Group 4: Madhav Rajesh, Evan Silvey, Hanna Koo, Parsa Bazargani, Yahli Hazan ECS 170
Professor Simmons June 12, 2024

Transcript Generation for the TV Show “Rick & Morty”

I. Introduction

Over the time that we have spent studying AI, we have learned about many different kinds of AI paradigms and algorithms. Currently, we are seeing an evolutionary rise in the areas of natural language processing, so we chose to spend this quarter taking part in a text generation-based project. Our project is a deep learning project based around natural language processing with the aim of producing unique transcripts that are based off of existing television scripts.

Our mission is to develop a custom script generator for the TV show Rick & Morty. By leveraging Recurrent Neural Networks (RNNs) and their variant with Long Short-Term Memory (LSTM) units, we aim to understand and replicate the show's distinctive comedic language and structure while ensuring the generated content is unique. Rick & Morty stands out as an excellent candidate for this task due to its consistent comedic style and stable use of grammar and word choice across episodes, despite the large variance in individual storylines. This consistency makes it a robust dataset for training our language models. Our goal is to produce scripts that not only mirror the structure, terminology, and language of Rick & Morty episodes but also introduce fresh and original storylines. By balancing similarity with originality, we aim to generate content that feels authentic to the show while contributing new and innovative narratives. Our motivation for choosing this project stems from the fun and engaging nature of working with TV show scripts, their applicability to real-life entertainment industries, and the universal appeal of television.

II. Background

In the realm of artificial intelligence and natural language processing, generating synthetic text that mimics the style and structure of existing works has been a significant area of research. For our project, we have chosen Rick & Morty as our focus due to its rich and varied content combined with a consistent comedic style. This consistency across episodes provides an ideal dataset for training our models.

We utilize several key metrics to evaluate our models' performance, which are loss, accuracy, and perplexity. Loss measures the difference between the original and generated transcripts, with lower values indicating better performance. We chose to use this metric because it can be an indicator of how far off or how close our generated transcript is to the existing scripts. If the loss is too high or too low, it can respectively mean that the model is doing a poor job generating these sequences or is potentially overfitted. Accuracy, defined as the inverse of loss, helps assess how closely the generated text matches the original scripts. Similarly to loss, accuracy can give an indicator to the model’s performance in generating transcripts so we decided to use that metric for evaluation. Perplexity is a common metric that is used in language processing, which evaluates the model's ability to generate meaningful responses by measuring predictive uncertainty per word or character. Lower values indicating more coherent outputs and in this project, we wanted to evaluate our model’s effectiveness in being able to generate these meaningful outputs. These three metrics allow us to evaluate the success of the model in generating similar, meaningful, but unique scripts.

To achieve our goal, we implemented both a standard RNN and an RNN with LSTM units. The LSTM-enhanced model addresses common issues such as vanishing and exploding gradients, which can impede the learning process in deep learning models dealing with sequential data. We anticipated that the LSTM model would outperform the standard RNN in terms of accuracy and overall performance. Our initial target is to achieve accuracy rate in the 60% - 80% range, ensuring that the generated transcripts are not mere replicas of the original episodes but still maintain a high level of similarity in terms of word usage and grammatical structure. Through this process, we aim to develop a robust script generator that captures the essence of Rick & Morty while contributing new and engaging content to the show's universe.

III. Methodology

Training Data

Since this task involves training data based on reading Rick & Morty transcripts, our aim was to figure out which episodes to use as training. Since there is no existing dataset, we would have to create the dataset where we place our transcripts in a uniform text file. In addition, we would need to process the data on our own. For shows with multiple seasons, it is also important to understand that the themes and the writing styles may vary from season to season. Some characteristics or recurring storylines may span over the course of the whole show, but there are many differences as well. A storyline that one may see in a debut season may be very different from what the subject matter could be in a future season.

Rick & Morty has seven seasons total with an average of about ten to eleven episodes per season, with each transcript spanning thousands of characters. Since this show is intended to be standalone where the comedic nature and the writing is consistent throughout the whole TV series, the need of worrying about the changes from season to season may be a slightly less concern. However, due to the size of the transcripts, we would need to combine them into a dataset and much of the work would go into preprocessing the writing, so that would invoke a much larger runtime. Therefore, for this project, we decided to direct our training data from using most of the episodes from the first season to achieve that consistency. This would be our approach if we used any other television show, so we are following that trend with Rick and Morty.

RNN Method

As a starting point, we chose to use a method that is commonly used for language processing: the Recurrent Neural Network (RNN). This algorithm is one that is used primarily in sequential data, which includes words and sentences. Since we are using TV scripts as our source of data and aiming to create a similar transcript as a result, the word sequences and format are crucial. In addition to the dialogue, other aspects of the transcript have to also make some sense, including the stage directions. The RNN allows for us to iterate over the same node, which is not the case in a feed forward neural network. As a result, the model has a higher likelihood of producing clear and coherent phrases and word patterns that follow trends from the data. Below is a diagram of the general RNN architecture, showing the iterative ability and how it works:

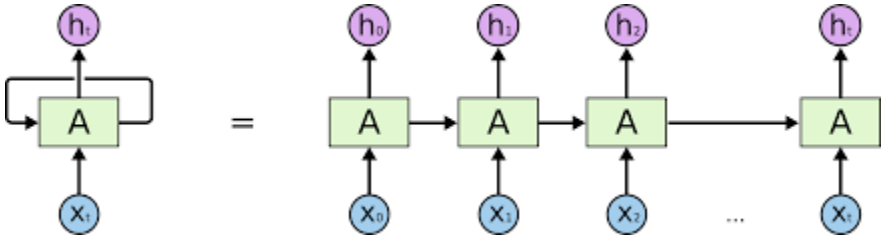

Figure 1: Diagram showing the standard neural network.

With an RNN, we use embeddings, which is common in many deep learning processes. The goal is to be able to convert these variables into vectors. This is what strengthens the model's ability to better understand the sequence and its meaning.

A downside with standard RNNs is the potential for our gradients to either decrease over time or increase exponentially, known as the vanishing or exploding gradient problem. With this deep learning model, the value of the weights can determine whether the learning rate will increase or decrease over time due to backpropagation. As a result, the weights will update too slowly for vanishing gradients and too much for exploding gradients. This will cause the model to make less progress in its learning.

RNN with LSTM

To bring forth more stability in the learning rate of the model, we would need variations that improve on the standard RNN. For our project, we also implemented the RNN with Long Short Term Memory (LSTM) and this is our primary model for the project that we aim to get our best results in. This approach fixes the gradient problems that a traditional RNN would have, as there would likely be a higher degree of convergence in the

learning rate. The gating mechanisms in an LSTM allow for the ability to choose what memory to forget and thus is likely to improve in our performance metrics, as seen in this diagram on the next page below for the general architecture.

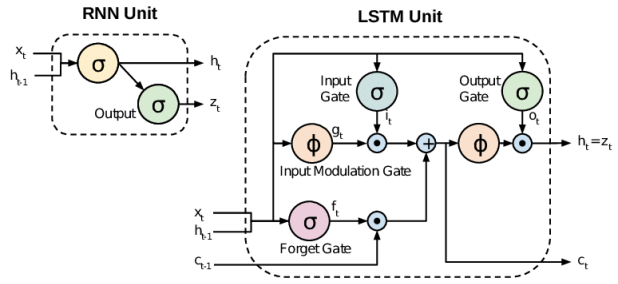

Figure 2: General architecture for an LSTM unit

The LSTM unit has more enhancements, including a forget gate that helps with selectively forgetting older memory. Here, the combination of both the gating and larger window size is what will lead to the model to improve in its performance. A key thing to take away is that there will be far more parameters compared to a standard RNN; however that will be just what we need in order to reach a significant level of improvement.

Data Preprocessing and its Tradeoff:

For this type of task, there are different ways to preprocess the data. For our job, we started by preprocessing based on words. A key benefit with this approach is that using words is likely to result in transcripts that use meaningful words that exist in the current vocabulary. Therefore, the transcript will better align with existing Rick & Morty episodes. However, this approach struggles in finding any unseen words and phrases, so the output can result in large amounts of regurgitation. Our metrics will imply high performance, but this can mean there is a degree of overfitting due to the lack of uniqueness in the script. This would fail to achieve the goal of our transcript appearing individually.

Another method to do the data preprocessing is to process based on individual characters. As a consequence, there would be less semantic meaning in the words that come out of the transcript. As a result, the performance is likely to take a hit. On the positive side, we may get more originality with the word choice, as the output could use unseen words while training. Here, the question that we have to ask ourselves is, “To what degree does the language take a step back compared to word processing, and is that drop off worth it in order to generate more originality in the script?”

IV. Results

When running grid search on our four different scenarios, we were able to examine the total number of trainable parameters used for our models. The table is shown below for the parameter count:

Model Type	Data Preprocessing Method	Number of Parameters
RNN	Word Preprocessing	1380089
RNN with LSTM	Word Preprocessing	3675641
RNN	Character Preprocessing	4399202
RNN with LSTM	Character Preprocessing	16994402

Table 1: Trainable parameters based on the four model combinations (RNN or RNN with LSTM with word or character preprocessing.

As expected, the RNN with LSTM has more than double the parameters compared to the RNN counterparts for word and character preprocessing. This is expected because with the LSTM, we have extra gating that helps with the transfer of information that is not there in a standard RNN. Thus, the number of trainable parameters

increases. We also can see that the character preprocessing RNN and RNN with LSTM has a much larger number of parameters compared to the word preprocessing.

Word Preprocessing:

1. RNN

The traditional RNN approach was our very first base model to start the training process, before going into the other approaches to improve results. Out of our four different models, the traditional RNN for word preprocessing performed the lowest. The graph below shows our results in the metrics:

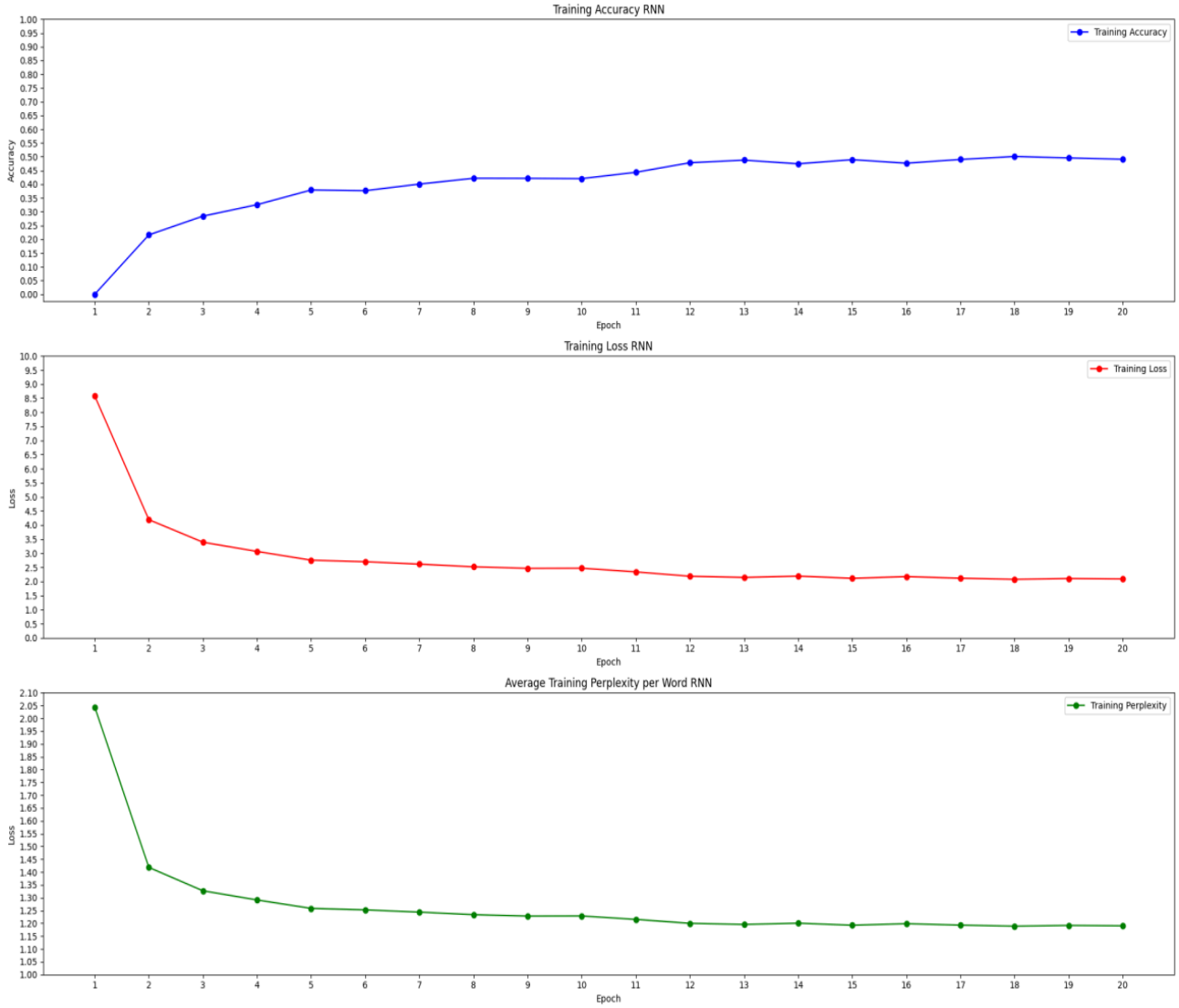

Figure 3: Result for Training Accuracy (top), Training Loss (middle) and Training Perplexity per word (bottom) over the course of 20 epochs for RNN using word preprocessing

As seen with the graph for word processing with an RNN without any LSTM, our model had a training accuracy of around 50%, a training loss of 2.05 after 20 epochs, and a perplexity score of around 1.2. Additionally, we can notice that there is some very clear flattening out of these graphs which could signify some overfitting which is common with RNNs. Adding our LSTM implementation is a great first step at getting better results while mitigating some of this overfitting. Aside from just our evaluation metrics, we can also look at our actual results in the form of the transcripts generated. On the next page is an excerpt from our RNN generated transcript with input prompt, 'rick: mmm panda express':

```
rick: mmm panda express. and you're. enters the jump him him the, with. it him and the,! the the where,
a start. both with into the? the that to. knocked? and him.. you're, with with with.. some., and him
him grab., with. you're? where that.. you're and and with? the?

jerry:, annie it with, the a with all? where is him a summer back.?, summer you, take and that snuffles
and?. a. the. some?, that? him.

morty: that looks all??.. go. with. ya. and...!, the the with, you're the.?,. the. and take jump the a
a some. here where, him,. through?.. with. that a a the; the their with

morty:,, a the?...? him morty on dr start him and the it a,? get a,? with into a. would, their a getting
with. him is a! a.?.. summer, with s a and is, him a the look takes tackle a?? and the at with and through
into, him? it.. a ( ) with him head an.

…
```

Figure 4: Portion of output transcript from the RNN using word processing.

It is immediately noticed that this transcript is relatively gibberish and has very flawed semantic and grammatical structure. The punctuation is misplaced and all over the place in this transcript. Additionally there is little to no conversation between the characters and therefore, would not pass as clear and coherent English. This led us to testing with the LSTM implementation.

2. RNN with LSTM

As stated earlier in this paper, LSTM helps bring forth stability in the learning rate, as the graduates are not too large or too small. In addition, we are able to train over a longer sequence interval. This means that the model may better understand the context behind the sequence of words due to a larger window. As a result, we expected that our performance metric results would improve and that our transcript would follow 1) language conventions and 2) meaningful dialogue. Based on the plot and transcript excerpt seen on the next two pages, this happens to be the case.

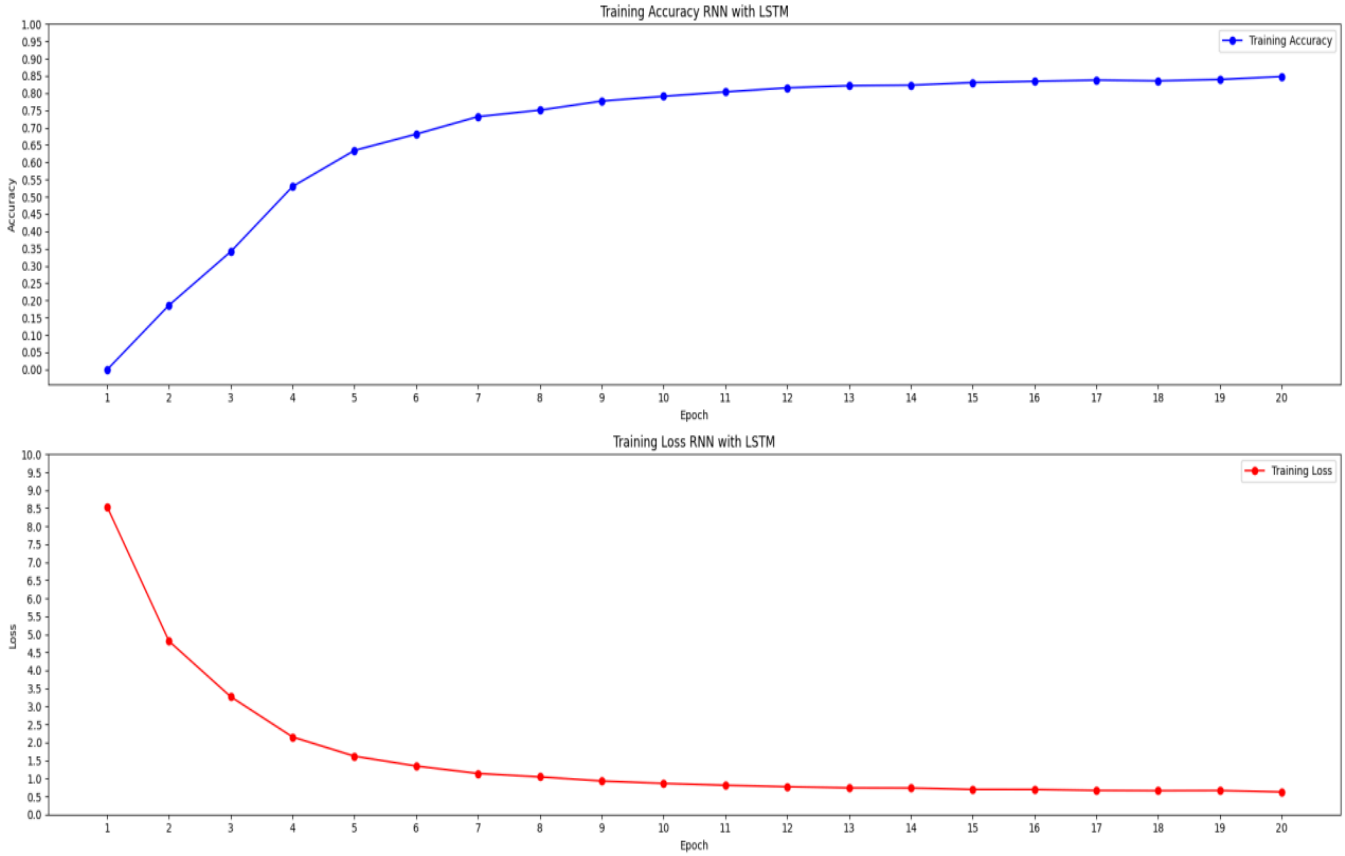

Figure 5a: Results for Training Accuracy (top) and Training Loss (middle) 20 epochs for the RNN with LSTM using word preprocessing.

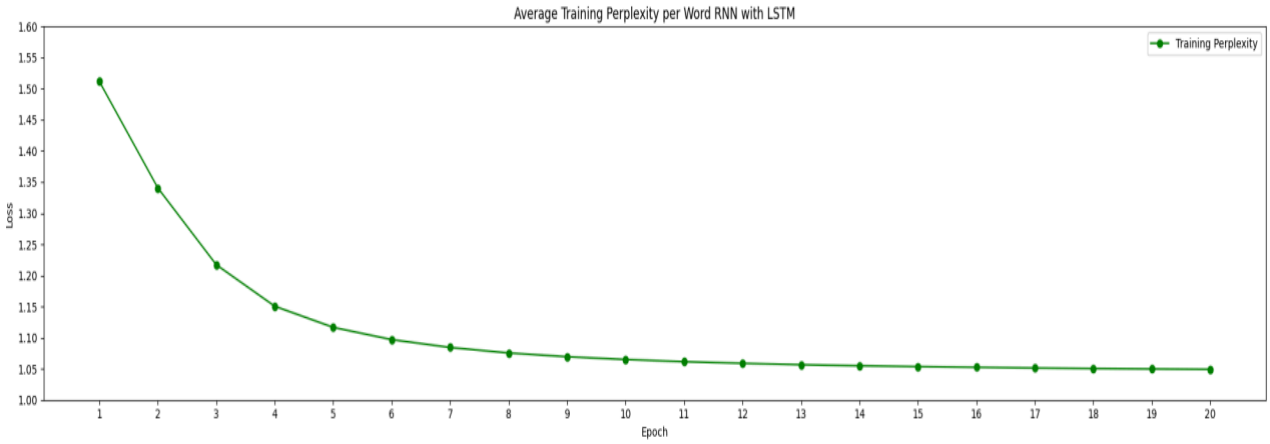

Figure 5b: Results for Training Perplexity per word (bottom) over 20 epochs for the RNN with LSTM using word preprocessing.

After using the RNN with LSTM for word preprocessing, there is immediate increase in performance, as shown in figures 5a and 5b. Training loss is about 0.68, Training accuracy is 83%, and Perplexity per word is 1.05. Additionally the graphs show that each epoch is consistently getting better, and towards epoch 10 our model begins to reach 80% and surpass it. As for the transcript generated with the RNN with LSTM on word generation, the transcript is a lot cleaner as well, this is an excerpt using the prompt, 'rick: kill summer, morty!':

```
rick: kill summer, morty! what are you doing, morty?! there's no time!

morty: jeez, rick. oh my god no my god! it's all over the place, morty.

morty: hey rick, that's pretty cool! it's just like garfield, only instead,
it's more meaningful than that.

mailman: my man!

jerry: there's not to worry about it, morty. th-these seeds aren't gonna get
through customs unless they're in someone's rectum, morty

morty: uuuh.
```

Figure 6: Output of part of our transcript for the RNN with LSTM using word preprocessing

With this transcript, our model has very well structured English, a flow of conversation, and a very ‘Rick and Morty’ feel to it which we found to be best evaluated by a human after the fact (us in this case!). Although this response is not very class appropriate, the differences in the RNN that implements the Long Short-Term Memory is astounding in terms of legibility, semantic structure, and quality of content.

With that said, our model failed to get a good accuracy any time there would be a completely unknown word, which would happen occasionally in such a sci-fi heavy show like Rick and Morty. This can lead to concerns of the model being overfitted because of its inability to understand words that are not part of the vocabulary of the existing transcripts. Because of this, we also implemented data preprocessing based on character instead of word. As a result, unknown words wouldn’t break the model. This can be seen in the plots of the next section.

<u>Character Preprocessing:</u>

1. RNN

In order to maintain consistency with our approach, we preprocessed the data based on character with both the traditional RNN plus the RNN with LSTM. On the next graph, we see that the accuracy, perplexity, and

loss improve in comparison to the word processing. However, in terms of the transcript, it was very unreadable to the point that we did not feel the need to show it in this paper. The plot of our metrics are on the next page:

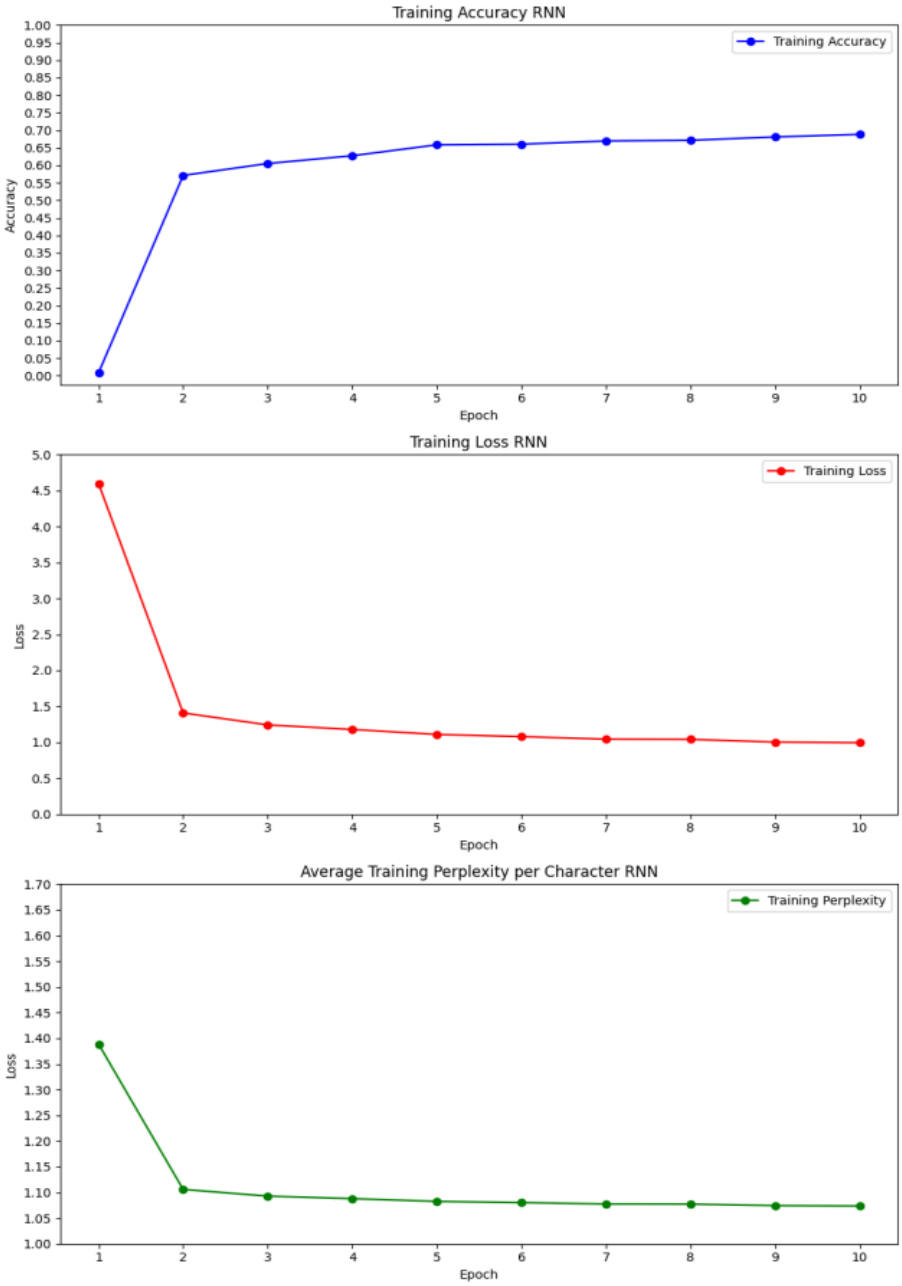

Figure 7: Results for Training Accuracy (top), Training Loss (middle), and Training Perplexity per character (bottom) over 10 epochs for the RNN.

Figure 7 above shows the results of the model when preprocessing was done by character and without using LSTM, and we see that the accuracy, loss, and perplexity are better than the word preprocessing without the LSTM. The semantic structure of these transcripts were fairly poor, and there was noticeably more use in stage directions and characters such as parentheses and brackets, rather than just dialogue which could be attributed to taking in each character instead of full words. This could be due to the use of line breaks, asterisks signifying stage directions, parenthesis for actions, etc. To summarize, this transcript did not follow the conventions of back and forth conversation as the formatting was incorrect.

2. RNN with LSTM

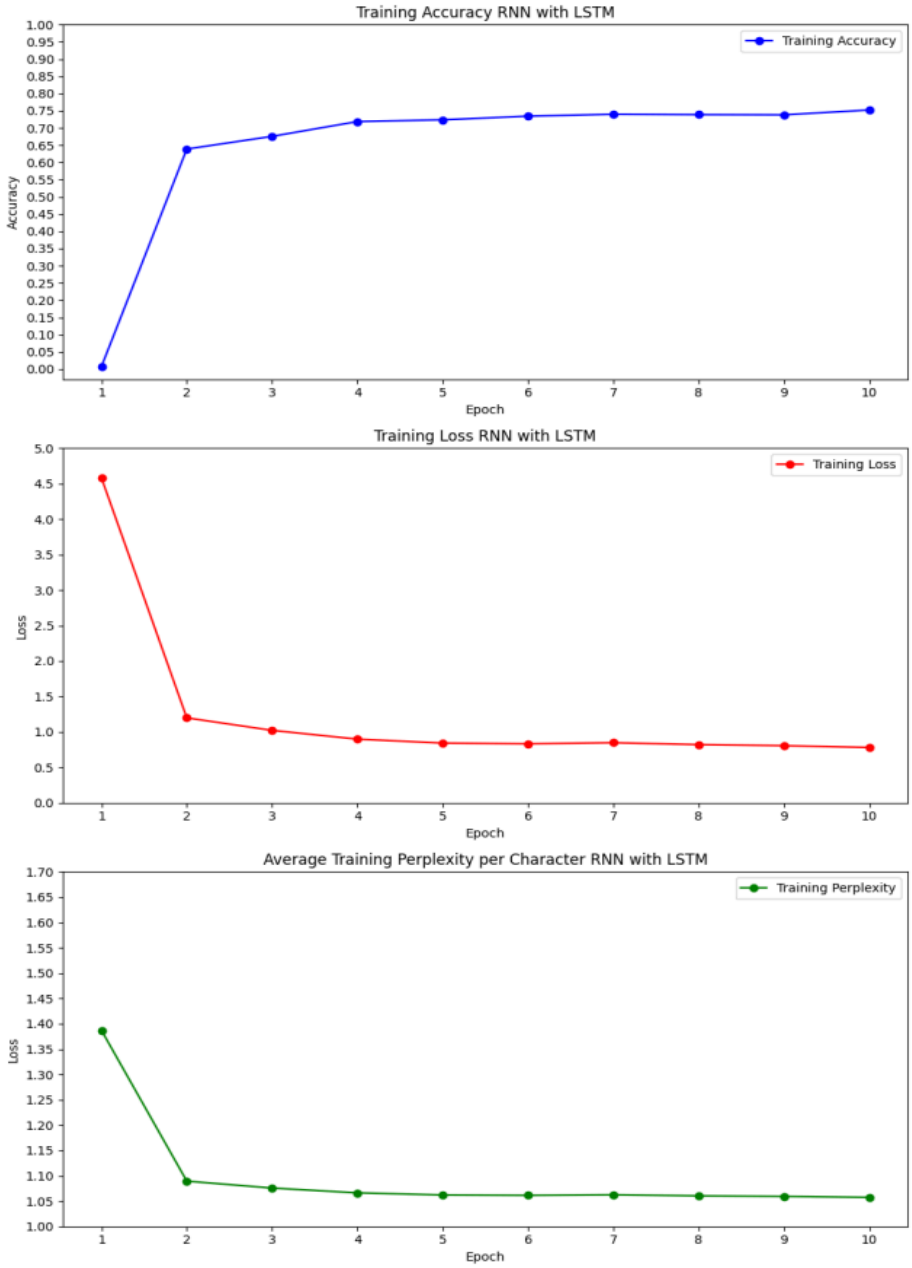

Figure 8: Training Accuracy (top), Training Loss (middle) and Training Perplexity per character (bottom) over 10 epochs for the RNN with LSTM

The final model we tested was the RNN with LSTM which was found to perform best for word preprocessing but this time after doing character preprocessing instead. As seen in figure 8, there was a Training loss of 1.3, Training accuracy of around 75%, and a Perplexity score of around 1.07. All of which would be a bit worse than the performance of the word preprocessing with the same model. On the next page (figure 7), we will see what the transcript looks like for the character preprocessing using LSTM, and we noticed that the dialogue was less coherent compared to the word preprocessing.

```
Here is an excerpt from our transcript for this model with input prompt, 'Rick: erm, what the sigma?\n\n\nMorty: ':

Rick: erm, what the sigma?

Morty: Ooooooooohhhhhh. (He and Rick are stopped by a bully, Frank Palicky. This is the surprise, a deer of shit.

Beth and Jacob giggle. Jerry starts tearing into the kitchen]

[Jerry, Summer and Beth run through reveal have man, isn't it?

Morty 30: It won't close to her cessors as you can keep simpait on the simulator.

Jerry: Wait, does it fleen Mr. Jelly Bean's closes them.)

Morty 2, 3 and 4: *Screams*

…
```

Figure 9: Figure 6: Output of part of our transcript for the RNN with LSTM using character preprocessing

We can observe that there are misspellings and confusing flow of conversation between the characters. This does not fully pass the 'Rick and Morty' feel test as it is pretty illegible. However, there was a degree of semantics being followed, such as complete sentences and a mostly proper use of punctuation and grammar conventions. It was however really interesting to see such drastic differences between our responses strictly based on character preprocessing and how there was some original content.

Potential Ethical Ramifications

There are two main points to consider regarding the ethical standpoint of transcript generation, and perhaps generative AI as a whole: copyright/intellectual property violation, and job security. The Writers Guild of America, a labor union representing writers in online media, went on strike from May 2 to September 27, 2023 protesting AI being used for scriptwriting. They cover many points regarding the use of generative AI in both commercial and non-commercial settings, and the logistics behind how AI might affect copyright.

The Writers Guild asserts that using copyrighted material to train AI models is not considered fair use in both commercial and non-commercial settings. This is because of the "potential use" that even non-commercial AI models might have, since AI organizations often do non-profit research that is later adapted into a commercial product (2). They make the key point about AI-generated transcripts that they are inherently exploitative of human writers' work—AI transcripts cannot exist without training on man-made scripts, often without the knowledge or consent of the owner. These AI generated scripts pit writers' work against them by using it to train the models, then creating competition in the market that leaves human writers at a disadvantage.

How copyright owners can prove that their work was copied by AI is also a topic of discussion. For example, our project directly seeks to mimic the writing style of Rick and Morty. If we were to pursue a large-scale project in this manner, would our project be considered fair use under copyright policies? The Writers Guild demands protection for this exact case, where they believe that AI systems should be required to disclose what kind of data and prompts were used to create a given generative output. Otherwise, a writer's "written voice" is at risk of being appropriated, which is concerning for those who "make their livelihoods from their skills, reputations, and distinctive style" (4).

Generative AI has already started to affect job security for writers, seeing as the WGA felt compelled to go on strike with it being one of the main topics at hand, and the strike being one of the longest the labor union has ever done. There are differing opinions on how far AI has progressed; some will say that AI has achieved lots, and others will say that it is far from extremes such as AI replacing all human jobs. But the strike provides concrete evidence that AI is, at the very least, threatening peoples' livelihoods.

Our project in particular is very small-scale and the transcripts are only used as an educational tool. This project is obviously not the concern of film and TV show writers today, but transcript generation is still very dangerous in principle. We do believe that there should be proper regulation in place for large AI systems and organizations so that all writers' creative works are protected. Placing such regulation will lead to writers receiving fair compensation for any contribution that they make to a piece of media, whether that contribution is direct or through training data used by AI models.

Discussion

Key Takeaways:

Doing this project allowed us to learn more about natural language processing and gain hands-on experience with creating such models. With complex machine learning models, it is important to find the best fit of hyperparameters, such as learning rate and batch size, so hyperparameter tuning is a big part of generating the best results.

Additionally, in a language processing task, decision making and choices that we make in the data preprocessing can play a big difference in the performance. This includes the decision to preprocess based on words vs characters, as it led to more original content in the dialogue. Additionally, in the beginning of training, we believed that our models would not reach a high enough accuracy and that was to be expected due to the model wanting to form original content. Initially, our LSTM model initially was performing poorly at around 30% accuracy. The main reason for this was because our c0 and h0 parameters were based on batch size instead of word window size. This allowed for a larger window to generate more coherent sequences, and that played a role in the improved performance. As a result, it is always important to start early on training because of the challenges of trying to identify potential flaws in the model.

Future Work:

If we were to continue this project in the future we would want to potentially build use transfer learning, where we would use an existing general LLM, and then move to a smaller dataset of Rick and Morty transcripts or whatever TV show we wish to mimic. In this sense, we would be aiming to develop a foundational model that works for general situations, and then migrate over into a more specific dataset.

Conclusion

This project required tedious desire and charisma as we dived into the wonderful world of natural language processing and script generation. Throughout the project, we traversed through the difficulties and intricacies of creating a model that could display the show's comedic style while also generating meaningful and authentic content. Our methodologies underscored the significant differences between using RNNs with and without LSTMs, while also exploring both word and character preprocessing techniques. We showcased the superior performance of LSTM-enhanced models, particularly their ability to address common issues like vanishing and exploding gradients effectively.

We also learned a lot about the trade-offs in text generation when it comes to finding the balance of similarity and originality. It was important to not undermine the importance of how different data processing choices can skew our results for better or worse, regardless of whether that was the coherence of our script or the content itself. While embracing these different aspects of our projects, it was important to also acknowledge the ethical impact of our model in creative industries, and how projects like ours can significantly put many careers in the creative industry at risk while also potentially violating ethical property violations. However, we did this project out of self interest and are not seeking to profit nor benefit from our results and work.

All in all, we had a wonderful time developing this project and making it out to be what it became. These projects not only allow us to explore our own interests, but also allow us to incorporate what we learn in class to topics we enjoy.

Contributions

Madhav Rajesh: Madhav was one of the leaders for organizing the report and the presentation slides, with his primary focus being on the methodology section for both the slides and report while checking other sections of the report as needed. He also viewed and ran through the code to evaluate and analyze the root causes of potential poor performance in the models. Discussed with the team to find ways to improve the performance.
Evan Silvey: Responsible for data collection, data tweaking, and programming the models. Ran grid-search for hyperparameter tuning and ran multiple training sessions to achieve the current status of the models. Created figures to describe the results and assisted as needed for the methodology sections of the report and presentation.
Hanna Koo: Mainly responsible for research on ethical impacts of the project. Assisted as needed on work for the project check-in, report, slides, and models.
Parsa Bazargani: Responsible for background research, analyzing the usage of NLP in creating transcripts, assessing the scope of our project. contributed significantly to the project check-in (research focused); did the slides and final report for the Introduction, Background Research and Conclusions section.
Yahli Hazan: Did a lot of the group organization in terms of communication and having everyone meet over discord. In terms of contributions, helped a lot with the results section and evaluating our different models performances.

References

Blum-Smith, Laura, et al. “Writers Guild of America West and Writers Guild of America East Comment On USCO Notice of Inquiry on Copyright & Artificial Intelligence.” *Wga.Org*, 2023, www.wga.org/uploadedfiles/news_and_events/public_policy/WGA_Comment_on_USCO_Artificial_Intelligence_and_Copyright.pdf.

Image Links

https://medium.com/@jianqiangma/all-about-recurrent-neural-networks-9e5ae2936f6e
https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/