PNet - ParseNet

Progress Report

Sai Krishna Karanam karanam.s@husky.neu.edu
Nischal Mahaveer Chand mahaveerchand.n@husky.neu.edu
Varun Sundar Rabindranath rabindranath.v@husky.neu.edu

I. Changes

We have renamed the project title to PNet, our implemention of a general purpose Natural Language (NL) to Abstract Syntax Tree (AST). We have looked at multiple papers and dataset. Have decided to go with base paper.

Apart from logistics, we have been working on our dataset uptill now. The xxx section describes the problems with the current data available and the changes and transformations we proposed.

II. Data Preprocessing

bla bla

III. Project Plan

Our initial project plan has significanlty changed to incorporate time into the data preprocessing step mentions in (section). The current project plan is as follows: For this project, we plan to have four different phases,

1) Phase 1: Literature Survey [DONE]

2) Phase 2: Data Preprocessing and Cleaning [Inprogress]

3) Phase 3: PNet
   a) Data Input
   b) Data Process (tokenization, etc..)
   c) Create model
   d) Training
   e) Testing

4) Phase 4: Evaluation on different data inputs
   Each model will be carefully evaluated and examined for potential optimizations. The best model will be used for further optimization and improvement.

IV. PNet

As presented by Maxim Rabinovich et. al. [1], PNet used a top-down recursive approach for the decoding process. We have decided to work with PyTorch (cite) for model creating, training, testing, and evaluation.

The basic data processing pipeline has been successfully implemented. Like most deep learning based systems, we are using a parallel text corpus based data input module to feed the input NL sequence into the encoder, and the output AST sequence to the decoder during training. About PTC: single source file, single target file, vocab files for both. Each line is a training example, NL from source text and AST from the target text.

Proposed decodig process as in (cite maxim). The problem with the model is the copy mechanism, for this, we plan to used a explicit copy module, based on (cite copy paper). We hypothise that integrating the above mentioned copy module should impove accuracy on the HS dataset.

For Djanjo dataset, bla bla bla.

V. Experiments

As mentioned in (section name), the data is (adj bad?). To show this, we run a seq2seq model on TensorFlow (cite). First we feed the data into the vanilla seq2seq module provided separately by Google (link to github repo). As expected, the results are not impressive, confirming our need for better a data corpus. Evaluation results can be found in (table ?).

VI. Proposed Plan for future phases of project

Presently, we are in the process of building the encoder-decoder model, defining architecute and parameters in PyTorch.

References

[1] M. Rabinovich, M. Stern, and D. Klein, "Abstract syntax networks for code generation and semantic parsing," arXiv preprint arXiv:1704.07535, 2017.