

# Jonny Li

ML/Software Engineer, Focus in Natural Language Processing

Website: <https://jonny.li> / Email: [jonny.li@mail.utoronto.ca](mailto:jonny.li@mail.utoronto.ca) / Github: [@jonnyli1125](https://github.com/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:

- UTokyo 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 Association Webmaster (2017-2018): Designed and managed club website with HTML/CSS/JavaScript.

## Skills

**Programming Languages:** Python, JavaScript, Java, Perl

**Technologies/Frameworks:** PyTorch, Keras/Tensorflow 2, scikit-learn, Colab/Jupyter, Git, SQL

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

## Employment

### Software Engineer, Language Modeling - SoundHound AI @ Toronto, Canada

Oct 2021 - Present

- Implemented and trained statistical and neural language models used in automatic speech recognition (ASR) pipeline.
- Optimized Hebrew ASR performance by 10% on key performance indicator benchmark through LM experiments.
- Streamlined development process through implementing/managing data collection, analysis, and build scripts.

### SDE Intern, Search - Amazon @ VirtualLocation

Jun 2020 - Sep 2020

- Optimized search engine efficiency and reduced engineering debt by implementing a dependency graph analyzer in Python to safely remove redundant search engine configuration objects.
- Implemented data processing pipelines for transferring data between AWS components.
- Devised methods to compress large data 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

### Webspeak to English Translator @ [github.com/jonnyli1125/piemaneese-translator](https://github.com/jonnyli1125/piemaneese-translator)

Feb 2022

- Implemented statistical machine translation model for translating webspeak to English using Python and NLTK.
- Formulated problem as SMT task and defined model components by derivation of Bayes theorem.
- Trained n-gram statistical language models on web chat corpus to predict corrected English words from context.
- Innovated solution for translation model due to lack of labelled training data, using linguistic domain knowledge.
- Implemented Discord chat bot interface using Discord API and Python, and deployed to cloud with Heroku.

### BERT-based Grammatical Error Correction @ [github.com/jonnyli1125/gector-ja](https://github.com/jonnyli1125/gector-ja)

Apr 2021 - Jun 2021

- Implemented grammatical error correction model for Japanese, as described in a [2020 Grammarly research paper](#).
- Achieved state of the art result with 10% relative performance increase on benchmark corpus for Japanese GEC.
- Implemented neural network for predicting token-level edit transformations, using pretrained BERT model from Huggingface library and Keras/Tensorflow 2.
- Implemented data processing and synthetic data generation pipelines for Wikipedia dump and [Lang8 Parallel Corpus](#), and training pipelines with Tensorflow and GCP/Colab.
- Created interactive web app demo using Python/Flask and HTML/CSS/JavaScript.

### Neural Shift-Reduce Dependency Parser @ [github.com/jonnyli1125/jp-srparser](https://github.com/jonnyli1125/jp-srparser)

Apr 2021

- Implemented shift-reduce dependency parser for Japanese with a feed-forward neural network to predict parser actions.
- Implemented model training and data processing pipelines for [UD Japanese GSD treebank corpus](#) in PyTorch.
- Implemented dependency tree visualization on interactive user input using spaCy and Colab.