Adversarial Attack for Semantic Parsing

Jingze Zhang, Shilin He(TA), Michael R. Lyu(Prof.)
Contact me at: jzzhang8@cse.cuhk.edu.hk

Abstract
Adversarial examples are common in Machine Learning technology. And surprisingly, adversarial examples are also common in our daily life. For example, you can try to Google with “Which team take the 3rd in world cup 2018?” and “Which team takes the 3rd in world cup 2018”, and you can get results with remarkable differences. To improve the robustness of semantic parsing models, we conduct a research on generating adversarial examples with various methods and attacking semantic parsing models with them. The result can give us some insight into an approach to building a more robust model.

Introduction
In the research, we train the model with Spider[1] dataset to generate SQL queries. For each question-query pair, extra data like database domain and expected structure for the SQL query was provided to help build the “schema” inside the model. Detailed information about Spider is presented in Table 1.[1]

Table: Details about Spider

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Spider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question Number</td>
<td>10,081</td>
</tr>
<tr>
<td>SQL Number</td>
<td>5,963</td>
</tr>
<tr>
<td>Database Number</td>
<td>200</td>
</tr>
<tr>
<td>Domain Number</td>
<td>138</td>
</tr>
<tr>
<td>Table Number/Database Number</td>
<td>5.1</td>
</tr>
</tbody>
</table>

GNN Model
GNN (Schema-based Graph Neural Network) model has 2 major features.[2]
- **Graph-based Network**: Compared to sequence, graph can represent the complex relation between tables more accurately and thus can serve as a more valid data representation.
- **Database Schema**: Database Schema is highly abstract data and rules for forming SQL query. It also contains the domains for generating GNN.

In general, GNN model has a higher stability in handling data related to several databases or tables and thus is more robust for complex input.

Methodology

Fast Gradient Sign Method (FGSM) with Approximation:
- **Find the gradient of input sentence \( x \) corresponding to the output \( y \) denoted by \( \text{grad} \)
- **Find the perturbed input sentence**
  \[ x_{\text{perturbed}} = x + \epsilon \cdot \text{sign} (\text{grad}) \]
- **Find the index \( i \) of the word with max|\( \text{grad}_{i} \)|
- **Find the word closest to \( x_{\text{perturbed}} \) denoted by \( x' \) with min|\( x' - x_{\text{perturbed}} \)| (Method 1) or max|\( \cos \angle x', x_{\text{perturbed}} \)| (Method 2)
- **Make \( x_{i} = x'_{i} \) and then generate a new sentence \( x' \)
- **Take \( x' \) as the input to the model and get the output \( y' \)
- **Compare the perturbed output \( y' \) with the original output \( y \) and observe the difference**

Synonym and Antonym Attack:
- **Calculate the gradient of input sentence \( x \) corresponding to the output \( y \) denoted by \( \text{grad} \)
- **Find the index \( i \) of the word with max|\( \text{grad}_{i} \)|
- **Replace the word at index \( i \) with its synonyms and antonyms according to the WordNet[3]
- **Test the structure of the new sentence with POS/Part of Speech) tagger by NLTK[4]
- **If the POS of word at index \( i \) remains the same, input it into the model and compare the output with the original output. Otherwise, continue to the next synonym/antonym**

FGSM Attack
We get the result through attacking the model with several values of \( \epsilon \). The the change of the successful rate (without considering the grammar correctness) can be presented with Figure 2. (The results for Method 1 and Method 2 are similar. The Figure 2 shows the result for Method 1.)

Figure 1: Structure for GNN Model

Figure 3: Distribution for Synonym Attack

Example #1 for similar input & output large differences:
- What ... with highest average attendance \( ? \Rightarrow \)
  select ... from stadium order by stadium.average desc limit 1
- What ... with greatest average attendance \( ? \Rightarrow \)
  select ... from stadium group by stadium.stadium_id
  order by avg (stadium.average) desc limit 1

Figure 4: Distribution for Antonym Attack

Example #2 for antonym with different SQL structure:
- How much ... younger dog weight \( ? \Rightarrow \)
  select weight from pets order by pet_age limit 1
- How much ... older dog weight \( ? \Rightarrow \)
  select count(*) from has_pet where has_pet.status = value

Conclusion
- **The model is robust enough to handle perturbation in a certain range, but not out of that range.**
- **The model can generally handle the synonym cases and recognize the semantic difference for antonyms.**
- **For the special cases, we have the hypothesis that the graph structure, while increases the robustness of the model, also increases the complexity inside the model. Thus, the complexity makes the weighting process hard to estimate and control, and the final weighing result may be unbalanced. Finally, when the word weighed too much in the input sentence get changed, the output will change with remarkable difference.**

Reference