MC REU Proposal – Summer 2023

Adity Kendre

axk6052@psu.edu

Undergraduate Student

Computer Science, Penn State Harrisburg

Next-Generation Language Model for On-Device Natural Language Processing

Supervisor

Dr. Hien Nguyen

Computer Science & Math Program, Penn State Harrisburg

Secondary supervisor

Dr. Faisal Kabir

Computer Science Program, Penn State Harrisburg

Objective

The objective of this project is to develop a large language model that can run natively on personal devices, such as smartphones and home assistants, in order to reduce latency, increase user privacy, and improve natural language processing efficiency and accuracy.

Introduction

Large language models, such as OpenAI's GPT-3 and Google's BERT, have made significant breakthroughs in natural language processing (NLP) tasks, such as language translation and text generation. However, these models are typically hosted on cloud servers and require an internet connection to function properly. This poses several challenges, including increased latency and reduced privacy for users who rely on these models. Moreover, internet connectivity is not always reliable, which can make it difficult to access these models when needed. These challenges have sparked interest in developing large language models that can run natively on personal devices, such as smartphones and home assistants.

According to a recent study by OpenAI, reducing latency in natural language processing is crucial for improving user experience and efficiency. The study found that reducing latency by even a few milliseconds could significantly improve user satisfaction and engagement. Moreover, running language models on personal devices can help increase user privacy by eliminating the need for data to be transmitted to cloud servers.

In addition to improving user experience and privacy, running large language models natively on personal devices can also lead to faster and more efficient processing of natural language tasks. This can be achieved by leveraging on-device machine learning techniques, such as transfer learning and model compression, as well as specialized hardware, such as GPUs and TPUs.

Several companies, including Apple and Google, are already exploring the potential of on-device machine learning for natural language processing. Apple's Siri and Google Assistant both use on-device machine learning techniques to provide voice-activated assistance, while Google's BERT has been optimized for mobile devices to reduce latency and improve user experience.

The limitations of current cloud-based language models, such as internet connectivity and latency, have sparked interest in developing large language models that can run natively on personal devices. By leveraging on-device machine learning techniques and specialized hardware, these models have the potential to significantly improve user experience, efficiency, and privacy in natural language processing tasks.

Literature Review

Large language models have become increasingly powerful and capable of performing complex tasks in natural language processing, such as generating creative text, answering comprehension questions, and even predicting protein structures. However, their widespread deployment has raised concerns about their deficiencies, including the generation of false information, propagation of social stereotypes, and production of toxic language.

Despite the potential benefits, full research access to these models remains limited due to the resources required to train and run such large models. This restriction has hindered progress on efforts to improve their robustness and mitigate known issues, such as bias, toxicity, and the potential for generating misinformation.

In recent years, researchers have made significant efforts to develop smaller, more performant models that require less computing power and resources. For example, Alpaca, a language model fine-tuned from Meta's LLaMA 7B model, has demonstrated many behaviors similar to OpenAI's text-davinci-003 but is also surprisingly small and easy/cheap to reproduce. Similarly, Databricks' Dolly, a large language model trained on the Databricks Machine Learning Platform, has demonstrated that a two-year-old open source model (GPT-J) can exhibit high-quality instruction following behavior when subjected to fine tuning on a focused corpus of 50k records (Stanford Alpaca).

These developments are significant because they demonstrate that the ability to create powerful artificial intelligence technologies is more accessible than previously realized. Smaller models trained on more tokens are easier to retrain and fine-tune for specific potential product use cases. For instance, LLaMA, a state-of-the-art foundational large language model, has been released publicly to enable researchers to advance their work in this subfield of AI. It is available at several sizes (7B, 13B, 33B, and 65B parameters) and is designed to help researchers explore new use cases, validate others' work, and test new approaches.

In terms of hardware, the most common way to run large language models is through cloud-based services, such as Amazon Web Services, Google Cloud, or Microsoft Azure, which provide access to powerful computing resources. However, recent developments in hardware have made it possible to run large language models on specialized hardware, such as graphics processing units (GPUs) or tensor processing units (TPUs), which can provide significant performance gains. For example, OpenAI's GPT-3 model can run on TPUs, which provide a 10x performance boost compared to GPUs.

In summary, current literature suggests that the development of smaller, more performant models such as Alpaca and LLaMA can democratize access to large language models and enable

researchers to study these models. In terms of hardware, cloud-based services remain the most common way to run large language models, but specialized hardware such as TPUs can provide significant performance gains.

Project Goals

1. Develop a large language model that can run natively on personal devices, such as smartphones and home assistants, to reduce latency and increase user privacy.
2. Utilize on-device machine learning techniques and specialized hardware for inference of large language model
3. Conduct extensive testing of the developed model across a range of use cases and scenarios to ensure its accuracy, reliability, and user satisfaction.
4. Compare the performance and user experience of the developed model with existing cloud-based language models and other on-device language models to demonstrate its advantages.
5. Provide recommendations for further research and development of on-device language models, including strategies for addressing challenges related to on-device machine learning and user privacy.

My Research Background

I am a third-year computer science student at Penn State Harrisburg and have completed classes such as Artificial Intelligence (CMPSC 441) which aided in creating a strong foundation for machine learning techniques. Furthermore, I am currently taking Applied Data Science (CMPSC 445) which has allowed me to further hone in on my skills.

I have had extensive experience in deep learning, working as an intern at Global Digital Innovation and York Exponential, where I designed and built internal cloud-based pipelines for training computer vision models using PyTorch Lightning and TensorFlow.

In terms of personal projects, I have created a heart sound abnormality detection app using a novel deep learning model with PyTorch Lightning. I also built a machine learning playground for visualizing different algorithms such as SVMs, K-NNs, Naive Bayes, and MLPs using PyTorch and scikit-learn.

In terms of research, I have led the development of models for EEG connectome analysis using architectures like ResNet and DenseNet, which yielded an average accuracy increase of 20% over previous methods.

Personal Statement

As a computer science student with a strong interest in deep learning and machine learning, I have developed a solid foundation with my coursework, job experience, personal projects, and research background.

Methodology

For this project, I propose a methodology that includes research design, data collection, data analysis, model development, and testing and evaluation. The research design will involve conducting a thorough literature review of current models to plan initial research.

Data collection and management will be a critical component of the methodology. I will collect and preprocess the data for use in training the model. PyTorch will be the primary tool used for model development, and I will select and implement a suitable model architecture based on the research findings.

The model development stage will involve training the model and fine-tuning it for optimal performance. I will also explore integrating on-device machine learning techniques to improve the model's efficiency.

The interface design stage will involve integrating the model with an interface that allows for easy user interaction. I will test the model on a range of use cases and scenarios to ensure it performs as intended. I will also analyze its performance and user experience during this stage.

Finally, I will wrap up the project by finalizing documentation and creating a report that summarizes the methodology used and the results obtained. Throughout the project, I will utilize various technologies and tools, such as PyTorch, for model development and data analysis, and personal hardware like smartphones and home assistants may be used for testing and evaluation.

Tentative Schedule

Week	Task
1	**Literature Review:** plan initial research and conduct study of current models
2	**Data Management:** collecting and preprocessing the data
3	**Model development:** selecting and implementing model architecture
4	**Model development:** training model
5	**Model development:** fine-tuning model
6	**Model development:** integrating on-device machine learning techniques
7	**Interface design**: integrating interface with model
8	**Testing and evaluation**: testing model on a range of use cases and scenarios
9	**Testing and evaluation**: analyzing model performance and user experience
10	**Wrap-up:** Finalizing documentation and creating report

Review

We aim to develop a methodology for training a large language model that can be integrated into personal devices. The model will be trained on a diverse range of data sources to enable it to understand natural language and generate human-like responses. The goal is to create a model that can operate efficiently on personal devices, such as smartphones and home assistants, to improve user experience and accessibility. Our methodology will be designed to ensure that the model is capable of learning from large datasets, and can be fine-tuned to improve performance over time. Furthermore, this methodology can be extended to other language models and personal devices.

References

Deep LearningNatural Language Processing Natural Language “Open Sourcing Bert: State-of-the-Art Pre-Training for Natural Language Processing.” – *Google AI Blog*, https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

“Hello Dolly: Democratizing the Magic of Chatgpt with Open Models.” *Databricks*, https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html.

“Introducing Llama: A Foundational, 65-Billion-Parameter Language Model.” *Introducing LLaMA: A Foundational, 65-Billion-Parameter Language Model*, https://ai.facebook.com/blog/large-language-model-llama-meta-ai/.

“Overview.” *Stanford CRFM*, https://crfm.stanford.edu/2023/03/13/alpaca.html.