

# YAPEI CHANG

**EMAIL** echang33@smith.edu **TEL** +1 413 801 1370

**GITHUB** <https://github.com/lilakk> **LINKEDIN** [www.linkedin.com/in/ella-yapei-chang](http://www.linkedin.com/in/ella-yapei-chang)

**EDUCATION** Smith College, Class of 2022, B.A. in Computer Science with Math Minor, GPA: 3.87/4.0, Major GPA: 3.91/4.0

**SKILLS** Python, NLP, ML/DL, PyTorch, scikit-learn, Tensorflow, Java, JavaScript, TypeScript, React, Golang, AWS technologies, GraphQL, Git, C, C++, Common Lisp

**COURSEWORK HIGHLIGHT** [Advanced Natural Language Processing](#) (grad-level), [Neural Networks: Neuroscience to AI](#) (grad-level), [Machine Learning](#), [Deep Learning Specialization by deeplearning.ai](#), [Deep Neural Networks with PyTorch by IBM](#), [Tensorflow in Practice Specialization by deeplearning.ai](#), [Udacity Natural Language Processing Nanodegree](#), Discrete Mathematics, Linear Algebra, Multivariable Calculus, Differential Equations, Linguistics, Cognitive Psychology, Neuroscience

## INTERNSHIP EXPERIENCE

Software Development Engineer Intern at Amazon Seattle May - Aug. 2021

- Worked as a full-stack developer in the Make on Demand (MoD) team on a project focused on printer quality assurance, which involved working with microservices and various AWS technologies. Key contributions are listed below in detail.
- Implemented popups using React and TypeScript in UX Service. Dynamically rendered UX elements by calling Device Service, which retrieves data from DynamoDB. Implemented functions to submit requests to Execution Service.
- Added new items to a DynamoDB table by following the single-table non-relational design principle.
- In Device Service, implemented functions in Golang that queries DynamoDB with GraphQL.
- In Execution Service, modified an AWS state machine to pass UX Service requests to Imposition Service.
- In Imposition Service, implemented a workflow in Java that based on the request from UX Service, downloads files from Amazon S3, imposes text and barcodes on them, then uploads them to S3.

## PUBLICATIONS

"RELIC: Retrieving Evidence from Literature in Context" Accepted into ACL 2022 ([link to preprint](#))

- Worked with Professor Mohit Iyyer from UMass Amherst. The paper presents a dataset (ReLiC) for the following literary evidence retrieval task: given an excerpt from a literary analysis surrounding a masked quotation, retrieve the quoted passage from the set of all passages in the quoted work. ReLiC contains 90K literary analysis excerpts.
- Crawled HathiTrust to retrieve full literary analysis documents that quote one or more of 79 widely-read English texts, extracted spans from those texts that constitute a major part of our dataset. Built a preliminary model in PyTorch by training BERT to choose the correct quote given the context surrounding a masked quote and 2 candidate quotes. Built and evaluated BM25 and ColBERT on our dataset by tailoring existing open-source code to our use case. Results from the evaluations were used to gauge performance on a simplified proxy task to compare against other models.

"A Broader Range for 'Meaning the Same Thing': Human Against Machine on Hard Paraphrase Detection Tasks" Accepted into ACS 2020 ([link to publication](#))

- Worked with Professor Jamie Macbeth from Smith College. The paper investigates how BERT and humans identify paraphrases in fundamentally different ways, bringing out the importance of knowledge representation.
- Conducted data analysis and visualization on a variety of semantic textual similarity metrics (including Jaccard and Levenshtein distances) in Python using Numpy, Pandas, Matplotlib, NLTK, and SciPy.

## RECENT RESEARCH / PROJECT EXPERIENCE

Unlikelihood Training for Neural Text Generation Sep. 2021 - Present

- Working with Professor Mohit Iyyer from UMass Amherst. The goal of the project is to implement a sequence-level finetuning object on top of a pre-trained language model to make it less prone to hallucinating random facts.
- Trained multiple models on a GPU cluster to compare performance. Conducted part-of-speech analysis on different models' outputs. Set up and ran experiments to evaluate selected models on SWAG. Implemented token-level beam search in PyTorch using a trained retriever for reranking.

Feedback Transformer-XL Feb. - May 2021

- Using PyTorch, incorporated a mechanism that passes information computed at previous higher layers to future lower layers into Transformer-XL.
- Model achieved a 1.4% improvement in perplexity on the lm1b dataset after 28 evaluation steps, and a 2% improvement in perplexity on the wikitext-103 dataset after 21 evaluation steps. The main limiting factor was that due to insufficient computational resources, we could not run full experiments.

English Language Interpreter (ELI) Apr. 2019 - Mar. 2021

- Worked with Professor Jamie Macbeth from Smith College on ELI, a symbolic semantic parsing system implemented in Common Lisp that takes an input sentence and outputs its conceptual dependency (CD) form. It can be used to generate paraphrases of a given sentence.
- Reengineered the existing system, implemented new CD primitives, polysemy resolution, and a question-answering module in Common Lisp.

Keyword Extraction with Clustering Algorithms Oct. - Dec. 2020

- Built keyword extraction pipelines in scikit-learn using affinity propagation, k-means clustering, and mean shift clustering respectively on FastText word embeddings. Applied them to 106 NLP paper abstracts on Arxiv. Evaluated them against RAKE and TextRank using Sørensen–Dice similarity.