



Code Presentation

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

Aula 8/9 InPars v1 and v2

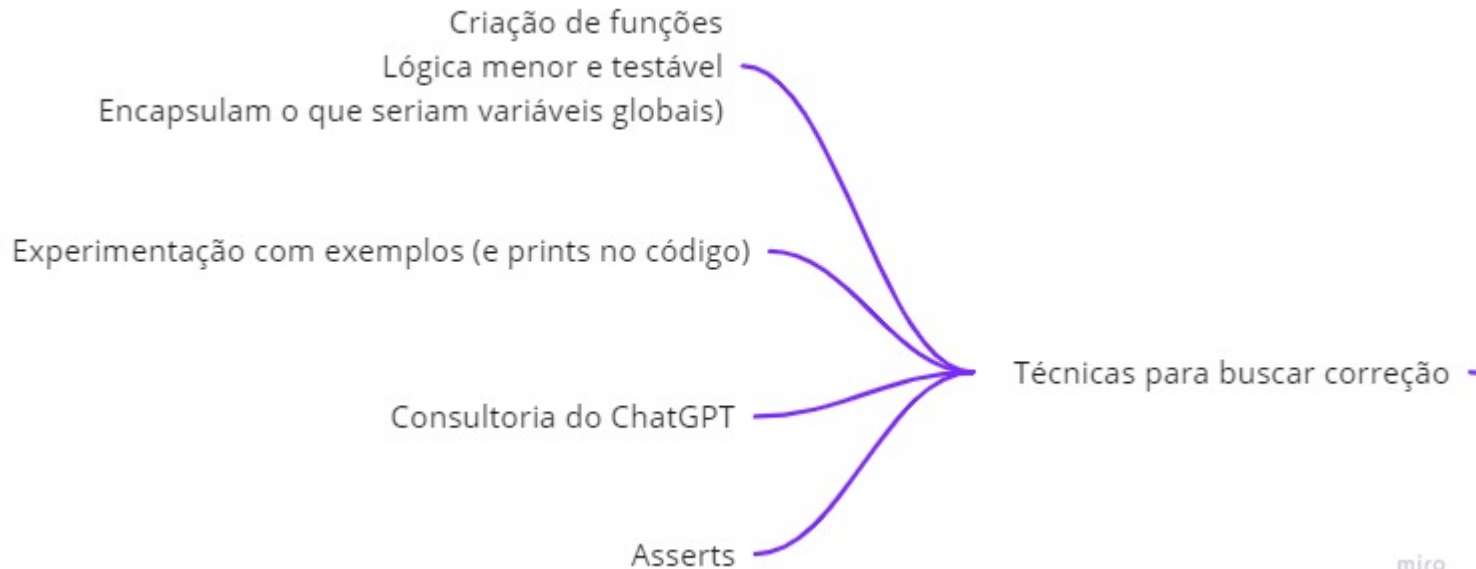
Processo InPars

[\(Ver apresentação do artigo\)](#)

<https://miro.com/app/board/uXjVO->

[OAf1w=?moveToWidget=3458764553067017236&cot=14](https://miro.com/app/board/uXjVO-OAf1w=?moveToWidget=3458764553067017236&cot=14)

Conceitos



Solução: mudei para o EleutherAI/gpt-j-6B

Bom que pude gerar mais queries!

Mas selecionei as 30k mais "relevantes"

Erro ao chamar api do "gpt-3.5-turbo"

*SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate
verify failed: unable to get local issuer certificate (_ssl.c:1129)*

Prompt constante construído com ajuda do chatGPT

Truques do código



```
instrucao = 'Instruction: Based on the text, generate just one question succinctly, '  
instrucao += 'answered by the text, avoiding repeating words. See examples below:\n\n'  
exemplo1 = f'Text: {shortened_text_shot_example_1}\n\nQuestion: {question_shot_example_1}\n\n'  
exemplo2 = f'Text: {shortened_text_shot_example_2}\n\nQuestion: {question_shot_example_2}\n\n'  
exemplo3 = f'Text: {shortened_text_shot_example_3}\n\nQuestion: {question_shot_example_3}\n\n'  
texto_a_completar = 'Text: {context}\n\nQuestion: '
```

Técnica para evitar palavras frequentes:

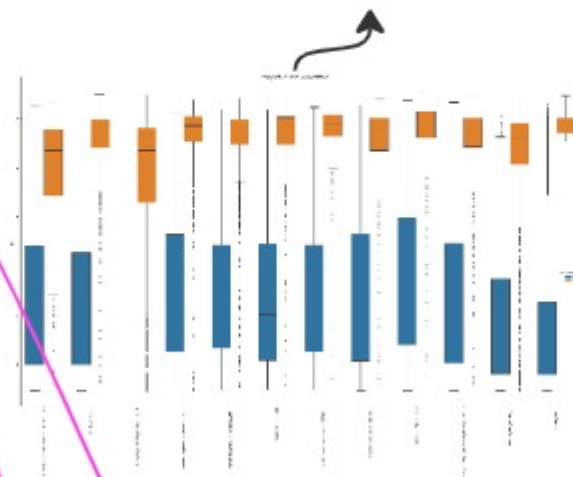


```
# para evitar gerar texto (máscara) com "_____"
# self.bad_words = ['_', '___', '____', '_____']
self.bad_words = [self.tokenizer.decode(pos) for pos in range(self.tokenizer.vocab_size) \
                  if '_' in self.tokenizer.decode(pos)]
self.bad_words_ids = [[self.tokenizer.get_vocab()[word]] \
                      for word in self.bad_words \
                      if word in self.tokenizer.get_vocab()]
self.bad_words_ids += [[1849]] # '\xa0'
self.bad_words_ids += [[12085]] # 'significance'
```

Como esperado, muitos "negativos" são relevantes (amostrados do bm25)

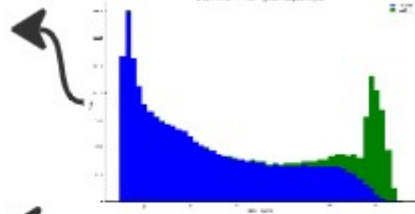
* conforme modelo que foi usado para filtrar: cross-encoder/ms-marco-TinyBERT-L-2

Resultados interesantes

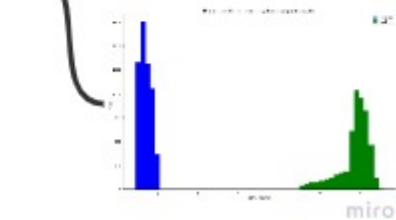


Experimentei "cortar" os dados: positivos e negativos

antes



depois

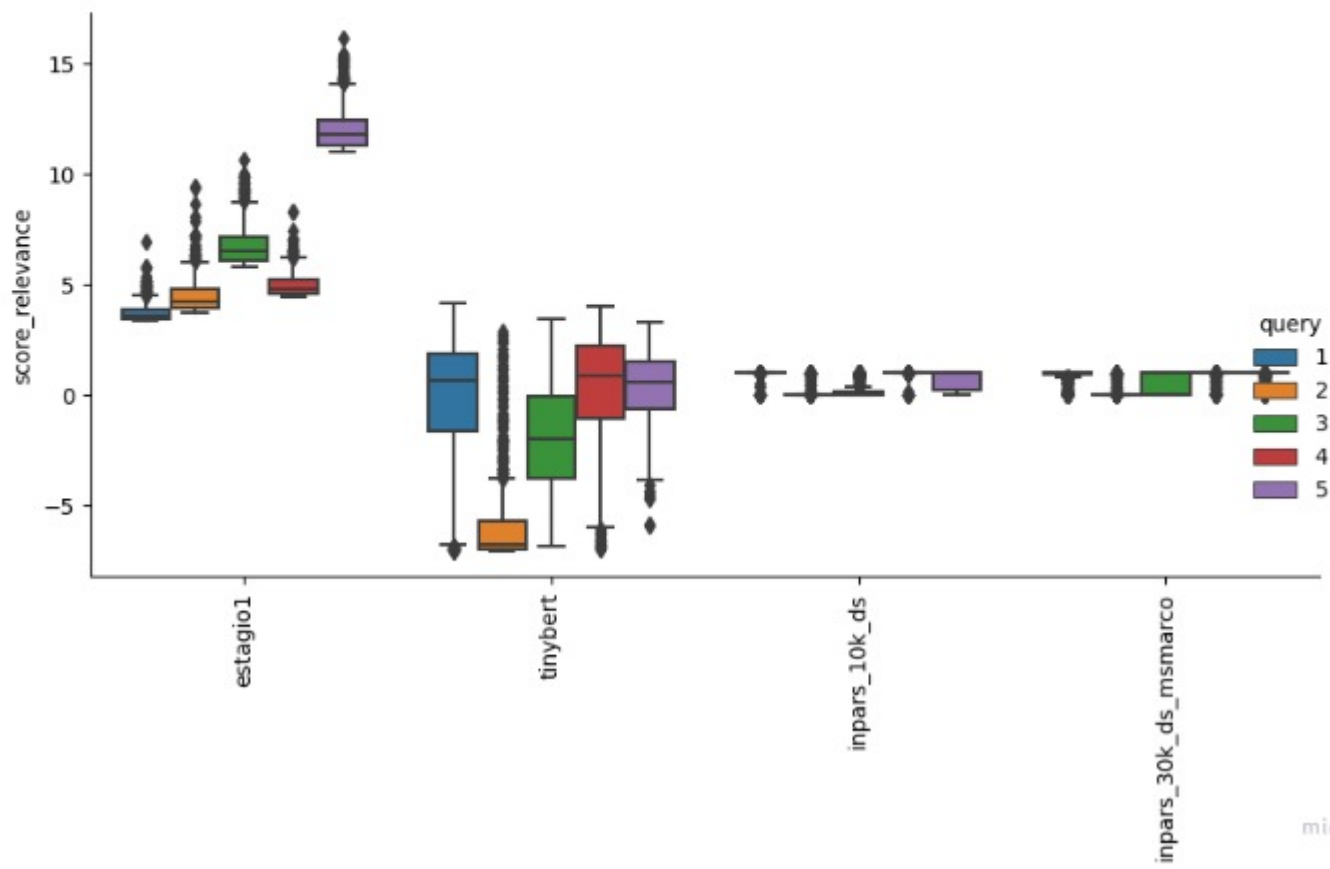


Alguns resultados



Curiosidade: se for fazer média entre scores dos retrievers é necessário normalizar antes

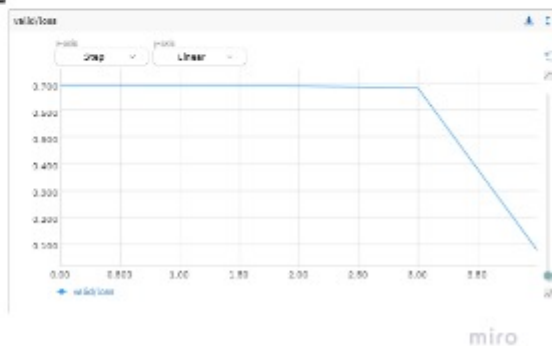
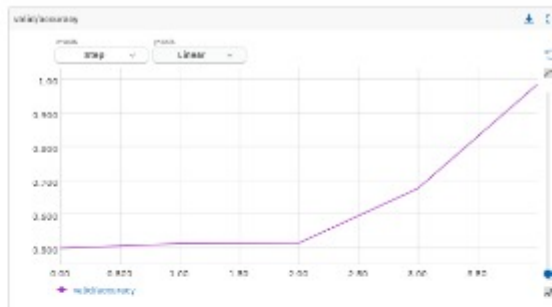
Faixa de valores dos scores de cada retriever



- Não entendi por quê no meu treinamento:
1. nas primeiras épocas pouco o modelo aprende?
 2. tem uma queda na loss a partir, geralmente, da época 3. O último treino, foi entre 4 e 5 (ndcg@10 dobrou de 32 para 60)

Uso `torch.optim.SGD(model.parameters(), lr=1e-3)` [Rastro no Neptune](#)

É como explicar que no InPars conseguiram em uma só época?
Experimentaram mais épocas?



Quais principais diferença entre:

`from transformers import AutoModelForSequenceClassification`

x

`from sentence_transformers import CrossEncoder`

transformers x sentence_transformers

Tópicos avançados

Qual modelo podemos usar para português
em substituição aos modelos abaixo:

EleutherAI/gpt-j-6B "LLM open-source"

microsoft/MiniLM-L12-H384-uncased

cross-encoder/ms-marco-TinyBERT_{v1.2}

Como o cross-encoder/ms-marco-TinyBERT-L-2 foi pré-treinado no ms-marco, ele não deveria ser um melhor reranqueador no TREC-DL20 do que no TREC-Covid?

Eu consegui ndcg@10: TREC-DL20=62,25 ; TREC-Covid=53,68

Parece-me que os colegas conseguiram mais de 70 nesse modelo sem finetuning no TREC-Covid.

- Faz sentido?

Provavelmente devo ter feito algo errado:

- . Posso criar o índice usando `LuceneSearcher.from_prebuilt_index('beir-v1.0.0-trec-covid.flat')`?