**1. Preparação dos Dados**

* **Criação do Dataset**: Cria um conjunto de dados balanceado com frases de sentimentos positivos e negativos, juntamente com seus rótulos correspondentes.
* **Criação do Vocabulário**: Gera um vocabulário a partir do conjunto de dados e atribua um número inteiro único a cada palavra.
* **Codificação (Encoding)**: Converte os dados textuais em sequências numéricas com base no vocabulário.
* **Padding**: Padroniza as sequências para o mesmo comprimento, garantindo compatibilidade com o processamento em lote no LSTM.

**2. Codificação dos Rótulos**

* Codifica os rótulos dos sentimentos ("positivo" e "negativo") em valores inteiros para classificação binária.

**3. Divisão dos Dados**

* Divide os dados codificados e preenchidos em conjuntos de treinamento e teste usando train\_test\_split.

**4. Conversão para Tensores**

* Converte os dados e os rótulos em tensores do PyTorch para entrada no modelo.

**5. Definição do Modelo**

* **Camada de Embedding**: Mapeia palavras (números inteiros) para vetores densos, capturando relações semânticas.
* **Camada LSTM**: Processa os dados sequenciais e aprende dependências temporais.
* **Camada Totalmente Conectada (Fully Connected)**: Mapeia a saída do estado oculto do LSTM para o resultado binário (sentimento positivo/negativo).
* **Ativação Sigmoide**: Produz probabilidades para a classificação binária.

**6. Inicialização do Modelo**

* Inicializa o modelo SentimentLSTM com hiperparâmetros adequados, como tamanho do vocabulário, dimensões do embedding, dimensões ocultas e dimensões de saída.
* Configura o otimizador (Adam) e a função de perda (BCELoss).

**7. Treinamento**

* Realiza um loop pelos *epochs*:
  + **Forward Pass**: Calcula as previsões no conjunto de treinamento.
  + **Cálculo da Perda**: usa a perda binária de entropia cruzada (*binary cross-entropy loss*) para medir a diferença entre as previsões e os rótulos reais.
  + **Backpropagation**: Calcula os gradientes e atualize os parâmetros do modelo.
  + **Cálculo de Precisão**: Computa a precisão do treinamento arredondando as previsões para valores binários.

**8. Avaliação**

* Coloca o modelo em modo de avaliação e faça previsões no conjunto de teste.
* Calcula a precisão no teste para avaliar o desempenho do modelo.

**9. Saída**

* Exibe a perda e a precisão do treinamento para cada *epoch*.
* Exibe a precisão final no teste para avaliar a capacidade de generalização do modelo.