**Fine-Tuning de um Modelo de Linguagem para Responder a Perguntas sobre Produtos da Amazon**

**Autores:**

**Data de Conclusão do Projeto:** 17/12/2024

**Sumário**

1. **Introdução**
   * **1.1. Contextualização**
   * **1.2. Objetivos**
   * **1.3. Estrutura do Documento**
2. **Preparação do Dataset**
   * **2.1. Descrição do Dataset AmazonTitles-1.3MM**
   * **2.2. Exploração e Limpeza dos Dados**
   * **2.3. Pré-processamento**
   * **2.4. Preparação dos Prompts para Fine-Tuning**
3. **Escolha e Configuração do Modelo Base**
   * **3.1. Critérios de Seleção do Modelo**
   * **3.2. Modelos Considerados**
   * **3.3. Modelo Escolhido e Justificativa**
4. **Configuração do Treinamento e Fine-Tuning**
   * **4.1. Ambiente de Treinamento**
   * **4.2. Configuração dos Hiperparâmetros**
   * **4.3. Execução do Fine-Tuning**
   * **4.4. Monitoramento e Salvamento de Checkpoints**
5. **Documentação dos Resultados e Avaliação de Cada Execução**
   * **5.1. Estratégia de Treinamento**
   * **5.2. Monitoramento do Treinamento**
   * **5.3. Avaliação do Modelo**
     + **5.3.1. Primeira Execução - Experimentos com Datasets Pequenos**
     + **5.3.2. Segunda Execução - Sem Otimizações Específicas**
     + **5.3.3. Terceira Execução - Com Otimizações Específicas**
     + **5.3.4. Quarta Execução - Versão Melhorada com Otimizações Específicas**
     + **5.3.5. Quinta Execução - Teste com LLaMA 3-8B Quantizado em 4 Bits (LoRA + Estratégia Econômica)**
   * **5.4. Conclusões das Execuções**
6. **Desenvolvimento da Interface de Usuário**
   * **6.1. Implementação da Interface de Linha de Comando**
   * **6.2. Configuração do Modelo para Geração de Respostas**
7. **Documentação e Conclusões Gerais**
   * **7.1. Conclusões**
   * **7.2. Trabalhos Futuros e Melhorias**
8. **Referências**
9. **Anexos**
   * **9.1. Código-Fonte**
   * **9.2. Logs de Treinamento**

**1. Introdução**

**1.1. Contextualização**

O Processamento de Linguagem Natural (PLN) é uma área da inteligência artificial que se concentra em permitir que máquinas compreendam e interajam com a linguagem humana de forma natural. Com o crescimento exponencial de dados textuais provenientes de diversas fontes, como redes sociais, e-commerce e comunicações digitais, o PLN tornou-se fundamental para extrair informações relevantes e aprimorar a interação entre humanos e máquinas.

Modelos de linguagem avançados, baseados em arquiteturas de redes neurais profundas como os Transformers, revolucionaram o PLN. Esses modelos, quando pré-treinados em grandes volumes de dados textuais, podem ser adaptados para tarefas específicas por meio do fine-tuning, refinando seus parâmetros para melhorar o desempenho em contextos particulares.

**1.2. Objetivos**

Desenvolver modelos capazes de responder a perguntas sobre títulos de produtos da Amazon, utilizando fine-tuning em modelos base pré-treinados GPT-2 e LLaMA 3-8B , adaptando-os para compreender e gerar respostas relevantes no contexto dos produtos

**1.3. Estrutura do Documento**

Este documento está organizado nas seguintes seções:

1. **Introdução:** Apresenta o contexto e os objetivos do projeto.
2. **Preparação do Dataset:** Descreve o dataset utilizado e as etapas de limpeza e pré-processamento.
3. **Escolha e Configuração do Modelo Base:** Detalha os critérios de seleção do modelo e a justificativa da escolha.
4. **Configuração do Treinamento e Fine-Tuning:** Explica como o treinamento foi configurado e executado.
5. **Documentação dos Resultados e Avaliação de Cada Execução:** Apresenta os resultados obtidos em cada execução e a análise de desempenho.
6. **Desenvolvimento da Interface de Usuário:** Descreve a implementação da interface para interação com o modelo.
7. **Documentação e Conclusões Gerais:** Apresenta as conclusões e sugestões para trabalhos futuros.
8. **Referências:** Lista as fontes consultadas.
9. **Anexos:** Inclui o código-fonte e logs de treinamento.

**2. Preparação do Dataset**

As etapas de limpeza, pré-processamento e formatação dos dados estão detalhadas na função process\_and\_format\_dataset no notebook.

**2.1. Descrição do Dataset AmazonTitles-1.3MM**

O dataset **AmazonTitles-1.3MM** contém consultas reais de usuários, títulos de produtos e suas descrições na Amazon, totalizando aproximadamente 3.218.856 registros (soma de trn.json e tst.json). Para este projeto, focamos nos campos:

* **title**: Título do produto.
* **content**: Descrição do produto.

**2.2. Exploração e Limpeza dos Dados**

Processamos os datasets de treinamento e teste, realizando as seguintes etapas:

* **Processando o dataset de treinamento** **(trn.json)**
  + Total de registros no arquivo: 2.248.619
  + Registros com 'title' e 'content': 2.248.619
  + Registros após criar DataFrame: 2.248.619
  + Registros após remover nulos: 2.248.619
  + Registros após remover strings vazias: 1.390.403
  + Registros após remover duplicatas: 1.367.131
  + **Total de registros após processamento: 1.367.131**
* **Processando o dataset de teste (**tst.json**):**
  + Total de registros no arquivo: 970.237
  + Registros com 'title' e 'content': 970.237
  + Registros após criar DataFrame: 970.237
  + Registros após remover nulos: 970.237
  + Registros após remover strings vazias: 599.743
  + Registros após remover duplicatas: 593.744
  + Total de registros após processamento: 593.744

**Resumo das etapas de limpeza:**

* **Remoção de valores nulos ou vazios:** Eliminamos registros com campos vazios em title ou content, reduzindo **de 3.218.856** para **1.990.146** registros (soma de trn.json com **1.390.403** e tst.json com **599.743** registros)**.**
* **Remoção de duplicatas**: Eliminamos registros duplicados para evitar redundâncias, resultando em **1.960.875** registros após a limpeza (soma de trn.json com **1.367.131** e tst.json com **593.744** registros).

**2.3. Pré-processamento**

Aplicamos uma função de normalização para:

* **Remover caracteres de controle**: Eliminamos caracteres não imprimíveis.
* **Remover espaços extras**: Padronizamos os espaços em branco.
* **Manter capitalização e pontuação**: Preservamos esses elementos para compatibilidade com o modelo,

Exemplo da função normalize\_text utilizada:

|  |
| --- |
| import re  import unicodedata  **# *Função para normalizar o texto***      def normalize\_text(text):  **# *Remove caracteres de controle***          text = ''.join(ch for ch in text if unicodedata.category(ch)[0] != 'C')  **# *Remove espaços extras***          text = re.sub(r'\s+', ' ', text).strip()          return text  **# *Aplica a função de normalização nas colunas 'title' e 'content'***      df['title'] = df['title'].apply(normalize\_text)      df['content'] = df['content'].apply(normalize\_text) |

**2.4. Preparação dos Prompts para Fine-Tuning**

Criamos pares de pergunta e resposta no seguinte formato:

* **Pergunta**: "What is 'Título do Produto'?"
* **Resposta**: "Descrição do produto correspondente."

|  |
| --- |
| df['instruction'] = "Answer the question based on the product title."  df['input'] = df['title'].apply(lambda x: f"What is '{x}'?")  df['output'] = df['content'] |

**3. Escolha e Configuração do Modelo Base**

**3.1. Critérios de Seleção do Modelo**

* **Tarefa Alvo**: Geração de respostas baseadas em perguntas sobre títulos de produtos.
* **Recursos Computacionais**: Limitações de hardware, como memória GPU
* **Licenciamento e Uso Comercial**: Conformidade com as restrições de uso dos modelos.

**3.2. Modelos Considerados**

* **BERT**: Ótimo para compreensão de linguagem, mas não ideal para geração de texto.
* **Llama**: Adequado para geração de texto
* **Mistral**: Compatível com geração de texto e recursos disponíveis.
* **GPT-2:** Modelo de linguagem da OpenAI, disponível em tamanhos menores

**3.3. Modelo Escolhido e Justificativa**

**Modelo Escolhido:** GPT-2

**Justificativa**:

* **Adequação à Tarefa**: Projetado para geração de texto.
* **Compatibilidade**: Pode ser treinado em GPUs com memória limitada.
* **Licenciamento**: Licença permissiva para uso educacional e de pesquisa.
* **Desempenho**: Oferece bom desempenho para tarefas de geração de texto.

**Modelo Escolhido:** LLaMA 3-8B

**Justificativa**:

Após o sucesso parcial das execuções anteriores com o GPT-2 , surgiu a demanda de testar outro modelo, neste caso o LLaMA 3-8B, mas em um cenário extremamente restritivo em termos de recursos, usando quantização em 4 bits e LoRA.

* **Adequação à Tarefa**: Projetado para geração de texto.
* **Compatibilidade**: Pode ser treinado em GPUs com memória limitada.
* **Licenciamento**: Licença permissiva para uso educacional e de pesquisa.
* **Desempenho**: Oferece bom desempenho para tarefas de geração de texto.

**4. Configuração do Treinamento e Fine-Tuning**

**4.1. Ambiente de Treinamento**

* **Hardware**:
  + **GPU**: Mínimo de 12 GB de VRAM.
  + **Memória RAM**: Suficiente para processar o dataset.
* **Software**:
* **Sistema Operacional**: Linux (Google Colab).
* **Bibliotecas**
  + **Python 3.x**
  + **PyTorch**
  + **Transformers (Hugging Face)**
  + **Datasets**
  + **Pandas, NumPy, etc.**

**4.2. Configuração dos Hiperparâmetros**

|  |  |
| --- | --- |
| **Parâmetro** | **Valor** |
| Taxa de Aprendizado | 5e-5 |
| Número de Épocas | Variável por execução |
| Tamanho do Lote | Variável por execução |
| Gradient Accumulation | Variável por execução |
| Otimizador | AdamW |
| Weight Decay | 0.01 |
| Scheduler | Linear |
| Warmup Steps | 500 |
| Seed | 42 |
| Uso de FP16 | Variável por execução |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  |  |

**Justificativa**:

* **Taxa de Aprendizado**: Valor padrão ajustado para estabilidade.
* **Número de Épocas** Baseado no tamanho do dataset e objetivos.
* **Tamanho do Lote**: Ajustado conforme os recursos
* **Uso de FP16**: Reduz consumo de memória e acelera o treinamento.

**4.3. Execução do Fine-Tuning**

* **Processo**:
  + Carregamento do dataset preparado.
  + Configuração do GPT-2/ LLaMA 3-8B para fine-tuning.
  + Treinamento com monitoramento de métricas.
* **Dificuldades**:
  + Limitações de memória: Ajustes no tamanho do lote e uso de FP16 para o GPT-2 e BF16 para o LLaMA 3-8B.
  + Tempo de treinamento significativo.

**4.4. Monitoramento e Salvamento de Checkpoints**

* **Monitoramento utilizado no treinamento com o GPT-2**:
  + Uso do TensorBoard para acompanhar métricas.
  + Registro de métricas a cada 100 passos.
* **Salvamento de Checkpoints no treinamento com o GPT-2**:
  + Salvamos a cada 200 passos
  + Manutenção dos 3 últimos checkpoints para economizar espaço.

**5. Documentação dos Resultados e Avaliação de Cada Execução**

**5.1. Estratégia de Treinamento**

* **Redução do Dataset**: Para o GPT-2 utilizamos 100.000 registros de treinamento e 10.000 de validação. Já para o treinamento utilizando LLaMA 3-8B utilizamos um numero menor de registros: 5000 para treinamento e 1.000 registros de validação.
* **Múltiplas Execuções**: Para o modelo GPT-2 realizamos 4 ( Quatro ) execuções com ajustes nos hiperparâmetros. Para o modelo LLaMA 3-8B realizamos 1 (Uma) execução.
* **Documentação Detalhada**: Registro das métricas e observações.

**5.2. Monitoramento do Treinamento**

* **Perda de Treinamento (train\_loss)**
* **Perda de Validação (eval\_loss)**
* **Taxa de Aprendizado**
* **Tempo por Época**
* **Uso de Memória GPU e CPU**

**5.3. Avaliação do Modelo**

**5.3.1 Primeira Execução - Experimentos com Datasets Pequenos**

* **Objetivo**: Avaliar o modelo com apenas 10 registros de treinamento e 5 de validação.
* **Configurações**:
  + **Épocas**: 5
  + **Tamanho do Lote**: 1
  + **Gradient Accumulation Steps**: 1
  + **Taxa de Aprendizado**: 5e-5
* **Observações** 
  + Métricas iniciais vazias devido ao número insuficiente de passos.
  + Ajustes em logging\_steps, evaluation\_strategy e save\_strategy.
  + O modelo não aprendeu padrões significativos.
* **Análise das Respostas Geradas**
  + Resposta repetitiva e irrelevante.
  + Indica falta de aprendizado efetivo.
  + Exemplode Resposta:

|  |
| --- |
| **Resposta gerada:**  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.  ***### Instruction:***  Answer the question based on the product title.  ***### Input:***  What is 'Girls Ballet Tutu Neon Pink'?  ***### Response:***  Hi, I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a girl. I'm a |

* **Conclusões :**
  + Necessidade de datasets maiores para um treinamento eficaz.
  + Impacto limitado de ajustes hiperparâmetros em datasets pequenos.
  + Ação futura: Aumentar o dataset para pelo menos 10.000 registros.

**5.3.2 Segunda Execução - Sem Otimizações Específicas**

* **Objetivo**: Avaliar o modelo com 10.000 registros de treinamento sem otimizações específicas.
* **Configurações**:
  + **Épocas:** 5
  + **Tamanho do Lote:** 8
  + **Gradient Accumulation Steps:** 1
  + **Taxa de Aprendizado**: 5e-5
  + **Sem regularização** (weight\_decay = 0)
  + **FP16:** desativado
* **Observações:**
  + Redução inicial da perda, mas estabilização precoce.
  + Possível overfitting identificado.
* **Análise da Resposta Gerada:**
  + Relevância parcial, mas com incoerências e repetições.
  + Qualidade geral média a baixa.
  + Exemplode Resposta:

|  |
| --- |
| **Resposta Gerada:**  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.  ***### Instruction:***  Answer the question based on the product title.  ***### Input:***  What is 'Girls Ballet Tutu Neon Pink'?  ***### Response:***  The girls' tutu is a fun and colorful ensemble that is perfect for any occasion. The ensemble is made of a soft, soft fabric that will keep your child entertained and entertained. It is also great for a party or a special occasion! The tutus are made from a durable, stretchy fabric and are perfect to wear for parties, parties or any special occasions. Each tutum measures approximately 3" long and measures 8" wide. Made of 100% cotton and 100 percent polyester. |

* **Conclusões:**
* Modelo captou características gerais, mas faltou qualidade nas respostas.
* Necessidade de otimizações específicas e datasets maiores.
* Ajustes recomendados: Ativar do\_sample durante a geração, utilizar datasets maiores, aplicar técnicas de regularização e revisar parâmetros de geração.

**5.3.3. Terceira Execução - Com Otimizações Específicas**

* **Configurações Adicionais:**
  + Dataset ampliado para 100.000 registros.
  + Aplicação de dropout nas embeddings, conexões residuais e atenção.
  + Uso de FP16 ativado.
  + Ajustes nos hiperparâmetros, incluindo gradient\_accumulation\_steps e weight\_decay.
* **Observações:**
  + Redução significativa das perdas de treinamento e validação.
  + Monitoramento eficaz com o TensorBoard.
* **Análise da Resposta Gerada:**
  + Relevância parcial, mas com incoerências e termos sem sentido.
  + Melhorias em relação às execuções anteriores, mas ainda insuficiente.

|  |
| --- |
| **Resposta Gerada:**  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request**.**  ***### Instruction:***  Answer the question based on the product title.  ***### Input:***  What is 'Girls Ballet Tutu Neon Pink'?  ***### Response:***  Girls' Tutus. Girls' tutus come in neon colors. They're not cute. And they're NOT the prettiest. So make them look sexy! Each tutu comes with a coordinating wig, tutela, and a matching tutelab. The fun is in the tutels. There is a cute tuteloab on each tuturi and it is easy to follow the directions. You can use the included scissors to cut the perfect tutelleab from the base of each of the |

* **Conclusões** 
  + As otimizações contribuíram para melhorias, mas o modelo ainda não atingiu a qualidade desejada.
  + Recomendações: Revisar a qualidade dos dados, experimentar modelos maiores e refinar hiperparâmetros.

**5.3.4. Quarta Execução - Versão Melhorada do Terceiro Notebook com Otimizações Específicas**

* **Configurações Adicionais:**
  + Implementação de EarlyStoppingCallback.
  + Ajuste do lr\_scheduler\_type para 'cosine'.
* **Observações:**
  + Early stopping acionado, evitando treinamento desnecessário.
  + Melhorias significativas na qualidade das respostas.
* **Análise da Resposta Gerada**
  + Resposta pertinente e informativa.
  + Terminação abrupta da resposta, indicando necessidade de ajuste em max\_new\_tokens.

|  |
| --- |
| **Resposta Gerada:**  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.  ***### Instruction:***  Answer the question based on the product title.  ***### Input:***  What is 'Girls Ballet Tutu Neon Pink'?  ***### Response:***  A high quality Neon pink tutu. Perfect for girls who want to express their creativity. The tutus are lightweight and comfortable. Easy to wash and care for. One size fits most. Made in USA. Available in Neon color. Measures 3" wide x 4" tall. Machine washable. Comes in a resealable plastic bag with instructions. Recommended for children ages 3 months and up. Colors may vary slightly from picture. Imported. Package of 2. Set includes one tutu ( |

* **Conclusões**
  + Otimizações resultaram em respostas de melhor qualidade.
  + Recomendações: Ajustar parâmetros de geração, testar com outros exemplos e considerar modelos maiores.

**5.3.5. Quinta Execução - Teste com LLaMA 3-8B Quantizado em 4 Bits (LoRA + Estratégia Econômica)**

* **Configurações do Teste:**
  + Modelo: unsloth/llama-3-8b-bnb-4bit
  + Quantização: 4 bits (para reduzir uso de memória)
  + Técnicas: LoRA (Low-Rank Adapters) para treinamento eficiente
  + Dataset: Ao invés de treinar com todo o conjunto de dados, utilizamos apenas 5.000 registros para treinamento e 1.000 para teste.
* **Hiperparâmetros:**
  + num\_train\_epochs=1 (apenas 1 época, visando demonstrar a viabilidade do treinamento)
  + per\_device\_train\_batch\_size=1
  + gradient\_accumulation\_steps=8 (simulando um batch efetivo maior sem estourar memória)
  + evaluation\_strategy="no" e save\_strategy="no" para acelerar o treinamento
  + learning\_rate=2e-4
  + bf16=True (uso de BFloat16 em GPUs A100 para reduzir memória)
  + optim="adamw\_8bit" (otimizador em 8 bits, reduzindo ainda mais o consumo de memória)

**Observações:**

* O objetivo desta execução não era obter a melhor qualidade de respostas, mas sim demonstrar a possibilidade de treinar um modelo LLaMA 3-8B em condições extremamente limitadas de GPU.
* O treinamento levou cerca de 37 minutos para 625 passos, com uma perda final próxima de 1.7166, considerando apenas uma época e um dataset reduzido.
* **Análise da Resposta Gerada**
  + Resposta obtida, embora simples e genérica, já demonstra coerência e relevância ao tema (um tutu de balé cor de rosa).
  + Em comparação com as execuções iniciais, nota-se que, mesmo com recursos reduzidos e apenas uma época de treinamento, o modelo LLaMA 3-8B em 4 bits + LoRA foi capaz de produzir uma resposta minimamente útil.

|  |
| --- |
| **Resposta Gerada:**  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.  ***### Instruction:***  Answer the question based on the product title.  ***### Input:***  What is 'Girls Ballet Tutu Neon Pink'?  ***### Response:***  This pink tutu is perfect for any ballerina. It is made from soft and stretchy spandex with a layer of tulle. It is easy to put on and off and is very comfortable. The elastic waistband is adjustable for a |

* **Conclusões**
  + Foi possível treinar um modelo LLaMA 3-8B quantizado em 4 bits com LoRA, utilizando apenas 5.000 registros e configurações extremamente restritivas.
  + Mesmo com as limitações, a resposta gerada foi coerente com a questão apresentada.
  + Ajustes adicionais (mais épocas, mais dados, parâmetros de geração, etc.) poderiam melhorar a qualidade, mas o objetivo principal de testar a viabilidade em ambiente limitado foi atingido

**5.4. Conclusões das Execuções**

**A conclusão das execuções pode ser resumida da seguinte forma:**

* Impacto das Otimizações: Ajustes nos hiperparâmetros, uso de técnicas como LoRA, early stopping e FP16/quantização demonstraram melhorias significativas no desempenho do modelo em relação às execuções iniciais. Com cada rodada de testes, tornou-se evidente que pequenas modificações nos parâmetros de treinamento e nas técnicas de fine-tuning podem levar a respostas mais coerentes e relevantes.
* Limitações de Recursos: A dificuldade em treinar modelos maiores como LLaMA ou aumentar drasticamente o conjunto de dados resultou da restrição computacional. A impossibilidade de utilizar GPUs mais potentes ou instâncias com mais memória limitou a exploração de arquiteturas mais complexas ou rodadas mais extensas de treinamento, restringindo o potencial desempenho do modelo.
* **Aprendizados:**
  + Ajustes Finos e Otimizações: Os resultados confirmam que ajustar hiperparâmetros (taxa de aprendizado, gradiente acumulado, tamanho do lote, entre outros) e aplicar técnicas de compactação ou quantização são fundamentais para obter resultados satisfatórios, especialmente em ambientes de recursos limitados.
  + Qualidade e Quantidade dos Dados: Um dos fatores mais determinantes para o sucesso do modelo foi a qualidade e a quantidade do dataset. Mais dados e dados mais limpos levam a um melhor aprendizado e, consequentemente, a respostas mais úteis e coerentes.
  + Desafios Computacionais: A experiência evidenciou que recursos computacionais limitados (tanto em termos de GPU quanto de tempo e memória) representam um obstáculo significativo. Para atingir resultados de ponta, é necessário acesso a recursos mais robustos ou a implementação de estratégias de otimização ainda mais agressivas (como quantizações mais complexas, reduções de tamanho de modelos ou estratégias como LoRA).

**Em resumo, as execuções mostraram que, embora seja possível produzir modelos funcionais em ambientes restritivos, o desempenho ideal depende tanto do controle preciso dos hiperparâmetros e técnicas de otimização quanto da disponibilidade de recursos computacionais adequados e dados de boa qualidade.**

**6. Desenvolvimento da Interface de Usuário**

**6.1. Implementação da Interface de Linha de Comando**

Desenvolvemos uma interface de linha de comando para interagir com o modelo:

* O usuário insere uma pergunta.
* O modelo gera uma resposta baseada na pergunta e no contexto aprendido.

**6.2. Configuração do Modelo para Geração de Respostas**

Implementamos a função generate\_response que:

* Formata a entrada do usuário.
* Utiliza o modelo treinado para gerar a resposta.
* Decodifica e retorna a resposta ao usuário.

**Exemplo de Implementação**:

|  |
| --- |
| ***def generate\_response(user\_question):***  ***# Formata o input do usuário***  input\_text = alpaca\_prompt.format(  "Answer the question based on the product title.",  user\_question  , ""  )  ***# Tokeniza a entrada***  inputs = tokenizer.encode(input\_text, return\_tensors='pt').to(device)  ***# Gera a resposta***  outputs = model.generate(inputs, max\_new\_tokens=100)  ***# Decodifica a resposta***  generated\_text = tokenizer.decode(outputs[0], skip\_special\_tokens=True)  ***# Extrai a resposta***  return generated\_text.split("### Response:")[1].strip() |

**7. Documentação e Conclusões Gerais**

**7.1. Conclusões**

* **Resultados Alcançados:** Desenvolvemos um modelo capaz de responder a perguntas sobre produtos da Amazon de forma relevante e informativa, com melhorias significativas na quarta execução.
* **Limitações Enfrentadas:** Restrições computacionais impediram a experimentação com modelos maiores e ajustes mais agressivos.
* **Aprendizados:** Evidenciamos a importância do ajuste fino de hiperparâmetros, do uso eficiente dos recursos e da qualidade dos dados de treinamento.

**7.2. Trabalhos Futuros e Melhorias**

* **Implementar as Sugestões Identificadas:** Realizar novas execuções com as alterações propostas, caso recursos permitam.
* **Explorar Modelos Maiores:** Utilizar modelos como o GPT-2 Medium ou Large para potencialmente melhorar o desempenho.
* **Aumentar a Quantidade de Dados:** Ampliar os dados de treinamento, validação e teste para aprimorar a capacidade de generalização.
* **Implementar Técnicas de Compressão ou Quantização:**
  + **Compressão de Modelos:** Reduzir o tamanho e a complexidade para facilitar a implantação em ambientes com recursos limitados.
  + **Quantização:** Diminuir a precisão dos pesos e ativações do modelo, economizando memória e aumentando a eficiência computacional.
* **Desenvolver uma Interface de Usuário:** Implementar uma interface web para facilitar o acesso ao modelo.
* **Avaliação Quantitativa e Qualitativa:**
  + **Métricas Automáticas:** Implementar métricas como ROUGE e BLEU para avaliação quantitativa.
  + **Avaliação Qualitativa:** Realizar análises mais abrangentes com diferentes tipos de prompts.

**8. Referências**

* **Documentação do Hugging Face Transformers**: Wolf, T., et al. (2020). *Transformers: State-of-the-Art Natural Language Processing*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Disponível em: <https://huggingface.co/docs/transformers/>
* **Dataset AmazonTitles-1.3MM disponível em**

**9. Anexos**

**9.1. Código-Fonte**

Github

**9.2. Logs de Treinamento**

* **Treinamento Segunda Execução**

|  |
| --- |
| **[6250/6250 14:16, Epoch 5/5]**  **Época Training Loss Validation Loss**  **1 2.895400 2.418449**  **2 2.382500 2.396800**  **3 2.304600 2.392149**  **4 2.131900 2.392719**  **5 2.272600 2.396315** |

* **Treinamento Terceira Execução**

|  |
| --- |
| **[31250/31250 2:09:44, Epoch 10/10]**  **Step Training Loss Validation Loss**  **500 2.540700 2.424012**  **1000 2.483300 2.386914**  **...**  **31000 2.132100 2.197310** |

Gráfico

Descrição gerada automaticamente

* **Treinamento Quarta Execução**

|  |
| --- |
| **[24500/31250 1:42:34 < 28:15, 3.98 it/s, Epoch 7/10]**  **Step Training Loss Validation Loss**  **500 2.540400 2.423739**  **1000 2.482800 2.386429**  **...**  **24500 2.135200 2.198745** |

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

* **Treinamento Quinta Execução**

|  |
| --- |
| **Num examples = 5,000 | Num Epochs = 1**  **Batch size per device = 1 | Gradient Accumulation steps = 8**  **Total batch size = 8 | Total steps = 625**  **Number of trainable parameters = 41,943,040**  **[625/625 37:21, Epoch 1/1]**  **Step Training Loss**  **50 1.823300**  **100 1.708600**  **150 1.726500**  **200 1.711400**  **250 1.665800**  **300 1.673000**  **350 1.720600**  **400 1.730000**  **450 1.721300**  **500 1.714200**  **550 1.670300**  **600 1.709300**  **TrainOutput(global\_step=625, training\_loss=1.7166249145507813, metrics={'train\_runtime': 2256.1676, 'train\_samples\_per\_second': 2.216, 'train\_steps\_per\_second': 0.277, 'total\_flos': 3.782108225748173e+16, 'train\_loss': 1.7166249145507813, 'epoch': 1.0})** |

**Ao final, a perda média se estabilizou em torno de 1.7166 após 625 passos.**