Exercício 3

Suponha que você tenha uma tarefa específica de processamento de linguagem natural (PLN) para a qual deseja criar um modelo de alto desempenho. Você tem acesso a um modelo de linguagem pré-treinado e deseja realizar o fine-tuning desse modelo para sua tarefa específica.

1. Escolha uma tarefa de NLP específica, de sua preferência, para a qual você deseja adaptar o modelo de linguagem, como classificação de sentimentos, resumo de texto, geração de texto ou qualquer outra tarefa.

Vou escolher **resumo de texto** como a tarefa específica de NLP para a qual faremos o fine-tuning. O resumo de texto é uma tarefa de geração de texto que visa criar um resumo conciso e informativo de um texto maior.

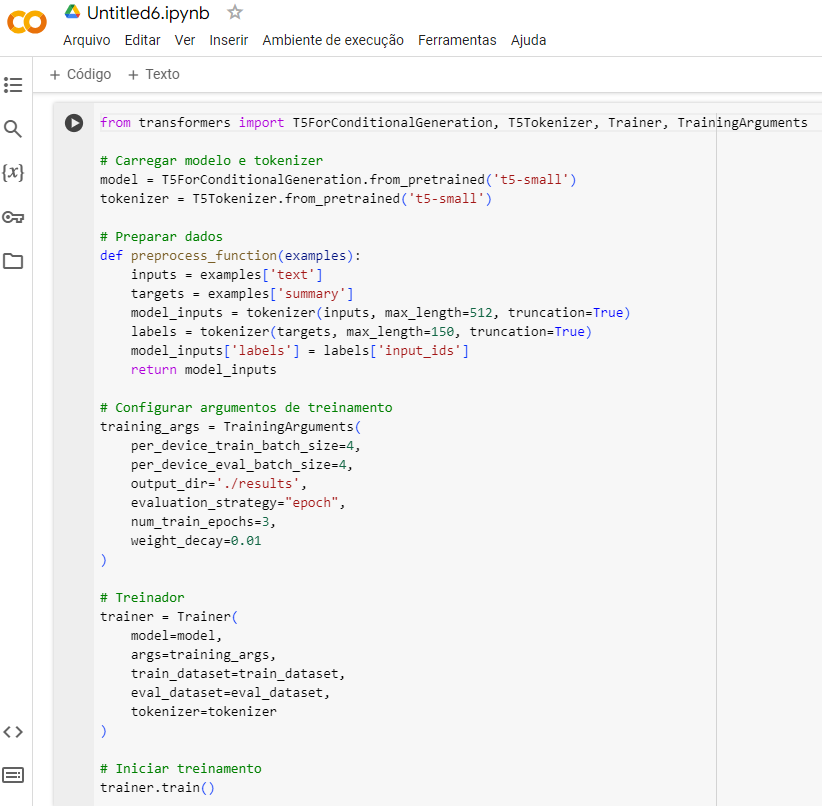
1. Como você faria a coleta e processamento desses dados?

**Coleta e Processamento de Dados**

1. **Coleta de Dados:**
   * **Fontes de Dados:** Para a tarefa de resumo, você pode usar conjuntos de dados já existentes, como o CNN/Daily Mail, XSum, ou datasets de resumos de artigos científicos como o arXiv. Alternativamente, você pode coletar dados de notícias, artigos científicos ou posts de blogs e criar seus próprios resumos.
   * **Métodos de Coleta:** Dependendo da fonte, você pode usar APIs (por exemplo, para coletar artigos de notícias), crawlers web (para blogs e sites de notícias) ou datasets públicos.
2. **Processamento de Dados:**
   * **Limpeza:** Remova informações irrelevantes, formatação excessiva, e normalize o texto (por exemplo, removendo caracteres especiais).
   * **Tokenização:** Divida o texto em tokens (palavras ou sub-palavras). Para modelos baseados em Transformers, como BERT ou GPT, a tokenização é feita de acordo com o vocabulário do modelo.
   * **Divisão:** Separe os dados em conjuntos de treinamento, validação e teste. Uma divisão comum é 80% treinamento, 10% validação e 10% teste.
   * **Resumo de Texto:** Se o dataset não tiver resumos, você precisará gerar resumos manualmente ou usar algoritmos de resumo automático para criar uma versão resumida dos textos.
3. Descreva o processo de fine-tuning do modelo de linguagem pré-treinado para a tarefa específica.

**Processo de Fine-Tuning do Modelo de Linguagem Pré-Treinado**

1. **Escolha do Modelo Pré-Treinado:**
   * Modelos como BERT (para tarefas de compreensão) e T5 ou GPT-3 (para tarefas de geração) são comuns para resumo de texto. Vou assumir que você escolheu T5, um modelo Transformer que pode ser adaptado para tarefas de geração de texto.
2. **Preparação dos Dados:**
   * **Formato de Entrada:** Prepare os dados no formato de entrada do modelo. Para T5, isso geralmente significa formatar o texto como pares de entrada e saída, onde a entrada é o texto a ser resumido e a saída é o resumo correspondente.
3. **Configuração do Fine-Tuning:**
   * **Configuração do Treinamento:** Defina hiperparâmetros como taxa de aprendizado, número de épocas e tamanho do lote.
   * **Treinamento:** Use um framework como Hugging Face Transformers para carregar o modelo T5 pré-treinado e realizar o fine-tuning com seus dados de resumo.



1. Como você avaliaria o desempenho do modelo fine-tuned, especifique as métricas.

**Avaliação do Desempenho do Modelo Fine-Tuned**

1. **Métricas Comuns para Resumo de Texto:**
   * **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** Métrica que compara a sobreposição de n-gramas entre o resumo gerado e o resumo de referência. ROUGE-N (para n-gramas), ROUGE-L (para a maior sequência comum) são comuns.
   * **BLEU (Bilingual Evaluation Understudy):** Avalia a precisão dos n-gramas no texto gerado comparado ao texto de referência, embora seja mais usada em tradução automática.



1. Descreva como você implantaria o modelo fine-tuned em um ambiente de produção para realizar previsões em novos dados.

**Implantação do Modelo Fine-Tuned em Ambiente de Produção**

1. **Criação de uma API:**
   * Utilize frameworks como FastAPI ou Flask para criar uma API que recebe texto e retorna o resumo gerado pelo modelo.



**Escalabilidade:**

* **Serviço em Nuvem:** Implante a API em serviços como AWS Lambda, Google Cloud Functions ou Azure Functions para escalabilidade automática.
* **Containers:** Use Docker para empacotar a aplicação e Kubernetes para orquestração se a carga de trabalho for alta.

1. EXTRA - você pode explorar o ajuste de hiperparâmetros do modelo fine-tuned, como o tamanho da rede, a taxa de aprendizado e o número de épocas de treinamento, para otimizar o desempenho.

**Ajuste de Hiperparâmetros (Extra)**

1. **Taxa de Aprendizado:**
   * Experimente diferentes taxas de aprendizado (por exemplo, 1e-5, 3e-5, 5e-5) para encontrar a mais adequada para o seu problema específico.
2. **Número de Épocas:**
   * Comece com um número moderado de épocas (por exemplo, 3-5) e ajuste conforme necessário. Use validação para evitar overfitting.
3. **Tamanho do Lote:**
   * Ajuste o tamanho do lote para equilibrar a memória e a performance. Valores comuns são 8, 16 ou 32, dependendo da capacidade de memória da GPU.
4. **Tamanho da Rede:**
   * Pode-se experimentar diferentes variantes do modelo, como T5-small, T5-base, e T5-large. Modelos maiores podem ter melhor desempenho, mas exigem mais recursos computacionais.
5. **Ajuste Fino de Hiperparâmetros:**
   * Utilize técnicas como grid search ou random search para encontrar a combinação ideal de hiperparâmetros. Ferramentas como Optuna ou Ray Tune podem ajudar nesse processo.