Week 4: Generating Shakespeare

Nate Bachmeier

TIM-8150: Artificial Intelligence

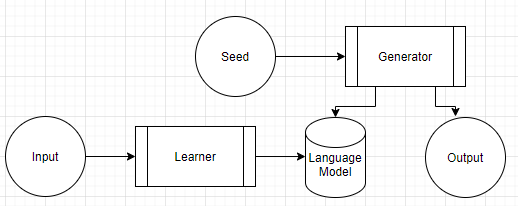
October 25, 2020

Northcentral University

# Generating Shakespeare

Natural Language Processing (NLP) is a broad topic that covers both the parsing and creation of human language (Sintoris & Vergidis, 2017). 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 impressive results by first generating *Shakespeare* *Plays*. While solutions range in complexity, they all inherently from 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 System Design



## 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). This small data appears in many tutorials and experiments due to being both open-source and fun.

|  |  |
| --- | --- |
| Figure 2: Excerpt of Input |  |

# Learning and Generation Processes

This work reviews two different 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 (Kahn Academy, 2014). GPT uses a Neural Network (N.N.) to predict the next token using weighted attention vectors (Hugging Face, 2020).

## Markov Decision Process

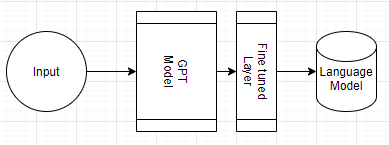
The first solution to the Shakespeare problem starts by building a map keyed on n-grams (e.g., ‘he says’) with a list of proceeding n-grams (see Figures 3&4). The list items are not distinct, enabling random selection from the list to maintain the statistical weights (see Figure 5). This naïve implementation produces decent results (see Figure 6) and represents a good mental model for how smarter systems could work. For example, the central data structure is string tuples that contain literal text. Instead, an encoding mechanism could incorporate pre- and post-processing to stem and lemmatize tokens (Edureka, 2018). This missing functionality leads to overfitting as the n-gram size increases due to insufficient options for the next term. It would be possible to introduce these capabilities to this MDP process but is outside the scope of this work.

|  |  |
| --- | --- |
| 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 compile the *GPT* *Shakespeare Model*.

Figure 7: Transfer Learning



Despite having a very long training time, the engineering effort to start using GPT is relatively small. The Python package gpt-2-simple (Woolf, 2019) encapsulates much of the complexity into the utility methods ‘finetune’ (see Figure 8) and ‘generate’ (see Figure 9). GPT also maintains whitespace and formatting, resulting in more realistic outputs.

|  |  |
| --- | --- |
| Figure 8: Fine Tuning | Figure 9: Generating |

# Challenges and Limitations

Researchers have several open problems across NLP generation before it can fully replace humans.

## Semantic versus Logical

One of the critical limitations of language models is that they are semantically correct, not logically sound (see Figure 10 & 11). For instance, there are connections between paragraphs (e.g., ‘so when’ and ‘at three’), but a more profound conversation is missing. This behavior is partially due to machines excel at enumerating token sequences. However, human-dialogs has multiple layers of context and behave closer to a LIFO (Last-In-First-Out) stack than an iterator (Fridman, 2020).

|  |  |
| --- | --- |
| Figure 10: Logical versus Semantic | Figure 11: Logical versus Semantic |

## Processing Power

Ubiquitous access to cloud computing resources enables more businesses to experiment with NLP, though it is still far from free. For example, the Amazon Web Service (AWS) costs to support this paper are nearly $150. While this is unlikely to be a barrier of entry for any large enterprise, it can meet resistance among smaller organizations after multiplying the expense by team size and across longer durations. Without sufficient processing capabilities, it is challenging to produce long relevant documents, such as full-length movie scripts and novels.

# Conclusions

NLP Generation follows a two-step process that first produces a language model, then uses it from a second Generator process. Using a simple MDP algorithm helps to demonstrate this concept with n-gram frequency maps. While the results look semantically correct, there are several limitations, such as overfitting and unnatural mixing of verb tenses. GPT overcomes some of these issues by bringing enormous data volumes to train billions of neural network parameters. Researchers can then use transfer learning to extend that model without needing to rebuild the base weights.

An effective strategy to master these concepts is through hands-on tutorials and experiments, like the *Shakespeare Models.* While modern-day tooling abstracts away much of the complexity, there are still several challenges to producing quality results. For instance, maintaining the depth and breadth of context across lengthy conversations is an open problem. These limitations prevent machine learning systems from writing the next best-selling novel, but they can author short scenes.

# References

Edureka. (2018, October 16). *Natural language processing in 10 minutes*. Retrieved from YouTube: https://www.youtube.com/watch?v=5ctbvkAMQO4

Fridman, L. (2020, January). *Deep Learning State of the Art*. (Massachusetts Institute of Technology (MIT)) Retrieved from YouTube: https://youtu.be/0VH1Lim8gL8

Hugging Face. (2020). *OpenAI GPT2*. Retrieved from Hugging Face: https://huggingface.co/transformers/model\_doc/gpt2.html

Kahn Academy. (2014). *Origin of Markov Chain*. Retrieved from Kahn Academy: https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/markov\_chains

Radford, A. W. (2018). *Language Models are Unsupervised, Multitask Learners*. Retrieved from OpenAI: https://cloudfront.net/better-language-models/language-models.pdf

Sintoris, K., & Vergidis, K. (2017). Extracting business process models using NLP techniques. *IEEE 19th Conference on Business Informatics* (pp. 135-139). Thessaloniki, Greece: Institute of Electrical and Electronics Engineers. doi:https://doi-org.proxy1.ncu.edu/10.1080/014492997119879

TensorFlow. (2020, October 9). *Tiny Shakespeare*. Retrieved from Tensorflow: https://www.tensorflow.org/datasets/catalog/tiny\_shakespeare

Thomas, R. (2019). The new era of NLP. *Scientific Computing with Python.* Austin, Texas: SciPy. Retrieved from https://youtu.be/KChtdexd5Jo

Woolf, M. (2019, September 4). *How To Make Custom AI-Generated Text With GPT-2*. Retrieved from Max Woolf's Blog: https://minimaxir.com/2019/09/howto-gpt2/