Code Presentation
IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas
Student: Marcus Vinícius Borela de Castro

Trade-off custo x desempenho de alguns pipelines

```
custo_indexacao_tempo = 0
## acumular custo por tempo
for tempo_valor in parm_dados['tempo_indexacao_segundo']:
    if tempo_valor['tipo'] == 'cpu':
        custo_indexacao_tempo +=  tempo_valor['valor'] * CUSTO_CPU_ALOCADA_SEGUNDO
    elif tempo_valor['tipo'] == 'gpu':
        custo_indexacao_tempo +=  tempo_valor['valor'] * CUSTO_CPU_ALOCADA_SEGUNDO
    else:
        raise Exception(f"Tipo de tempo deveria ser cpu ou gpu e não {tempo_valor['tipo']}")
# tem que deixar cpu disponível 24h
custo_cpu_dia = 24 * CUSTO_CPU_ALOCADA_HORA
# índice tem que ficar em memória
custo_memoria_dia = 24 * parm_dados['memoria_indice_byte_ram'] * CUSTO_RAM_CPU_HORA_BYTE
custo_dia = custo_memoria_dia + custo_cpu_dia
custo_gpu_dia = 0
if parm_contexto == 'utilizacao_perfeita': #(assim que terminou de processar uma query, já tem outra)
    if parm_dados['se_retrieval_usa_gpu']:
        custo_gpu_dia = 24 * 3600 * custo_gpu_segundo[parm_tipo_gpu]
        custo_dia += custo_gpu_dia
        print(f"para {parm_contexto} custo gpu dia: {custo_gpu_dia}")
    num_queries_dia = (24 * 3600) / parm_dados['retrieval_tempo_medio_por_query']
    custo_query = round(custo_dia / num_queries_dia, 10)
elif parm_contexto == 'utilizacao_precaria_100': #(100 queries/dia)
    if parm_dados['se_retrieval_usa_gpu']:
        custo_gpu_dia = 100 * parm_dados['retrieval_tempo_medio_por_query'] * custo_gpu_segundo[parm_tipo_gpu]
        custo_dia += custo_gpu_dia
        print(f"para {parm_contexto} custo gpu dia: {custo_gpu_dia}")
    num_queries_dia = 100
    custo_query = round(custo_dia / num_queries_dia, 10)
return {'usd_query': custo_query,
        'usd_dia': custo_dia,
        'usd_gpu_dia': custo_gpu_dia,
        'usd_mes': (30 * custo_dia),
        'usd_indexacao_tempo': custo_indexacao_tempo,
        }
```

Conceitos

Cálculo


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

Salvando informações a cada experimento — Truques do código

```
resultado_execucao['ndcg_10'] = round(100*results['NDCG@10'],2)
```

```
resultado_pipeline[nome_pipeline] = resultado_execucao
print(resultado_pipeline[nome_pipeline])
```

```
{'tempo_indexacao_segundo': [{'tipo': 'cpu', 'valor': 0.109634}], 'memoria_indice_byte_ram': 269772727, 'se_retrieval_usa_gpu':
True, 'retrieval_tempo_medio_por_query': 1.722126, 'ndcg_10': 71.25}
```

```
avaliacao_pipeline_contexto[nome_pipeline] = retorna_calculo_contexto(resultado_execucao, parm_tipo_gpu='3090')
print(avaliacao_pipeline_contexto[nome_pipeline] )
```

```
para utilizacao_perfeita custo gpu dia: 5.999616
para utilizacao_precaria_100 custo gpu dia: 0.011958442944000001
{'utilizacao_perfeita': {'usd_query': 0.0001345363, 'usd_dia': 6.749761483605887, 'usd_gpu_dia': 5.999616, 'usd_mes':
202.49284450817663, 'usd_indexacao_tempo': 9.135801219999999e-07}, 'utilizacao_precaria_100': {'usd_query': 0.0076210393,
'usd_dia': 0.7621039265498879, 'usd_gpu_dia': 0.011958442944000001, 'usd_mes': 22.86311779649664, 'usd_indexacao_tempo':
9.135801219999999e-07}}
```

Desejava executar novamente o InPars com modelo correto. Mas Hugging Face estava fora. Então optei por experimentar mudanças no Splade.

Problemas encontrados e soluções

Estimativa de custo para Gpu 3090

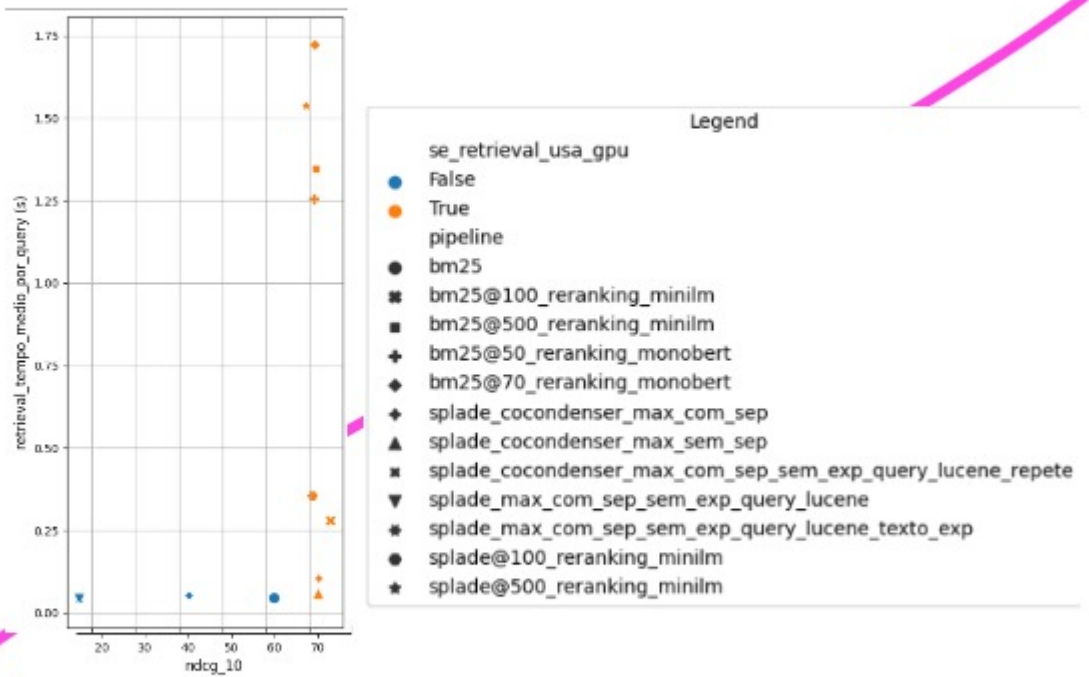
Resultados interessantes

Alguns números

	pipeline	ndcg_10	retrieval_tempo_medio_por_query
1	bm25@100_reranking_minilm	74.85	0.279581
5	splade_cocondenser_max_sem_sep	72.14	0.104524
6	splade_cocondenser_max_sem_sep	72.00	0.058798
2	bm25@500_reranking_minilm	71.47	1.347039
4	bm25@70_reranking_monobert	71.25	1.722126
3	bm25@50_reranking_monobert	70.97	1.255988
10	splade@100_reranking_minilm	70.68	0.354909
11	splade@500_reranking_minilm	69.25	1.537312
0	bm25	61.88	0.044992
9	splade_max_com_sep_sem_exp_query_lucene_texto_exp	42.31	0.052678
7	splade_cocondenser_max_com_sep_sem_exp_query_l...	17.13	0.040139
8	splade_max_com_sep_sem_exp_query_lucene	17.13	0.042195

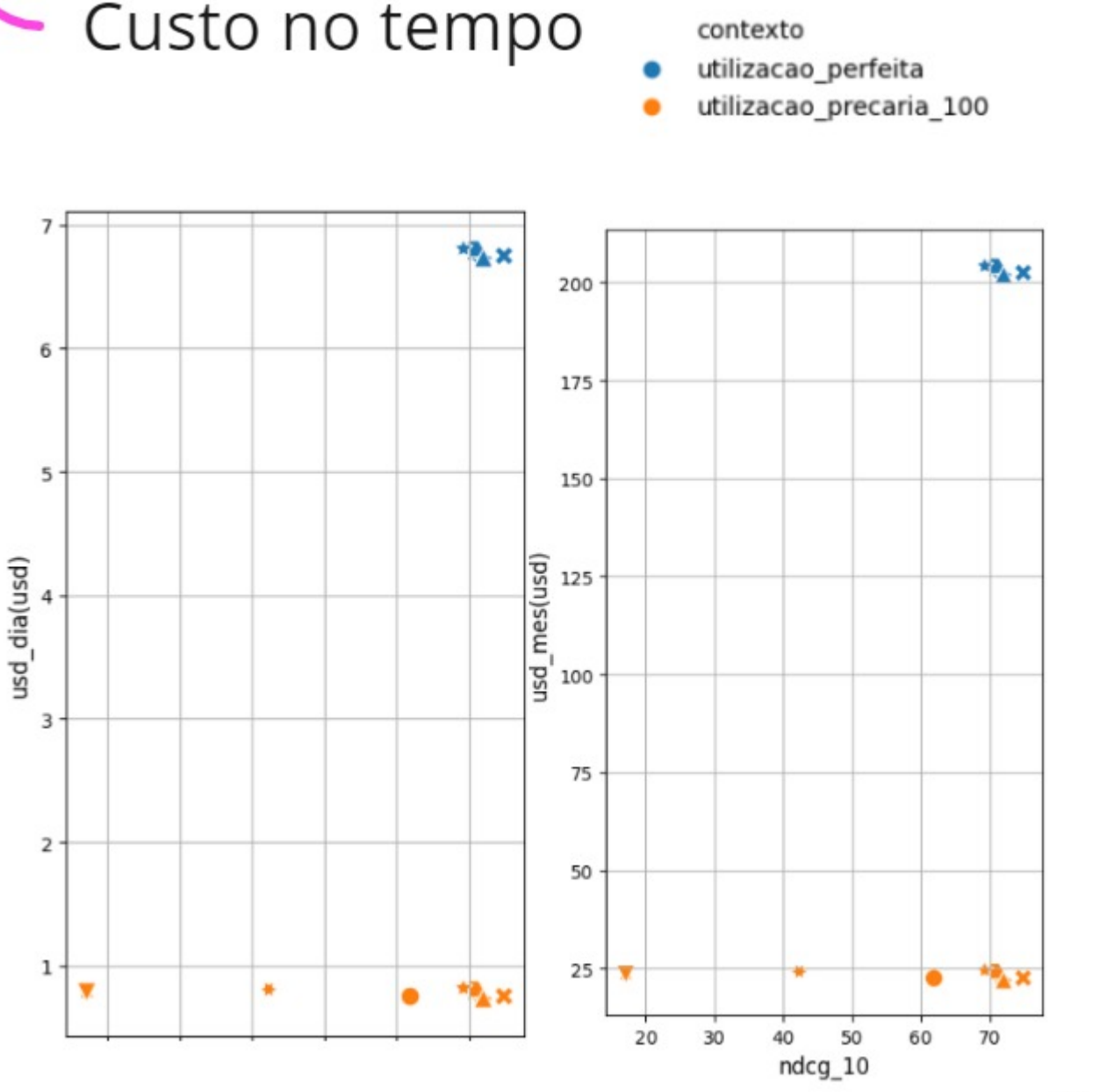
	pipeline	ndcg_10	retrieval_tempo_medio_por_query	se_retrieval_usa_gpu
splade_cocondenser_max_com_sep_sem_exp_query_l...	17.13	0.040139	False
splade_max_com_sep_sem_exp_query_lucene	17.13	0.042195	False
bm25	61.88	0.044992	False
splade_max_com_sep_sem_exp_query_lucene_texto_exp	42.31	0.052678	False
splade_cocondenser_max_sem_sep	72.00	0.058798	True
splade_cocondenser_max_com_sep	72.14	0.104524	True
bm25@100_reranking_minilm	74.85	0.279581	True
splade@100_reranking_minilm	70.68	0.354909	True
bm25@50_reranking_monobert	70.97	1.255988	True
bm25@500_reranking_minilm	71.47	1.347039	True
splade@500_reranking_minilm	69.25	1.537312	True
bm25@70_reranking_monobert	71.25	1.722126	True

Visualizando no gráfico é melhor

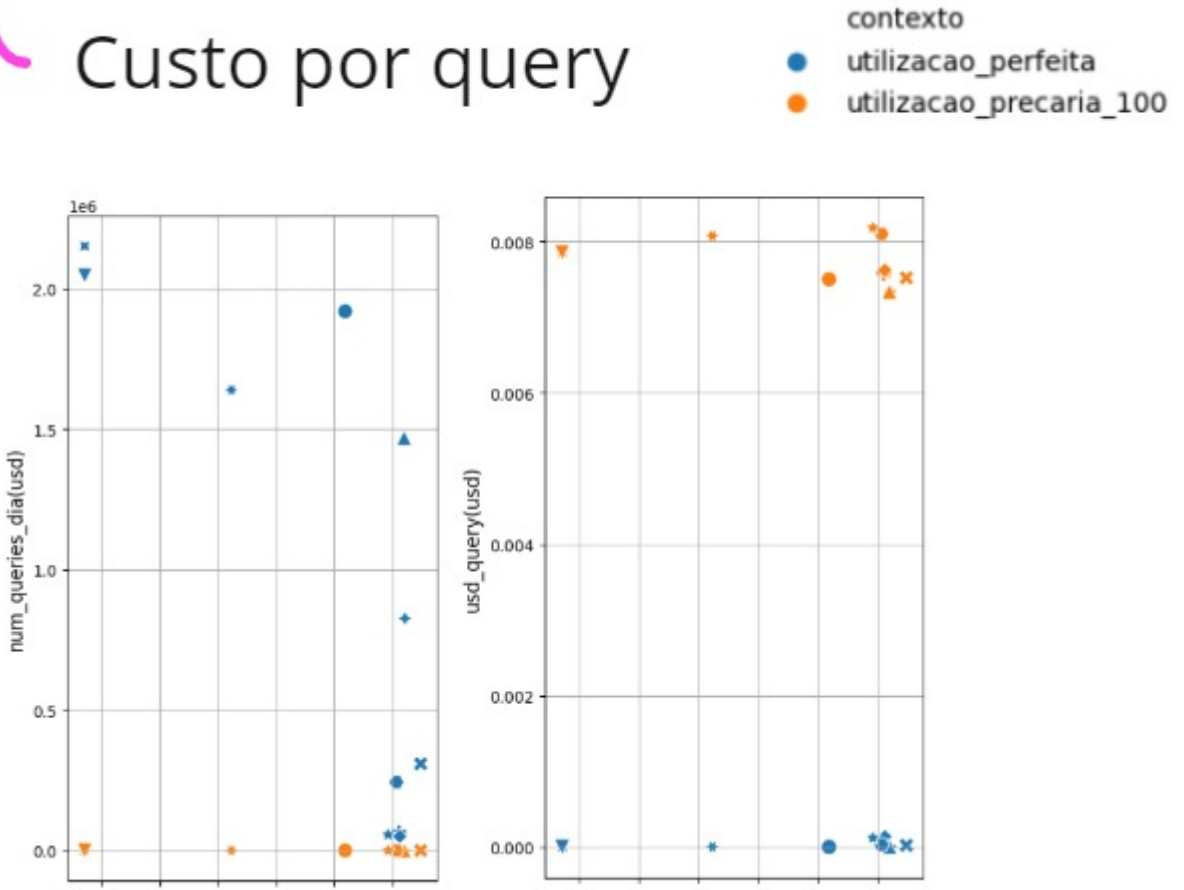


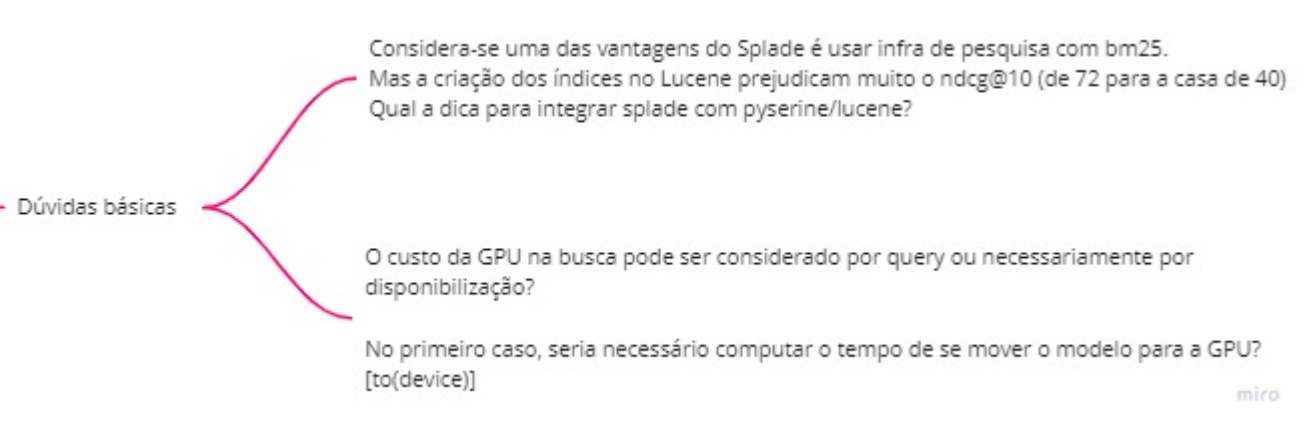
legenda e eixo x em comum

Custo no tempo



Custo por query




Dúvidas básicas
Considera-se uma das vantagens do Splade é usar infra de pesquisa com bm25.
Mas a criação dos índices no Lucene prejudicam muito o ndcg@10 (de 72 para a casa de 40)
Qual a dica para integrar splade com pyserine/lucene?
O custo da GPU na busca pode ser considerado por query ou necessariamente por disponibilização?
No primeiro caso, seria necessário computar o tempo de se mover o modelo para a GPU? [to(device)]
miro

Tópicos avançados

Dado que temos um modelo treinado localmente, quais os passos para portar ele e o serviço para a nuvem (exemplo: Azure ou AWS)