Assignment 3: Transition Parsing with Neural Networks

CSE 538 Fall 2019
Released: Oct 30
Due: Nov 13 11:59pm

1 Transition Parsing System

In this assignment you will implement the Neural Network based Transition Parsing system described in [Chen and Manning, 2014]. This assignment consists of following components:

1.1 Parsing Algorithm

In `parsing_system.py`, there is an overall wrapper that begins with the initial state, and asks the classifier to provide a distribution over the available actions, selects the next available legal action, and then changes the state by applying the action.

You will be implementing the ARC standard system rather than the eager version. This is a slight modification with only three types of actions (shift, left, right) instead of the four. The paper[Nivre, 2004] describes in detail.

1.2 Neural Network Classifier using TensorFlow

You will implement the training and test functionality for the transition classifier. You will generate the features given the (sentence, tree) pairs and write the loss function to use for training the network.

2 Getting Started

As well as reading this document, which covers the general tasks being asked of you, you should follow the README in the code repository. The README outlines each part of the code, file by file, as well as explains what should be turned in on submission. This assignment pdf serves only to give you an introduction to the assignment and what to put in your report, the details are in the README!

3 Model Implementation [60 pts]

You will implement a neural Dependency Parsing model by writing code for the following:
From Incrementality in Deterministic Dependency Parsing (2004, Nivre)

- the arc-standard algorithm

From A Fast and Accurate Dependency Parser using Neural Networks (2014, Danqi and Manning)

- feature extraction
- the neural network architecture including activation function
- loss function

4 Experiments and Analysis [30 pts]

You will carry out experiments to figure out the effects of following on the model performance:

- activations (cubic vs tanh vs sigmoid)
- pretrained embeddings
- tunability of embeddings

and write your results and explain your findings in the report. We have provided you with a script to run these experiments. Please look at the readme for details.

5 Running on GPU

This assignment should be doable on your personal laptops. But if you are facing problems, you can run the experiments on google colab. Please look at `colab_notes.md` in the code for instructions on how to do it.

6 Report

The write up should be 3-4 pages of length. You should have the following 2 sections:

1. Model Implementation: Explain high level idea of how you implemented the code.
2. Experiments: Present the results of the experiments and discuss your findings.

7 Grading Rubric

- 60 points – Implementation
- 30 points – Experiments and Analysis
- 10 points – Evaluation of the test predictions. (Relative grading)

References

[Chen and Manning, 2014] Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

[Nivre, 2004] Nivre, J. (2004). Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57. Association for Computational Linguistics.