Code Presentation

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

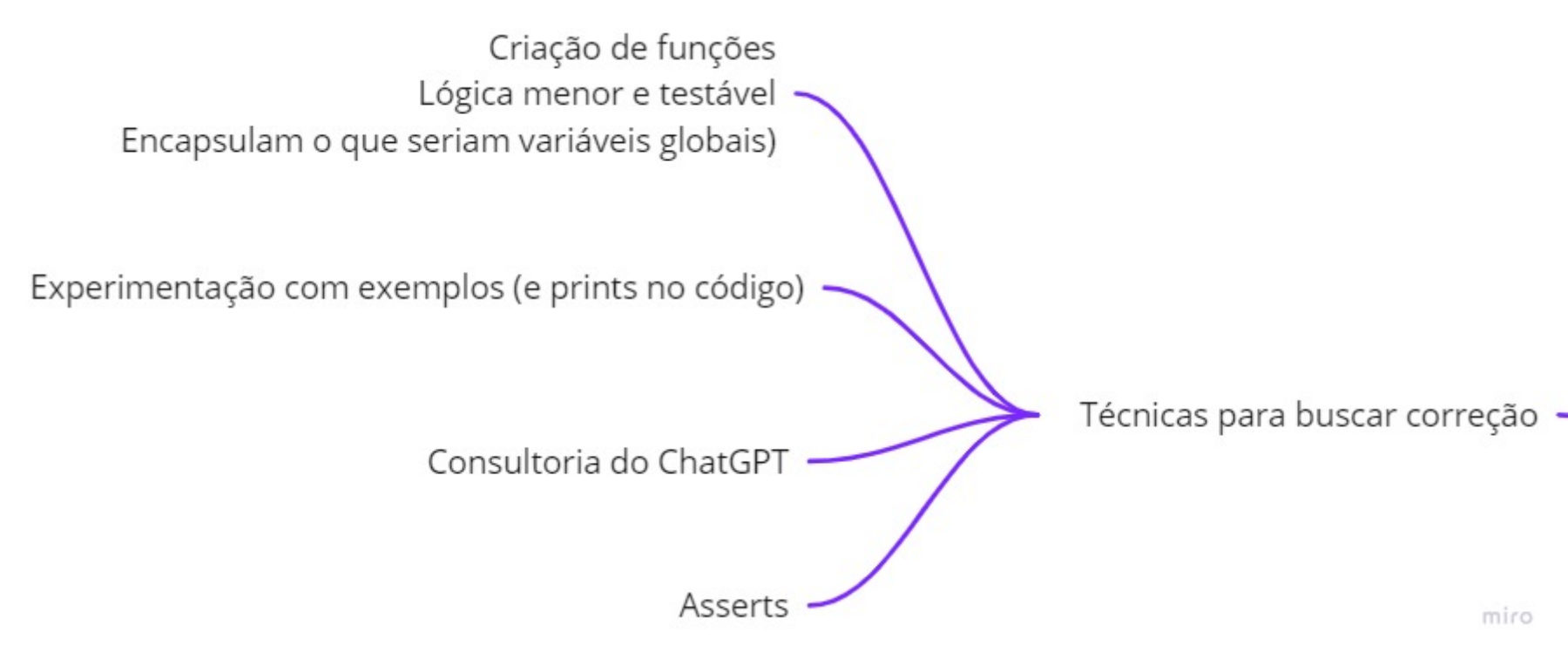
Aula 8/9 InPars v1 and v2

Processo InPars

(Ver arpresentação do artigo)

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

Conceitos


Criação de funções
Lógica menor e testável
Encapsulam o que seriam variáveis globais)
Experimentação com exemplos (e prints no código)
Consultoria do ChatGPT
Asserts
Técnicas para buscar correção
miro

No dataset, ajustei max_seq_len para pegar do tokenizador (parâmetro)

Mas, no tokenizador do microsoft/MiniLM-L12-H384-uncased, seu valor é:
1000000000000000019884624838653

```
self.max_seq_length = tokenizer.model_max_length # model.config.max_position_embeddings
if self.max_seq_length > 64000:
  print(f"Valor de self.max_seq_length  {self.max_seq_length} indica que deve ser usado
  self.max_seq_length =  512
```

Problemas encontrados e soluções

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: '
```

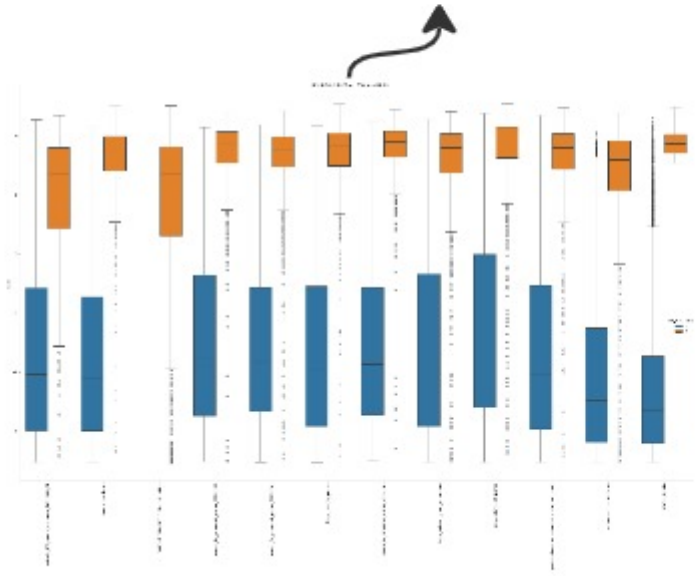
Técnica para evitar palavras frequentes:

```
# 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'
```

Resultados interessantes

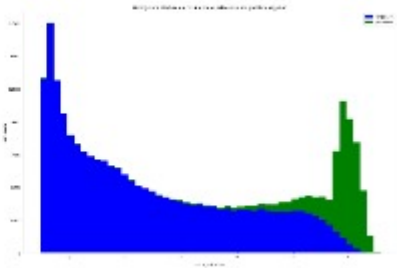
Visualização da distribuição dos dados (quanto à relevância*
Como esperado, muitos "negativos" são relevantes (amostrados do bm25)

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

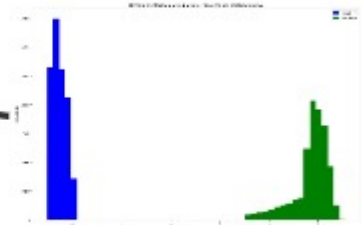


Experimentei "cortar" os dados: positivos e negativos

antes

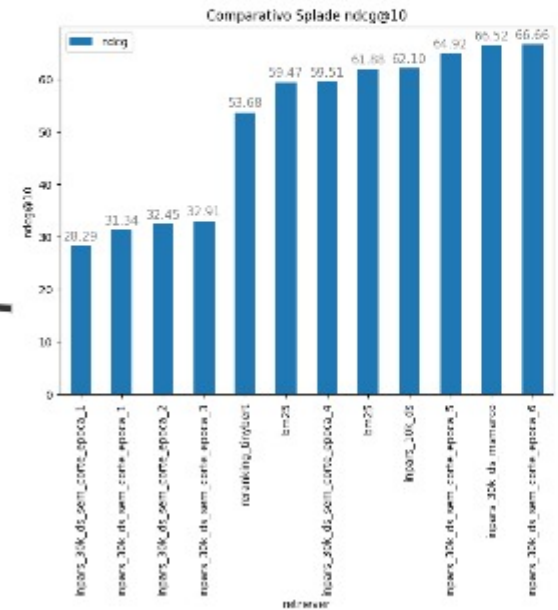


depois



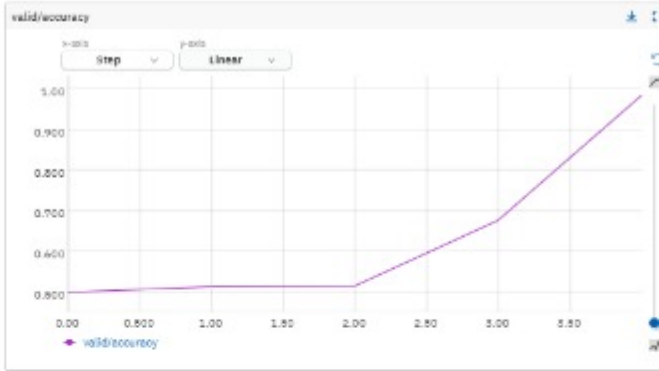
Mas, aparentemente, os resultados foram piores (em execução).
Mas falta separar a causa: uso do ms-marco ou corte?

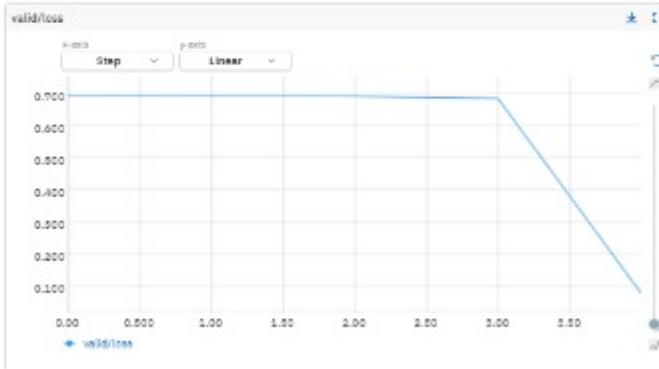
Alguns resultados



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]



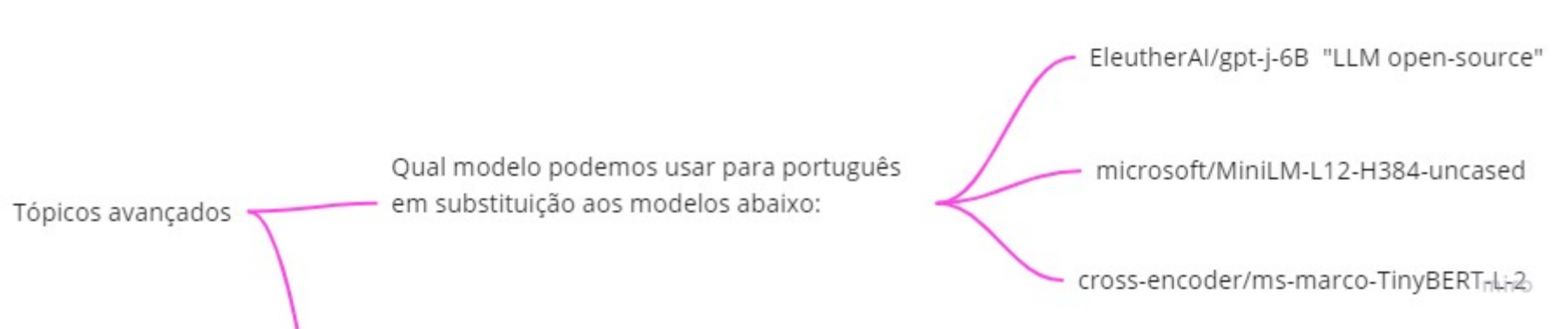


É como explicar que no InPars conseguiram em uma só época?
Experimentaram mais épocas?

Quais principais diferença entre:

- from tranformers import AutoModelForSequenceClassification
- x
- from sentence_transformers import CrossEncoder

tranformers 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-L-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 finetunning no TREC-Covid.
Faz sentido?

Provavelmente devo ter feito alto errado:
. Posso criar o índice usando LuceneSearcher.from_prebuilt_index('beir-v1.0.0-trec-covid.flat')?