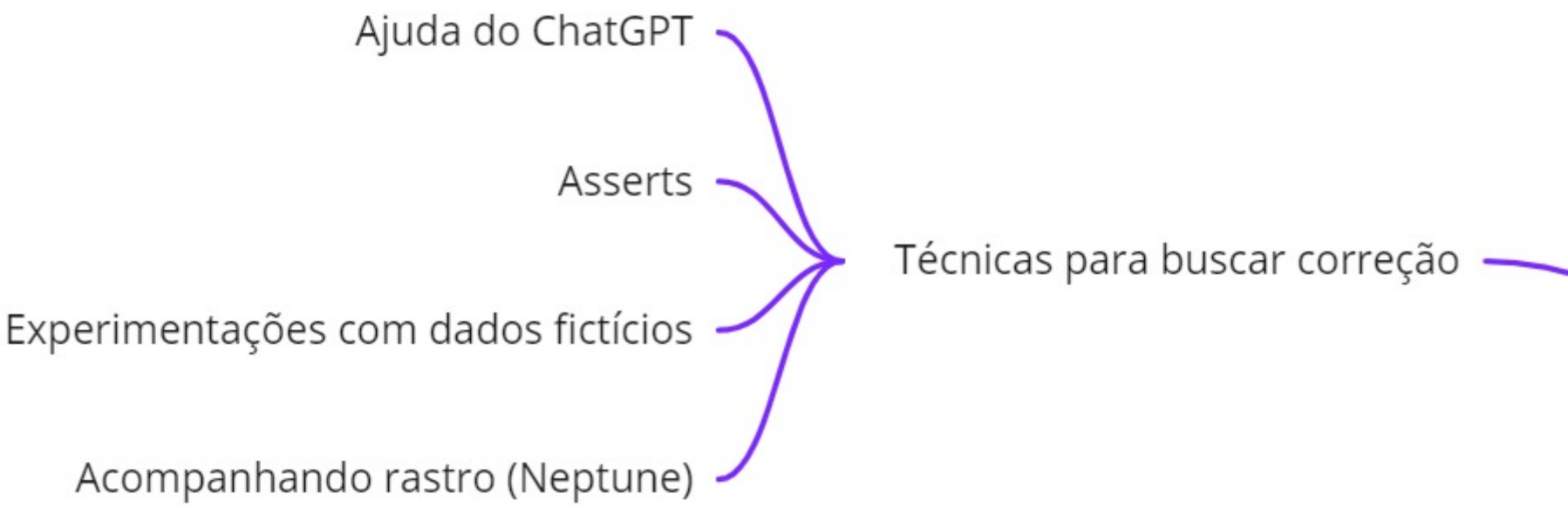
Code Presentation

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

Aula 6 - Doc2Query


Ajuda do ChatGPT
Asserts
Experimentações com dados fictícios
Acompanhando rastro (Neptune)
Técnicas para buscar correção

Dessa vez, optei por usar HF.Trainner

os.environ['NEPTUNE_ALLOW_SELF_SIGNED_CERTIFICATE'] = 'TRUE'
os.environ['NEPTUNE_PROJECT'] = 'marcusborela/IA386DD'
os.environ['NEPTUNE_API_TOKEN'] = getpass.getpass('Informe NEPTUNE_API_TOKEN'

training_args = Seq2SeqTrainingArguments(report_to="neptune", („„)(

https://app.neptune.ai/marcusborela/IA386DD/e/IAD-72

Integração Trainner com Neptune

# Imprimindo estatísticas do índice
index_reader = IndexReader(DIRETORIO_INDICE)
print(index_reader.stats())

Visualizando dados dos índices

Truques do código

```
pipe = pipeline("text2text-generation", model=model, tokenizer=tokenizer, device=0,
                max_length=512, repetition_penalty=1.3)
```

Uso de HF.pipeline
Não precisamos preocupar com detalhes!

```python
def remove_duplicatas(generated_text, max_sequences):
    frases_filtradas = []
    for i, text in enumerate(generated_text):
        frases_geradas = set()  # Conjunto para armazenar frases geradas e garantir que sejam distintas
        for retorno in text:
            frase = retorno["generated_text"]
            if frase not in frases_geradas:  # Verificar se a frase já foi gerada antes
                frases_geradas.add(frase)  # Adicionar à lista de frases geradas
                if len(frases_geradas) == max_sequences:  # Parar de adicionar frases quando atingir o limite máximo
                    break
        frases_filtradas.append(list(frases_geradas))  # Adicionar lista de frases geradas (set) a frases filtradas
    return frases_filtradas
```

```
+ Code  + Markdown
```

```python
def gerar_texto(lista_texto, parm_num_return_sequences:int=5, parm_batch_size:int=8):
    generated_text = pipe(lista_texto, num_workers=8, batch_size=parm_batch_size, top_k=50,
                          do_sample=True, num_return_sequences=parm_num_return_sequences*2)
    return remove_duplicatas(generated_text, max_sequences = parm_num_return_sequences)
```

Estruturação de experimentos

```
dict_experimentos = {
    "expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e sem expansão em sem texto": ['corpus_com_texto_expansao_5_queries_bleu_1990', 'corpus_sem_texto_sem_expansao'],
    "expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e com expansão em sem texto": ['corpus_com_texto_expansao_5_queries_bleu_1990', 'corpus_sem_texto_com_expansao_10_queries_bleu_1990'],
    "expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e sem documentos sem texto": ['corpus_com_texto_expansao_5_queries_bleu_1990'],
    "expansão em documentos com texto de até 10 queries (modelo com BLEU 21.1) e sem expansão em sem texto": ['corpus_com_texto_expansao_10_queries_bleu_211', 'corpus_sem_texto_sem_expansao'],
    "expansão em documentos com texto de até 10 queries (modelo com BLEU 21.1) e com expansão em sem texto": ['corpus_com_texto_expansao_10_queries_bleu_211', 'corpus_sem_texto_com_expansao_10_queries_bleu_1990'],
    "expansão em documentos com texto de até 10 queries (modelo com BLEU 21.1) e sem documentos sem texto": ['corpus_com_texto_expansao_10_queries_bleu_211'],
}
```

ChatCPT funcionou

Problemas encontrados e soluções

Marcus Borela 13h05

Agora sim, o barco do treinamento está indo de "vento em popa"...

O problema era realmente no dataset.__get_item__:

```
        saida = {"input_ids": self.tokenized_texts["input_ids"][index],
                 "attention_mask": self.tokenized_texts["attention_mask"][index],
                 "labels": self.tokenized_~~texts~~queries["input_ids"][index]}
```

Gratidão a todos que sugeriram algum ajuste (Eduardo, Gustavo, Marcos, Mirelle e Monique, ordem alfabética 😇)... É o chatCPT funcionando (Colegas Para o Trabalho) 🥰

```
2023/04/10 12:54:38     Em compute_metrics: bleu=15.07811773134163
2023/04/10 12:54:38
2023/04/10 12:54:38     Predicao[0] :how many steps does a mile walk
2023/04/10 12:54:38     Ground_truth[0]:['how many fitbit steps equal a mile']
2023/04/10 12:54:38     {'eval_loss': 1.8028239011764526, 'eval_bleu': 15.07811773134163,
'eval_runtime': 34.1723, 'eval_samples_per_second': 29.264, 'eval_steps_per_second': 1.844, 'epoch':
3.0}
```

Resultados interessantes

67.23 - expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e sem expansão em sem texto
66.92 - expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e com expansão em sem texto
67.92 - expansão em documentos com texto de até 5 queries (modelo com BLEU 19.9) e sem documentos sem texto
66.46 - expansão em documentos com texto de até 10 queries (modelo com BLEU 20.1) e sem expansão em sem texto
66.48 - expansão em documentos com texto de até 10 queries (modelo com BLEU 20.1) e com expansão em sem texto
67.05 - expansão em documentos com texto de até 10 queries (modelo com BLEU 20.1) e sem documentos sem texto

Obs.:
O modelo que gerou até 5 queries, truncou a entrada em 450 caracteres.
O modelo que gerou até 10 queries, truncou a entrada em 480 tokens.

- Dúvidas básicas
  - Quais os parâmetros usar na geração de texto?
    - Fiz alguns experimentos seguindo link
    - https://colab.research.google.com/github/huggingface/blog/blob/main/notebooks/02_how_to_generate.ipynb#scrollTo=AZ6xs-KLi9jT
  - O texto não pré-processado não é salvo no índice?

```
bm25_score_genotyp = index_reader.compute_bm25_term_weight('zjufx4fo', 'genotyp', analyzer=None)
bm25_score_genotype = index_reader.compute_bm25_term_weight('zjufx4fo', 'genotype', analyzer=None)
print(f"bm25_score_genotype: {bm25_score_genotype}, bm25_score_genotyp:{bm25_score_genotyp}")

=> bm25_score_genotype: 0.0, bm25_score_genotyp:2.7530345916748047
```

Tópicos avançados

Por quê a loss vai tão mal nos dados de avaliação e mesmo assim o BLeu continua subindo?

Desacoplação entre o Bleu e a loss