Supplementing Neural Semantic Parsing Training with Accuracy
15-300, Fall 2019
Webpage: andrew.cmu.edu/ dchao1

David Chao
November 1, 2019

1 Description

My project proposal concerns neural semantic parsing, in specific, code generation. Code generation in this context is the process of taking a natural language input and converting it into code. Recent publications in code generation have framed the problem as a neural sequence to sequence (seq2seq) task. This proposal was not created from talking to a project mentor, and as such, may not be feasible or as well versed in the field as it could be.

There have been numerous advances in the field of neural semantic parsing in recent years. Recent literature has advanced state of the art models by providing the ability to copy from source, the framing of general purpose code as an abstract syntax tree and applying grammar, and the ability to abstract out the associated grammar from the model. However, these advances have not tackled a persistent problem in neural semantic parsing, which is the accuracy of the generated code on complicated situations.

To discuss this, some background on metrics should be given. There are two main metrics used to determine the quality of generated code. The first is BLEU, which measures how similar the code looks to the reference. The second is accuracy, which indicates if the code actually does what it is intended to do. Current neural model’s train in a manner that attempts to maximize both BLEU and accuracy. They do so by maximizing the model’s chances of outputting the exact reference. However, for a given task, there are multiple ways to write accurate code. It doesn’t make sense to exclude all other accurate outputs in favor of only the reference when training. We propose a few approaches that may help neural models prioritize accuracy, which we hope will ultimately result in models that produce more accurate code.

We have thought of a few approaches that hypothetically could improve model accuracy. The first one involves applying reinforcement learning techniques to this task, where accuracy of generated code can be interpreted as a reward function. The second approach involves altering the likelihood function of outputs to be aware of this reward function. The last approach involves modifications to the decoding steps in the neural model via re-ranking. Further discussion of these approaches will be provided in the literature review.

2 Goals

The goal of this project is to supplement current state of the art semantic parsing models. Two obvious metrics result. The first metric measures whether or not the proposed changes actually
benefits current models. The second metric measures the viability of proposed changes in terms of computation speed and implementation.

**75% Goal:** If things go more slowly than expected, I should have determined a viable method of improving accuracy. I should have implemented the method and been able to start testing on current data sets. The method itself may not be fully implemented correctly, or it’s results end up being worse than current models, or the viability of the implementation is poor due to large computation time and confusing implementation.

**100% Goal:** If things go smoothly, I should have implemented a method of improving accuracy. I should have done sufficient testing on data sets and hopefully see improvements in model accuracy. The method itself may still be hard to implement or time consuming, but the results and conclusions should be clear.

**125% Goal:** If things go faster than expected, I should have implemented a method or multiple methods of improving accuracy. These should be extensively tested on data sets and should show improvement or insights in current models. The method should hopefully be easy to implement or viable for use in future models.

## 3 Milestones

Due to how the actual content has not been decided for this project, it’s hard to determine concrete milestones for progress.

**15-300 Milestone:** Do research into current proposed methods as well as look into new methods. Decide upon initial method to try out.

**January 27th:** Get state of the art model to train and output results on chosen data sets.

**February 10th:** Augment framework for running data sets as needed for method.

**February 24th:** Have first implementation of method running on data set.

**March 16th:** No milestone. Refine data set and method as needed.

**March 30th:** No milestone. Refine data set and method as needed.

**April 13th:** Have data set and methods in place to start performing final testing to obtain results.

**April 27th:** Have final test results and appropriate analysis done.

## 4 Literature Search

There are a few papers regarding current neural semantic parsing models. These include models introducing the copy feature [2], models that utilize abstract syntax trees to constrain the decoding space [4], and models that abstract away the grammar [5]. These papers should help in providing baselines to augment.

The first discussed method is introducing reinforcement learning techniques to the neural model. There has been research performed in this area, and Keneshlee et al. [1] provides resources to help determine which concepts are applicable and easy to implement, as well as pros and cons. We expect this paper to provide further insight into the challenges and benefits of RL techniques in seq2seq models. The paper also provides source code implementing it’s discussed models, which will be a useful reference.

The second discussed method is changing the loss function of the neural models to be reward aware. The standard loss function used in many models is negative log likelihood. Norouzi et al. [3] introduced a new loss function, called Reward Augmented Maximum Likelihood (RAML),
and applied it on seq2seq models for speech recognition and machine translations, showing notable improvements. Attempting to implement this for neural semantic parsing could prove beneficial.

The last discussed method involves changing the decoding of the neural model such that it produces higher accuracy code. There are many ways to approach this. One way is proposed by Yin et al. [6], and could be used as a baseline to build further upon or use in conjunction with the other methods.

For the duration of this project, further literature review should be done such that various methods of improving the neural models are considered. Further understanding of the various methods is also needed to determine potential benefits as well as viability of each method before committing to any one.

5 Resources Needed

Sufficient compute will be needed. This can be done either through AWS or other cloud services, or through use of local GPU systems. I plan on acquiring a GPU for compute, which should hopefully be sufficient seeing as I don’t currently have a mentor for this proposal.

References


