Week 4: Generating Shakespear

Nate Bachmeier

TIM-8150: Artificial Intelligence

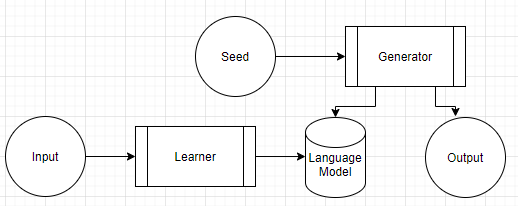
October 25, 2020

Northcentral University

# Generating Shakespeare

Natural Language Processing (NLP) is a broad topic that covers both the parsing and creation of human language. When Artificially Intelligent (A.I.) systems generate content, such as movie dialog, it virally spreads across social media and our imagination (Thomas, 2019). Many practitioners begin their journey toward these results by first generating *Shakespeare* *Plays*. While solutions range in complexity, they all inherently use the same design architecture (see Figure 1). First, training example data feeds into a learning algorithm to produce a statistical model of the grammar. Second, a generator process takes seed information and emits words (tokens) that are likely to follow.

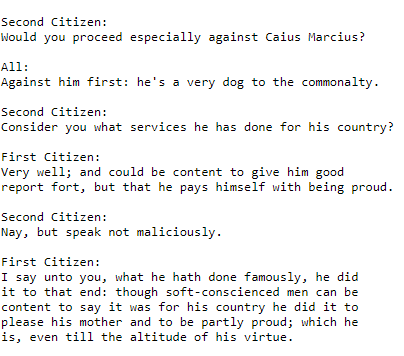
Figure 1: Generator



## Dataset Details

The *Tiny Shakespeare Dataset* is approximately one megabyte in size and contains 40,000 lines from various plays (TensorFlow, 2020). Each entry begins with the actor’s name, followed by their passage (see Figure 2).

Figure 2: Excerpt of Input



# Learning Process

This work reviews two learning processes called Markov Decision Processes (MDP) and Generative Pretrained Transformer (GPT). MDP models the probabilities that two dependent actions will take place by running experiments and measuring the outcomes. GPT uses a Neural Network (N.N.) to predict the next token using weighted attention vectors.

## Markov Decision Process

The first solution to the Shakespeare problem starts by building a map keyed on n-grams (e.g., ‘he says’) with a list of proceeding n-grams (see Figures 3&4). The list items are not distinct, enabling random selection from the list to maintain the statistical weights (see Figure 5). This naïve implementation produces decent results (see Figure 6) and is easy to comprehend.

|  |  |
| --- | --- |
| Figure 3: MDP Design | Figure 4: MDP Implementation |
| Figure 5: Generator | Figure 6: Results |

## Generative Pretrained Transformer

Open A.I.’s GPT-2 uses a more complex data structure to model the relationship between words while accounting for various aspects like verb tense and parts of speech (Radford et al. 2018). They train the learning model’s billions of parameters by crawling Reddit.com and consuming any referenced articles. Researchers incorporate GPT through *transfer learning* by adding a *fine-tuned* layer to the processing pipeline (see Figure 7). While this approach reduces the total training time and improves the quality of results, it can still take hours of CPU (Central Processing Unit) time. For instance, an Amazon EC2 ml.c5.9xlarge instance with 32 cores and 72 GiB required nearly three hours to build the *GPT* *Shakespeare Model*.

Figure 7: Transfer Learning

