Comparative Study of Deep Neural Language Models for Text Generation

Shubham Dodia
Department of Computer Engineering, VIIT, Pune
Pune, Maharashtra, India
shubham.dodia8@gmail.com

Madhavi Sonawane
Department of Computer Engineering, VIIT, Pune
Pune, Maharashtra, India
madhavison06@gmail.com

Janai Kasle
Department of Computer Engineering, VIIT, Pune
Pune, Maharashtra, India
kaslejanai1@gmail.com

Shruti Kamble
Department of Computer Engineering, VIIT, Pune
Pune, Maharashtra, India
shrutikamble3032@gmail.com

Pooja Vishwakarma
Department of Computer Engineering, VIIT, Pune
Pune, Maharashtra, India
vishwakarmapooja1826@gmail.com

***Abstract*— Natural Language Generation (NLG) has emerged as a leading artificial intelligence tool, particularly for text generation. This work examines deep neural language models, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU), from the standpoint of text synthesis. The study offers a comprehensive comparison of the effectiveness of each model, including their design, methods of training, and general performance in various text creation scenarios. Issues including vanishing gradients, comprehending the trade-offs between model complexity and performance, and determining the model's adaptation to other domains are all addressed throughout the examination. We investigate how well these models capture contextual subtleties and produce coherent, contextually relevant text across a wide range of applications, including chatbots, content generation, and summarization. We delve into the impact of model size, training data, and architectural nuances on the quality and efficiency of text generation. The results show that in terms of weighted average, accuracy, loss, and perplexity, GRU performs better than CNN, RNN, and LSTM.**

Keywords—Deep Learning, Machine Learning, Recurrent Neural Networks (RNN), Natural Language Processing, Transformer Models

I. INTRODUCTION

Natural language processing has seen a radical change in the last few years due to the development of deep learning techniques, particularly in the field of text generation. According to Market.us, the global market for AI text generators was valued at about USD 360 million in 2022 and is projected to reach roughly USD 1,808 million by 2032 [12]. The growth is projected to maintain a Compound Annual Growth Rate (CAGR) slightly exceeding 18% from 2023 onward. Google recently revealed its intentions to integrate an AI text generation feature into Chrome, enabling users to compose content more efficiently while avoiding grammatical errors. This functionality, named "Help me write," aims to streamline the writing process for users by minimizing the need for content rewriting, saving both time and effort.

Text generation, a foundational aspect of natural language processing (NLP), holds significant importance across diverse applications, including chatbots, machine translation, content creation, and text summarization. Presently, these technologies find applications in education, entertainment, media, healthcare, e-commerce, and manufacturing. Text generation tools such as Jasper AI, Anyword, Writecream, Rytr, GrowthBar, and ChatGPT, among others, have the capability to produce web copy within minutes based on a set of keywords [9]. NLP, as a discipline, contains two essential processes: Natural Language Understanding (NLU) and Natural Language Generation (NLG). While NLU focuses on deciphering and interpreting human language, NLG is dedicated to the generation of coherent and contextually relevant text. This study focuses on Natural Language Generation (NLG), with a specific emphasis on investigating the text generation capabilities of deep neural language models [2]. The market for AI text generators is expected to growand trend in the next years, as shown in Figure 1.

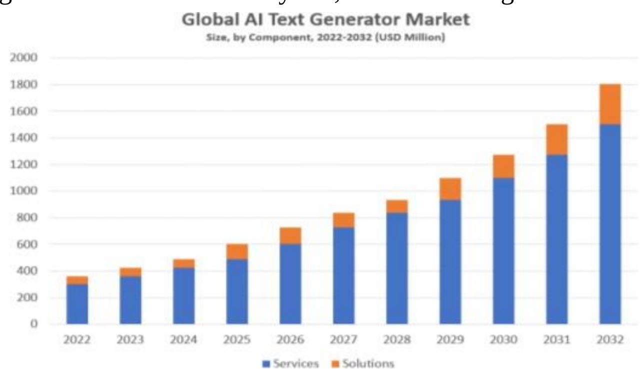

Figure 1. AI Text Generator Market in upcoming years

To delve into the complexities of Natural Language Generation (NLG), a varied array of deep learning models has been utilized, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), and the conventional Recurrent Neural Network (RNN). Each of these models presents unique architectural variation, influencing their ability to capture and generate meaningful sequences of text.

The dataset chosen for this comparative study is the Shakespeare dataset, a rich collection of all of Shakespeare's plays, characters, lines and stylistic elements. Using this dataset allows us to assess the performance of our selected models in the context of complex language patterns and. stylistic variations. The sequential steps involved in using Pre-trained Language Models (PLMs) for text generation are shown in Figure 2.

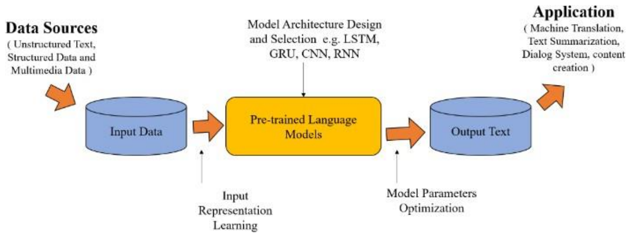

Figure 2. Process of Pre-trained Language Models (PLM's) for text generation

Briefly, the Pre-trained Language Models (PLMs) under consideration are:

1. **CNN (Convolutional Neural Network):** This deep learning architecture is crafted for handling structured grid data, especially images. It employs convolutional layers to autonomously acquire hierarchical representations, capturing spatial hierarchies for efficient feature extraction. It usually consists of fully connected layers for classification and pooling layers for dimensionality reduction [10].
2. **LSTM (Long Short-Term Memory):** The vanishing gradient problem is what this recurrent neural network (RNN) architecture aims to resolve. In order to accomplish this, memory cells with input, forget, and output gates are integrated. This enables the model to effectively recognize and maintain long-range dependencies in sequential data [3].
3. **GRU (Gated Recurrent Unit):** With update and reset gates to regulate information flow, it is a variation of recurrent neural network (RNN) architecture that integrates memory and computation into a single unit. This makes it easier to expose learning to temporal dependencies while reducing some of the complications associated with long short-term memory (LSTM) networks [6].
4. **RNN (Recurrent Neural Network):** This type of neural network architecture is intended to handle sequential data by integrating recurrent connections, which enable the network to retain and employ data from earlier time steps. This makes it appropriate for tasks involving time-series data, sequential patterns, and temporal dependencies, though it frequently encounters difficulties like vanishing or exploding gradients.

This research embarks on a comprehensive comparative analysis of prominent deep neural language models, delving into their architectures, training methodologies, and performance across various text generation scenarios. A thorough grasp of each model's applicability and influence on the field of text creation in artificial intelligence will be possible thanks to the comparison results and debates that will illuminate the distinctive features of each model [1].

II. Literature Review

A comprehensive literature survey for Comparative Study of Deep Neural Language Models For Text Generation is presented in this section. Table 1 represents literature survey which mainly includes the datasets, Pretrained Model/Algorithms used, Performance Matrix, and the remark of each paper.

Table 1. Literature Review on Comparison of Deep Neural Language Models

Ref. No	Paper Title	Dataset	Pretrained Model/Algorithms used	Performance Matrix	Remark
1	A comparative study of deep learning based language representation learning models	CARER-Emotion, DailyDialog, CrowdFlower, Isear	Word2vec and glove, fastText, BERT, RoBERTa, ALBERT, XLNet, ELMo, ULMFiT, GPT-2	Accuracy, loss, precison, recall, F1-score, training time, number of parameters	With 86.12% accuracy, BERT outperforms other NLP techniques in sentiment analysis, a research study that successfully tackles NLP's shortcomings. Enhancing the paper's practical significance and potential impact is the proposed use of BERT in online messaging for security.
2	Research and Implementation of Text Generation Based on Text Augmentation and Knowledge Understanding	The corresponding author can provide the datasets used and analyzed in this study upon reasonable request.	Transformer, GPT2, BERT, BoB, EDA-BoB	Accuracy, Perplexity, BLEU, ROUGE, and ME-TEOR, and the WOEM	This paper skillfully addresses NLG issues by putting forth a workable plan that combines knowledge understanding and text augmentation. The introduction of the lightweight language model (EDA-BoB) shows encouraging results.
3	Research on Text Generation Based on LSTM	-	LSTM (peephole), LSTM, GRU, CBOW	BERT score, BLUERT	This study compares and contrasts LSTM with visual hole connection and GRU in terms of long text generation performance, emphasizing LSTM's superiority. The paper's insightful conclusion highlights the continued relevance of LSTM in text generation despite the advances in pre-training models such as BERT.

4	A Systematic Literature Review on Text Generation Using Deep Neural Network Models	Detailed summary of datasets. Table 7of paper[4]	GPT2, GPT3, LSTM, GRU,	CIDEr, NIST, WER, Word Perplexity, and BERTScore, BLEU, ROUGE, and ME-TEOR	This comprehensive analysis of the literature highlights the growing significance of text generation in artificial intelligence. The paper is a priceless resource for scholars, practitioners, and educators interested in the changing field of text generation.
5	BLEURT: Learning Robust Metrics for Text Generation	WMT	sentBLEU, BERTscore w/ BERT, BERTscore w/ roBERTa, Meteor++, RUSE, YiSi1, BLEURTbase,	precision, recall and F-score, BERTScore, BLEU, ROUGE, BLEURT	In order to address expressivity and robustness issues in natural language generation, this work introduces BLEURT as a useful metric. The novel pre-training scheme, improved with synthetic data, represents a significant breakthrough, proving the generalizability of BLEURT for assessing neural encoder-decoder models in NLG in various domains.
6	Comparison of Evaluation Metrics for Short Story Generation	-	GRU, Pretrained GPT-2, Finetuned GPT-2), N-gram, CBOW	Perplexity, BLEU score, the number of grammatical errors, Self-BLEU score, ROUGE score, BERTScore, and Word Mover's Distance (WMD).	This work provides important insights into the performance of the N-gram, CBOW, GRU, Pretrained GPT-2, and Finetuned GPT-2 models by conducting a systematic comparison of short story generation across these models using a variety of automatic evaluation metrics. The paper is more significant because of its emphasis on recommending the use of metrics for particular goals and correlating them.
7	CONTEXT BASED TEXT-GENERATION USING LSTM NETWORKS	The Lord of the Rings(LOR) dataset	Word2Vec, Bidirectional LSTM	BLEU, F-measure, WER, accuracy, perplexity,	This work creatively introduces context vectors to address shortcomings in LSTM-based language models for text generation, most notably showing enhanced semantic consistency when using the word clustering method. Context-enhanced language generation models can be advanced by taking into account future considerations.
8	Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet	-	GPT, BERT, and XLNet	-	Through a deft navigation of the terrain of Transformer-based models (GPT, BERT, and XLNet), this paper highlights the capabilities of GPT-3, the bidirectional nature of BERT in Google search, and the goal of XLNet to improve BERT.
9	Pre-trained Language Models for Text Generation: A Survey	WMT'16 German-English, WMT'14 English-French, CNN/DailyMail, XSum, GigaWord, SAMSum, DSTC7-AVSD, SQuAD, LDC2017T10, WebNLG, E2E	XLM, mBERT, UniLM, MASS , T5 , BART and PEGASUS, DialoGPT, Meena, PLATO (utilized the Seq2Seq loss), SC-GPT, GPT2,	SacreBLEU, ROUGE, BERTScore, Perplexity, Inform, Success, Distinct, METEOR, chrF++	This survey covers several aspects of the research on text generation using Pre-trained Language Models (PLMs), including input representation learning, model architecture, and parameter optimization. Language-agnostic PLMs are essential, andit addresses issues like controllablegeneration and suggests future directions like multi-grained control and optimizationexploration.
10	A Comparative Study on Word Embeddings in Deep Learning for Text Classification	20NewsGroup, SST-2, AAPD, Reuters	CoVe, Flair, ELMo and BERT, CNN and BiLSTM and the methods used are Word2vec, GloVe, FastText	Accuracy macro-F1	In order to classify texts, this study employs CNN and BiLSTM encoders to compare contextualized and classic word embeddings methodically. Important results include the overall dominance of CNN, the superior performance of BERT over ELMo, and recommendations for best pairings based on dataset properties.

III. Comparative Analysis of CNN, RNN, LSTM & GRU

A. CNN: Originally intended for image processing, convolutional neural networks (CNNs) have shown promise in other domains as well. Text generation and other activities involving natural language processing are among the many tasks for which CNNs are being used. In text generation task, CNNs are used differently from more traditional sequential models like Gated Recurrent Unit(GRU), Long Short-Term Memory(LSTM), and Recurrent Neural Networks(RNNs). In Figure 3, the architecture and workflow of a Convolutional Neural Network(CNN) pipeline are schematically digrammed.

Here's a simplified formula for a basic convolutional operation in a text generation CNN. Assuming we have a 1-dimensional input sequence x of length n, and we apply a convolution operation with a filter w of size k.

$$y_i = f(w * x_{i:i+k-1} + b) \tag{1}$$

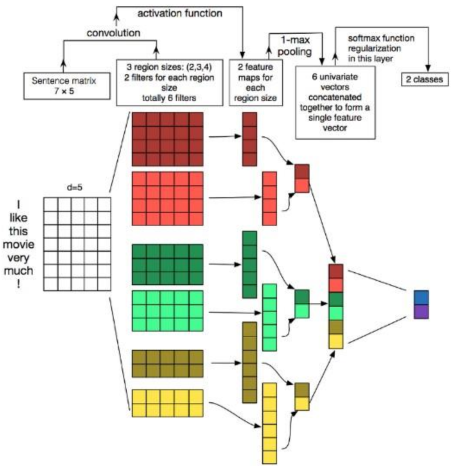

Figure 3. Schematic Diagram of CNN Pipeline

- Convolutional Layers: key elements that carry out convolution operations to extract hierarchical and local patterns from the incoming data.
- Pooling Layers: Shrink the spatial dimensions to highlight salient characteristics and support translation invariance.
- Fully Connected Layers: Traditional dense layers that connect every neuron in one layer to every neuron in the next layer.

B. RNN: Recurrent Neural Networks (RNNs) are becoming very useful tools in natural language processing (NLP), especially when it comes to text production. RNNs are superior to typical neural networks for tasks like language modeling, text production, and text prediction because of their sequential character, which enables them to capture dependencies in sequential input. The internal workings of a Simple Recurrent Neural Network (RNN) during information processing are depicted in Figure 4.

Hidden State Update

$$h_t = \sigma(W_{hh} * h_{t-1} + W_{xh} * x_t + b_h) \tag{2}$$

Output Calculation

$$y_t = softmax(W_{hy} * h_t + b_y) \tag{3}$$

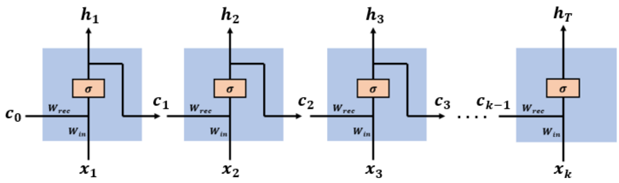

Figure 4. Simple RNN Internal Operation

The key components of an RNN are its hidden layers. RNN can remember the word (data) sequence and use the sequence pattern for prediction with the aid of hidden layers. It has been applied to speech recognition and other NLP jobs where word order is important. RNN functions similarly to a memory that retains the sequence since it accepts input in the form of time series, or word sequences.

C. LSTM: The internal architecture of a Long Short-Term Memory (LSTM) model is depicted in Figure 5, which also highlights its gating mechanisms and specialized memory cells for enhanced information processing and retention.

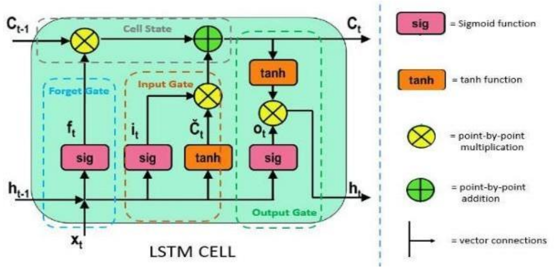

Figure 5. Internal architecture of LSTM model

The RecurrentNeural Network (RNN), which specializes in Long Short- Term Memory (LSTM), is a powerful architecture for text production. Essentially, each LSTM network hasthree gates to regulate information flow and cells to store data. Information is carried by the Cell States from early to later time steps without disappearing. Tanh activation, often known as sigmoid activation, is a technique used by gates. Tanh activation values vary from 0 to 1.

- Forget Gate **-** Decides which data from the prior state should be removed.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{4}$$

- Input Gate: Adds new data to change the state of the cell.

$$i_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{5}$$

$$\hat{C}_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{6}$$

Cell state Update:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{7}$$

- Output Gate **-** Produces the final output based on the modified cell state.

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

D. GRU: The internal architecture of a Gated Recurrent Unit (GRU) model for effective sequential data processing is shown in Figure 6. This model has fewer parameters than an LSTM, making it simpler to design.

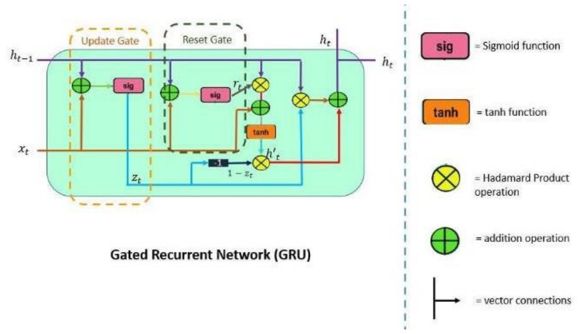

Figure 6. Internal architecture of GRU model

The goal of GRU, a specific type of recurrent neural network (RNN), is to enhance training efficacy and solve issues like the vanishing gradient issue. Long Short Term Memory (LSTM) networks and GRU are similar in that they require less computing power, which makes GRU a good option when computing resources are limited [13].

GRU has Update Gate, Reset Gate and Hidden State. Below is the simplified version of the GRUformula for a single time step in the context of text generation [5]:

Reset gate

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r) \quad (10)$$

Update gate

$$z_t = \sigma(W_z * [h_{t-1}, x_t] + b_z) \quad (11)$$

Candidate Activation

$$\hat{h}_t = tanh(W * [r_t \odot h_{t-1}, x_t] + b) \quad (12)$$

Hidden State Update

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (13)$$

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r) \quad (14)$$

A detailed comparative analysis of RNN, LSTM, CNN, and GRU text generation models is given in Table 2. The table illustrates its coverage of important features. When analyzing each neural network architecture's advantages and disadvantages in relation to text generation tasks, this table is a useful resource [11].

Table 2. Comparison of CNN, RNN, LSTM & GRU for Text Generation

Feature	CNN	RNN	LSTM	GRU
Architecture	Convolutional	Recurrent	Recurrent	Recurrent
Sequential Processing	No	Yes	Yes	Yes
Memory Handling	Limited to local patterns	Limited by vanishing gradient	Good for long dependencies	Effective for sequential data
Computational Complexity	Moderate	High	High	Moderate
Long-Term Dependencies	No	Challenging due to vanishing gradient	Yes	Yes
Local Pattern Recognition	Yes	No	Limited	Limited
Gating Mechanisms	N/A	N/A	Input, Forget, Output gates	Update, Reset gates
Training Time	Shorter	Longer	Longer	Shorter
Common Use Cases	Image classification, Text classification	Time series prediction, Speech recognition	Text generation, Translation	Text generation, Translation

IV. Experiments and Results

A. **Hardware**:

o **CPU-** Intel Core i5 or i7 and AMD Ryzen 5, Ryzen 7

o **RAM**-Minimum 8 GB, Recommended 16 GB or more for better performance

o **GPU**-NVIDIA GeForce GTX 1660, RTX 2060

B. **Evaluation metrics**:

o **Accuracy -** It speaks about how well a machine learning model performs when applied to a validation dataset.

o **Loss -** Error or discrepancy between the intended values and the anticipated result.

o **Perplexity -** It is a measure of the probability that a sentence produced by a model trained on a dataset will be encountered.

As we trained four different models for the text generation such as CNN, RNN, LSTM, GRU and these carefully trained models are able to achieve the accuracy as shown in the following Figure 7. Out of all the four models GRU gives more accuracy which 68.81%. And the accuracy of other models i.e. CNN, RNN and LSTM are 59.42%, 61.18% and 65.48%. Similarly the comparative loss analysis for these models is shown in figure 8. The loss values of CNN, RNN, LSTM, GRU are 1.3821, 1.3421, 1.1332 and 1.0046. Figure 9 and Figure 10 compares the accuracy and loss of the CNN, RNN, LSTM, and GRU text generation models in a heatmap, highlighting their differences. A rapid visual evaluation of each architecture's performance across various evaluation metrics is made possible by the color-coded representation. Warmer tones indicate areas with higher accuracy, while cooler tones indicate areas with lower accuracy or higher loss. With regard to text generation tasks, this thorough visualization facilitates comprehension of the relative advantages and disadvantages of each neural network model [7].

Table 3. Results of Deep Learning Models on Shakespeare Dataset

Models	Accuracy	Loss	Perplexity
CNN	59.42%	1.3821	5.06
RNN	61.18%	1.4002	4.65
LSTM	65.48%	1.1332	3.11
GRU	68.81%	1.0046	2.43

Future Research Directions

Equipped with an extensive set of pre-trained parameters that enable it to comprehend complex linguistic patterns, OpenAI's top transformer-based model, GPT-2, performs exceptionally well in text generation.

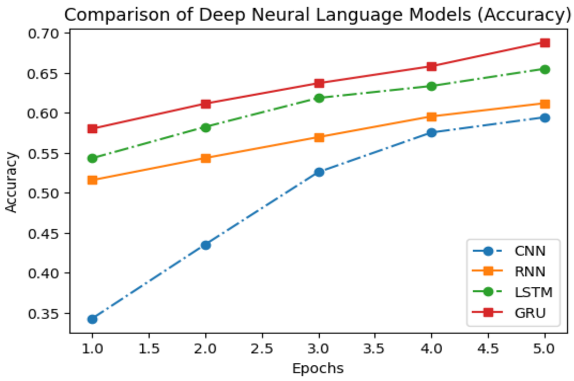

Figure 7. Comparative Accuracy Analysis of CNN, RNN, LSTM, and GRU

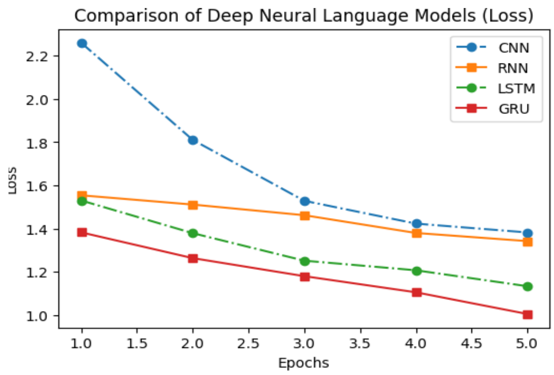

Figure 8. Comparative Loss Analysis of CNN, RNN, LSTM, and

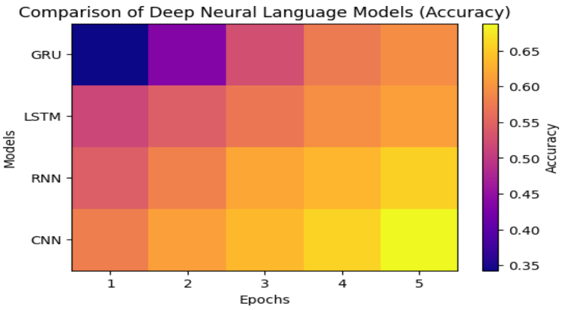

Figure 9. Heatmap comparison of Accuracy of CNN, RNN, LSTM, and GRU

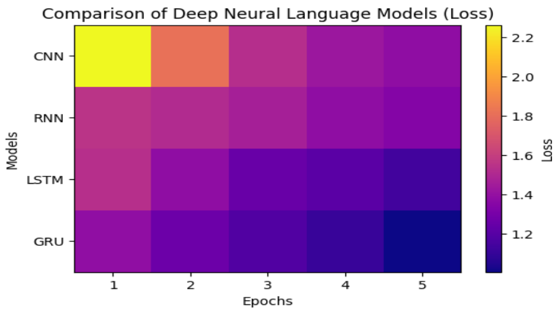

Figure 10. Heatmap comparison of Loss of CNN, RNN, LSTM, and GRU

GPT-3, an OpenAI model with 175 billion parameters, generates text better than its predecessors and does a better job of capturing complex language patterns and dependencies. Its enormous scale adds a crucial dimension to upcoming GPT-2 and XLNet comparative analyses [8].

- Develop and use proper quality metrics that encompass a broader spectrum of language generation nuances, ensuring a more comprehensive evaluation of model performance.
- Explore strategies for optimizing the use of computational resources, including GPUs and TPUs, to enhance accessibility and scalability of deep neural language models.
- Integrate state-of-the-art models like GPT-3 and XLNet. Evaluate and compare the performance of GPT-3 and XLNet in various text generation tasks, exploring their strengths, weaknesses, and potential synergies for improved language understanding and generation.

Conclusion

As a result, we have carefully trained four different text generation models: CNN, RNN, LSTM, and GRU. These trainings have produced informative performance differences. This comprehensive study examines the efficacy of deep neural language models, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), for text synthesis using the Shakespeare dataset. The literature review highlights the growing importance of text generation in artificial intelligence by presenting insights from a variety of studies. The comparative analysis on the Shakespeare dataset shows that GRU performs better than CNN, RNN, and LSTM in terms of accuracy, loss, perplexity, and weighted average. The tabular representation summarizes the main findings from each study. The gaps that have been found point to potential avenues for future research, such as improving models for multilingual text generation and tracking difficulties in producing complex language constructs.

References

[1] Boukabous, M., & Azizi, M. (2021). A comparative study of deep learning based language representation learning models. Indonesian Journal of Electrical Engineering and Computer Science, 22(2).

[2] Liu, L., Sun, Y., Liu, Y., Roxas, R. E. O., & Raga, R. C. (2022). Research and Implementation of Text Generation Based on Text Augmentation and Knowledge Understanding. Computational Intelligence and Neuroscience, 2022.

[3] Li, L., & Zhang, T. (2021). Research on text generation based on LSTM. Int. Core J. Eng., 7(5), 525-535.

[4] Fatima, N., Imran, A. S., Kastrati, Z., Daudpota, S. M., & Soomro, A. (2022). A systematic literature review on text generation using deep neural network models. IEEE Access, 10, 53490-53503.

[5] Sellam, T., Das, D., & Parikh, A. P. (2020). BLEURT: Learning robust metrics for text generation. arXiv preprint arXiv:2004.04696.

[6] Netisopakul, P., & Taoto, U. (2023). Comparison of Evaluation Metrics for Short Story Generation. IEEE Access.

[7] Santhanam, S. (2020). Context based text-generation using lstm networks. arXiv preprint arXiv:2005.00048.

[8] Topal, M. O., Bas, A., & van Heerden, I. (2021). Exploring transformers in natural language generation: Gpt, bert, and xlnet. arXiv preprint arXiv:2102.08036.

[9] Li, J., Tang, T., Zhao, W. X., Nie, J. Y., & Wen, J. R. (2022). Pretrained language models for text generation: A survey. arXiv preprint arXiv:2201.05273.

[10] Wang, C., Nulty, P., & Lillis, D. (2020, December). A comparative study on word embeddings in deep learning for text classification. In Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval (pp. 37-46).

[11] Iqbal, T., & Qureshi, S. (2022). The survey: Text generation models in deep learning. Journal of King Saud University-Computer and Information Sciences, 34(6), 2515-2528.

[12] https://www.globenewswire.com/en/news-release/2023/03/14/2626533/0/en/AI-Text-Generator-Market-Size-to-Reach-USD-1-808-Million-by-2032-CAGR-18.html

[13] https://tech.hindustantimes.com/how-to/how-to-benefit-from-google-chrome-ai-text-generation-feature-help-me-write- 71701942953666.html