Project Report

i200796 Muhammad Hamza Shahzad

May 2024

1 Title: String Based Text Generation

2 Authors and Affiliation

Muhammad Hamza Shahzad
i200796@nu.edu.pk
Department of Computer Science, National University of Computer and Engineering, Islamabad, Pakistan"

3 Abstract

The main purpose of this project is to dive deeper into the world of text generation using different models with distinct architectures. The aim was to generate coherent text sequences similar to that of the famous TV show Game of Thrones' and another renowned corpus that goes by the name of Shakespeare-Texts. Multiple models were used including Sentence Transformers, BiDirectional Short Term Long Memory (BiLSTM) and Recurrent Neural Networks (RNNS). RNNS were used with different architectures in order to capture all the intricate details and differences to further boost our findings. Furthermore, multiple changes were made throughout the project that helped the models learn the patterns in a better way and paving way for better generation. All models were further compared on the basis of their performance, quality, quantity, and majority of the performance metrics to find out their pros and cons.Our study offers valuable lessons for future experiments in text generation and natural language processing research due to our findings, comparative analyses and shedding light on the effectiveness of various generative AI approaches.

4 Keywords

1) Recurrent Neural Network
2) Gated Neural Network
3) Bidirectional Long short term memory
4) Transformers

5 Introduction

The Domain NLP is at an all time high when it comes to advancements specially in the field of text generation.However, there exist numerous problems like generating text that blends in with the context, basically, enabling models to create contextually coherent text. In this paper, we will cover how different models learn sophisticated patterns and talk like characters from game of thrones or generate sayings similar to that of shakespeare. By using technologies like transformers to help our model better grasp the semantics. Furthermore, models like Recurrent Neural Networks with different architectures and Bidirectional Long

short term memory are used to train data and mimic the dialogues. With continuous experimentation and numerous methods used, this paper provides insightful findings regarding the domain of Text Generation.

6 Related Work

6.1 Literature Review 1:

Title: Word Level LSTM and Recurrent Neural Network for Automatic Text Generation

6.1.1 Summary of Research Question or Problem Addressed

The research paper mainly aims to tackle the problem of coherent and semantically relevant text, meaning generating texting that fits in the context. The models used in this paper include Recurrent Neural Networks (RNNs) and long term short memory (LSTM) in order to tackle one of the main problems in Nlp; Text generation.

6.1.2 Methods Proposed and Used in the Research

The experiment uses a LSTM-RNN architecture for the generation of text. LSTM is used to mitigate the problems that occur when only using traditional RNNs. These issues include vanishing gradient and exploding gradient descent. Furthermore, the methodology for the word level LSTM-RNN is that firstly data is preprocessed, then trained using the above mentioned model which is followed by semantically coherent text generation.

6.1.3 Datasets Utilized

The dataset which is used in this research paper is called 'The Republic of Plato' which comprises a story from the Greek Philosopher Plato. Furthermore, preprocessing is done by removing punctuation marks etc and dividing the dataset into equal parts for training.

6.1.4 Accuracy Metrics Used

The accuracy matrix used in this paper is training and validation accuracy.Training accuracy is used to tell how well the model works on the training dataset and the validation accuracy basically tells how well the model performs on unseen data.

6.1.5 Discussion on Contributions and Potential Solutions

The paper outlines the power of LSTM-RNNs in the text generation field and how well they perform in comparison with the outdated traditional RNNs. This is because they can capture long term memory and generate text that is more suitable regarding the context.

6.2 Literature Review 2:

Title: Deep Learning Models in NLP

6.2.1 Summary of Research Question or Problem Addressed

The research paper focuses on the role of deep neural networks in the field of Natural Language processing (NLP). It also tackles various questions regarding effective text generation, architectural models and methods used for evaluation of models.

6.2.2 Methods Proposed and Used in the Research

There are various models which are used in this paper which include Recurrent Neural Networks (RNNs) which are used for sequence wise text generation and language modeling. Secondly, Convolution Neural Networks (CNNs) are used for text classification. Thirdly, Variational AutoEncoders (VAEs) for latent representation and Generative adversarial Networks (GAN) for text generation.

6.2.3 Datasets Utilized

The dataset used is News Group Movie Review Sentiment Classification.

6.2.4 Accuracy Metrics Used

Human evaluation was used which checks the quality and coherency of text. Creativity and diversity of text generation is checked and the perplexity. Perplexity refers to how well the model is predicting the words.

6.2.5 Discussion on Contributions and Potential Solutions

The final teachings of the paper stated that RNNs are more suitable for short sequences but are exploited by the exploding gradient issue. CNNs aren't very useful when it comes to text generation. VAEs are more suitable for text generation and Gans are good in learning complex patterns however, they lack diversity.

6.3 Literature Review 3:

Title: Enhancing Text Generation with GANs and Knowledge Distillation

6.3.1 Summary of Research Question or Problem Addressed

The main functionality of the research paper includes how Generative Adversarial networks (GANs) and knowledge distillation can be used in eradicating the challenges faced during text generation. Furthermore, it addresses how these generative Ai methods can be used to generate quality and coherent text.

6.3.2 Methods Proposed and Used in the Research

The method proposed in this model basically targeted to remove the one hot representation by the help of variational autoencoders that derive continuous and smooth representation of texts which is then fed into the GAN discriminator. Then, Knowledge distillation is used where the student (generator model) learns to synthesize the text that aligns with smooth representation derived by the autoencoder teacher model

6.3.3 Datasets Utilized

Two datasets were used in this experiment. The first was the SNLI Dataset and the second was Google dataset.

6.3.4 Accuracy Metrics Used

The Accuracy was measured using BLEU-2, BLEU-3, BLEU-4 scores. Jensen Shannon Distance was also used to provide insight on the quality and coherence of the generated text.

6.3.5 Discussion on Contributions and Potential Solutions

This paper has a significant contribution to the field of NLP and text generation as it emphasized on the use of GANs with knowledge distillation. These findings of the experiment helped in ensuring how well this model performed as compared to simple GANs that fall into the trap of lack of diversity and coherency in text generation.

6.4 Literature Review 4:

Title: Challenges of GANs in Text Generation

6.4.1 Summary of Research Question or Problem Addressed

The research paper emphasizes on how GANs work for text generation and the challenges that are faced along the way. The paper also explores all methods that can be used to improve text generation using GANs.

6.4.2 Methods Proposed and Used in the Research

The paper emphasizes three main techniques that are used to enhance GAN performance in the field of generation, especially for discrete data like natural language. These approaches are, Gumbel-Softmax Differentiation, Reinforcement Learning and Modified training objectives. These approaches help in optimizing discrete sequence generation and make the handle discrete data in order to better suit text generation.

6.4.3 Datasets Utilized

The dataset used is called Common Crawl.

6.4.4 Accuracy Metrics Used

The paper used a perplexity metric to measure the quality of text generation. BLEU and ROGUE were also used to measure the overlap between reference and generated text.

6.4.5 Discussion on Contributions and Potential Solutions

All of the mentioned approaches contribute significantly when it comes to text generation using GANs for discrete data. Reinforcement learning improves the training stability and Modified objective improves the overall performance. Gumbel-Softmax provides differentiable relaxation. Therefore, all these models jointly contribute in generating text that is meaningful and can be used in applications like chatbots and language models.

6.5 Literature Review 1:

Title: Hypernetworks for Neural Network Architectures

6.5.1 Summary of Research Question or Problem Addressed

The research paper contributes by introducing hypernetworks, these networks generate weights for other neural networks dynamically. This helps in improved training and leads to better results.

6.5.2 Methods Proposed and Used in the Research

Hypernetworks, these networks generate weights for other neural networks dynamically based on input data.

6.5.3 Datasets Utilized

The paper does not mention a dataset as it enforces more on theoretical concepts.

6.5.4 Accuracy Metrics Used

No specific accuracy metric is used as the paper focuses on the potential use of hypernetworks.

6.5.5 Discussion on Contributions and Potential Solutions

The paper mainly enforces on the teachings of hypernetworks and how they are used in neural networks to make them more flexible and adaptive to learning, hence paving way for improved generations in the field of natural language.

7 Materials and Methodology

There were 2 datasets that were used in the project. The first one was Game of Thrones and the second one was Shakespeare's texts. Game of Thrones includes subtitles from 7 seasons of the show. The data is in JSON format and is preprocessed first, this is done by removing regular expressions and html tags in order to train smoothly. Also, data is converted to numbers, each character is given a unique number in order to make the models learn complex patterns in a better way.

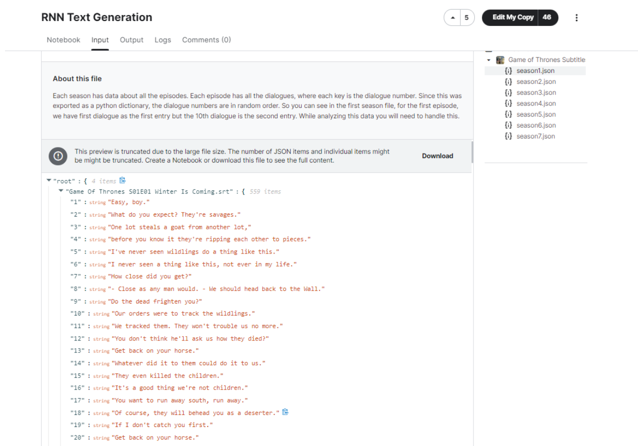


Figure 1: Game of Thrones dataset.

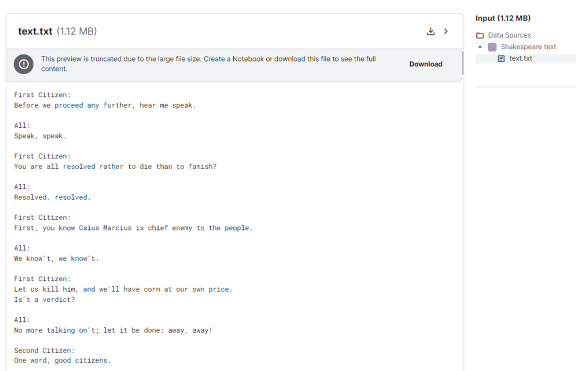


Figure 2: Shakespeare dataset.

In this project, numerous models were used including RNNs, GRUs, LSTM-RNNS and Transformers. RNNs with different architecture were used in order to experiment with all the nitty gritty differences.

Model 1: It uses Recurrent neural networks with gated recurrent units. The architecture comprises an embedding layer followed by GRU layers which is followed by a dense layer which has softmax activation function. Loss is calculated using the formula:
loss=tf.keras.losses.sparse$_{c}ategorical_{c}rossentropy(labels, logits, from_{l}ogits = True)$

Model 2: It uses Recurrent neural networks with gated recurrent units. The architecture comprises an embedding layer followed by GRU layers with 1024 units which is followed by a dense layer which has softmax activation function. Loss is calculated using the formula:
loss=tf.keras.losses.sparse$_{c}ategorical_{c}rossentropy(labels, logits, from_{l}ogits = True)$

Model 3: It uses a transformer with a self attention mechanism. The architecture consists of an embedding layer followed by a multihead attention layer. Lastly, there is a dense layer with a softmax activation function which is used in prediction. Loss is calculated through the following formula:
loss=tf.keras.losses.sparse$_{c}ategorical_{c}rossentropy(labels, logits, from_{l}ogits = True)$

Model 4: In this model, we use the LSTM-RNN model. The architecture includes an Embedding layer followed by a single LSTM layer with 1024 units, and finally a Dense layer for character prediction. Loss is calculated using:
loss=tf.keras.losses.sparse$_{c}ategorical_{c}rossentropy(labels, logits, from_{l}ogits = True)$

8 Results and Discussion

8.1 RNN Model

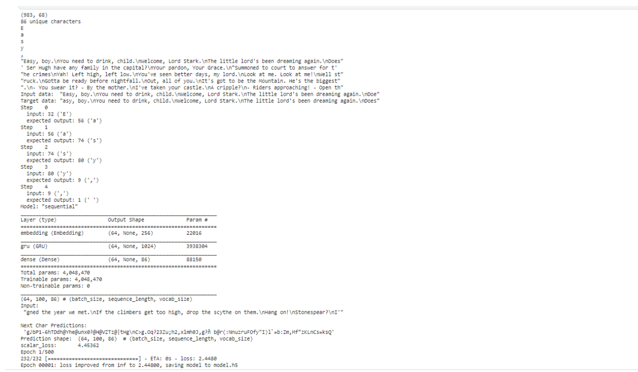


Figure 3: RNN training.

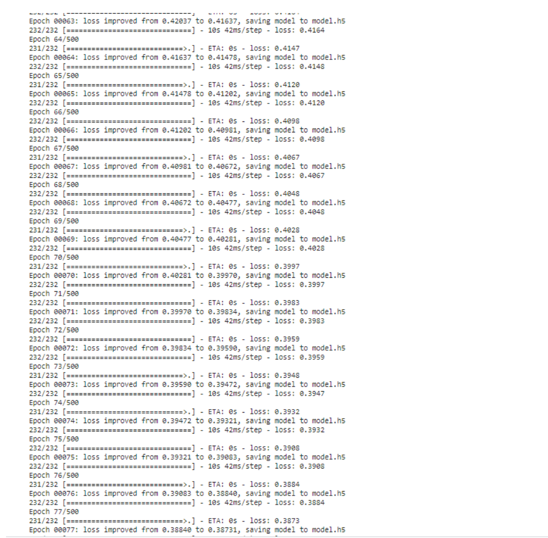


Figure 4: RNN Model.

```
Epoch 00188: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-11.
232/232 [==============================] - 10s 42ms/step - loss: 0.3128
Epoch 189/500
231/232 [============================>.] - ETA: 0s - loss: 0.3137
Epoch 00189: loss did not improve from 0.31265
232/232 [==============================] - 10s 42ms/step - loss: 0.3137
Epoch 00189: early stopping
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (1, None, 256)            22016
_________________________________________________________________
gru_1 (GRU)                  (1, None, 1024)           3938304
_________________________________________________________________
dense_1 (Dense)              (1, None, 86)             88150
=================================================================
Total params: 4,048,470
Trainable params: 4,048,470
Non-trainable params: 0
```

Figure 5: RNN Result (ES)

```
_________________________________________________________________
Seven Hells!
Please take all of Dragons: You see in frandsome flower.
Ser Jaime, I'm not afraid.
Where I come from, thank you.
When Jon Snow returns with the wildlings, though the innocents will be the new many of them we hold.
Have you heard a difference between shut your mother and the find Lord Petyr Baelish, ser.
- I want to. They need st.
Well, he's nothing more than mel nothing,
They're the first our door.
Hold the door!
You can risk your home, my lady.
Now, I have a trial's Daughter.
Queen Margaery!
We're transporting any further.
Eving men are not my enemy.
one day,
But it doesn't matter now. He's dead,
When attacked the Wall.
Your family at least.
And we took those cands and the King's son.
You understand me?
ready to dear myself.
threw too much revenge. We need allies?
Then of his name.
I remember what it feels like.
It's a long way to King's Landing. They won't go any fathers,
Who needs to be done. - What?
- You're not from wry.
Forgive me, Khaleesi, but you don't know.
Gone, along wit
```

Figure 6: RNN text generation.

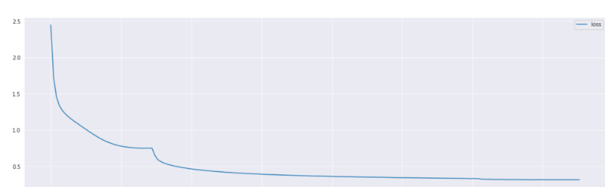


Figure 7: RNN loss plot.

Discussion of the RNN model

This recurrent neural network Model (RNN) worked best for the dataset which basically concludes that even though traditional RNNs are exploited by issues like vanishing gradients, however, they work best when it comes to capturing short sequences. The text generated is by far the best when it comes to performance metrics like diversity, quality and accuracy. The generated text is coherent and quality of text is really good (Can be seen in figure 6). Furthermore, it took the least amount of time. Early stopping mechanism made sure that the trainig process stops as soon as the loss does not improve for a certain number of epoochs as shown in figure 5.

8.2 LSTM-RNN Model

```
/kaggle/input/shakespeare-text/text.txt
First Citizen:
Before we proceed any further, hear me speak.

All:
Speak, speak.

First Citizen:
You are all resolved rather to die than to famish?

All:
Resolved. resolved.

First Citizen:
First, you
No. of unique characters: 65
Character to Index:

  '\n':   0
  ' ' :   1
  '!' :   2
  '$' :   3
  '&' :   4
  "'" :   5
  ',' :   6
  '-' :   7
  '.' :   8
  '3' :   9
  ':' :  10
  ';' :  11
  '?' :  12
  'A' :  13
  'B' :  14
  'C' :  15
  'D' :  16
  'E' :  17
  'F' :  18
  'G' :  19
  'H' :  20
  'I' :  21
  'J' :  22
  'K' :  23
  'L' :  24
  'M' :  25
  'N' :  26
  'O' :  27
  'P' :  28
  'Q' :  29
  'R' :  30
  'S' :  31
  'T' :  32
  'U' :  33
  'V' :  34
  'W' :  35
  'X' :  36
  'Y' :  37
  'Z' :  38
  'a' :  39
```

Figure 8: LSTM-RNN preprocessing.

```
Input text to Integer:

'First Citizen:\nBefor' mapped to [18 47 56 57 58  1 15 47 58 47 64 43 52 10  0 14 43 44 53 56]
Character Stream:

F
i
r
s
t

C
i
t
i
```

Figure 9: Word embedding.

```
Sequence:

'First Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou are all resolved rather to die than to famish?\n\nAl'
"l:\nResolved. resolved.\n\nFirst Citizen:\nFirst, you know Caius Marcius is chief enemy to the people.\n\nAll:\nWe know't, we know't.\n\nFirst Citizen:\nLet us k"
"ill him, and we'll have corn at our own price.\nIs't a verdict?\n\nAll:\nNo more talking on't; let it be done: away, away!\n\nSecond Citizen:\nOne word, good "
'citizens.\n\nFirst Citizen:\nWe are accounted poor citizens, the patricians good.\nWhat authority surfeits on would relieve us: if they\nwould yield us but '
'the superfluity, while it were\nwholesome, we might guess they relieved us humanely;\nbut they think we are too dear: the leanness that\nafflicts us, the '
'object of our misery, is as an\ninventory to particularise their abundance; our\nsufferance is a gain to them Let us revenge this with\nour pikes, ere we '
'become rakes: for the gods know I\nspeak this in hunger for bread, not in thirst for revenge.\n\nSecond Citizen:\nWould you proceed especially against Caiu'
"s Marcius?\n\nAll:\nAgainst him first: he's a very dog to the commonalty.\n\nSecond Citizen:\nConsider you what services he has done for his country?\n\nFirst "
'Citizen:\nVery well; and could be content to give him good\nreport fort, but that he pays himself with being proud.\n\nSecond Citizen:\nNay, but speak not m'
'aliciously.\n\nFirst Citizen:\nI say unto you, what he hath done famously, he did\nit to that end: though soft-conscienced men can be\ncontent to say it was'
Input data:  'First Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou are all resolved rather to die than to famish?\n\nA'
Target data: 'irst Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou are all resolved rather to die than to famish?\n\nAl'
Prediction shape:  (64, 150, 65)
Loss:       4.175309
Epoch 1/50
115/115 [==============================] - 9s 78ms/step - loss: 2.9704
Epoch 2/50
115/115 [==============================] - 9s 77ms/step - loss: 2.2387
```

Figure 10: LSTM-RNN sample.

```
115/115 [==============================] - 9s 76ms/step - loss: 0.7655
Epoch 44/50
115/115 [==============================] - 9s 76ms/step - loss: 0.7459
Epoch 45/50
115/115 [==============================] - 9s 77ms/step - loss: 0.7241
Epoch 46/50
115/115 [==============================] - 9s 76ms/step - loss: 0.7052
Epoch 47/50
115/115 [==============================] - 9s 77ms/step - loss: 0.6862
Epoch 48/50
115/115 [==============================] - 9s 77ms/step - loss: 0.6689
Epoch 49/50
115/115 [==============================] - 9s 77ms/step - loss: 0.6502
Epoch 50/50
115/115 [==============================] - 9s 76ms/step - loss: 0.6339
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (1, None, 256)            16640
_________________________________________________________________
lstm_3 (LSTM)                (1, None, 1024)           5246976
_________________________________________________________________
dense_3 (Dense)              (1, None, 65)             66625
=================================================================
Total params: 5,330,241
Trainable params: 5,330,241
Non-trainable params: 0
```

Figure 11: Trainig Results LSTM-RNN.

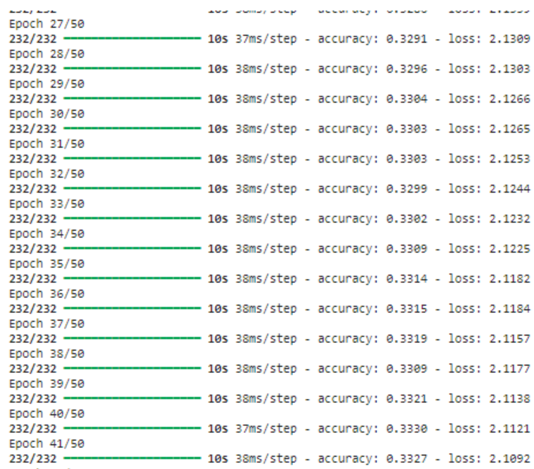
```
Enter your starting string:  romeo
romeos cannot oppress the
courtesy, and they are croop'dou do so increasy
Than can my banisher.

PETRUCHIO:
Whose word is this forlight story to be,
The late of England, there they mut and bid
The priest is not in the mutiate of the house,
When I have might have hid more resolved with
the sharp to the angel, which may make her the deed
As throne and holds upon him as the morn;
For she is secure a stranger, and most accursed
And all the justice of your mistress ere have done
But darkness in this place of hell his works it is.

NORTHUMBERLAND:
Nor I.

CLARENCE:
We have anoney bore me?

BIONDELLO:
When we breathed depated the morning dried.
And, here's no more of them all; then she may, and such as you
Have the nightingale: it was too sudden'd with the night,
When calm death in heaven.

HENRY BOLINGBROKE:
Cousin, stand from me, sir; and therefore fire
Priers of the mind to play at the goose?

MERCUTIO:
Why, sir, I spake like not of many of mine: then, as I love myself,
That I might steal and c
```

Figure 12: Text generation LSTM-RNN.

Discussion for the LSTM-RNN model

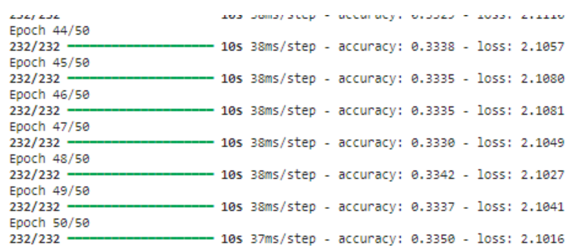
The long short term memory recurrent neural network works increadiby when it comes to generating text and learning complicated patterns due to its mechanism of remembering long sequences of data. However, diversity is one of the significant issues in text generation when it comes to LSTM-RNNs which can be seen in figure 12.

8.3 Transformer Model



```
Epoch 27/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 37ms/step - accuracy: 0.3291 - loss: 2.1309
Epoch 28/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3296 - loss: 2.1303
Epoch 29/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3304 - loss: 2.1266
Epoch 30/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3303 - loss: 2.1265
Epoch 31/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3303 - loss: 2.1253
Epoch 32/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3299 - loss: 2.1244
Epoch 33/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3302 - loss: 2.1232
Epoch 34/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3309 - loss: 2.1225
Epoch 35/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3314 - loss: 2.1182
Epoch 36/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3315 - loss: 2.1184
Epoch 37/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3319 - loss: 2.1157
Epoch 38/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3309 - loss: 2.1177
Epoch 39/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3321 - loss: 2.1138
Epoch 40/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 37ms/step - accuracy: 0.3330 - loss: 2.1121
Epoch 41/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3327 - loss: 2.1092
```

Figure 13: Transformer trainig.



```
Epoch 44/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3338 - loss: 2.1057
Epoch 45/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3335 - loss: 2.1080
Epoch 46/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3335 - loss: 2.1081
Epoch 47/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3330 - loss: 2.1049
Epoch 48/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3342 - loss: 2.1027
Epoch 49/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 38ms/step - accuracy: 0.3337 - loss: 2.1041
Epoch 50/50
232/232 ━━━━━━━━━━━━━━━━━━━━ 10s 37ms/step - accuracy: 0.3350 - loss: 2.1016
```

Figure 14: Transformer training result.

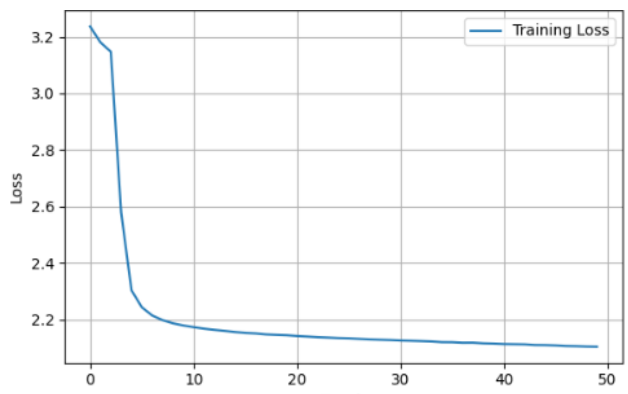


Figure 15: Transformer loss plot.

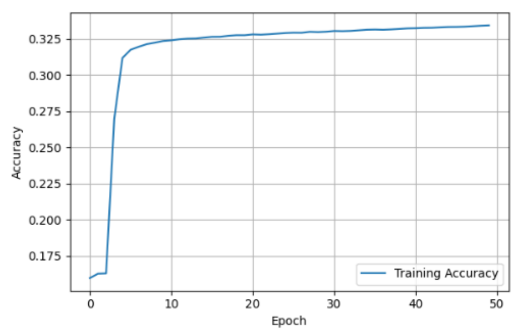


Figure 16: Transformer accuracy plot.

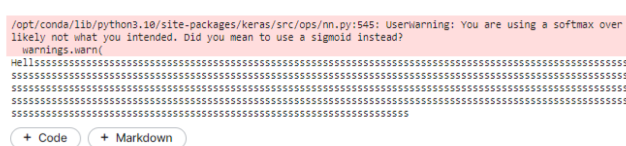


Figure 17: Text generation Transformer.

Discussion for transformer Model Transformer models have emerged as one of the most powerful AI mechanisms to generate text and are a integral part of Natural Language processing (NLP). However, they also are targeted by alot of issues. Starting off, they require a lot of memory. Secondly, the core feature : self attention mechanism which basically requires pairwise comparison will all input makes the model computationally complex and pushes the time complexity to a quadratic time with respect to the sequence length.

9 Conclusion

To summarize the main findings of the project, it can be understood that RNNs can play a vital role in the domain of text generation. The traditional RNNs can be cumbersome to use in certain scenarios as compared to LSTM-RNNs as they are exploited by issues like vanishing gradients etc. However, they can be considered more efficient and better in text generation when it comes to short sequences. Whereas on the other hand, LSTM-RNNS are one of the best models when it comes to text generation as they are most suitable for capturing long term sequences and hence, generate coherent and quality text. Their flexible architecture and long term short memory mechanism helps in understanding complex patterns and improving textual semantics. Transformers on the other hand work really well in the field of NLP, especially text generation. However, they also face alot of issues like overfitting on almost all small and noisy datasets. Furthermore, transformers require a lot of memory. Most importantly, the self attention mechanism makes the model computationally complex as it causes quadratic time complexity with respect to the sequence length.

10 Citations

References

[1] Krishna Sai Buddana, H. V., Manogna, P. V. S., Sai Kaushik, S., & Shijin Kumar, P. S. (2021). Word Level LSTM and Recurrent Neural Network for Automatic Text Generation. In Proceedings of the International Conference on Artificial Intelligence (pp. 1-10). IEEE. https://ieeexplore.ieee.org/abstract/document/9402488

[2] Burin, D., Kilteni, K., Rabuffetti, M., Slater, M., & Pia, L. (2019). Article title. Journal Name, Volume(Issue), Article Number. DOI or URL. https://www.sciencedirect.com/science/article/pii/S1319157820303360

[3] Haidar, M. A., & Rezagholizadeh, M. (Year). TextKD-GAN: Text Generation Using Knowledge Distillation and Generative Adversarial Networks. *Journal Name*, *Volume(Issue)*, Page range. DOI or URL. https://link.springer.com/chapter/10.1007/978-3-030-18305-9$_9$

[4] de Rosa, G. H., & Papa, J. P. (Year). A survey on text generation using generative adversarial networks. *Journal Name*, *Volume(Issue)*, Page range. DOI or URL. https://www.sciencedirect.com/science/article/abs/pii/S0031320321002855

[5] Semeniuta, S., Severyn, A., & Barth, E. (Year). A Hybrid Convolutional Variational Autoencoder for Text Generation. *Journal Name*, *Volume(Issue)*, Page range. DOI or URL. https://paperswithcode.com/paper/a-hybrid-convolutional-variational