Predição da próxima palavra utilizando LSTM

Modelo de predição da próxima palavra da sentença utilizando a rede neural recorrente *Long Short Term Memory*

1st Gustavo Lázaro
Engenharia da Computação
Universidade Tecnológica Federal do Paraná
Apucarana, Brasil
guiandisou@gmail.com

2nd Yuri Getaruck
Engenharia da Computação
Universidade Tecnológica Federal do Paraná
Apucarana, Brasil
yurigetaruck@alunos.utfpr.edu.br

***Resumo*—Este artigo apresenta um projeto de predição da próxima palavra utilizando LSTM, uma arquitetura de rede neural recorrente. O objetivo é desenvolver uma inteligência artificial capaz de sugerir a palavra mais provável a seguir uma sentença dada pelo usuário. Para isso, o artigo explica os conceitos teóricos das redes neurais recorrentes e da Long Short Term Memory, bem como os desafios e as vantagens de aplicá-las nesse tipo de tarefa.**

***Index Terms*—RRN, LSTM, rede neural, predição, NWP**

I. Introdução

A previsão da próxima palavra é uma aplicação fundamental do Processamento de Linguagem Natural (NLP), que tem como objetivo melhorar a eficiência da digitação e fornecer sugestões contextuais enquanto se digita. Esta tecnologia é amplamente utilizada em várias aplicações, como correção automática em e-mails e mensagens de texto, bem como em ferramentas de pesquisa como o Google Search e o MS Word.

Este artigo se concentra na implementação de um sistema de Inteligência Artificial (IA) para a previsão da próxima palavra, utilizando Redes Neurais Recorrentes (RNNs) com Memória de Longo e Curto Prazo (LSTM).

As RNNs são uma classe de redes neurais que são especialmente eficazes para lidar com sequências de dados, como séries temporais ou texto. Elas têm a capacidade única de reter informações do passado e usá-las para influenciar as previsões futuras. No entanto, as RNNs tradicionais lutam para lidar com dependências de longo prazo devido ao problema do desaparecimento do gradiente.

Aqui é onde a LSTM entra em cena. A LSTM é uma variante especializada da RNN que inclui uma 'memória celular' e portões de controle para regular o fluxo de informações dentro da rede. Isso permite que a LSTM lide efetivamente com dependências de longo prazo, tornando-a ideal para tarefas como a previsão da próxima palavra.

Neste projeto, exploraremos como as RNNs equipadas com LSTM podem ser usadas para construir um modelo eficaz de previsão da próxima palavra. Através deste estudo, esperamos contribuir para o campo do NLP e fornecer insights valiosos sobre a aplicação prática das RNNs e LSTM na previsão da próxima palavra.

II. Objetivo

O principal propósito deste projeto é criar um sistema de Inteligência Artificial (IA) dedicado à previsão da palavra seguinte, empregando Redes Neurais Recorrentes (RNNs) atreladas a Memória de Longo e Curto Prazo (LSTM).

O sistema será projetado para prever a próxima palavra em uma sequência de texto, com base no contexto fornecido pelas palavras anteriores. Isso pode ser extremamente útil em várias aplicações, como correção automática, assistentes de digitação e interfaces de usuário preditivas.

Especificamente, o projeto tem os seguintes objetivos:

1) **Compreender e aplicar RNNs e LSTM**: Vamos explorar a teoria por trás das RNNs e LSTM e como elas podem ser aplicadas para modelar sequências de dados. Isso inclui entender como as RNNs podem capturar dependências temporais nos dados e como a LSTM resolve o problema do desaparecimento do gradiente nas RNNs tradicionais.
2) **Desenvolver um modelo de previsão da próxima palavra**: Vamos construir um modelo que possa prever a próxima palavra em uma sequência de texto. O modelo será treinado em um grande corpus de texto e aprenderá a capturar as dependências contextuais entre as palavras.
3) **Avaliar o desempenho do modelo**: Após o treinamento, vamos avaliar o desempenho do nosso modelo em um conjunto de testes separado. Isso nos permitirá entender quão bem nosso modelo aprendeu a generalizar a partir dos dados de treinamento.
4) **Aplicar o modelo em um cenário do mundo real**: Finalmente, vamos explorar como nosso modelo pode ser integrado em uma aplicação do mundo real, como um assistente de digitação ou uma interface de usuário preditiva.

Através deste projeto, esperamos não apenas desenvolver um sistema eficaz de previsão da próxima palavra, mas também ganhar insights valiosos sobre a aplicação prática das RNNs e LSTM no campo do Processamento de Linguagem Natural.

III. Motivação e justificativa

A motivação para este projeto surge da crescente necessidade de sistemas eficientes de previsão de palavras em

várias aplicações. Com o advento da digitalização, a digitação tornou-se uma parte integral de nossas vidas diárias. Seja para enviar um e-mail, escrever um relatório ou simplesmente navegar na web, a digitação é uma habilidade essencial na era digital. No entanto, a digitação pode ser uma tarefa demorada e propensa a erros, especialmente em dispositivos móveis com teclados pequenos.

A previsão da próxima palavra pode melhorar significativamente a eficiência da digitação, fornecendo sugestões contextuais enquanto se digita. Isso não só acelera o processo de digitação, mas também ajuda a reduzir erros de digitação. Além disso, a previsão da próxima palavra pode ser usada para desenvolver interfaces de usuário mais intuitivas e preditivas, melhorando a experiência do usuário.

Contudo, a antecipação da palavra subsequente representa uma tarefa complexa que demanda uma compreensão profunda do contexto e das relações temporais presentes nos dados. As Redes Neurais Recorrentes (RNNs) com Memória de Longo e Curto Prazo (LSTM) demonstraram ser eficazes na abordagem desses desafios. Ao explorar essas RNNs e as LSTM neste projeto, almejamos não somente criar um sistema eficiente de previsão da próxima palavra, mas também acrescentar ao campo do Processamento de Linguagem Natural (NLP).

Em última análise, acreditamos que este projeto tem o potencial de fazer avanços significativos no campo do NLP e abrir novas possibilidades para a aplicação da IA na melhoria da eficiência da digitação e no desenvolvimento de interfaces de usuário mais intuitivas.

IV. Trabalhos relacionados

Esta seção tem como objetivo fornecer um contexto amplo e abrangente sobre o cenário atual de pesquisa e desenvolvimento no campo da previsão da próxima palavra, com ênfase nas abordagens baseadas em Redes Neurais Recorrentes (RNNs) e na utilização da Memória de Longo e Curto Prazo (LSTM). Esta seção explorará os principais estudos, técnicas e descobertas que moldaram o campo, oferecendo um panorama das realizações e desafios existentes no âmbito da previsão da próxima palavra com base em IA.

A. Next Word Prediction Using Deep Learning

A pesquisa *Next Word Prediction Using Deep Learning* [TSY22] se concentra no problema da previsão da próxima palavra na língua hindi usando técnicas de aprendizado profundo. Os autores coletaram dados do corpus paralelo inglês-hindi do IIT Bombay, que contém 15,61,841 frases, e utilizaram 5000 frases com mais de seis palavras para seu trabalho. Eles dividiram o conjunto de dados em uma proporção de 90:10 para treinamento e teste, e, posteriormente, dividiram o conjunto de treinamento em uma proporção de 80:20 para validação. Os autores utilizaram as arquiteturas de redes neurais *Long Short Term Memory* (LSTM) e *Bidirectional Long Short Term Memory* (BiLSTM) para prever a próxima palavra em hindi. Eles também exploraram o uso de cadeias de Markov e um modelo híbrido de LSTM e cadeias de Markov para fins de comparação. Os autores constataram que o modelo proposto alcançou uma aucrácia de 88.55% utilizando a arquitetura LSTM e 89.48% utilizando a arquitetura BiLSTM, superando os métodos existentes. O artigo também aborda o pré-processamento dos dados e a avaliação do desempenho do modelo. No geral, esta pesquisa oferece uma valiosa contribuição para o campo da previsão da próxima palavra na língua hindi utilizando técnicas de aprendizado profundo.

B. Next Word Prediction in Telugu using RNN Mechanism

O projeto de pesquisa *Next Word Prediction in Telugu using RNN Mechanism* [R+22] propõe um modelo de rede Long Short-Term Memory (LSTM) para previsão da próxima palavra na língua Telugu. Os autores utilizaram o conjunto de dados Kaggle denominado telugu_books.csv, que contém 25.108 registros extraídos de diversos livros em Telugu. Eles selecionaram 3.000 registros únicos do conjunto de dados original, com uma proporção de divisão de 80:20 para treinamento e teste. O modelo proposto utiliza as três últimas palavras de uma sentença como entrada e prevê a próxima palavra. Os autores não utilizaram modelos de linguagem probabilísticos (PLMs) ou Informação de Ganho Antecipado (LIG) em seu trabalho. Eles compararam a precisão de seu modelo com o modelo existente que utilizava LSTM para prever a próxima palavra na língua Assamesa. Os autores não utilizaram fonética em seu modelo, uma vez que cada palavra é única em Telugu, ao contrário do Assamese, que possui sinônimos equivalentes para algumas palavras. O modelo proposto alcançou uma acurácia de 95,43%, superior à precisão do modelo existente. Os autores concluíram que seu modelo é eficiente e altamente preciso para a previsão da próxima palavra na língua Telugu.

C. A Machine Learning Approach to predict the Next Word in a Statement

O artigo entitulado *A Machine Learning Approach to predict the Next Word in a Statement* [RY23] explora o uso de aprendizado de máquina para prever a próxima palavra em uma declaração. Os autores empregam duas abordagens usando modelos de memória de longo e curto prazo e LSTM bidirecional para alcançar alta precisão na previsão da próxima palavra. Eles também utilizam técnicas de ”raspagem de web”para adquirir dados de diversos canais e bibliotecas, como TensorFlow, Keras, NumPy e Matplotlib, a fim de obter os resultados.

Os autores utilizaram o conjunto de dados medium-article-dataset, que continha informações de diferentes artigos da plataforma Medium, em seus experimentos. Eles extraíram os dados do site por meio de técnicas de raspagem da web, ao mesmo tempo em que observaram as questões éticas e legais envolvidas, além de manter a qualidade dos dados. O conjunto de dados consistia em 6.508 registros e 10 campos diferentes, incluindo id, URL, título, subtítulo, imagem, aplausos, respostas, tempo de leitura, publicação e data.

Os autores treinaram os modelos LSTM e BiLSTM no conjunto de dados pré-processado e os utilizaram para pre-

ver a próxima palavra em uma declaração. A precisão e a perda de ambos os modelos foram comparadas após 50 épocas de treinamento. Os resultados mostraram que o modelo BiLSTM superou o modelo LSTM, com uma acurácia de aproximadamente 85%, enquanto o modelo LSTM obteve aproximadamente 57%.

Em resumo, este artigo de pesquisa apresenta um estudo abrangente sobre a previsão da próxima palavra usando aprendizado de máquina. Os autores empregam técnicas de raspagem da web e diversas bibliotecas para obter os resultados. Eles comparam a precisão e a perda de duas abordagens usando modelos de memória de longo e curto prazo e LSTM bidirecional, concluindo que o modelo BiLSTM supera o modelo LSTM.

Em resumo, os trabalhos de pesquisa mencionados forneceram uma visão abrangente e direcionamento detalhado sobre as abordagens baseadas em LSTM na previsão de palavras. Mesmo em idiomas desafiadores, como o hindi e o telugu, essas abordagens demonstraram alcançar níveis satisfatórios de precisão. Além disso, além de destacar o uso de Redes Neurais Recorrentes, esses estudos também lançaram luz sobre as técnicas de pré-processamento de dados, revelando lições valiosas que podem ser aplicadas em nossa própria pesquisa.

V. Metodologia

Nesta seção, será descrita em detalhes a metodologia usada para desenvolver o modelo de previsão da próxima palavra com base em redes neurais recorrentes (RNN) com memória de longo e curto prazo (LSTM). O processo pode ser dividido nas seguintes etapas:

A. Importação de Bibliotecas

No início, foram importadas as bibliotecas necessárias no ambiente Python, incluindo TensorFlow, Keras, pickle, NumPy, pandas e matplotlib.

B. Coleta e Carregamento do Dataset

Foi selecionado um corpus em português brasileiro atualizado, coletando dados de várias fontes, como livros, artigos, sites, etc. Em seguida, o dataset foi carregado no ambiente de desenvolvimento.

C. Pré-processamento do Dataset

Antes de usar os dados para treinar o modelo, foi realizado um pré-processamento. Isso incluiu:

- Conversão dos dados brutos em uma lista de sentenças ou frases.
- Limpeza do texto para remover caracteres especiais, espaços extras e pontuações desnecessárias.
- Padronização da capitalização para evitar duplicação de palavras devido a diferentes caixas.
- Divisão do texto em tokens ou palavras individuais.

D. Tokenização e Preparação dos Dados

Foi utilizado um tokenizer para converter palavras em representações numéricas. Foi criado um dicionário que mapeia palavras para números e vice-versa. Em seguida, os dados foram divididos em sequências de tokens de entrada (por exemplo, as últimas 3 palavras da frase) e os tokens de saída (a palavra seguinte). O conjunto de dados foi dividido em treinamento e teste.

E. Criação do Modelo LSTM

Foi definida a arquitetura da rede LSTM, incluindo uma camada de embedding, camadas LSTM com células de memória de longo e curto prazo e uma camada de saída que gera uma distribuição de probabilidade sobre o vocabulário.

F. Visualização do Modelo

Foi utilizada a biblioteca Matplotlib para criar uma representação gráfica do modelo, mostrando camadas, conexões e fluxo de dados.

G. Treinamento do Modelo

O modelo foi treinado com os dados de treinamento, ajustando os hiperparâmetros, como o número de épocas e o tamanho do lote. O progresso do treinamento foi monitorado, acompanhando métricas de perda e acurácia.

H. Previsão

O modelo treinado foi utilizado para prever a próxima palavra com base nas últimas 3 palavras de uma frase de entrada. O desempenho do modelo foi avaliado usando dados de teste e métricas relevantes, como precisão.

I. Integração em uma Aplicação do Mundo Real

Após a validação do modelo, o modelo foi integrado em uma aplicação do mundo real, como um assistente de digitação ou interface preditiva. A IA foi testada em cenários reais para avaliar sua eficácia e otimizada conforme necessário.

Esta metodologia fornece uma estrutura clara para o desenvolvimento do modelo de previsão da próxima palavra baseado em LSTM e permite a replicação dos resultados. Todos os experimentos foram realizados no ambiente Python, com a ajuda das bibliotecas mencionadas.

VI. Resultados

A. Modelo 1 - 6520 palavras

O modelo 1, que utilizou um banco de frases com 6520 palavras, apresentou uma progressiva melhoria ao longo das 50 épocas de treinamento, como visto na figura 1. Inicialmente, a perda (loss) estava alta, aproximadamente 6.9 como visto no gráfico da figura 2, enquanto a precisão (accuracy) era baixa, em torno de 4%. No entanto, ao longo das iterações, houve um avanço considerável. A perda diminuiu significativamente, atingindo aproximadamente 0.15, enquanto a precisão aumentou para cerca de 99%. Esses resultados indicam que o modelo foi capaz de aprender e se ajustar aos dados de treinamento de maneira eficiente.

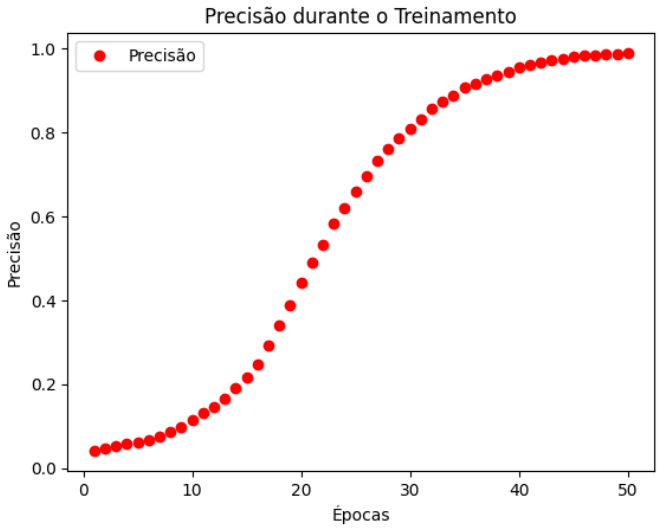

Figura 1. Melhoria por época do modelo 1 com banco de 6520 palavras

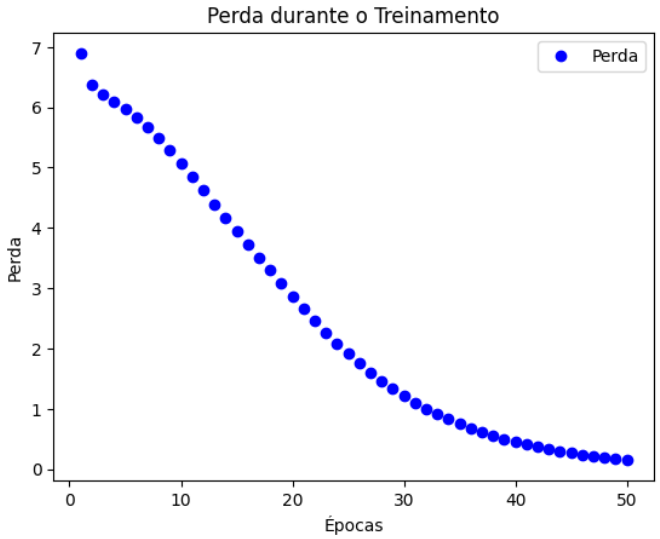

Figura 2. Perda por época do modelo 1 com banco de dados de 6520 palavras

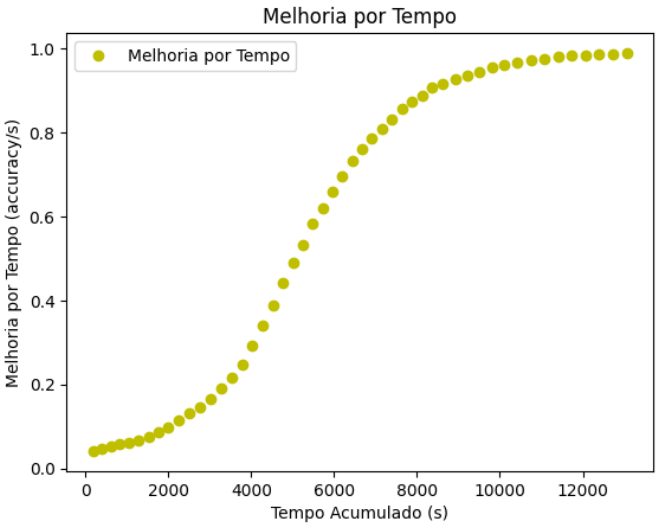

Figura 3. Precisão por tempo do modelo 1 com banco de 6520 palavras

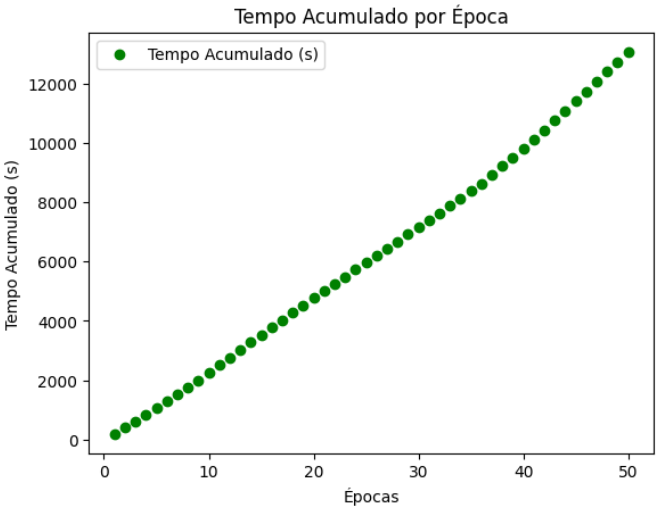

Figura 4. Tempo por época do modelo 1

A alta precisão e a baixa perda ao final das 50 épocas são indicadores positivos da capacidade preditiva do modelo. No entanto, é importante considerar a possibilidade de overfitting, onde o modelo pode estar se ajustando em excesso aos dados de treinamento, comprometendo sua capacidade de generalização para novos dados. Visto que o conjunto de teste é o mesmo do de treinamento, optamos por assim fazer devido ao sistema utilizado em celulares para prever a próxima palavra que vamos inserir no teclado, ele prevê baseado em nosso próprio histórico, fazendo justamente o teste no mesmo conjunto de treinamento.

Seria recomendável avaliar o desempenho do modelo em dados de validação ou teste, a fim de compreender melhor sua habilidade de generalização para dados não observados durante o treinamento. Isso contribuiria para uma avaliação mais abrangente do desempenho do modelo em cenários variados.

Em quesitos de performance é importante avaliar também o desempenho em relação ao tempo, pois no modelo 2, discutido mais a frente, isso foi uma grande dificuldade. Observando a figura 4, podemos notar que o tempo médio por época é constante e por volta de 240 segundos, o que nos leva a aproximadamente 3 horas e meia para realizar o treinamento desse modelo. Já a melhoria por tempo segue a mesmo padrão da melhoria por época, já que possuem a mesma proporção devido à constância do tempo por época.

Adicionalmente, ajustes como técnicas de regularização ou ajustes de hiperparâmetros podem ser considerados para potencialmente otimizar ainda mais o desempenho do modelo ou diminuir problemas de overfitting.

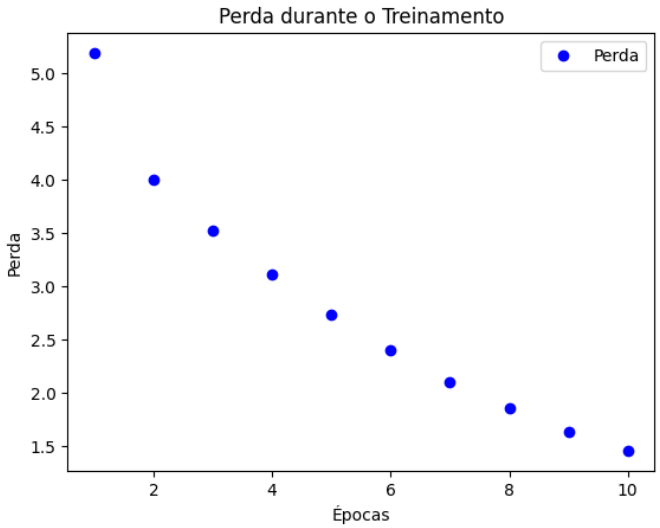

Figura 5. Perda por época do modelo 2 de 223784 palavras

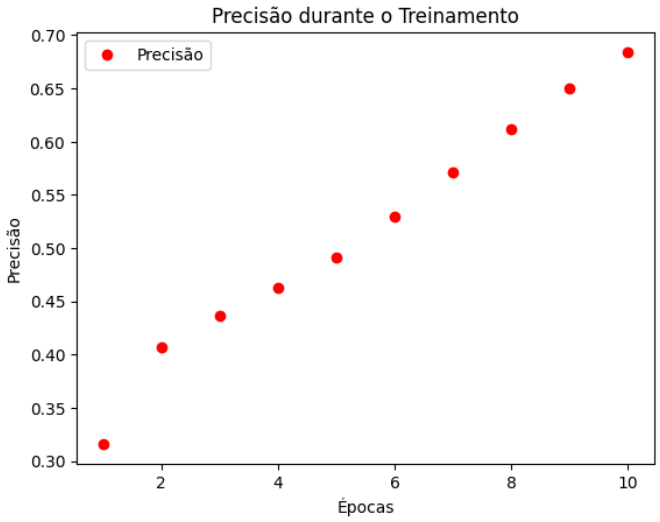

Figura 6. Precisão por época do modelo 2 de 223784 palavras

B. *Modelo 2 - 223784 palavras*

O histórico de treinamento do modelo 2, que utilizou um banco de frases com 223784 palavras, apresentou um progresso notável ao longo das 10 épocas. Inicialmente, a perda (loss) foi registrada em torno de 5.19, como na figura 5, com uma precisão (accuracy) de aproximadamente 31.62% na primeira época, figura 6. Ao prosseguir com o treinamento, houve uma melhoria consistente nos resultados. A perda diminuiu gradualmente, alcançando cerca de 1.46 ao final das 10 épocas, enquanto a precisão aumentou para aproximadamente 68.42%.

Esses resultados indicam um aprendizado significativo do modelo ao longo das épocas. A redução da perda e o aumento na precisão sugerem que o modelo foi capaz de capturar os padrões nos dados de treinamento de forma mais precisa, como

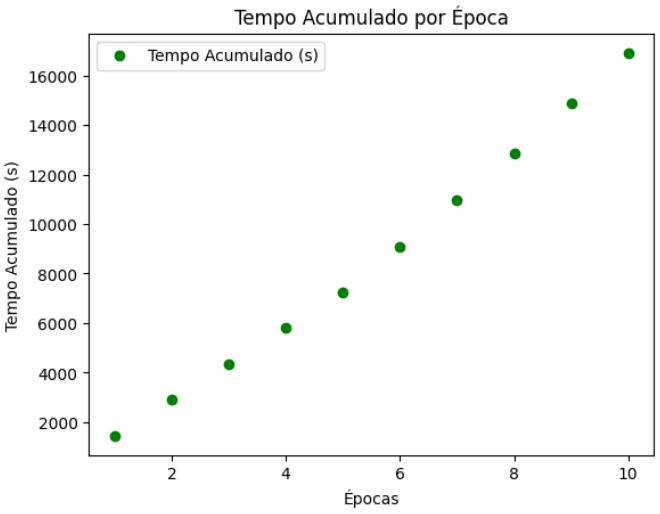

Figura 7. Tempo por época do modelo 2 de 223784 palavras

também demonstrado no modelo 1.

O overfitting pode ocorrer nesse modelo pelo mesmo motivo do modelo 1, onde o conjunto de teste é o mesmo de treinamento.

Porem, no caso desse modelo precisamos observar seu desempenho em relação ao tempo, onde ele obteve um custo computacional mais alto que o anterior, na figura 7 podemos notar que cada época possuia um custo muito maior para ser processada, levando por volta de 1400 segundos nas primeiras 5, e após isso chegando até a 2000 segundos, onde o tempo de processamento dele levou entorno de 5 horas.

Contudo, com os modelos prontos, foram salvos em arquivos para poderem ser carregados mais rapidamente e com isso foi feito um script interativo onde o usuário pode solicitar que ele preveja a próxima palavra conforme ele vai digitando, e também que ele de uma entrada e diga o número de palavras a frente que ele deseja prever, para que o algoritmo use o modelo de sua escolha para o fazer.

VII. Conclusão

No contexto deste projeto, concluímos que esse tipo de inteligência artificial, focada na previsão da próxima palavra, mostra-se altamente eficaz quando aplicado em conjuntos de treinamento e teste semelhantes. Por exemplo, ao usarmos nosso próprio celular, observamos que o modelo é capaz de acertar a próxima palavra com precisão, especialmente em casos frequentes de conectivos, artigos e pronomes presentes em muitos textos. Esses elementos comuns são reconhecidos com sucesso por ambos os modelos, o que foi verificado em testes manuais utilizando o script interativo.

Entretanto, ao utilizar o celular de outra pessoa, ou seja, um conjunto de treinamento diferente do de teste, a precisão do modelo pode variar significativamente. Isso ressalta a importância da similaridade dos conjuntos de dados utilizados no treinamento e teste do modelo para alcançar resultados mais precisos. A presença de termos específicos ou estilos de escrita

distintos entre diferentes usuários pode impactar a capacidade do modelo de prever com precisão a próxima palavra.

Essa variação na precisão destaca a necessidade contínua de aprimoramento e adaptação desses modelos para lidar com a diversidade e particularidades encontradas em diferentes contextos de uso. Estratégias adicionais de adaptação e refinamento do modelo podem ser exploradas para melhorar sua capacidade de generalização em diferentes situações de uso, visando alcançar resultados mais consistentes e precisos.

Referências

[R+22] Vijaya Saraswathi R et al. "Next Word Prediction in Telugu using RNN Mechanism". Em: *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. 2022, pp. 98–104. DOI: 10.1109/ICAISS55157.2022.10010963.

[TSY22] Aditya Tiwari, Neha Sengar e Vrinda Yadav. "Next Word Prediction Using Deep Learning". Em: *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)*. 2022, pp. 1–6. DOI: 10.1109/GlobConPT57482.2022.9938153.

[RY23] Vishal Rathee e Sakshi Yede. "A Machine Learning Approach to predict the Next Word in a Statement". Em: *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. 2023, pp. 1604–1607. DOI: 10.1109/ICESC57686.2023.10193001.