Seq2SQL Using Seq2seq with Attention Model

LILY Spring 2018 Workshop

Dongxu Wang

Methods

- Seq2seq methods
  - Seq2seq with Attention over Input (Bahdanau, 2015)
  - Augmented pointer networks with SQL structure (Victor Zhong, 2017)
  - Seq2seq with Attention over Input, schema, copy (Catherine Finegan-Dollak et al., 2018)
- Seq2tree methods
  - Attention-based seq2Tree (AST) (Dong and Lapata, 2016)
  - Attention-based seq2Tree (AST) with copy on PL (Yin et al. 2017)
- Oracle Entity Assumption?
  - Dong and Latapa, 2016, Catherine Finegan-Dollak et al., 2018

Seq2SQL Using Seq2Seq with Attention

From…

Return the id and type code of the template that is used for the greatest number of documents .

To…

```
SELECT T1 . template_id , T2 . Template_Type_Code
FROM Documents AS T1 JOIN Templates AS T2
ON T1 . template_id = T2 . template_id
GROUP BY T1 . template_id ORDER BY count ( * ) DESC LIMIT 1
```

How to predict tables / fields right?

- Peek at the schema data!
  - Database name, field name, table name, is primary key, is foreign key…
- Or copy… Pointer network

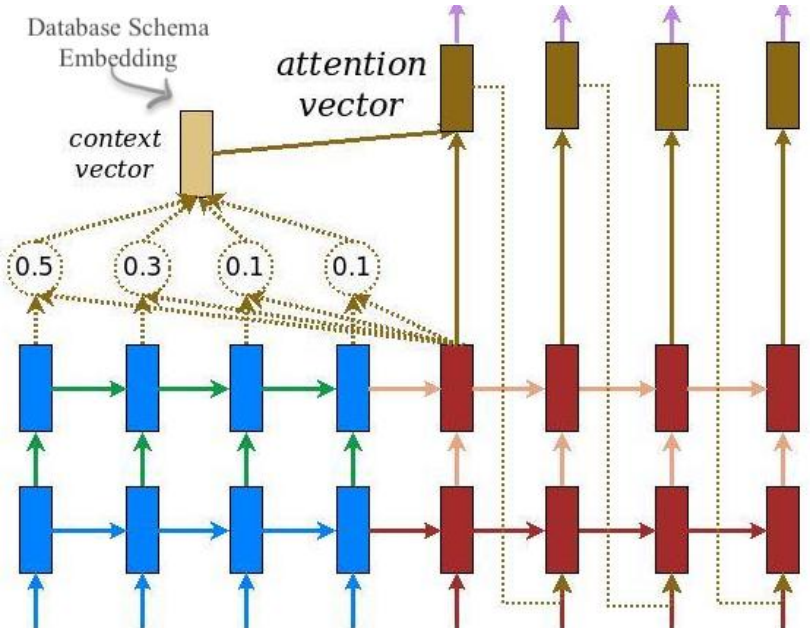

Figure 1. Model Structure of Sequence-to-Sequence Model with Attention of Database Schema

Figure adapted from https://www.tensorflow.org/tutorials/seq2seq

Data

- nl2SQL pairs dataset created by LILY project members.
- 4204 pairs in train and 632 pairs in validation, 49 databases in train, and 7 database in validation.

Results

- Mostly can be parsed into AST (Abstract Syntax Tree)
- Not very good with complex structure
- Not very good at predicting new database schema data

Components	Accuracy	Matched	Gold	Predicted
order	8.3871 %	13	155	103
group	1.1976%	2	167	194
compound	0.0000%	0	46	0
where nested	0.0000%	0	50	43
where expression	0.0000	0	409	423
where operator	7.3171%	6	82	42
select agg	31.9328%	76	238	197
select without agg	6.0032%	38	633	794
select	1.02564%	60	585	639
having	11.1111%	4	36	33

Table 1. Accuracy of each components on our mix data

Conclusion

- Seq2seq model with attention on the input and database schema can transform natural language to SQL queries.
- Accuracy is low when:
  - Train, dev, test data set are drawn from different databases: HARD to predict new fields! May use copy mechanics…
  - Complex queries: HARD to understand the meaning for nested queries and predict correct grammar. May need grammar or hand-engineered features.

Reference

- Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generat- ing structured queries from natural language using reinforcement learning. CoRR, abs/1709.00103, 2017.
- Improving Text-to-SQL Evaluation Methodology Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan Dhanalakshmi Ramanathan, Sesh Sadasivam, Rui Zhang and Dragomir Radev. Proceedings of ACL 2018
- Li Dong and Mirella Lapata. Language to logical form with neural attention. In Association for Computational Linguistics (ACL), 2016
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

Seq2SQL Using Seq2seq with Attention Model

Dongxu Wang[1]
[1]LILY Lab, Yale University

LILY Lab

Introduction

For the task of converting natural language to SQL (nl2SQL) problem, many previous works have developed various methods, e.g. training sequence-to-sequence model with attention on the input sequence and schema, or translate the natural language to AST (Abstract Syntax Tree) of SQL first. These methods can also be improved by copy from input.
However, the modeling and testing are both hard because of existing data sets are sparse and general form SQL. So in the project we developed a new data set: a large amount of databases with natural language and SQL pairs. The new data set will have more flexible natural language sentences and standardized SQL components. By now more than half of the data set has been finished, and a smaller sample data set have been finished.
My work includes 1. labeling SQL and natural language pairs and 2. reviewing the SQL files created by others. 3. experimenting with a sequence-to-sequence model with attention on on database schema and over the input, using the data set we build.

Data Labeling

In the project we plan to label approximately 150 databases, each with 50 - 60 natural language to SQL pairs: around 30 pairs will cover different SQL components, and each may have a natural language paraphrase.
We chose many different components in the annotated data. We aim to generalize the data, and also standardize it for possible models to learn. The components are shown in table 1 by the order of priority, if they can be used to do the same task.
By now, we have created a subset of the ultimate data set, which conforms to the same constraints and contains 4204 pairs in train and 632 pairs in validation, 49 databases in train, and 7 database in validation.

Table 1. SQL Components and Labeling Priority

SQL Components
SELECT (multi)
FROM(multi)
DISTINCT(multi)
WHERE
JOIN
UNION, INTERSECT, EXCEPT
NESTED
EXISTS / NOT IN
ORDER BY / LIMIT
GROUP BY / HAVING
Expressions: Arithmetic Math (+-*/) String (LIKE, %) Opertors (NOT, AND, OR, !=, >=, <=) Aggregation (COUNT, SUM, AVG, MAX, MIN ...)

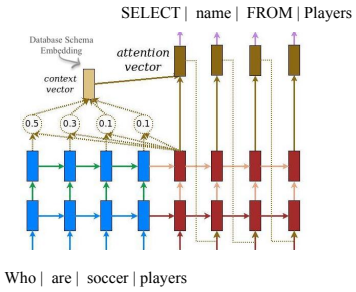

Figure 1. Sequence-to-Sequence Model with Attention of Database Schema

Approach

Experimented the sequence-to-sequence model with attention on the database schema and over the input. The model structure is shown in Figure 1, using our newly built dataset, whose train and test data are drawn from separate databases.
The encoder uses bidirectional RNN. The decoder attention layer uses Bahdanau attention layer, and the database schemas embedding is constructed with the natural language interpretation of table names and column names.
Previous work using sequence-to-sequence model in includes applying sequence-to-sequence model with attention over input, schema, with copy mechanics. Or applying augmented pointer network on different components of SQL structure. Or applying attention-based seq2tree (AST) method on programming languages.

Evaluation

Here we use a evaluation method based on AST(abstract Syntax Tree). That is transforming generated SQL queries and gold SQL queries into ASTs. Then we compared the different components of the two syntax trees. We split SQL queries into the following components and compute the recall, precision and F1-scores for:
a) select columns
b) select all aggregation functions
c) select all without aggregation functions
d) where expressions
e) operations in where
f) nested queries in where
g) group by
h) having
i) order by
j) except, union, intersect

Results

The result is shown in table 2. Nearly all the sentences are in correct SQL structures, but the ability of predicting the correct column names and predicting complex SQL of the model is poor. The reason that it cannot predict columns correctly is that these column names are not shown in training data, even with schema attention. Because the columns only in the test set is not 'paid' attention during training.
And the complex nested structures are harder for sequence-to-sequence models to learn.

Table 2. Predicted Accuracy of Different SQL Components

Components	Accuracy	Matched	Gold	Predicted
order	8.3871 %	13	155	103
group	1.1976%	2	167	194
compound	0.0000%	0	46	0
where nested	0.0000%	0	50	43
where expression	0.0000	0	409	423
where operator	7.3171%	6	82	42
select agg	31.9328%	76	238	197
select without agg	6.0032%	38	633	794
select	1.02564%	60	585	639
having	11.1111%	4	36	33

Conclusion

In nl2SQL task, I experimented the sequence-to-sequence with schema attention model used by [1]. However, the data set used by [1] differs from ours in that the training, validation, and test data set are drawn from single, identical database during each experiment. So the increased number of databases in our data set lead to more bigger search space, the different corpus and vocabulary during training and testing is also a severe problem.
As a result, the sequence-to-sequence model can learn a well-structured SQL, but is poor at predicting new databases' columns and tables and more complex structures. The schema attention during training is not working well.

Reference

[1] Improving Text-to-SQL Evaluation Methodology Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Dhanalakshmi Ramanathan, Sesh Sadasivam, Rui Zhang and Dragomir Radev Proceedings of ACL 2018