HackPressIO, Content Generation

Executive Summary

## Presented by

Jed Chang

Bridger Norman

Celeste Popoca

Luke Russell

Kelin Tang

Ella Yang

1. Summary

The goal of this research project was to select a recursive neural network architecture that ensures high-quality results while generating excitement among stakeholders and investors. We examined the use of LSTM and GRU layers in text generation networks to improve the performance of existing code from the company. We explored strategies such as increasing training iterations, adding more layers, and experimenting with different cell types. By considering the strengths and trade-offs of LSTM and GRU networks, we can optimize text generation models for enhanced performance and market appeal. The conclusion of our research is that a model with GRU and LSTM layers had the best text output.

1. Methodology

Process:

1. Gather Data on various Authors
2. Combine data to create the larger set for the authors
3. Brainstorm metrics to measure model accuracy
4. Create a vanilla model
5. Compared GRU layers vs LSTM layers
6. Testing the model we create
7. Determined how well the model outputs text in an authors style
8. Use metrics to determine our final model

The data that we chose to gather came from the three authors, Jane Austin, Mary Shelly, and Mark Twain. *Below are the links to the data that we obtained from Gutenberg and Github.*

[Mary Shelly](https://www.gutenberg.org/ebooks/84)

[Mark Twain](https://raw.githubusercontent.com/exoden1/myrepo/master/marktwain_Mississippi.txt)

*Note: Jane Austin was provided by HackPressIO*

The metrics we used to judge our text outputs were spelling and grammar accuracy. For example, we would take the total word count then subtract all misspelled words, then divide the number of correctly spelled words divided by all words to get our accuracy numbers, which allows us to compare by ratio because each output has different word lengths. To make sure these results were consistent and not biased by our human error, We pasted the output into Microsoft word and counted the spelling errors by the red lines and the grammatical errors by the blue lines. Though this approach is not perfect it allows us to have consistency in our error because it is interpreted by the same program making our metrics comparable.

We used the basic GRU model provided by the previous data science team to create a base to compare how well we improved our text outputs to generate new books.

The training process utilized early stopping to ensure that we did not overfit allowing us to find the best version of our model.

After comparing many model variations as shown below in Table 1, we chose a model that was a mix of GRU and LSTM with layers as shown in Figure 1

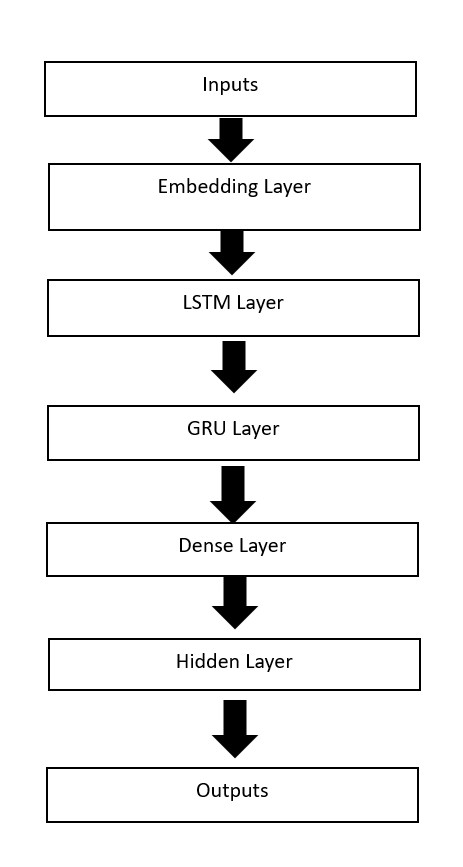


Figure 1

The model can be deployed for text generation for any of the three chosen authors. Allowing us to instantly recreate their style with unique text. Overall, the model shows great potential for enhancing operational efficiency related to our goal of content creation.

1. Results, Action Items, and Limitations

We chose to test several different ways to train Recurrent Neural Networks (CNN): GRU, LSTM, GRU Vanilla, LSTM Vanilla. The LSTM&GRU model had Spelling and grammar accuracy both are over 96%, the LSTM vanilla model had spelling and grammar accuracy scores of 96 and 97, and the GRU vanilla model had spelling and grammar accuracy scores of 94 and 96, we decide the GRU & LSTM model had a high spelling and LSTM accuracy score, which means it has roughly a 97% correlation with the variation in all the road signs [see Table 1]. This parameter comes from testing the neural network on different models.

|  | Vanilla GRU | Vanilla LSTM | LSTM\*2 | GRU & LSTM -Ella | GRU Dense\*2 |
| --- | --- | --- | --- | --- | --- |
| Spelling Accuracy | 94% | 96% | 43% | 98% | 96% |
| Grammar Accuracy | 96% | 97% | 98%\* | 96% | 96% |
| Epochs | 20 | 52 | 20 | 70 | 20 |

(Table 1)

*\*grammar accuracy for LSTM is not valid because it was gibberish*

Based on these results, the RNN model that worked the best was the GRU&LSTM architecture. This demonstrates a promising future for text generating in the company. The model can provide reasonably accurate estimates for determining what each road sign means. It will assist the company in providing a good text generating model with their style and voice.

1. Action Items

Based on the results, HackPressIO should continue to get more authors’ book data. It would be especially helpful if the model was trained on a large dataset. Convert the text into a numerical representation that the model can understand. Tokenization involves assigning a unique numerical value (index) to each word or character in the text.

1. Limitations

One of the big limitations we had in this model is the lack of open research data. It may be difficult or impossible to access a sizable body of writing by a single author, particularly if they are modern or protected by copyright. Due to this restriction, assembling a large dataset for Recurrent Neural Networks training is challenging.

Another limitation is GPU memory, we use Google Colab the GPU available may be limited. This can restrict the scale or speed at which you can train or generate text using RNN models. However, using hardware with more GPU memory will help.

1. Q&A

**What would give us the most promise for both a quality model, but also something that could get people excited?**

Combine text generation with other modalities like images, audio, or video to create more immersive and engaging experiences. For instance, generate text captions for images or produce dialogue based on audio inputs. Multimodal models can generate more diverse and captivating content.

**I'm wondering what your views are on using a teacher forcing strategy compared to a curriculum learning strategy?**

We thought the curriculum learning strategy would be the ideal way to train the model, especially when it came to grammar. However, we wanted the output to be in the style of Jane Austin and sentences only increasing in complexity wouldn’t reflect her writing style well. So, we used the teacher forcing strategy.

**Our previous team used logits in the output layer and then used Sparse Categorical Cross Entropy as the loss function. Are you planning to use that approach as well?**

For our model, we decided to keep both the logits and Sparse Categorical Cross Entropy. We chose to adapt it into our model architecture.

**Which of the following would you recommend?**

1. **We should not use any additional text, because it will change the style of the generated text.**
2. **We should only use works that are out of copyright and now in the public domain, such as Jane Austen, or other older works.**
3. **We could use all of Wikipedia, or other creative commons works.**
4. **We could use all of the Internet, since we will not be copying any of the text verbatim.**
5. Python Notebooks

Below is Github Gist link to the notebook we used during this case study:

<https://colab.research.google.com/drive/1I-sLAFbBtpGPs6XdnGjTpG1Pxq6b-Njq>