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 FROMLOCATIONS
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 ply it to the remainin se

X 3 → 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: ↓

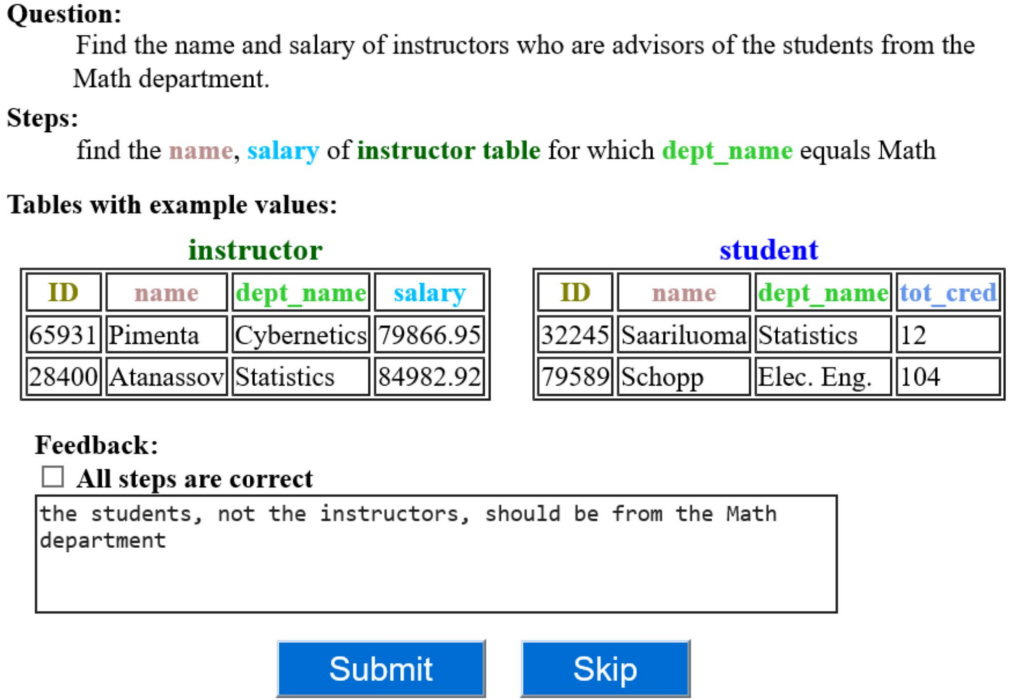
Step 1: Find the number of rows of each value of id in browser table.

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

Number of	Train	Dev	Test
Examples	7,481	871	962
Databases	111	9	20
Uniq. Questions	2,775	290	506
Uniq. Wrong Parses	2,840	383	325
Uniq. Gold Parses	1,781	305	194
Uniq. Feedbacks	7,350	860	948
Feedback tokens (Avg.)	13.9	13.8	13.1

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

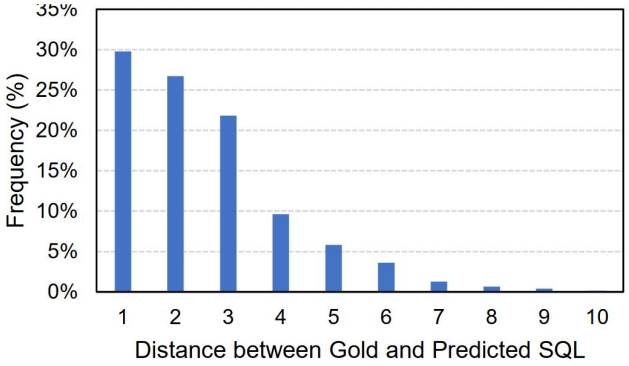


Figure 4: A histogram of the distance between the gold and the predicted SQL.

78%+ within a distance of 3 or less

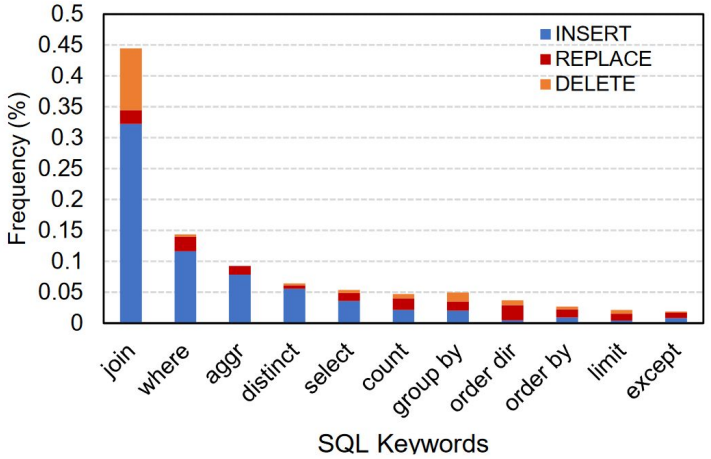


Figure 5: A histogram of different SQL keywords appearing in edits (between the gold and predicted SQL) and their distribution across edit types (replace, insert or delete).

Questions that require a join is harder and more error prone

SPLASH Dataset Analysis

- Feedback Characteristics (sample 200 examples)

Complete Feedback: [81.5%]	
Question:	Show the types of schools that have two schools.
Pred. SQL:	`SELECT TYPE FROM school GROUP BY TYPE HAVING count(*) >= 2`
Feedback:	You should not use greater than.
Partial Feedback: [13.5%]	
Question:	What are the names of all races held between 2009 and 2011?
Pred. SQL:	`SELECT country FROM circuits WHERE lat BETWEEN 2009 AND 2011`
Feedback:	You should use races table.
Paraphrase Feedback: [5.0%]	
Question:	What zip codes have a station with a max temperature greater than or equal to 80 and when did it reach that temperature?
Pred. SQL:	`SELECT zip_code FROM weather WHERE min_temperature_f > 80 OR min_sea_level_pressure_inches > 80`
Feedback:	Find date , zip code whose max temperature f greater than or equals 80.

SPLASH Dataset Analysis

- Feedback Characteristics (sample 200 examples)

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

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

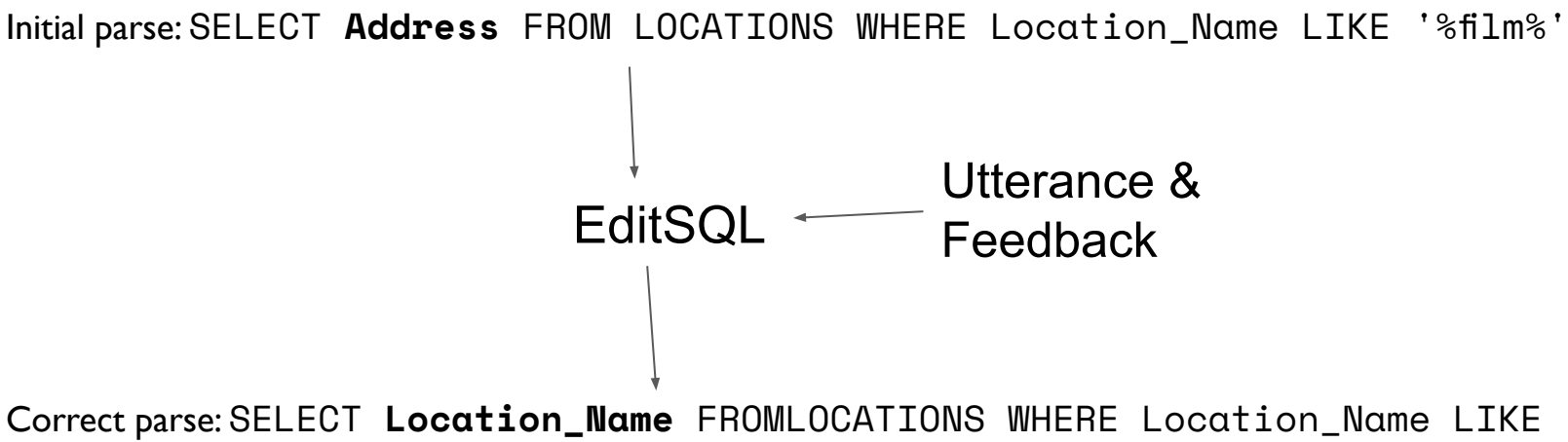
Appending the feedback to the question for each training example in SPLASH

Note: Seq2Struct + Feedback does not use the mispredicted parses

Baselines

- EditSQL + Feedback

SOTA model for conversational text-to-SQL



Baselines - Results

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

Baseline	Exact Match Accuracy (%) Correction	End-to-End
Without Feedback		
⇒ Seq2Struct	N/A	41.30
⇒ Re-ranking: Uniform	2.39	42.48
⇒ Re-ranking: Parser score	11.26	46.86
⇒ Re-ranking: Second Best	11.85	47.15
With Feedback		
⇒ Re-ranking: Handcrafted	16.63	49.51
⇒ Seq2Struct+Feedback	13.72	48.08
⇒ EditSQL+Feedback	**25.16**	**53.73**
Re-ranking Upper Bound	36.38	59.27
Estimated Human Accuracy	81.50	81.57

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?