Pengfei **He**

Software Developer · Machine Learning Engineer · NLP Researcher

1+ year of NLP research & work experience, and 3+ years of hands-on experience on machine learning tools.

425-772-7623 | hepengfe@uw.edu | hepengfei.ml | feipenghe | hepengfe | hepengfe

Education

University of Washington *Seattle, WA*

B.S. Applied Computational & Mathematical Sciences, Data Science & Statistics. Minor in Linguistics. *Sept. 2018 - June. 2021*

Work Experience

Petuum Inc. *Sunnyvale, CA*

Machine Learning Engineer (NLP), Director: Jayesh Gada and Hector Liu *Aug. 2021 - Present*

- Designed and implemented a **Distributed Faiss Passage Indexer** with **unit testing** that coordinates nodes with limited computing resources to inference passage embeddings and collect kNN search results from multiple nodes given passage queries in the distributed environment. This method enables **selecting global negatives** from the large scale index which cannot fit into one machine's RAM and GPU memory in a distributed setting of low per-node-RAM(64GB) and per-node-GPU-memory(12GB).
- Researched and implemented **Asynchronous Index Refresh** from **ANCE** based on forte project framework for QA applications, and it enables training the wikipedia passage embeddings on a cluster with low per-GPU-memory(12GB) and makes the **training significantly more efficient** than the traditional sequential pipeline of training and inference.
- Implemented a dataset class for general classification tasks and enabled **open-source project forte** loading various classification datasets where users can **customize the input entries and labels** based on training tasks by editing configuration files.

H2 Lab at University of Washington *Seattle, WA*

Undergraduate Research Assistant, Advisor: Aida Amini, Sewon Min and Hanna Hajishirizi *Sept. 2020 - March. 2021*

- Adopted **sequence-to-sequence Transformer** models using Huggingface for **question answering systems** to generate a sequence of answers, and **pre-trained** them on NQ dataset and **fine-tuned** them on AmbigQA dataset to improve its performance on downstream tasks.
- Developed a **clustering-assisted question answering system** to address **question ambiguities** and to improve **answer diversity**, and our model achieved **higher recall** than the baseline model.
- Implemented **parallel model inference** on multiple GPU and utilized all CPU threads to prepare batch data, and it **speeds up the evaluation process by 4 times** and the whole training process significantly under 2-GPU settings.
- Implemented **dynamic data augmentation** on AmbigQA dataset for multi-answer QA pairs and low-frequency answers such as dates and numbers to address the difficulty of Reader outputting multiple answers and the long-tail distribution in the multi-answer QA setting.
- Built **Python scripts** to **parallelize pre-processing** unstructured document-level medical data at scale and to manage data fields at different level properly as input data for an end-to-end medical relation extraction system. The inference relations help our biologist collaborators to read biomedical literature more efficiently.

Courses

Deep Learning	CSE543 Deep Learning, CSE599I Generative Model
Machine Learning	CSE547 Machine Learning for Big Data, CSE546 Machine Learning
Natural Language Processing	CSE517 Natural Language Processing, CSE599D1 Multilingual NLP Seminar, LING572 Statistical NLP
	LING 571 Deep Processing Techniques for NLP, LING 472 Introduction to Computational Linguistics
Prescriptive Analytics	CSE542 Reinforcement Learning, CSE573 Artificial Intelligence
Data Analytics	CSE414 Database System, SOC225 Data & Society
Algorithm	CSE521 Advanced Algorithms, CSE373 Data Structure & Algorithm
Optimization	MATH407 Linear Optimization, MATH164 Optimization(UCLA)
	AMATH352 Applied Linear Algebra & Numerical Analysis, AMATH301 Beginning Scientific Computing

Course numbers above 500 represent **graduate levels*

Skills

Programming Python, Huggingface, PyTorch, Multiprocessing, Java, Numpy, Spark, MapReduce, C, SQL, Linux, Jupyter Notebook

Mathematics Optimization, Linear Programming, Probability and Statistics, Numerical Analysis

Projects

Sentence Corruption Classifier/Generator *Seattle, WA*

Personal Project *Summer 2020*

- Implemented **Byte Pair Encoding** to encode text and to reduce the text noise, and built a specialized **collate function** to unify sentence batch.
- Built a pipeline of RNN/LSTM/GRU to classify corrupted sentence and seq2seq model to generate corrupted sentence using **PyTorch**.
- Wrote **Linux** scripts and utilized **DataParallel** to run experiments on multiple GPUs.

Survey of Spontaneous Emergent Discrete, Compositional and Point-Symmetrical Signals *Seattle, WA*

ACMS Honor Thesis Advisor: Shane Steinert-Threlkeld *Spring 2020 - Fall 2020*

- Implemented the cross-entropy loss function and adjusted model output accordingly for existing experiments using PyTorch.
- Analyzed clustering results of the intermediate layer of autoencoder under the new training conditions in **Jupyter Notebooks** and **discovered a point symmetry** phenomenon for min/max functions, and presented the finding in the group meeting.

Honors & Awards

2018~2021 **Dean's List**, Undergraduate academic scholarship over six quarters. *Seattle, WA*

2021 **ACMS Honors Student**, Departmental Honors for students with academic excellence and an honor thesis. *Seattle, WA*