SQLNET: GENERATING STRUCTURED QUERIES FROM NATURAL LANGUAGE

INTRODUCTION

Synthesizing SQL queries from natural language is a long-standing open problem. Toward solving the problem, the de facto approach is to employ a sequence-to-sequence model. Such an approach requires the SQL queries to be serialized. Since the same SQL query may have multiple equivalent serializations, training a sequence-to-sequence model is sensitive to the choice from one of them. This phenomenon is documented as the “order-matters” problem.

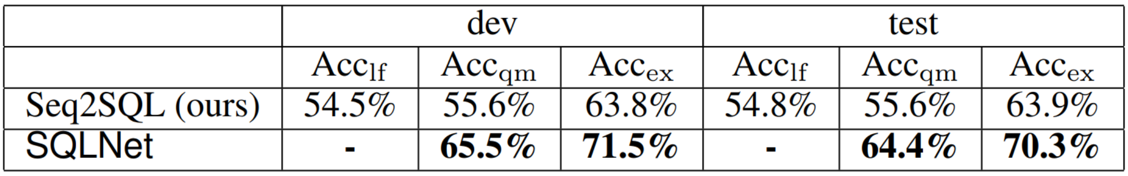
This paper propose a novel approach, i.e., SQLNet, to fundamentally solve this problem by avoiding the sequence-to-sequence structure when the order does not matter. In particular, we employ a sketch-based approach where the sketch contains a dependency graph so that one prediction can be done by taking into consideration only the previous predictions that it depends on. In addition, we propose a sequence-to-set model as well as the column attention mechanism to synthesize the query based on the sketch.

DATA

WikiSQL - A large crowd-sourced dataset for developing natural language interfaces for relational databases. WikiSQL is the dataset released along with the paper Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning.

METHODOLOGY

This paper proposes SQLNet to fundamentally solve this issue by avoiding the sequence-to-sequence structure when the order does not matter. The paper employ a sketch-based approach to generate a SQL query from a sketch. The sketch aligns naturally to the syntactical structure of a SQL query. A neural network, called SQLNet, is then used to predict the content for each slot in the sketch. Our approach can be viewed as a neural network alternative to the traditional sketch based program synthesis approaches.



	dev			test		
	Acc_{lf}	Acc_{qm}	Acc_{ex}	Acc_{lf}	Acc_{qm}	Acc_{ex}
Seq2SQL (ours)	54.5%	55.6%	63.8%	54.8%	55.6%	63.9%
SQLNet	-	**65.5%**	**71.5%**	-	**64.4%**	**70.3%**

To summarize, the main contributions in this work are three-fold. First, we propose a novel principled approach to handle the sequence-to-set generation problem. Our approachvoids the “order-matters” problems in a sequence-to-sequence model and thus avoids the necessity to employ a reinforcement learning algorithm and achieves a better performance than existing sequence-to-sequence based approach. Second, we propose a novel attention structure called column attention, and show that this helps to further boost the performance over a raw sequence-to-set model. Last, we design SQLNet which bypasses the previous state-of-the-art approach by 9 to 13 points on the WikiSQL dataset and yield the new state-of-the-art on an NL2SQL task.

MODEL

In SQLNET, Natural language descriptions and column names are treated as a sequence of tokens. We use the Stanford CoreNLP tokenizer (Manning et al., 2014) to parse the sentence. Each token is represented as a one-hot vector and fed into a word embedding vector before feeding them into the bi-directional LSTM. To this end, we use the GloVe word embedding. The size of the hidden states is kept 100. Adam optimizer is used with a learning rate 0.001. We train the model for 200 epochs and the batch size is 64. We randomly re-shuffle the training data in each epoch.

In Seq2SQL, the word embedding for tokens appearing in GloVe should be fixed during training. However, In this it was observed that performance can be boosted by 2 points when we allow the word embedding to be updated during training. Therefore, we initialize the word embedding with GloVe as discussed above and allow them to be trained during the Adam updates after 100 epochs.

EVALUATION

We compare our work with Seq2SQL, the state-of-the-art approach on the WikiSQL task. We compare SQLNet with Seq2SQL using three metrics to evaluate the query synthesis accuracy:

1. Logical-form accuracy: We directly compare the synthesized SQL query with the ground truth to check whether they match each other.

2. Query-match accuracy: We convert the synthesized SQL query and the ground truth into a canonical representation and compare whether two SQL queries match exactly. This metric can eliminate the false negatives due to only the ordering issue.

3. Execution accuracy: We execute both the synthesized query and the ground truth query and compare whether the results match to each other.

CONCLUSION

Our approach results in the exact query-match accuracy of 61.5% and the result-match accuracy of 68.3% on the WikiSQL testset. In other words, SQLNet can achieve exact query-match and query-result-match accuracy of 7.5 points and 8.9 points higher than the corresponding metrics of Seq2SQL respectively, yielding the new state-of-the-art on the WikiSQL dataset.