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]

Spider Dataset

Dataset	Spider
Question Number	10, 181
SQL Number	5, 693
Database Number	200
Domain Number	138
Table Number/Database Number	5.1

Table 1: Details about Spider

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.

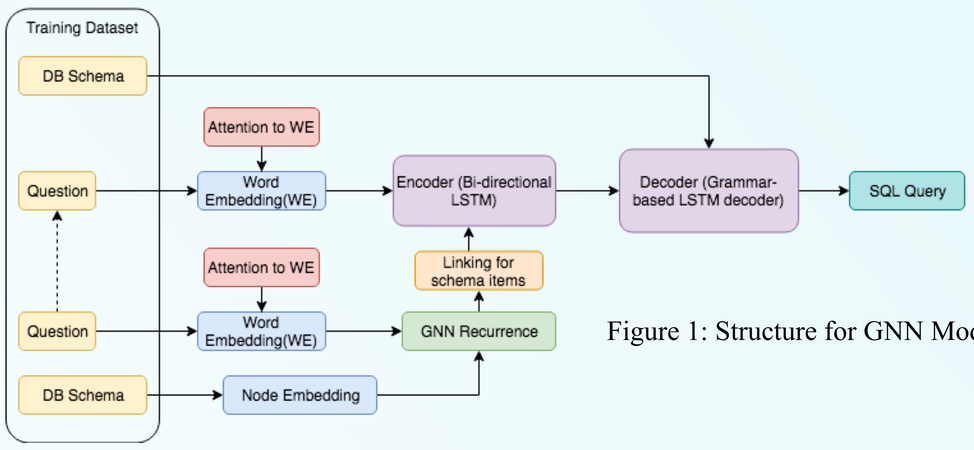


Figure 1: Structure for GNN Model

Methodology

Fast Gradient Sign Method (FGSM) with Approximation:

- Find the gradient of input sentence x corresponding to the output y denoted by $grad$
- Find the perturbed input sentence $x_{perturbed} = x + \varepsilon\, sign(grad)$ with the coefficient ε
- Find the index i of the word with $\max|grad_i|$
- Find the word closest to $x_{perturbed,i}$ denoted by x'_i with $\min|x'_i - x_{perturbed,i}|$ (Method 1) or $\max(cos < x'_i, x_{perturbed,i} >)$ (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 $grad$
- Find the index i of the word with $\max|grad_i|$
- 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)

Results

FGSM Attack

We get the result through attacking the model with several values of ε. 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.)

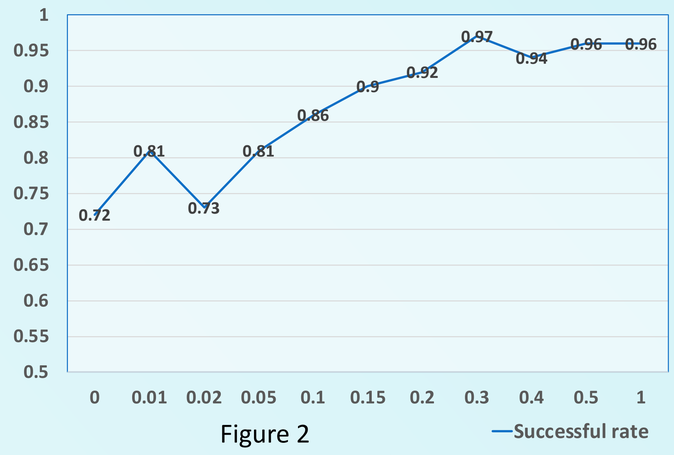


Figure 2
(x-axis is the value of ε)

Figure 3: Distribution for Synonym Attack

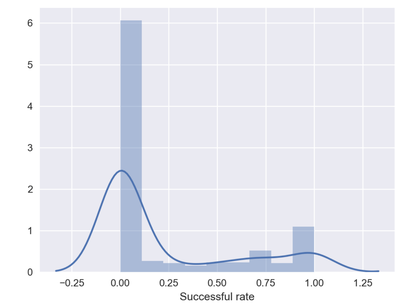
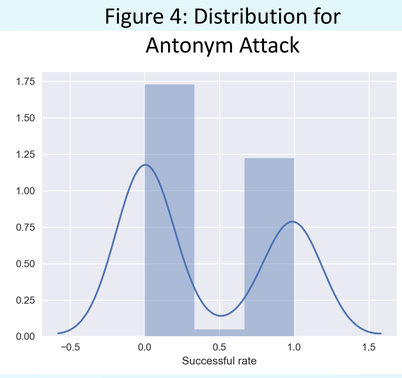


Figure 4: Distribution for Antonym Attack



Example #1 for similar input & output large difference:
What ... with highest average attendance ? =>
select … from stadium order by stadium.average desc limit 1
What ... with greatest average attendance ? =>
select … from stadium group by stadium.stadium_id order by avg (stadium.average) desc limit 1

Synonym and Antonym Attack

We attacked the model with the Synonym and Antonym method and counted the successful rate for all examples. For each example, the successful rate refers to the rate of the number of all successfully attacked synonyms(antonyms) to the number of all synonyms(antonyms). Figure 3 and Figure 4 shows the distribution of examples for different successful rates. In synonym part we in total tested 611 examples and got a final overall successful rate 61.1% for examples while in antonym we tested 125 examples and got overall successful rate 21.3% for examples.

Special Examples

During the attacking process, we also manually checked the attacked examples. Specially, the prediction result for some cases are uncommon compared to similar cases. (See Example #1 and #2)

Example #2 for antonym with different SQL structure:
How much ... youngest dog weigh ?=>
select weight from pets order by pet_age limit 1
How much ... oldest dog weigh ?=>
select count (*) from has_pet where has_pet.stuid = ' value '

Conclusion

FGSM Attack	• The model is robust enough to handle perturbation in a certain range, but not out of that range.
Synonym and Antonym Attack	• The model can generally handle the synonym cases and recognize the semantic difference for antonyms.
Special Examples	• 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 weighing 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

[1] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In EMNLP, 2018

[2] Ben Bogin, Matt Gardner, and Jonathan Berant. Representing schema structure with graph neural networks for text-to-sql parsing, 2019.

[3] George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[4] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.