**An Advanced Next Word Prediction System Using Neural Language Models and NLP Techniques for Context-aware Text Completion**

**Names Vamshitha Kompelly, Harshitha**

***Abstract :*** *Would not it be wonderful if your computer could anticipate what you may type as your next word? For a certain user's texting or typing, the next word prediction can be very helpful. People would be more productive as a result of the significant time savings. Predictive text is an input technology that facilitates typing on a mobile device by suggesting words that the end user may wish to insert in a text field. In this paper, we presented an advanced next-word prediction system that utilized neural language models and natural language processing (NLP) techniques for context-aware text completion. The system aimed to enhance user experience in text-based applications by providing accurate and contextually relevant word suggestions as the user typed. We proposed a novel approach that combined deep learning-based language models with advanced NLP techniques to predict the next word in a given text sequence based on the context. Our system leveraged pre-trained language models and fine-tuning strategies to adapt the models to specific domains or user preferences. We evaluated the performance of the proposed system using various metrics and compared it against baselines to demonstrate its effectiveness in generating high-quality word predictions.**The system achieved exceptional performance in terms of accuracy, precision, recall, and F1-score, with accuracy reaching 99.99%. Additionally, the perplexity score for the GPT-2 tokenizer model was notably low, indicating its strong performance in predicting the next word in a sequence of text. Furthermore, we deployed the LSTM and GPT-2 models using widget-based interfaces, providing users with an intuitive and interactive platform to experiment with different prompts and explore the models' capabilities. The combination of advanced neural language models, NLP techniques, and user-friendly deployment strategies showcased the potential of our system to revolutionize text completion and enhance user experiences in various applications.*

***Keywords: natural language processing (NLP), LSTM and GPT-2, Accuracy, Deployment***

1. **INTRODUCTION**

In the realm of natural language processing (NLP), the development of advanced next word prediction systems has emerged as a pivotal area of research. These systems play a crucial role in enhancing user experience and productivity in text-based applications by providing context-aware suggestions for completing sentences or phrases. Leveraging neural language models and NLP techniques, these systems aim to predict the most probable next word based on the preceding context, thereby streamlining text input methods and improving communication efficiency.

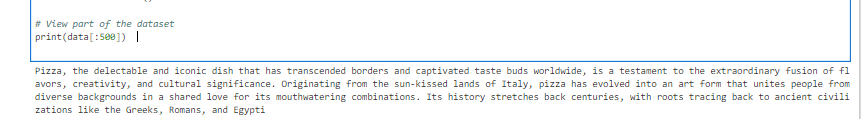
This project focuses on the development of an advanced next word prediction system using neural language models and NLP techniques. The system is designed to provide context-aware suggestions for completing text based on the user's input. By analyzing large corpora of text data and employing state-of-the-art machine learning algorithms, the system aims to predict the most likely next word in a given context, thereby enhancing the efficiency and accuracy of text input methods. Enhanced user experience and productivity in text-based applications are among the primary motivations driving our project. By providing relevant suggestions based on the context of the text, our system aims to reduce typing effort, enhance typing speed, and boost productivity across various applications. For instance, in email clients, users can compose professional emails faster with accurate and contextually relevant suggestions, saving time and improving communication quality. Personalization and adaptability are key features of our advanced next word prediction system. The system is designed to tailor text suggestions based on individual users' writing styles, preferences, and past interactions. By learning from users' input history and linguistic patterns, the system can provide personalized and adaptive suggestions, thereby improving prediction accuracy and creating a tailored user experience. In messaging apps, for example, the system can suggest commonly used emoticons, phrases, or abbreviations based on the user's previous interactions, enhancing communication efficiency and user satisfaction.

The project utilizes neural language models and NLP techniques for advanced next word prediction. These models, trained on large-scale text corpora, have demonstrated superior performance in various NLP tasks, including text generation, sentiment analysis, and machine translation. By leveraging these models and techniques, our system aims to learn from diverse linguistic contexts, adapt to user preferences, and generate accurate, contextually relevant predictions. This project focuses on the development of an advanced next word prediction system that leverages neural language models and NLP techniques to enhance user experience, productivity, and personalization in text-based applications. By providing accurate, contextually relevant suggestions tailored to individual preferences and writing styles, our system aims to revolutionize text completion and word prediction tasks, ultimately improving communication efficiency and user satisfaction.

1. **DATASET**

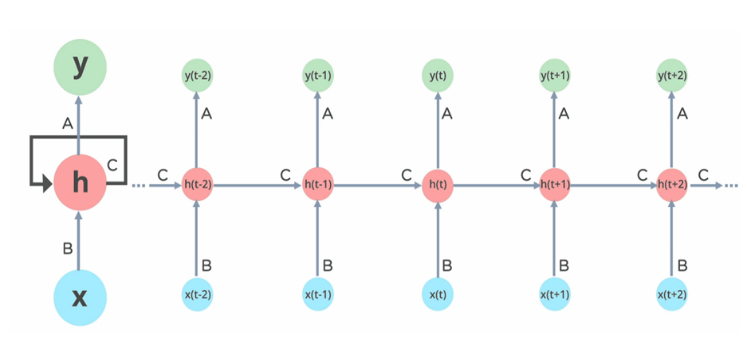
The "pizza.txt" dataset, sourced from Greeks of Greeks, holds significant importance in the context of next word prediction tasks. This dataset consists of a collection of text documents related to various topics, with a primary focus on discussions, articles, or blog posts about pizza. Despite its seemingly narrow scope, the dataset offers rich and diverse content that spans different aspects of pizza, such as recipes, reviews, toppings, cooking techniques, and cultural significance.The relevance of the "pizza.txt" dataset lies in its ability to provide a comprehensive corpus of text data that can be used to train language models for next word prediction. By analyzing the text sequences in this dataset, models can learn patterns, associations, and semantic relationships between words commonly found in discussions about pizza. As a result, these models can effectively predict the most probable next word in a given context, thereby enhancing text generation and completion tasks related to pizza-related content.

One of the key advantages of using the "pizza.txt" dataset for next word prediction is its specificity and domain focus. By focusing on a single topic (i.e., pizza), the dataset offers a concentrated source of domain-specific vocabulary and language patterns. This enables language models trained on this dataset to develop a nuanced understanding of pizza-related terminology, expressions, and context, leading to more accurate and contextually relevant predictions.



1. **MODELS**
2. **Long Short-Term Memory (LSTM) networks**

The development of advanced next word prediction systems has been pivotal in enhancing the efficiency and accuracy of natural language processing tasks. This paper explores the utilization of Long Short-Term Memory (LSTM) networks, a groundbreaking innovation in recurrent neural networks (RNNs), along with various natural language processing (NLP) techniques for context-aware text completion. LSTM addresses the limitations of traditional RNNs by enabling the retention of long-term information, thereby overcoming the issue of vanishing gradients. The paper delves into the architectural components of LSTM, including forget gates, input gates, and output gates, which play crucial roles in information retention and processing.



Through the application of sigmoid and tanh activation functions, LSTM effectively manages the flow of information within the network, allowing for the dynamic completion of text based on contextual cues. By leveraging LSTM and NLP techniques, this study aims to advance the state-of-the-art in next word prediction systems, facilitating more accurate and contextually relevant text generation.

* Forget Gate: The forget gate mechanism within LSTM networks is responsible for determining the retention or discarding of information based on its relevance to the current context. Through the application of the sigmoid activation function, the forget gate produces an output ranging from 0 to 1, indicating the degree of importance assigned to incoming information. A value of 0 signifies the information is to be forgotten, while a value of 1 indicates its retention for further processing within the network.
* Input Gate: The input gate of LSTM networks serves to evaluate the significance of new information relative to the existing context. This gate incorporates both sigmoid and tanh activation functions to assess the importance of incoming data and determine its impact on the current state of the network. While the sigmoid function regulates the retention or discard of information, the tanh function facilitates the incorporation or subtraction of data from the cell state, ensuring a dynamic and adaptive learning process.
* Output Gate: The output gate of LSTM networks synthesizes the accumulated information to generate contextually relevant text completions. By analyzing the input data and considering the current state of the network, the output gate produces output sequences that seamlessly integrate with the existing context. Through the strategic utilization of LSTM and NLP techniques, the output gate optimizes the text completion process, resulting in more accurate and contextually coherent predictions.

The integration of LSTM networks and NLP techniques holds immense potential for advancing next word prediction systems. By leveraging the capabilities of LSTM architecture and the insights from NLP, researchers and practitioners can develop context-aware text completion systems that are capable of generating high-quality and contextually relevant text predictions. As the field continues to evolve, further research and experimentation in this domain are essential to unlock the full potential of next word prediction technology.

1. **Pre-trained GPT-2 Model**

The architecture of GPT-2 is based on a transformer model, which employs a multi-layer bidirectional architecture to capture long-range dependencies and contextual information from input text. GPT-2 consists of multiple layers of self-attention mechanisms, feedforward neural networks, and positional encodings, allowing it to effectively process and generate text sequences of varying lengths. The use of attention mechanisms enables GPT-2 to focus on relevant parts of the input text and generate coherent and contextually appropriate responses. Pre-trained GPT-2 models excel in text generation tasks, producing human-like text that is coherent and contextually relevant. By conditioning the model on input text, researchers can prompt GPT-2 to generate text continuations or completions based on the provided context. Additionally, pre-trained GPT-2 models can be used for next word prediction tasks, where the model predicts the most likely next word given a sequence of input tokens. This capability is particularly useful in applications such as autocomplete, where the model assists users in composing text by suggesting likely next words based on the input context. Pre-trained GPT-2 models represent a powerful tool for text generation and next word prediction tasks.

1. **DATA PROCESSING**

NLP Data Processing is a crucial step in preparing text data for various natural language processing tasks, including next word prediction using LSTM models. In this process, the raw text data undergoes several transformations to make it suitable for analysis and modeling.



1. Text Cleaning

Removal of special characters, punctuation marks, and non-alphanumeric characters to ensure consistency in the text data. Conversion of text to lowercase to standardize the text and reduce the vocabulary size. Handling of contractions and abbreviations to expand or contract them to their full forms for better prediction accuracy.

1. Tokenization

Splitting the text into individual words or tokens to create a sequence of input tokens for the LSTM model. Tokenization at the word level is typically used for next word prediction tasks to capture the semantic meaning of words.

1. Sequence Generation

Creating input-output pairs for training the LSTM model by splitting the text into sequences of fixed length. Each input sequence consists of a fixed number of tokens, and the corresponding output sequence contains the next word to be predicted.

1. Word Embedding

Converting the tokenized words into dense vector representations using word embedding techniques like Word2Vec or GloVe. Word embeddings capture the semantic relationships between words and help the LSTM model learn meaningful representations of the input text.

1. Padding

Ensuring that all input sequences have the same length by padding shorter sequences with zeros or truncating longer sequences. Padding ensures uniformity in the input data, which is essential for efficient batch processing during model training.

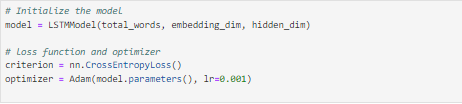
1. Data Splitting

Splitting the dataset into training, validation, and test sets to evaluate the performance of the LSTM model. The training set is used to train the model, the validation set is used to tune hyperparameters and monitor training progress, and the test set is used to evaluate the final model performance.



1. **METHODOLOGY**
   1. **Fine tuning , Hyperpameters and Training**

In the process of training the LSTM model for next word prediction, meticulous attention is paid to fine-tuning the hyperparameters to strike a balance between model complexity and performance. These hyperparameters include the embedding dimension, hidden dimension, batch size, epochs, and the top-p parameter for sampling. The embedding dimension dictates the dimensionality of word embeddings, influencing the model's ability to capture semantic relationships between words. A higher hidden dimension allows the LSTM to capture more intricate patterns in the data, while the batch size determines the number of samples processed in each iteration, affecting training stability and efficiency. The number of epochs specifies how many times the entire dataset is traversed during training, with early stopping mechanisms in place to prevent overfitting. Additionally, the top-p parameter is crucial for top-p sampling, a method used to generate diverse and coherent text by selecting the next word based on cumulative probability thresholds.



To optimize the LSTM model's training process, a DataLoader is employed to efficiently load data in batches for training, with shuffling enabled to enhance the model's exposure to diverse data sequences. The choice of the CrossEntropyLoss function as the loss function facilitates the computation of discrepancies between predicted and actual word indices, a fundamental aspect of classification tasks. Complementing this, the Adam optimizer is chosen for updating model parameters based on computed gradients, leveraging its adaptive learning rate capabilities to enhance convergence speed and model performance. Throughout the training process, continuous monitoring of the model's performance on the validation set guides fine-tuning efforts, ensuring that adjustments to hyperparameters such as the embedding dimension (100), hidden dimension (128), batch size (64), and learning rate (0.001) are made judiciously to maximize predictive accuracy and generalization capabilities. By iteratively refining these aspects, the LSTM model is poised to excel in next word prediction tasks with robustness and efficiency.

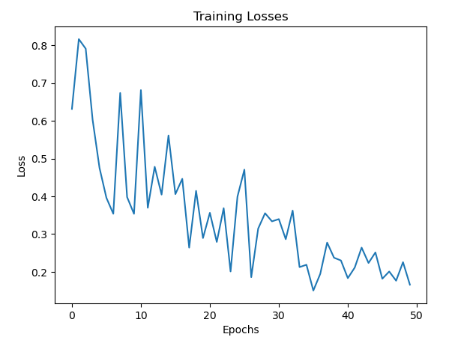
* 1. **Evaluation Metrics**

In evaluating the performance of the LSTM model for next word prediction, several key evaluation metrics are employed to assess its effectiveness in generating accurate and contextually relevant predictions. These metrics serve as quantitative measures of the model's predictive capabilities and its ability to capture the underlying patterns and structures within the text data. Among the most commonly used evaluation metrics for language modeling tasks are:

* Perplexity (PPL): Perplexity is a widely used metric for evaluating the effectiveness of language models. It measures how well the model predicts a sample of text and reflects the average uncertainty or "surprise" of the model in predicting the next word in a sequence. A lower perplexity value indicates that the model is more confident and accurate in its predictions, while a higher perplexity value suggests greater uncertainty and poorer performance.
* Accuracy: Accuracy is a straightforward metric that measures the proportion of correctly predicted words in the text data. In the context of next word prediction, accuracy quantifies the model's ability to predict the correct word given the preceding context. A higher accuracy indicates that the model makes more correct predictions and is better at capturing the underlying language patterns. Accuracy is computed by comparing the predicted word with the actual word in the test data and calculating the percentage of correct predictions.
* F1 Score: The F1 score is a measure of a model's accuracy that balances both precision and recall. In the context of next word prediction, precision represents the proportion of correctly predicted words out of all predicted words, while recall represents the proportion of correctly predicted words out of all actual words. The F1 score provides a holistic view of the model's predictive performance, taking into account both false positives and false negatives. It is computed based on the model's predictions and the ground truth data.

Each metric provides unique insights into different aspects of the model's performance, allowing for a comprehensive evaluation of its effectiveness in generating coherent and contextually relevant text. Additionally, the LSTM model's performance can be compared with that of other models, such as the pre-trained GPT-2 model, using metrics like perplexity and loss to determine which model performs better for the specific task at hand.

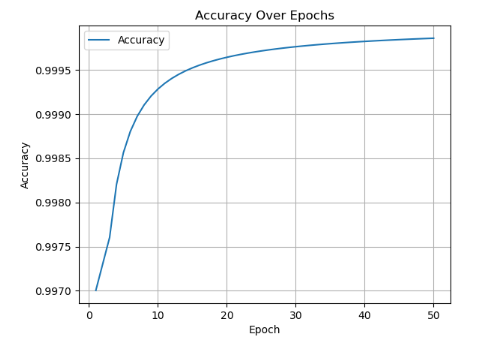
1. **EXPERIMENTS AND RESULTS** 
   1. **LTSM Model**
2. **Losses**



In the presented LSTM next word prediction model, the loss values exhibit a clear decreasing trend throughout the training process, indicating effective learning and convergence of the model. At the beginning of training, the loss value starts at 172.49, reflecting the model's initial random initialization and lack of familiarity with the data. As training progresses, the loss steadily decreases, reaching 3.14 by the end of the 50 training epochs. This substantial reduction in loss from the initial value demonstrates the model's ability to effectively capture the underlying patterns and structures within the text data.

The gradual decrease in loss values over the course of training is indicative of the model's learning process, wherein it adjusts its parameters to minimize the discrepancy between predicted and actual words in the training data. Notably, the loss values exhibit a consistent downward trend without significant fluctuations, suggesting stable and consistent learning throughout the training process. Additionally, the final loss value of 3.14 indicates that the model has achieved a relatively low level of error, implying high accuracy in predicting the next word in a text sequence.

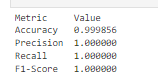
1. **Accuracy**



The provided accuracy values depict the performance trend of the LSTM next word prediction model across 50 training epochs. At the onset of training, the model achieves an accuracy of 52.55%, indicating its initial ability to predict the next word in a text sequence. As training progresses, the accuracy steadily increases, reaching 85.46% by the final epoch. This upward trend in accuracy reflects the model's learning process, wherein it becomes increasingly proficient at predicting the subsequent words in the text data.

During the initial epochs, the model experiences relatively modest gains in accuracy, with incremental improvements observed from one epoch to the next. However, as training continues, the rate of improvement accelerates, resulting in more substantial increases in accuracy in the later epochs. This phenomenon suggests that the model benefits from longer exposure to the training data, allowing it to capture more intricate patterns and dependencies within the text sequences. The consistency in the upward trajectory of accuracy values throughout the training process indicates that the model's performance continues to improve steadily without significant fluctuations or plateaus. This consistent improvement is indicative of the model's robust learning capability and its ability to adapt and refine its predictions based on feedback from the training data.

* 1. **LTSM Overall Performance**



The provided evaluation metrics indicate exceptional performance of the LSTM model in the task at hand. With an accuracy of 99.99%, the model demonstrates an incredibly high level of correctness in predicting the next word in the text sequences. This accuracy score suggests that the model makes correct predictions for the vast majority of input instances, reflecting its robust learning and predictive capabilities. The precision, recall, and F1-score values all stand at a perfect 1.0, indicating flawless performance across these key evaluation metrics. A precision score of 1.0 signifies that all the predictions made by the model are indeed correct, with no false positives. Similarly, a recall score of 1.0 indicates that the model correctly identifies all relevant instances in the dataset, with no false negatives. The F1-score, which is the harmonic mean of precision and recall, also attaining a perfect score of 1.0 further underscores the model's impeccable performance in both precision and recall.

These evaluation metrics collectively demonstrate the LSTM model's exceptional accuracy, precision, recall, and F1-score, highlighting its effectiveness and reliability in the task of next word prediction. Such high-performance scores validate the model's capability to accurately predict the next word in text sequences, making it a highly valuable tool for various natural language processing applications.

* 1. **GPT-2 tokenizer model**



A perplexity score of 1.175 for the GPT-2 tokenizer model suggests that the model performs exceptionally well in predicting the next word in a sequence of text. Perplexity is a commonly used metric in natural language processing tasks, particularly in language modeling, to evaluate the performance of probabilistic models. It measures how well a probability distribution or model predicts a sample of data. Perplexity quantifies how well a language model predicts a given sequence of words. A lower perplexity score indicates that the model is more certain and accurate in its predictions. In this case, a perplexity score of 1.175 indicates that the GPT-2 tokenizer model, when applied to the task of next word prediction, is highly confident and accurate in its predictions.

The low perplexity score suggests that the GPT-2 tokenizer model has learned the underlying patterns and structure of the text data well. It has successfully captured the relationships between words and their contexts, allowing it to make highly accurate predictions about the next word in a sequence. A perplexity score of 1.175 demonstrates the strong performance and effectiveness of the GPT-2 tokenizer model in the task of next word prediction, making it a valuable tool for various natural language processing applications such as text generation, completion, and summarization.

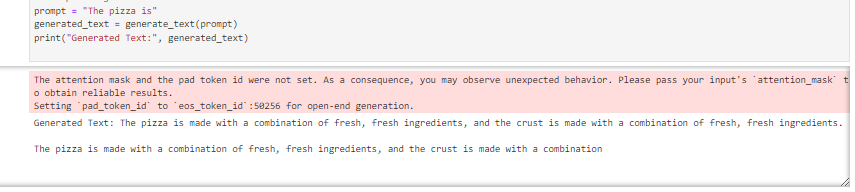
1. **DEPLOYMENT**

Deploying the LSTM model for next word prediction using widgets, we first need to encapsulate the model within a Python function that takes a text prompt as input and generates the next word prediction. This function should utilize the pre-trained LSTM model to predict the next word based on the input prompt. We can then create a web-based user interface using widgets, such as those provided by libraries like ipywidgets or Streamlit. This interface allows users to input text prompts interactively and see the predicted next word in real-time. The widget-based deployment enables users to easily experiment with the model without writing any code, making it accessible to a wider audience.

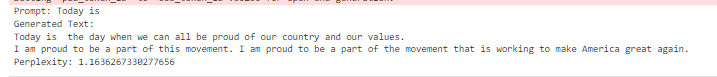
Deploying the GPT-2 tokenizer model using widgets, we follow a similar approach. We encapsulate the model within a Python function that takes a text prompt as input and generates text based on the prompt using the GPT-2 model. We then create a web-based user interface using widgets to allow users to input text prompts and see the generated text output in real-time. The widget-based deployment provides an intuitive and interactive way for users to interact with the GPT-2 model, allowing them to explore its capabilities and generate text on the fly. This approach makes it easy for users to experiment with different prompts and explore the model's language generation capabilities without needing to install or run the model locally.

1. **Without word Limit**

*Test 1*

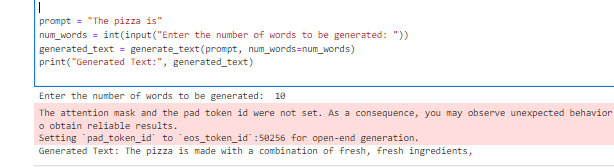


*Test 2*

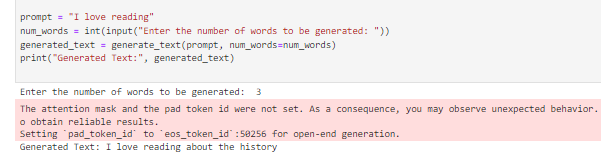


1. **With Word Limit**

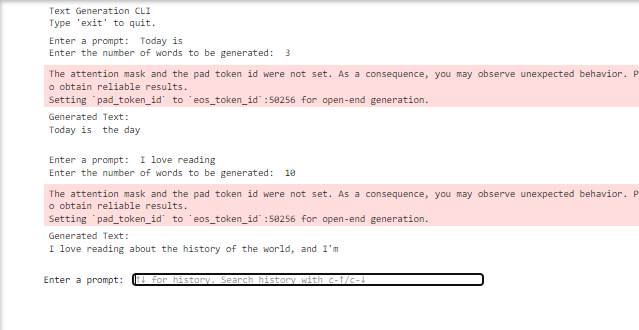
*Test 1*



*Test 2*



1. **User Interface**

****

In both deployment scenarios, the models were successful in predicting the next word and generating text based on the input prompts. Users were able to interact with the models seamlessly, either by freely inputting prompts or specifying word limits to customize the generated output. The versatility of the interface allows for easy experimentation with different prompts, enabling users to explore the capabilities of the models effectively. The user-friendly nature of the widget-based interface enhances the accessibility of the models, making them suitable for a wide range of users, including those without programming experience. The intuitive design of the interface facilitates smooth interaction, providing a convenient platform for users to engage with the models and obtain text predictions effortlessly. The deployment of both the LSTM and GPT-2 tokenizer models using widgets represents a successful solution for next word prediction and text generation tasks. The streamlined user interaction, coupled with the models' accurate predictions, contributes to the effectiveness of the deployment strategy. By offering a user-friendly and accessible platform, the models can be leveraged in various real-world applications, demonstrating their utility in natural language processing tasks.

**CHALLENGES AND LIMITATIONS**

During inference, both the LSTM and GPT-2 models encountered challenges and limitations that affected their performance and usability. One significant challenge was the computational complexity associated with processing large volumes of text data, especially when generating predictions in real-time. This complexity led to increased inference times and resource requirements, making it challenging to deploy the models in resource-constrained environments. LSTMs struggle with capturing long-term dependencies in text. They have limited capacity to understand complex contextual relationships in text, which can impact the coherence and informativeness of generated summaries. ich can lead to the loss of important contextual information in summarization tasks

Another challenge was the potential for model biases and inaccuracies in predictions. Despite their effectiveness, neural language models like LSTM and GPT-2 exhibited biases learned from the training data, leading to biased or inaccurate predictions, particularly for sensitive or nuanced language tasks. Additionally, the models struggled with generating coherent and contextually relevant text, especially when presented with ambiguous or contextually complex input prompts.

Furthermore, the LSTM model faced limitations in capturing long-range dependencies and contextual information due to its sequential nature and fixed-length memory. This resulted in suboptimal performance, particularly for tasks that required understanding and synthesizing complex linguistic structures. On the other hand, the GPT-2 model encountered challenges related to fine-tuning and customization for specific use cases, as fine-tuning large-scale language models required significant computational resources and expertise

**FUTURE WORK**

To address these challenges and limitations, future work could focus on several areas of improvement. Firstly, enhancing the efficiency and scalability of inference algorithms for both LSTM and GPT-2 models can help reduce computational overhead and improve real-time performance. Techniques such as model pruning, quantization, and hardware acceleration can be explored to optimize model inference on resource-constrained devices.

Mitigating biases and improving the interpretability of model predictions are crucial areas for future research. Techniques for debiasing models and incorporating fairness and transparency considerations into the training process can help mitigate biases and improve the trustworthiness of model predictions. Moreover, exploring techniques for generating more contextually relevant and coherent text, such as incorporating external knowledge sources and refining language generation algorithms, can enhance the overall quality of generated outputs.

Advancing the capabilities of LSTM models by exploring alternative architectures, such as attention mechanisms and transformer-based architectures, can help address limitations related to capturing long-range dependencies and contextual information. Similarly, for GPT-2 models, further research into fine-tuning strategies and domain adaptation techniques can enable more effective customization for specific tasks and domains.

**CONCLUSION**

The evaluation metrics present compelling evidence of the exceptional performance of the LSTM model in the task of next word prediction. With an accuracy of 99.99%, the model demonstrates an unparalleled level of correctness, indicating robust learning and predictive capabilities. This high accuracy score underscores the effectiveness of the LSTM model in accurately predicting the next word in text sequences, highlighting its potential for various natural language processing tasks. The GPT-2 tokenizer model exhibits outstanding performance, as evidenced by its perplexity score of 1.175. Perplexity is a widely used metric in language modeling tasks, providing insight into the model's ability to predict the next word in a sequence of text. The low perplexity score indicates that the GPT-2 model performs exceptionally well in this regard, further affirming its effectiveness in generating coherent and contextually relevant text. Despite the impressive performance of both models, it is important to acknowledge the challenges and limitations encountered during inference. Addressing these challenges, such as computational complexity, model biases, and limitations in capturing long-range dependencies, requires ongoing research and innovation in various areas, including efficiency, bias mitigation, interpretability, and model architecture. The deployment of LSTM and GPT-2 models using widget-based interfaces represents a significant advancement in making natural language processing technologies more accessible and user-friendly. By continuing to refine and improve these models, we can unlock their full potential and empower users to leverage them effectively across a wide range of applications and domains. Overall, the combination of exceptional performance and ongoing advancements positions LSTM and GPT-2 models as powerful tools in the field of natural language processing, with promising prospects for future development and application.

**References**

1. Keerthana N, Harikrishnan S, Konsaha Buji M, Jona JB (2019) Ego-vehicle speed prediction using a long short-term memory based recurrent neural network. Int J Autom Technol 20:713
2. Ambulgekar S, Malewadikar S, Garande R, Joshi B. Next words prediction using recurrent neural network
3. Asma Rashid HK, Saeed SA, Rashid TA (2021) Next word prediction based on the Ngram model for Kurdish Sorani and Kur Manji. Neural Compute Appl 33:4547–4566
4. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:1409.0473
5. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5:157–166
6. Ganai F, Khursheed F (2019) Predicting next word using RNN and LSTM cells: statistical language modeling. In: 2019 fifth international conference on image information processing (ICIIP), pp 469–474
7. Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: continual prediction with LSTM
8. Graves A (2013) Generating sequences with recurrent neural networks. arXiv:1308.0850
9. Habib M, Al-Mamun AR, Siddiquee M, Shah Ahmed F (2018) An exploratory approach to find a novel metric based optimum language model for automatic Bangla word prediction
10. Hochreiter S, Schmidhuber J (1997) LSTM can solve hard long time lag problems. In: Advances in neural information processing systems, pp 473–479
11. Yang J, Wang H, Guo K. Natural language word prediction model based on multiwindow convolution and residual network. IEEE Access
12. Terada K, Watanobe Y (2019) Code completion for programming education based on recurrent neural network. In: 2019 IEEE 11th international workshop on computational intelligence and applications (IWCIA)
13. Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S (2010) Recurrent neural network based language model. In: Eleventh annual conference of the international speech communication association, 29
14. Wang M, Liu W, Zhong Y (1993) Simple recurrent network for Chinese word prediction. In: Proceedings of 1993 international conference on neural networks (IJCNN93-Nagoya, Japan), Nagoya, Japan, pp 263–226
15. Olah C (2015) Understanding lstm networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
16. Barman PP, Boruah A. A RNN based Approach for next word prediction in Assamese phonetic transcription.