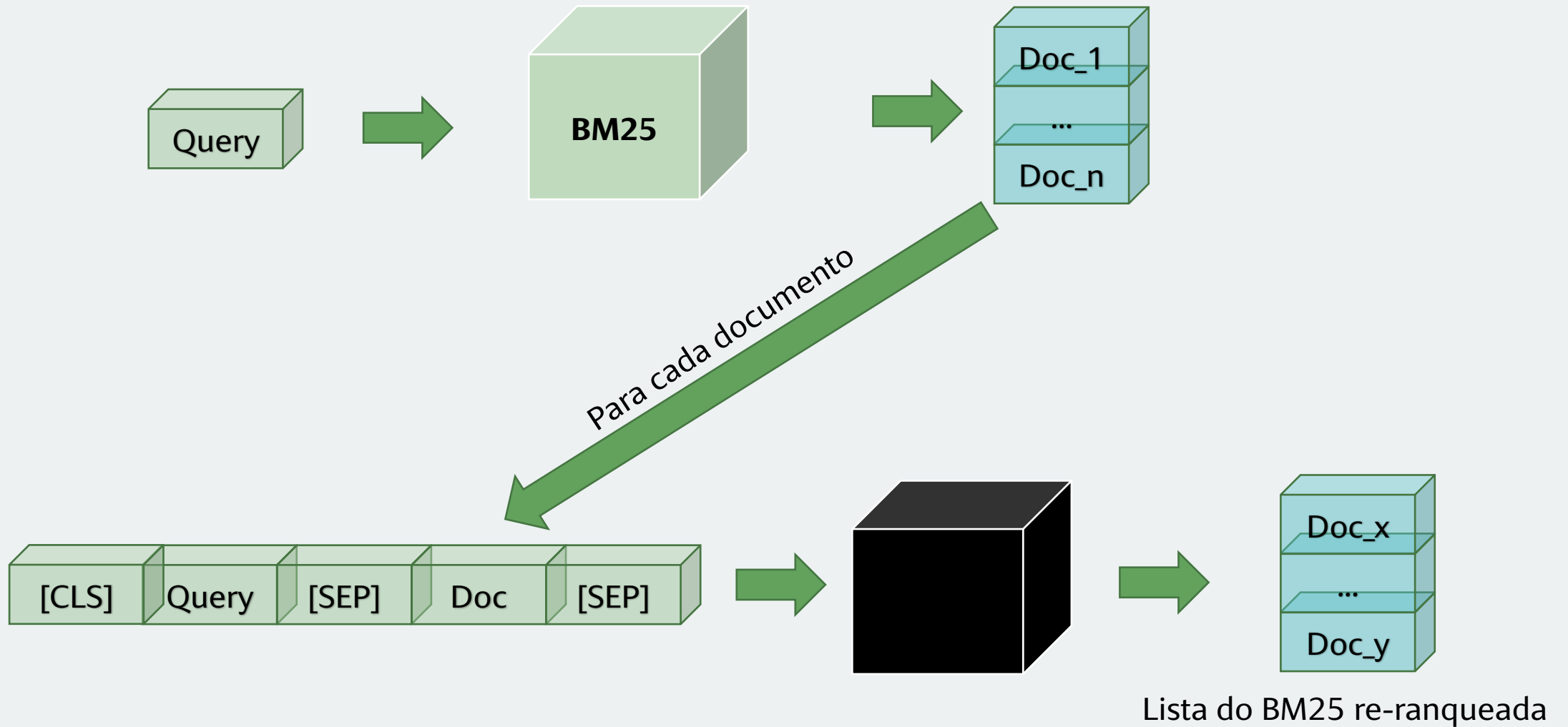


Notebook – Inpars

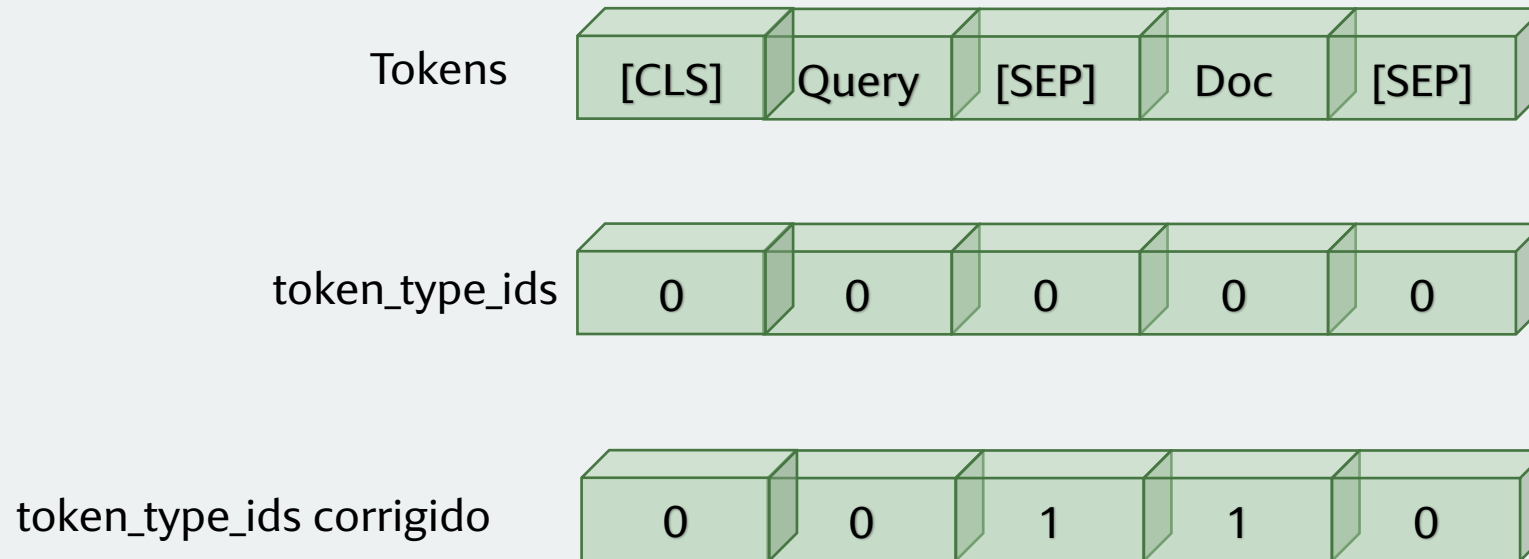
Leandro Carísio

Conceitos do exercício



Problemas e soluções no desenvolvimento

1. Primeira implementação sem separar a query do documento via token_type_ids:

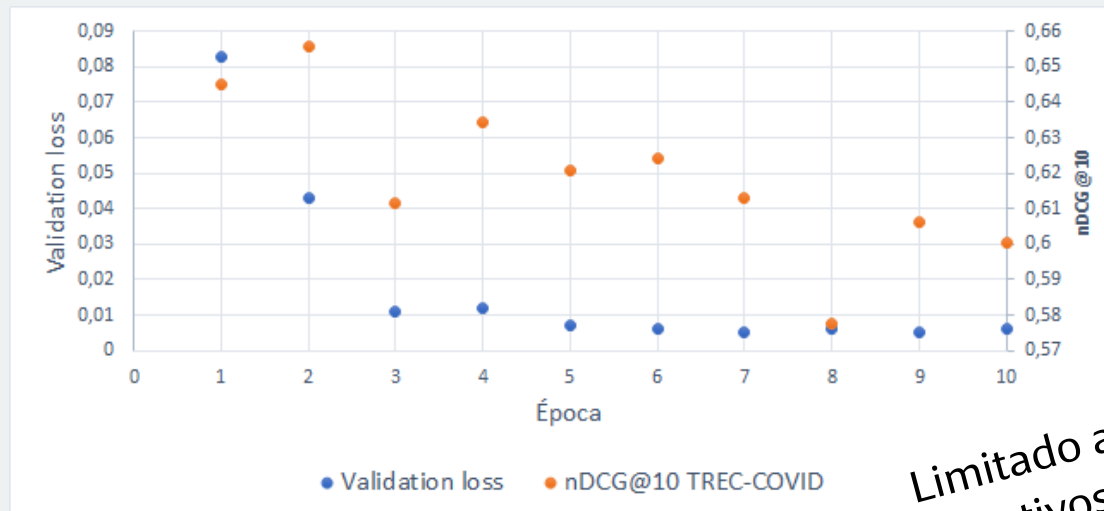


	nDCG@10	nDCG@10 após correção
cross-encoder/ms-marco-MiniLM-L-6-v2 (sem fine-tuning)	0,2949	0,7131

Problemas e soluções no desenvolvimento

2. Fine-tuning com `AutoModelForSequenceClassification` não funcionou e eu não faço ideia do motivo

	Antes do fine-tuning	Depois do fine-tuning
Sem corrigir <code>token_type_ids</code>	0,2949	0,6236
Corrigindo <code>token_type_ids</code>	0,7131	0,6557



Limitado a 5 exemplos negativos por query

Problemas e soluções no desenvolvimento

2. Fine-tuning com AutoModelForSequenceClassification não funcionou e eu não faço ideia do motivo

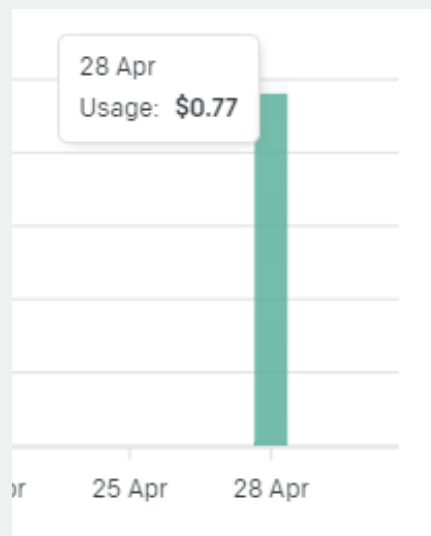
Tentativas pra resolver a questão:

- Limitar a no máximo 5 exemplos negativos por query
- Balancear o dataset (limitar a 1 exemplo negativo por query)
- Testar só com o meu dataset
- Reranking com 100 hits em vez de 1000 hits

A maior diferença foi em fazer o reranking só em 100 hits (chegou a 5 pontos no nDCG ***dependendo*** do modelo)

Resultados

Custo de US\$ 0.77 para gerar 1.000 queries usando gpt-3.5-turbo



nDCG@10

0,7131 sem fine-tuning

0,6798 com fine-tuning e reranking em 1000 documentos

0,6827 com fine-tuning e reranking em 100 documentos

Dúvida básica (?)

Qual o truque pra fazer o fine-tuning funcionar usando
`AutoModelForSequenceClassification`?

Pode ter sido poucas épocas?

And **disadvantage** of `MarginMSELoss` is the slower training time: We need way more epochs to get good results. In `MultipleNegativesRankingLoss`, with a batch size of 64, we compare one query against 128 passages. With `MarginMSELoss`, we compare a query only against two passages.

Obrigado

Leandro Carísio
carisio@gmail.com