Introduction
The objective of this project is to generate bash commands from natural language using a deep neural network. Novitiate and even experienced engineers can often find the terminal interface perplexing and are quickly overwhelmed by the syntax of bash commands. This project aims to ease that burden on new and experienced users alike.

Dataset
The dataset we used is from the "The NL2CMD Competition" consisting of 10,000 parallel translations of English (invocation) and bash (cmd). We further parsed the bash command into the corresponding template form for easier generalization during training.

Evaluation Metric
We used the cross-entropy loss to train the model, but to measure the model performance, we used the metric defined by the competition.

\[
S(p) = \sum_{x,y \in D} \frac{1}{m} \left( \log P(y|x) + \frac{1}{N} \left( X - \log P(y|x) \right) \right)
\]

- \( U(x) \): a sequence of bash utilities in a command \( x \)
- \( \epsilon \): predicted bash command; \( C \): ground truth bash command
- \( X = 2 \times |P(U(C)) \cap P(U(\epsilon))| - |P(U(C)) \cup P(U(\epsilon))| \)
- \( F(x) \): the maximum length between \( U(x) \) and \( |C| \)
- \( X \): the maximum size between \( F(x) \) and \( |C| \)

Methods
We experimented with several models, including GPT2, BART, and T5, as well as different tokenization schemes to improve model performance on the NL2CMD dataset.

- **Casual Language Modeling: GPT2**

  ![GPT2 Diagram]

  **INPUT:** <bos_token> <source_token> <invocation> <target_token> <template cmd> <eos_token>

  **OUTPUT:** <predicted cmd> <eos_token>

- **Seq2Seq Language Modeling: BART/T5**

  ![BART/T5 Diagram]

  **INPUT:** <invocation>

  **TRAINING**

  **LABEL:** <template cmd>

  **TESTING**

  **OUTPUT:** <predicted cmd>

  Tokenization mechanics play an enormous role in model performance, especially when the model you are training was originally trained for a different task, like GPT2.

Results & Analysis
We found that T5 performed best for this prediction task. It even outperformed the GPT3 baseline provided by the competition. It also continued to improve with training time. On the other hand, our GPT2 and BART models plateaued rather quickly, and never approached the GPT3 baseline using the NL2CMD scoring metric.

While cross-entropy loss appears to be the worst for T5, T5 actually performed the best when scoring according to the competition metric. While cross-entropy loss is effective for training, it clearly doesn't parallel performance for our task.