

# Facebook/OPT-125M-Portuguese LLM



Open in Colab

# 1. Questões

1. **Explicação de conceitos importantes do exercício feito**
2. ~~Técnicas para garantir que a implementação está correta~~
3. ~~Truques de código que funcionaram~~
4. **Problemas e soluções no desenvolvimento**
5. ~~Resultados interessantes/inesperados~~
6. **Uma dúvida "básica" que você ou os colegas possam ter**
7. ~~Um tópico "avançado" para discutirmos~~

## 2. Conceitos

**Perplexity** is closely related to the cross entropy that is directly minimized during training (intrinsic). Another metrics are **BLEU** and **ROUGE** that are more related to, e.g., classification accuracy (extrinsic). **BLEU** is a precision-like score to evaluate the quality of a translated text. **ROUGE** is a recall-like score to evaluate summarized text.

A high perplexity indicates that a language model is worse at predicting the next word in a sequence, implying greater uncertainty in its predictions. Therefore a lower perplexity indicates better performance in predicting the next word in a sequence, reflecting a more accurate understanding of language patterns and context.

- Ou seja, se o seu modelo de linguagem tem perplexidade de  $N$ , significa que ele é igual a um classificador aleatório de  $N$  classes


## 2.1 Conceitos

- por que usar PPL ao invés de entropia cruzada? <https://towardsdatascience.com/perplexity-intuition-and-derivation-105dd481c8f3> ver no chatgpt tbm
- Diferença entre logits e probabilidades
  - logits são valores numéricos não normalizados brutos produzidos pelo modelo na saída da sua última camada
  - probabilidades são valores normalizados dos logits que indicam a probabilidade relativa de cada palavra na sequência ou outro tipo de tarefa, caracterizando uma distribuição de probabilidade
  - Ambas as medidas podem ser usadas para avaliar a qualidade de um modelo de linguagem e para gerar previsões de palavras futuras em uma sequência.

### 3. Problemas e soluções no desenvolvimento

- primeira vez treinando/finetuning um LLM
- **Dataset** do HF é muito bom. Customização deles pra fazer preprocessing e outras coisas
- notebooks/apresentações interessantes
  - **Mirelle** e outros colegas
    - dataset de teste para overfitting
  - **Marcos Vieira**
    - truques de código
    - tempo gasto artigo + notebook (meu caso foi por parecido também, com ~20h de estudo na semana)
  - **Eduardo Oliveira**
    - treino com length de 2048
    - batchs em disco

# 3.1 Problemas e soluções no desenvolvimento

- calcular perplexidade inicial do dataset = número de palavras no vocabulário
- HF facilita demais o uso de transformers, mas é importante ficar atento a outros aspectos do modelo e tokenizador
- usei T4 16Gb no Colab Pro
  - treino demorado
  - +1 epoch = +2h-3h de treino
- no final consegui pegar um A100 pra treinar. ~~40 créditos que valem do que dinheiro~~ 
- teste de overfit com dataset pegando os 100 primeiros textos
- salvar `tokenized dataset` no disco

## 4. Resultados

split train/validation 80/20 (200.000, 50.000)

seq length	epochs	batch size	gpu	time/epoch	val_ppl
256	1	4	T4 16GB	~2h	31.67
512	1	4	T4 16GB	OOM 💣	-
512	1	8	A100 40GB	45min	10.96
512	1+1	8	A100 40GB	colab fail 🤔	-
512	2	8	A100 40GB	~2h	<b>9.67</b> 🏆
1024	1	16	A100 40GB	OOM 💣	-

epoch	train loss	val loss
1	2.571000	2.339543
2	2.327500	2.269327

## 5. Uma dúvida "básica" que você ou os colegas possam ter

- concatenação de sentenças para evitar padding e melhorar uso da gpu? discussão do slack
- não precisaria ajustar o vocab\_size da config do modelo para o novo dataset tokenizado?
- salvar processamento/tokenizer e outras coisas em disco para depois habilitar gpu e treinar - salvei o dataset tokenizado
- como calcular a perplexidade sem treino?
  - vi no notebook do **Thiago Laitz** que ele usa a loss da primeira iteração do finetuning
- como faria pra ajustar a função de custo pra usar ppl?