Archit Uniyal

+1 (434)-833-0881 | a.uniyal@virginia.edu | linkedin.com/in/uniyalarchit | github.com/archit31uniyal

Education

University of Virginia — Charlottesville, VA
Masters in Computer Science — *Aug. 2022 – Dec. 2024*

Panjab University — Chandigarh, India
Bachelors in Engineering in Computer Science — *Aug. 2018 – July 2022*

Experience

Software Developer Intern — May 2024 – Aug. 2024
OmniVision Technologies — *Santa Clara, CA*

- Deployed and debugged using GDB industry-standard automated testing methods like Crest and KLEE for C and C++ codebase
- Migrated these methods to Python, reducing software dependencies which come along with C++
- Developed a novel instrument-free method for automated unit test generation which reduced processing time by 10 times and is currently being used by the algorithm team internally

Research Scientist Intern — May 2023 – Aug. 2023
Oracle — *Burlington, MA*

- Lead research in the privacy group focusing on prevention of training data leaks in large language models (LLMs)
- Fine-tuned LLMs using HuggingFace DeepSpeed and techniques such as LoRA in Pytorch
- Patented a new concept of **entity-relationship privacy** in LLMs which targets a more realistic definition of privacy at the entity-level instead of measuring verbatim training data outputs

Research Intern — Jan. 2022 – July 2022
Samsung Research Institute — *Noida, India*

- Lead research on improving the existing time series-based human activity recognition algorithms in Galaxy watch5 and contributed 10K lines of code to the codebase
- Fine-tuned generative adversarial networks in Pytorch and reduced noise in the videos by 5 times compared to traditional denoising filters in Samsung smartphones

Projects

An Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models
Python, PyTorch, Matplotlib, HuggingFace transformers, large language models

- Created a fine-tuning dataset by adding synthetic data to MIMIC-III dataset
- Demonstrated increased memorization in large language models on the fine-tuning dataset
- Analyzed different fine-tuning methods such as LLM head tuning, the full LLM tuning and adapters
- OpenAI GPT-2 was used for the experiments on Wikipedia $wikitext-2-raw-v1$, Penn Treebank $ptb-text-only$ and enron email dataset

Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks
Python, PyTorch, Matplotlib, HuggingFace transformers, large language models

- Invented an efficient membership inference attack (MIA) on masked language models (MLM) which were previously thought to be immune to MIA
- Formulated a likelihood ratio hypothesis test which involves an additional reference MLM to more accurately quantify the privacy risks of memorization in MLMs
- ClinicalBERT was used as the target model trained on MIMIC-III dataset and Pubmed-BERT as the reference model trained on PubMed texts

Indian Sign Language
Python, OpenCV, Tensorflow, Pyaudio, Pyttsx3

- Built an interface to detect Indian Sign Language (ISL) alphabets and convert them to text using InceptionResnetV3
- Engineered LSTMs to generate text sequences from the classified ISL alphabets
- Integrated Google Text-to-Speech API for audio output

Publications

- An Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models (*EMNLP, 2022*)
- Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks (*EMNLP, 2022*)
- DP-SGD vs PATE: Which Has Less Disparate Impact on Model Accuracy? (*ICML, 2021*)
- GC-NET for classification of glaucoma in the retinal fundus image (*Machine Vision and Applications, 2020*)
- DC-Gnet for detection of glaucoma in retinal fundus imaging (*Machine Vision and Applications, 2020*)

Technical Skills

Languages: Python, C/C++, Java, SQL (Postgres), JavaScript
Frameworks: Pytorch, Tensorflow
Developer Tools: Git, Docker, Google Cloud Platform, Flask, React, VS Code, Bash, Linux, Git, Azure, AWS, HPC clusters, Hugging Face, OpenAI API, LangChain, Apache arrow
Concepts: Differential Privacy, Privacy-preserving ML, Natural Language Processing, Computer Vision, Pruning Algorithms