SQLGAN: Adversarial Training Methods for Text-to-SQL Generation

Yale

Eric Xue,[1] Dragomir Radev PhD[2]

[1]Yale College and [2]Department of Computer Science, Yale University

LILY Lab

Introduction

We propose the use of adversarial training methods to improve the generalization ability of existing Text-to-SQL models. The motivation is that MLE may encourage a model to generate the locally but not globally optimal tokens for the current time step, as well as overfitting. However, a discriminative network that automatically learns what correct SQL queries look like may provide better guidance for the model than MLE. Following this this motivation, we augment EditSQL with a discriminator and jointly train to two networks in an adversarial manner. During training, the existing model seeks to generate SQL queries that are realistic to fool the discriminator, who tries to differentiate between real and fake. More specifically, during training, we employ the discriminator to evaluate the generated SQL queries and feedback the evaluations to guide the learning of the generator. By doing this, we hope to provide more informative rewards for the generator that allow it to improve its generalization ability.

Methods

We augment EditSQL with a discriminative neural network and trained the two in an adversarial manner on the Spider dataset. The discriminator is implemented as a CNN that takes a natural language utterance and its corresponding SQL query and returns the probability of the query being correct. We use this discriminator to reward the generator according to the following objective function:

$$J(\theta) = \sum_{y} G_\theta(y_1|x) * Q_D^{G_\theta}(x, y_1)$$

where

$$Q_D^{G_\theta}((x, y_{1:m-1}), y_m = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D(x, y_{1:T_n}^{n}) \ \forall j < T \\ D(x, y) \qquad O.W. \end{cases}$$

Once we get more realistic SQL queries, we update the discriminator. We improve the generator iteratively in this way. To stabilize the adversarial training, we adopt teacher forcing, during which we assign each token in the gold query a reward of 1.

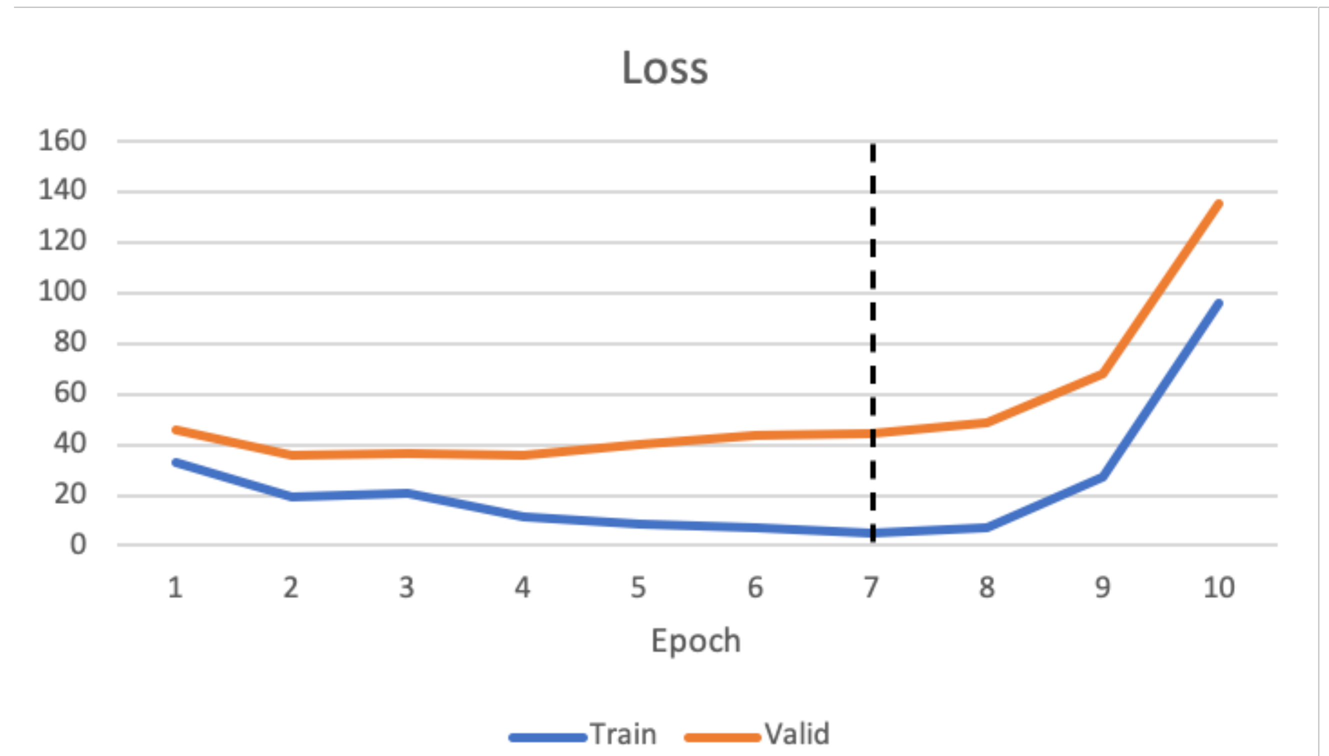

Figure 1. Generator loss.

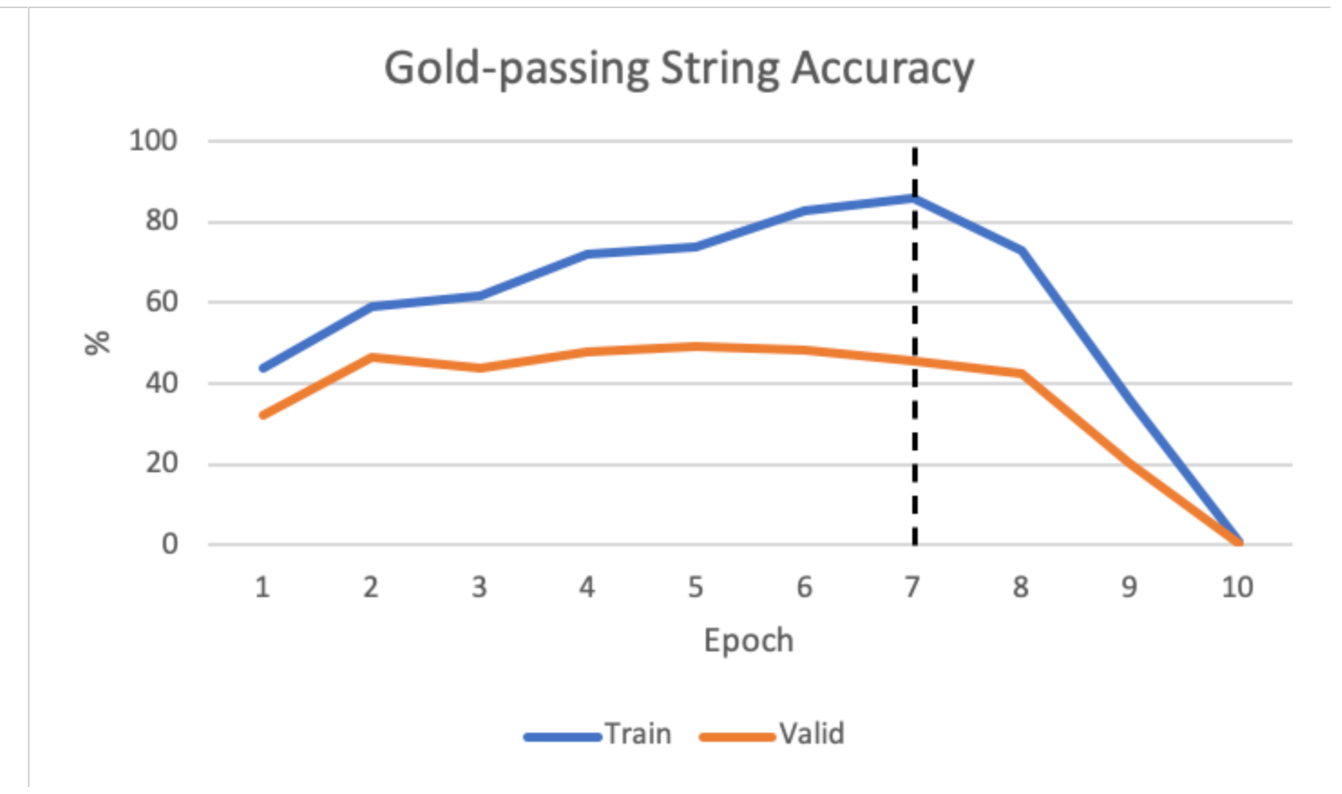

Figure 2. Generator string accuracy.

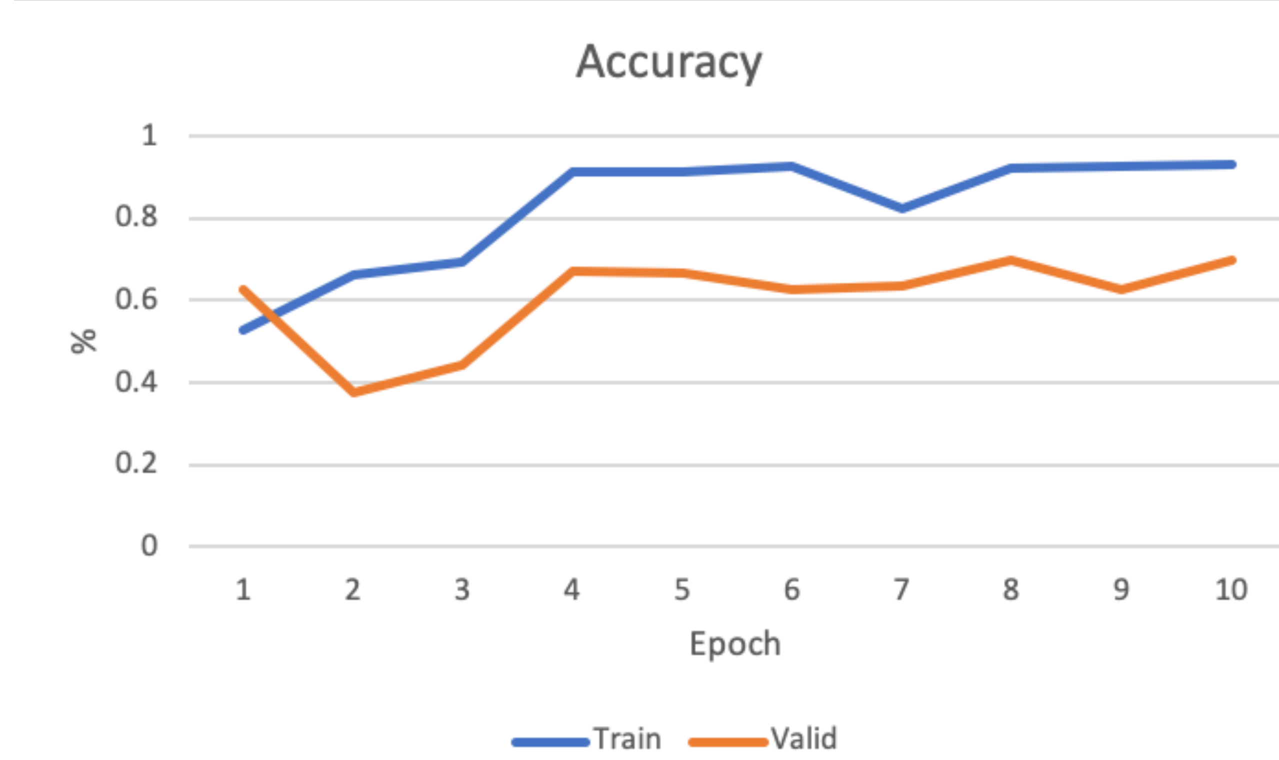

Figure 3. Discriminator accuracy

Figure 4. Discriminator confidence.

Figure 5. Discriminator loss.

Results

Unfortunately, we were unable to get the adversarial training process to converge. We hypothesize that a primary reason for the generator's divergence in adversarial training is the inaccuracy of the discriminator. Certainly, if the discriminator provides incorrect information to the generator, then the generator cannot learn how to generate correct SQL queries. Another concern is what distribution the discriminator is actually learning. It is possible that the discriminator is simply learning how to tell "correct-looking" queries apart from "incorrect-looking" queries, rather than determining whether the logic of the query matches the NL utterance as we had hoped. To address the questionable performance of the discriminator, we believe that its architecture should be augmented to include several features., such as including the table schema as input and computing attention between the convolutions on the NL question and the table schema. We hypothesize that rewards from the discriminator are not enough to achieve convergence. Rather than sequentially updating the generator's parameters with the rewards provided by the discriminator, then teacher forcing, we believe that integrating the two together may provide more stability for the model.

Conclusion

In this paper, we propose the use of adversarial training methods to improve the generalization ability of Text-to-SQL models. While we were unable to obtain results on the performance of this model, our work lays a foundation for future work in this subject area. Moreover, we were able to gain insight on the shortcomings of our current model, which will help better inform our design decisions of adversarial methods in the future. The next steps would be to implement the improvements mentioned in Result to see if we can achieve convergence.

Acknowledgement

I would like to thank Professor Radev and his graduate students for their support and advising throughout this research project.