feedback

Appending the feedback to the question for each training example in SPLASH

Note: Seq2Struct + Feedback does not use the mispredicted parses
Baselines

- EditSQL + Feedback

SOTA model for conversational text-to-SQL

Initial parse: `SELECT Address FROM LOCATIONS WHERE Location_Name LIKE '%film%'`

Correct parse: `SELECT Location_Name FROM LOCATIONS WHERE Location_Name LIKE '%film%'`
## Baselines - Results

**Correction Accuracy:**
the percentage of the testing examples that are correct

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Exact Match Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Without Feedback</strong></td>
<td></td>
</tr>
<tr>
<td>⇒ Seq2Struct</td>
<td>N/A</td>
</tr>
<tr>
<td>⇒ Re-ranking: Uniform</td>
<td>2.39</td>
</tr>
<tr>
<td>⇒ Re-ranking: Parser score</td>
<td>11.26</td>
</tr>
<tr>
<td>⇒ Re-ranking: Second Best</td>
<td>11.85</td>
</tr>
<tr>
<td><strong>With Feedback</strong></td>
<td></td>
</tr>
<tr>
<td>⇒ Re-ranking: Handcrafted</td>
<td>16.63</td>
</tr>
<tr>
<td>⇒ Seq2Struct+Feedback</td>
<td>13.72</td>
</tr>
<tr>
<td>⇒ EditSQL+Feedback</td>
<td><strong>25.16</strong></td>
</tr>
<tr>
<td>Re-ranking Upper Bound</td>
<td>36.38</td>
</tr>
<tr>
<td>Estimated Human Accuracy</td>
<td>81.50</td>
</tr>
</tbody>
</table>
Conclusions

1. Introduce the task of SQL parse correction using natural language feedback
2. Compare baseline models and show that natural language feedback is effective for correcting parses
3. But still SOTA models struggle to solve the task
Future Work

1. Explore improving the correction models
2. Leveraging logs of natural language feedback to improve text-to-SQL parsers
3. Expanding the dataset to include multiple turns of correction
Thank you ! Questions?