Jonny Li

ML/Software Engineer / https://jonny.li / jonny.li@mail.utoronto.ca / Github: @jonnyli1125 / LinkedIn: jonnylii

Education

University of Toronto Graduated Jun 2021

Honours Bachelor of Science, Computer Science and Linguistics
Courses: Machine Learning, AI, Computational Linguistics, NLP, Data Structures, Algorithms, Operating Systems
Activities:
- Foreign Exchange (2018-2019): Studied Computer Science and Linguistics at University of Tokyo.
- UofT Neurotech Workshops Lead (2017-2018): Created tutorial workshops for programming basics (i.e. Python, Git, etc).
- UofT Japan Assocation Webmaster (2017-2018): Designed and managed club website with HTML/CSS/JavaScript.

Skills

Programming Languages: Python, JavaScript, Java, C#, Ruby

Technologies/Frameworks: PyTorch, Keras/Tensorflow 2, scikit-learn, Colab/Jupyter Notebooks, Git, SQL, Ruby on Rails

Other: Machine learning, Neural networks, NLP, Agile methodologies, Test suite frameworks, CI/CD tools

Employment

SDE Intern - Amazon @ VirtualLocation Jun 2020 - Sep 2020

- Developed command-line tools for dependency chain analysis of search engine configuration metadata.
- Implemented DFS-like graph search with memoization optimizations on dense dependency graph in Python.
- Conducted analysis on real configuration metadata to determine relevant dependency relations of configuration objects.
- Devised efficient methods to store large output in Amazon S3 using data structures and compression algorithms.

Software Intern - Mitsucari @ Tokyo, Japan Sep 2018 - Aug 2019

- Designed and implemented model/view/controller components with Ruby on Rails, PostgreSQL, jQuery, Bootstrap, SASS.
- Wrote test suites and utilized CI/CD tools to deploy Ruby on Rails app hosted on Heroku.

Projects

BERT+MCTS-based Chess AI @ github.com/jonnyli1125/chess-bert-mcts Jun 2021

- Developed AlphaZero-style chess AI model in PyTorch by combining BERT with policy/value network heads.
- Implemented Monte Carlo Tree Search agent and universal chess engine interface protocol (UCI) in Python.
- Implemented data modules for pre-training BERT for masked language modeling on chess game states.
- Implemented data preprocessing/training pipelines for supervised learning of policy/value networks on CCRL computer chess games dataset using HDF5 dataset format, PyTorch, and GCP/Colab.
- Integrated bot with online chess server lichess.org for live online play, using Python and Lichess Bot API.

BERT-based Grammatical Error Correction @ github.com/jonnyli1125/gector-ja Apr 2021 - Jun 2021

- Implemented grammatical error correction model for Japanese, described in the Grammarly research paper "GECToR -- Grammatical Error Correction: Tag, Not Rewrite" (Omelianchuk et al. 2020).
- Implemented token classification neural network for predicting token-level edit transformations, using pretrained BERT model from Huggingface library and Keras/Tensorflow 2.
- Implemented data preprocessing/training pipelines for synthetic error corpus generation + token label computation on Wikipedia dump and NAIST Lang8 Learner Corpora, using TFRecords dataset format, Tensorflow, and GCP/Colab.
- Utilized linguistic domain knowledge to optimize synthetic error corpus generation for improved model training.
- Created interactive web demo app to get grammatical error correction on user input, using Flask and HTML/CSS/JavaScript.

Shift-Reduce Dependency Parser @ github.com/jonnyli1125/jp-srparser Apr 2021

- Implemented shift-reduce dependency parser for Japanese, with a feed-forward neural network to predict the next parser action given a parser state, using PyTorch and pretrained word embedding layer from Wikipedia2vec.
- Implemented data preprocessing/training pipelines for UD Japanese GSD treebank corpus in PyTorch.
- Implemented dependency tree visualization on interactive user input using spaCy and Colab.

Language Learning Discord Bot @ github.com/jonnyli1125/jp-bot Dec 2019

- Developed Discord chat bot for Japanese language learning, written in Python and deployed on Heroku.
- Implemented dictionary lookup/translation commands by interacting with RESTful APIs and scraping online dictionary websites using HTML parsers and regular expressions.
- Implemented server moderation features to keep track of punished/banned users by integrating user data with MongoDB.