EP 2– Tradução Automática de Baixo Recurso

Implementação, Treinamento e Avaliação de Modelos Baseados em LLM para Tradução Português ↔ Tupi Antigo

Entrega improrrogável: 07/12/2025

Este EP pode ser feito em dupla

O objetivo deste exercício programa (EP) é treinar e avaliar modelos de tradução automática em um cenário de baixo recurso, utilizando um córpus paralelo Português-Tupi Antigo e abordagens baseadas em modelos de linguagem de grande porte (LLMs). Vocês deverão produzir experimentos completos nos regimes *zero-shot learning* e *few-shot learning*, em ambas as direções de tradução.

Quando se fala de tarefas de processamento de texto utilizando a tecnologia de geração de texto, existem dois modos básicos de operação. O modo *zero-shot* (sem dicas) indica que a tarefa tem que ser realizada apenas por solicitação, sem apresentar qualquer exemplo. É a tarefa que o programa realiza só com o treinamento genérico a que foi submetido. Já o modo *few-shot* (poucas dicas) indica que alguns exemplos, em geral um número não muito grande, deverão ser fornecidos antes da realização da tarefa. Nesse caso, a gente diz que o modelo foi *refinado* (*fine-tuned*) para a realização da tarefa. Nesse exercício vamos tratar de gerar tradutores tanto no modo *zero-shot* quanto no modo *few-shot*. [1]. E vamos comparar as saídas dos dois modos.

Não existem medidas automáticas muito boas para se avaliar uma tradução. Mas algumas medidas são usadas tradicionalmente, e devem servir para comparar a qualidade de 2 métodos diferentes. Neste exercício vamos utilizar as seguintes medidas:

- **BLEU (Bilingual Evaluation Understudy)** avalia a qualidade de textos traduzidos automaticamente de um idioma para outro. As pontuações são calculadas para segmentos traduzidos individualmente — geralmente frases — comparando seus n-gramas com um conjunto de traduções de referência de boa qualidade (padrão ouro). Essas pontuações são então calculadas em média para todo o córpus, resultando num número entre 0 e 1 que indica o quão semelhante o texto candidato é aos textos de referência, com valores mais próximos de 1 representando textos mais semelhantes. [2]

- **chrF (CHaRacter-level F-score)** é uma métrica para avaliação de tradução automática que calcula a similaridade entre uma tradução automática e uma tradução de referência usando n-gramas de caracteres, e não n-gramas de palavras. Métricas baseadas em n-gramas de palavras são especialmente problemáticas para línguas com morfologia complexa.

[1] É importante dizer que recentemente *few-shot* se tornou popular com os modelos decoder tipo GPT onde ao invés de realizar o *fine-tuning*, o modelo recebe exemplos diretamente no prompt, isto é, não há propagação de gradientes e o modelo aprende com base no contexto (*In-Context learning*)

[2] Poucas traduções atingirão uma pontuação de 1, pois isso indicaria que o texto candidato é idêntico a uma das traduções de referência.

- Variantes de BLEU (VERT, ROUGE) e chrF (crhF++) existem e podem ser usadas para comparar a versões do seu tradutor.

Detalharemos as medidas abaixo, mas elas provavelmente existem como bibliotecas em algum pacote ou código na internet. Você é incentivado a usar versões de biblioteca em vez de reimplementar do zero.

BLEU

A medida pontuação BLEU recebe dois argumentos: uma string candidata $\hat{y}$ (em geral gerada pelo tradutor automático) e uma lista de strings de referência $(y^{(1)}, \ldots, y^{(n)})$ de possíveis traduções. Nossos dados só apresentam uma única tradução, então vamos considerar apenas a tradução padrão-ouro y.

Dado um string y, seja $G_n(y)$ o conjunto[3] de seus n-gramas. Dados dois strings s e y, seja a contagem de substrings $C(s, y)$ o número de ocorrências de s como um substring de y. Por exemplo, $C(ab, abcab) = 2$. Seja

$$p_n(\hat{y}; y) = \frac{\sum_{s \in G_n(\hat{y})} \min(C(s, \hat{y}), C(s, y))}{\sum_{s \in G_n(\hat{y})} C(s, \hat{y})}$$

a função que mede o grau de coincidência de n-gramas entre $\hat{y}$ e y.

Seja S o córpus padrão ouro e $\hat{S}$ o córpus gerado na tradução. Para punir as cadeias de strings que são muito curtas, define-se a penalidade de brevidade como:

$$BP(\hat{y}; y) = e^{-max(0, \frac{r}{c} - 1)}$$

onde $c = |\hat{y}^i|$ é o comprimento da tradução e $r = |y^i|$ é o comprimento da referência. A penalidade pela brevidade varia para cada uma das sentenças de um córpus.

Não há uma única definição de BLEU, mas uma família inteira delas, parametrizada pelo vetor de ponderação $w = (w_1, w_2, \ldots)$ que é uma distribuição de probabilidade discreta, $w_i \geq 0$, $\sum w_i = 1$.

$$BLEU(\hat{y}; y) = BP(\hat{y}; y) \cdot e^{\sum_{i=1}^{\infty} w_n \ln p_n(\hat{y}; y)}.$$

Tipicamente, $w_1 = w_2 = w_3 = w_4 = \frac{1}{4}$. Ou seja, a medida p_n é calculada apenas para $n \leq 4$. A medida $BLEU(\hat{S}, S)$ para um par de córpus tradução-referência $(\hat{S}, S)$ é a média das medidas $BLEU(\hat{y}; y)$ para todo par $(\hat{y}, y) \in (\hat{S}, S)$.

$$BLEU(\hat{S}; S) = \frac{1}{|S|} \sum_{(\hat{y}, y) \in (\hat{S}, S)} BP(\hat{y}; y) \cdot e^{\sum_{i=1}^{\infty} w_n \ln p_n(\hat{y}; y)}.$$

crhF

A fórmula geral para a pontuação CHRF é:

$$\textsc{chrF}_\beta = (1 + \beta^2) \frac{\textsc{chrP} \cdot \textsc{chrR}}{\beta^2 \textsc{chrP} + \textsc{chrPR}} \tag{1}$$

onde $\textsc{chrP}$ e $\textsc{chrR}$ representam a precisão e a cobertura de n-gramas de caracteres, calculadas aritmeticamente pela média em todos os n-gramas:

[3]Por ser conjunto, não admite duplicação.

- chrP é a percentagem de n-gramas na hipótese $\hat{y}$ que têm uma contraparte na referência y;
- chrR é a percentagem de n-gramas de caracteres na referência y que também estão presentes na hipótese $\hat{y}$.

e β é um parâmetro que atribui β vezes mais importância à cobertura do que à precisão — se $\beta = 1$, elas têm a mesma importância. Tipicamente, $\beta = 1$ ou $\beta = 3$.

Descrição Geral da Tarefa

Este EP envolve quatro tarefas principais:

a) Tradução Português → Tupi Antigo em regime **zero-shot**;

b) Tradução Português → Tupi Antigo em regime **few-shot** com *fine-tuning*;

c) Tradução Tupi Antigo → Português em regime **zero-shot**;

d) Tradução Tupi Antigo → Português em regime **few-shot** com *fine-tuning*.

Cada dupla deverá implementar, treinar e avaliar todos os cenários acima, fornecendo métricas numéricas e saídas de exemplo.

Dados

O córpus para esta atividade encontra-se disponível no repositório oficial:

`https://github.com/CalebeRezende/oldtupi_dataset/blob/main/C%C3%B3pia%20de%20portugues-guarani-tupi%20antigo.xlsx`

Vocês deverão realizar todas as etapas de preparação, limpeza, organização e divisão dos dados.

Metodologia

Detalhamos a seguir as principais etapas da metodologia de desenvolvimento de um tradutor, englobando o pré-processamento dos textos, a seleção de um modelo de tradução, a implementação dos modelos no modo zero-shot e few-shot, a realização das medidas e a comparação dos resultados.

Preparação do Córpus

Depois de baixar a planilha com os textos parelelos em Português[4] e Tupi Antigo, você deve separar em sub-córpus de treino, validação e teste. A parte de teste será usada sempre, mas o treino e validação apenas no modo few-shot. Portanto você deve:

[4]Você pode notar que os textos também aparentam estar em um português antigo. Se quiser, pode usar uma LLM para reescrever os textos em português mais moderno. Verifique se isso afeta o desempenho do tradutor.

- realizar leitura e limpeza do arquivo Excel, pois alguns caracteres estranhos podem estar nos textos;
- dividir o córpus em três subconjuntos: treino (70%), validação (15%) e teste (15%);
- documentar as decisões metodológicas de curadoria e reportá-las no relatório a ser entregue.

Cada subconjunto deve ser salvo em arquivo próprio para posterior uso nos experimentos.

Seleção e Justificativa do Modelo

Você deverá selecionar um modelo base pré-treinado adequado ao cenário de baixo recurso, tais como:

- `facebook/mbart-large-50-many-to-many-mmt`;
- `facebook/nllb-200-distilled-600M`;
- modelos do tipo `mt5-small` ou `t5-small`.

A sua escolha deve ter uma justificativa, a qual deve abordar características da arquitetura e do treinamento dos modelos. Apresente a justificativa no relatório a ser entregue.

Implementação Zero-shot

Aqui você deverá ver as instruções do tradutor que baixou e usá-lo diretamente. Veja como pré-selecionar as línguas de origem e destino.

- implementar um pipeline de tradução automática sem qualquer *fine-tuning*;
- aplicar o modelo nas duas direções (PT→TA e TA→PT);
- registrar as traduções do conjunto de teste;
- calcular as métricas **BLEU**, **chrF1** e **chrF3** (e outras, se quiser);
- salvar os resultados em arquivo próprio (`results_zero_shot`) ou em uma tabela que será parte do relatório final.

Implementação Few-shot com Fine-tuning

Aqui você deverá ver as instruções do tradutor que baixou e treiná-lo (*fine-tuning*) com os pares de dados do córpus de treinamento.

- utilizar o córpus para treinamento (*few-shot*);
- treinar o modelo com hiperparâmetros de baixo recurso (p.ex.: `batch size` = 4-8, `lr` = 5e-5);
- empregar *early stopping* com base na perda de validação;
- realizar o mesmo processo para as duas direções de tradução;
- salvar os resultados em `results_few_shot` ou em uma tabela que será parte do relatório final.

Avaliação Comparativa

Em seu relatório deve conter uma matriz comparativa contendo:

- métricas zero-shot vs. few-shot para as duas direções;
- análise qualitativa de 5 a 10 exemplos significativos, tanto positivos quanto negativos;

Entrega

Ao final da atividade, cada dupla deverá entregar:

1. **Repositório de código** contendo todos os scripts utilizados:
2. **Relatório técnico** (6–10 páginas) incluindo:
    - introdução;
    - descrição do córpus;
    - descrição da metodologia zero-shot e few-shot;
    - análise dos resulados métricas automáticas;
    - análise linguística qualitativa;
    - conclusões e limitações.