Text Generation using Markov And RNN LSTM For Random Text Documents

Aarnav Shankarram
Computer Science and Engineering
Nirma University
Ahmedabad, India
21bce003@nirmauni.ac.in

Vani Balani
Computer Science and Engineering
Nirma University
Ahmedabad, India
21bce021@nirmauni.ac.in

Krishi Desai
Computer Science and Engineering
Nirma University
Ahmedabad, India
21bce129@nirmauni.ac.in

***Abstract*—This research provides a novel approach to text generation based on the fascinating world of Harry Potter, utilizing Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architecture. Taking use of the Harry Potter series' depth and appeal, our goal is to replicate its distinct linguistic patterns and style through the use modern computational methods. We provide a novel method that efficiently imitates the unique linguistic features of the Harry Potter universe by using RNN LSTM networks.The study starts out by examining RNN LSTM networks and highlighting how well-suited they are for managing sequential data and identifying long-term dependencies. Our suggested methodology is framed by a review of previous studies in the field, which highlights current developments in text production with comparable techniques. We formulate the problem statement, presenting the goals of optimization, the mathematical structure, and the standards by which our model is judged. Three layers make up our architecture: an application layer that uses learned representations to generate relevant text, an intelligence layer that consists of multiple LSTM units to understand the of the Harry Potter dataset,and a data layer that preprocesses input text from the Harry Potter dataset. In conclusion, our research represents a significant step forward in natural language generation, offering exciting approach for exploring the intersection of literature and artificial intelligence using the text documents related to harry potter.**

***Index Terms*—text generation, RNN LSTM, Harry Potter, natural language processing**

I. Introduction

One of the core tasks of Natural Language Processing (NLP) is text generation, which is producing logical and contextually relevant text from an input or prompt. Prior to the generalization of deep learning, this function was frequently served by more conventional techniques like Markov Models and N-gram Language Models. For example, N-gram Language Models used statistical patterns in the training data to create text sequences by estimating a word's likelihood based on the n-1 words that came before it [1]. In a similar way, Markov Models adopted the idea of Markov chains and used the current word alone to predict the likelihood of the subsequent word [2]. These techniques, in spite of their simplicity, worked well to capture the statistical connections seen in the dataset and produce coherent text.

Some significant advances led to the use of Deep Learning (DL) to Natural Language Processing (NLP) applications. At first, word representations could be learned quickly and effectively with the use of algorithms like word2vec, which helped with semantic meaning comprehension [3]. Afterwards, the development of Long Short-Term Memory networks (LSTMs) and Recurrent Neural Networks (RNNs) made it possible to represent sequential data, which is essential for understanding language context. These findings, along with the availability of large datasets and pre-trained models, accelerated the acceptance of DL in NLP and completely changed the field [4].

RNNs are crucial for text generation because they understand and replicate the sequential nature of language. They take input one step at a time, remembering previous actions, which is necessary to identify long-term dependencies in text. They efficiently use data across long sequences with versions like as LSTMs and GRUs [5]. RNNs are also flexible enough to handle a wide range of text creation tasks because they can manage variable-length inputs and outputs. They are very useful for text generation overall because they can generate language that is similar to that of humans by using their comprehension of sequential patterns [6].

In the following sections, we will provide a detailed description of our approach. This will include information about our character-based RNN's architecture, the preprocessing steps we took to arrange the Harry Potter dataset, and the training strategy we employed to maximize the model's performance. We will also offer a thorough study of the created text in addition to this. This investigation will contrast RNN and Markov model text generation.

A. Motivation

The goal of the study is probably to assess and compare how well Markov model and LSTM model text production perform in generating text that is both logical and appropriate for the context. This is why the study compares these two

approaches' performance using the Harry Potter book dataset. LSTM models and Markov models show various methods for text generation: While LSTM models use recurrent neural networks to capture sequential dependencies for more complex text production, Markov models rely on statistical patterns in the data.

Our goal is to gain an understanding of the advantages and disadvantages of each method when producing text from a well-known and thoroughly researched corpus by conducting this comparison using the Harry Potter book dataset. The Harry Potter dataset was chosen because of its huge appeal and depth of linguistic variation, which make it a perfect testing ground for text generating algorithms. Through an analysis of how well each model mimics the writing style of J.K. Rowling and keeps continuity with the themes and characters of the Harry Potter series, the study may offer useful information for many scholars interested in natural language production.

B. Research Contribution

Our research attempts to improve text generation methods for the Harry Potter book dataset by presenting an architecture of a character-based RNN that is specially designed to reflect the unique linguistic style of JK Rowling. Furthermore, we have improved the training procedure by utilizing adaptive learning techniques. We have generated sentences from the dataset and conducted comparisons between conventional and new models.

C. Organization

The paper is structured in the following way: Section 2 provides a background report on RNN and LSTM networks. Section 3 studies related careers in the field. Section 4 formulates the problem and outlines the mathematical framework. Section 5 describes our proposed architecture. Section 6 presents exploratory results and discussions. Finally, Section 7 completes the paper and traces future research directions.

II. Background

RNN is widely used for sequential data. They are quite helpful in areas such as language comprehension and other areas. Unlike ordinary neural networks, which just process data in a linear fashion, RNNs are able to retain previous information, allowing them to gain a deeper understanding of the situation as it unfolds. Due to this RNN can be used for language modelling, speech recognition, and time series prediction tasks. However, standard RNNs face an issue called the vanishing gradient issue, which limits their ability to learn long-range dependencies in sequential data [7].
LSTM networks are a type of artificial neural network designed to handle sequential data, like sentences in natural language. By utilizing unique memory cells and gate mechanisms, they surpass the constraints of conventional neural networks. These characteristics enable LSTMs to selectively retain or forget data throughout lengthy sequences. This capacity allows LSTMs to understand complex patterns and context, which makes them very useful for jobs like developing and understanding spoken language. LSTM networks, in particular, provide a robust framework for understanding the complex structure and meaning of language, which makes them appropriate for text generation through the harry potter dataset [8].

III. Related Work

Table 1 summarizes recent advancements in text generation using RNN LSTM networks, highlighting key contributions and methodologies employed.

Within this section, we reviewed pre-existing surveys that explored various aspects of text generation and its applications of it. However, we found several studies with shortcomings due to the absence of a comprehensive and detailed survey on text techniques used in NLP. In [9], the Markov chain model used did not have good accuracy as it could not handle the incorrectly labeled data. [10] focused entirely on using the n-gram approach which does not have good accuracy and [11] had a problem with the dataset as it was not large enough, leading to a lack of good results. In [12] and [13], the generated sentence need not necessarily be meaningful. In [5], even when a different DL approach like generative adversarial network is used it has many shortcomings. Here CS-GAN was used. So when in [14], simple RNN is used especially word-based RNN then it has a very large vocab size and thus causing the length of one hot encoded vector to be very big and hence instead of word-based approach we have used character-based approach in our review paper from LSTM. TABLE **??** shows the comparison of existing surveys with the proposed survey.

TABLE I
COMPARISON OF SURVEY PAPERS

Year	References	Contributions	Merits	Demerits
2024	Proposed implementation	Text generation using Markov and RNN LSTM for random text documents here harry potter dataset and did a comparison between LSTM and Markov model	LSTM showed higher accuracy and used a character-based approach	–
2022	[12]	Uses residual learning method that improves the performance of neural language models by addressing information gaps left by n-gram models.	This approach consistently enhances performance across language modeling, machine translation, and summarization tasks	Does not guarantee that all generated sentences will be meaningful
2022	[13]	Text generation in Hindi and focusing on predicting subsequent words for completing sentences	achieved 0.69 and 0.61 accuracy using Trigram and Bigram models respectively	Does not guarantee that all generated sentences will be meaningful
2021	[15]	Introduced the Average Repetition Probability (ARP) and established upper bounds for it and revealed that current methods are akin to modifying word probability distributions.	Had high accuracy and solved repetition problem	The repetition problem caused by high inflow words in the language is not solved.
2021	[16]	It focuses on sentence generation using word level RNN	This approach focuses on summarization tasks	Does not guarantee that all generated sentences will be meaningful
2020	[10]	Markov transformer is a new way of generating text quickly while still keeping it accurate.Used for machine translation by making the process faster without sacrificing quality.	Had high accuracy, reduced repetitions, and minimized artifacts.	Using n-gram scores instead of max-marginals in text generation may result in lower accuracy
2018	[9]	Text generation in the target domain to provide labeled data for sentiment classification and did comparision between LSTM, GRU and markov model. and	LSTM and GRU showed higher accuracy and fscore values. These models can handle limited labeled samples in the target domain effectively.	Markov chain-based text generators have little tolerance for incorrectly labeled data, impacting their performance
2019	[14]	Used to predict the next word in bengali language	High accuracy of 0.987 with a loss of 0.0430 after approximately 3 hours of training	cannot create arbitrary length content and requires defining the content length
2018	[17]	Implemented story scrambler system using RNN and LSTM and getting accuracy of 0.63	Evaluated generated stories based on grammar correctness, linkage of events, interest level, and uniqueness	Accuracy is evaluated based on human ratings which introduces bias in the process.
2018	[5]	Used CS-GAN to generate more realistic sentences with labels	Generates realistic sentences	Used smaller dataset
2015	[11]	Focuses on the ability to generate texts with specific sentiments	Hidden Markov model based text generation achieves less accuracy than Markov chain based text generation but can generate a higher number of distinct texts	Used smaller dataset.

IV. PROBLEM FORMULATION

A. Mathematical Framework

1) Input Representation: The input sequences $x_1, x_2, ..., x_T$ are depicted as vectors encoded in one-hot format. In this representation, each each token x_t stands for a specific word or character in the text.

2) LSTM Cell Processing: Within each LSTM cell:

$$f_t = \sigma(W_f \cdot [a_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [a_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}t = \tanh(W_C \cdot [at - 1, x_t] + b_C) \quad (3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [a_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

Here, σ denotes the sigmoid activation function, $\tanh$ denotes the hyperbolic tangent activation function, W_f, W_i, W_C, W_o are weight matrices, and b_f, b_i, b_C, b_o are

bias vectors. h_{t-1} denotes the previous hidden state, and $[a_{t-1}, x_t]$ represents the concatenation of the previous hidden state and the current input.

3) Output Generation: The final hidden state a_T is passed through a linear layer followed by a softmax activation function to generate the output probability distribution over the vocabulary.

B. Optimization Objective

The goal of optimizing text generation with RNNs and LSTMs is twofold: to minimize loss and enhance text coherence. This involves adjusting model parameters iteratively to better align the predicted probability distribution over the vocabulary with the actual distribution of the next character in the sequence. Techniques like backpropagation through time (BPTT) aid in estimating gradients, while optimizers like Adam update parameters to enhance model performance. Ultimately, the objective is to empower the model to generate text that captures the underlying patterns in the training data and produces meaningful sentences based on the Harry Potter dataset.

C. Evaluation Metrics

V. Proposed Architecture

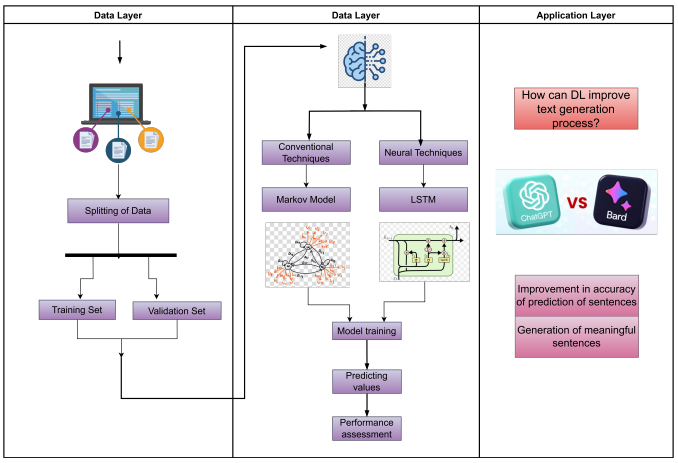

Fig. 1. Survey organization.

A. Data Layer

The data layer of this research paper establishes the framework for acquiring, preprocessing, and organizing the Harry Potter text data for the analysis. First, the text content is taken from random text files realted to the Harry Potter movies. The dataset is shuffled after data loading in order to introduce randomization and reduce any biases. The goal of this stage is to increase the dataset's diversity, which is essential for reliable model training. The dataset is then divided into separate training and validation subsets, taking care to ensure that the text data is distributed in a representative manner throughout both sets [18].Finally, the text input is standardized for use in further natural language processing tasks by applying fundamental preprocessing techniques including tokenization, lowercasing, and punctuation removal.

B. Intelligence Layer

Within the intelligence layer of this implementation focus was on text generation from the Harry Potter dataset using both LSTM (Long Short-Term Memory) and Markov models. Through iterative training, the LSTM model, a recurrent neural network architecture, generates sentences by skillfully capturing long-range dependencies and considering sequential patterns in the text. Markov model was used before LSTM which used probabilistic transition probabilities between neighbouring words or characters to produce text sequences that accurately reflect the statistical regularities seen in the dataset [19].

C. Application Layer

The application layer generates output text based on the learned representations from the intelligence layer, this is used in applications in ChatGPT and BARD [20].

VI. Results and Discussions

A. Experiment Setup and Roles

We need access to a cloud-based platform with CPU and GPU choices in order to run a deep learning model on Kaggle efficiently. GPUs are recommended due to their parallel processing capabilities, which shorten training durations. While enough storage space is required to store datasets, model weights, and intermediate outcomes and RAM is necessary to handle huge data sets during model training. Kaggle's platform, datasets, and community resources can all be accessed with stable internet connectivity and pre-configured software environments that include widely used DL frameworks and libraries.

B. Evaluation Metrics

We report results in terms of perplexity, BLEU score, and cosine similarity, comparing our model against baselines and state-of-the-art approaches.

VII. Conclusion and Future Scope

In conclusion, we have presented a novel approach to text generation on Harry Potter dataset. Our test findings show how effective the suggested approach is, providing new opportunities for natural language generation research. Future research could examine how to apply our methodology to different literary genres and languages.

References

[1] S. L. Aouragh, A. Yousfi, S. Laaroussi, H. Gueddah, and M. Nejja, "A new estimate of the n-gram language model," *Procedia Computer Science*, vol. 189, pp. 211–215, 2021. AI in Computational Linguistics.

[2] T. Brants, "Estimating markov model structures," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, vol. 2, pp. 893–896 vol.2, 1996.

[3] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[4] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part A, pp. 2515–2528, 2022.

[5] Y. Li, Q. Pan, S. Wang, T. Yang, and E. Cambria, "A generative model for category text generation," *Information Sciences*, vol. 450, pp. 301–315, 2018.

[6] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota, and A. Soomro, "A systematic literature review on text generation using deep neural network models," *IEEE Access*, vol. 10, pp. 53490–53503, 2022.

[7] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE International Conference on Neural Networks*, pp. 1183–1188 vol.3, March 1993.

[8] I. Dhall, S. Vashisth, and S. Saraswat, "Text generation using long short-term memory networks," in *Micro-Electronics and Telecommunication Engineering* (D. K. Sharma, V. E. Balas, L. H. Son, R. Sharma, and K. Cengiz, eds.), (Singapore), pp. 649–657, Springer Singapore, 2020.

[9] O. Abdelwahab and A. Elmaghraby, "Deep learning based vs. markov chain based text generation for cross domain adaptation for sentiment classification," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 252–255, July 2018.

[10] Y. Deng and A. Rush, "Cascaded text generation with markov transformers," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 170–181, Curran Associates, Inc., 2020.

[11] U. Maqsud, "Synthetic text generation for sentiment analysis," in *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (A. Balahur, E. van der Goot, P. Vossen, and A. Montoyo, eds.), (Lisboa, Portugal), pp. 156–161, Association for Computational Linguistics, Sept. 2015.

[12] H. Li, D. Cai, J. Xu, and T. Watanabe, "Residual learning of neural text generation with n-gram language model," in *Findings of the Association for Computational Linguistics: EMNLP 2022* (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), (Abu Dhabi, United Arab Emirates), pp. 1523–1533, Association for Computational Linguistics, Dec. 2022.

[13] Ghude, Tejashree, Chauhan, Roshni, Dahake, Krushna, Bhosale, Atharv, and Ghorpade, Tushar, "N-gram models for text generation in hindi language," *ITM Web Conf.*, vol. 44, p. 03062, 2022.

[14] S. Abujar, A. K. M. Masum, S. M. M. H. Chowdhury, M. Hasan, and S. A. Hossain, "Bengali text generation using bi-directional rnn," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, 2019.

[15] Z. Fu, W. Lam, A. M.-C. So, and B. Shi, "A theoretical analysis of the repetition problem in text generation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 12848–12856, May 2021.

[16] H. V. K. S. Buddana, S. S. Kaushik, P. Manogna, and S. K. P.S., "Word level lstm and recurrent neural network for automatic text generation," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–4, 2021.

[17] D. Pawade, A. Sakhapara, M. Jain, N. Jain, and K. Gada, "Story scrambler-automatic text generation using word level rnn-lstm," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 10, no. 6, pp. 44–53, 2018.

[18] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: Better text generation via filling in the
$''_{2018}$.

[19] L. Li and T. Zhang, "Research on text generation based on lstm," *International Core Journal of Engineering*, vol. 7, no. 5, pp. 525–535, 2021.

[20] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122–1136, 2023.