Cross-encoder Reranking

Marcos Piau Vieira

1. Explicação de conceitos importantes do exercício feito - fine tune

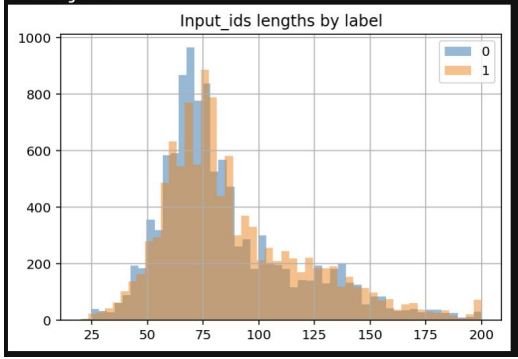
- finetune de modelo cross encoder
- serialização: [CLS] query [SEP] doc [SEP]
- 20 épocas, partindo do nreimers/MiniLM-L6-H384-uncased, com 10% para validação
- checkpoint epoch 11, 0.934091 de acurácia em dev
- learning rate constante AdamW de 5e-5, batch size de 32

1. Explicação de conceitos importantes do exercício feito - finetune

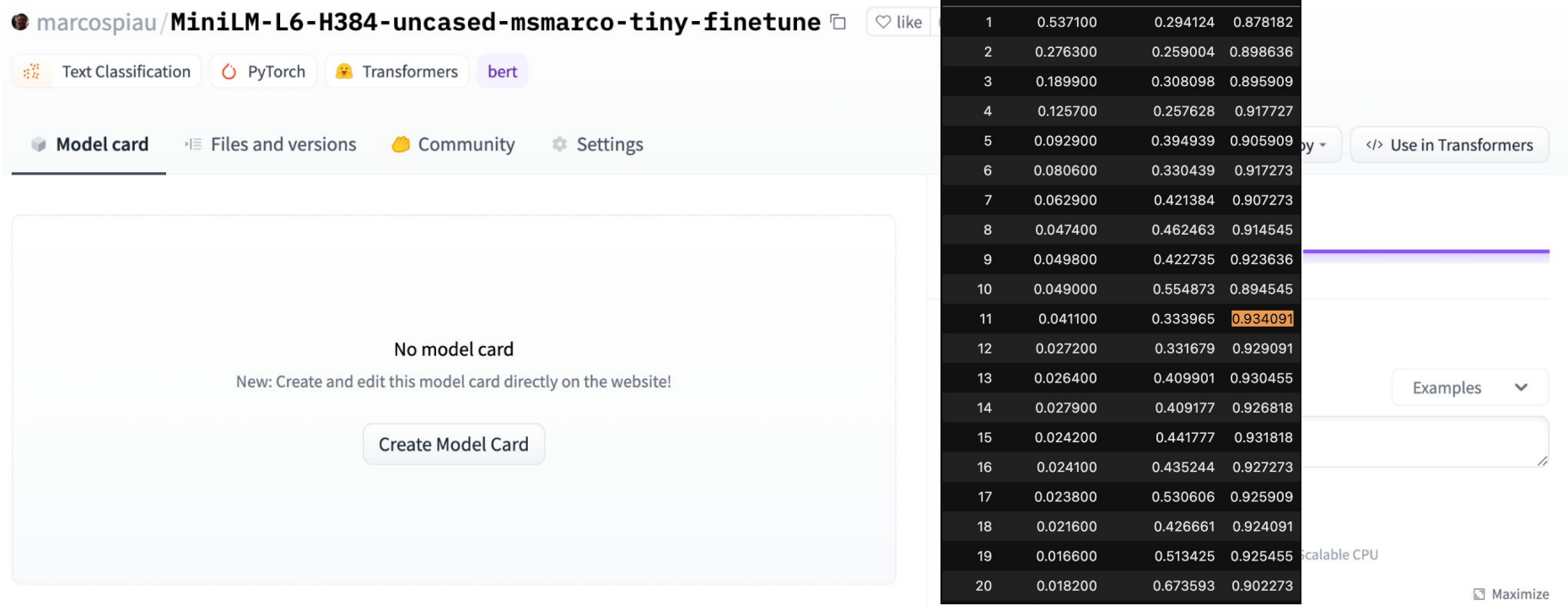
- seq_length de 200 tokens (+ que P99) com padding dinâmico + group_by_length *

```
********************************************************************************
Analyzing by label

Describe
         count        mean         std   min   25%   50%    75%    99%    max
label
0      11000.0   85.437364   31.035549  21.0  65.0  76.0  100.0  181.0  200.0
1      11000.0   88.630909   32.345934  20.0  66.0  80.0  106.0  186.0  200.0
Histogram
```



1. Explicação de conceitos importantes do exercício feito - finetune



[12380/12380 21:51, Epoch 20/20]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.537100	0.294124	0.878182
2	0.276300	0.259004	0.898636
3	0.189900	0.308098	0.895909
4	0.125700	0.257628	0.917727
5	0.092900	0.394939	0.905909
6	0.080600	0.330439	0.917273
7	0.062900	0.421384	0.907273
8	0.047400	0.462463	0.914545
9	0.049800	0.422735	0.923636
10	0.049000	0.554873	0.894545
11	0.041100	0.333965	0.934091
12	0.027200	0.331679	0.929091
13	0.026400	0.409901	0.930455
14	0.027900	0.409177	0.926818
15	0.024200	0.441777	0.931818
16	0.024100	0.435244	0.927273
17	0.023800	0.530606	0.925909
18	0.021600	0.426661	0.924091
19	0.016600	0.513425	0.925455
20	0.018200	0.673593	0.902273

1. Explicação de conceitos importantes do exercício feito - rerank

First stage	Reranker	ndcg_cut_10
BM25 (k1=0.9, b=0.4)	N/A	0.4796
BM25 (k1=0.9, b=0.4)	marcospiau/MiniLM-L6-H384-uncased-msmarco-tiny-finetune	0.4146
BM25 (k1=0.9, b=0.4)	cross-encoder/ms-marco-TinyBERT-L-2	0.6235
BM25 (k1=0.9, b=0.4)	shuffle scores BM25	0.0344

2. Técnicas para garantir que a implementação está correta

- finetune
  - overfit de uma amostra
  - overfit de um batch
- rerank:
  - usar rerank com modelo já treinado no msmarco como referência (cross-encoder/ms-marco-TinyBERT-L-2)
  - gerar uma run com scores aleatórios (shuffle dos scores do BM25)
  - gerar run com scores constantes (não fiz por falta de tempo)
  - olhar scores que saem do rerank

3. Truques de código que funcionaram

- utilizar indexes e funções pyserini (exemplo: utilizar indexes e funções pyserini ex) (searcher.batch_doc(docids, threads=threads))
- usar bibliotecas de dataframe para manipulação de dados ao invés de fazer tudo mão deixa código mais simples e (possivelmente eficiente)
- usar ecossistema huggingface (datasets, trainer etc)
- usar group_by_length=True no training_arguments reduziu tempo de 20 epochs de 51 min pra 20 min
- usar lib fix_text pra limpar mojibakes e erro de decoding
- no reranking, ordenar exemplos por sum(words_query+words_doc) diminuiu tempo de execução

3. Truques de código que funcionaram

qid	q0	docid	rank	score	run_id	query	document
i64	str	i64	i64	f64	str	str	str
23849	"Q0"	4348282	1	10.0663	"Anserini"	"are naturaliza...	"Civil Records ...
23849	"Q0"	2674124	2	9.8655	"Anserini"	"are naturaliza...	"See our FAQ's ...
23849	"Q0"	7119957	3	9.6442	"Anserini"	"are naturaliza...	"Yes, in most c...
23849	"Q0"	8133127	4	9.4317	"Anserini"	"are naturaliza...	"Spokeo pulls d...
23849	"Q0"	542113	5	9.3852	"Anserini"	"are naturaliza...	"Public Records...

3. Truques de código que funcionaram

qid	q0	docid	score	run_id	rank
---	---	---	---	---	---
i64	str	i64	f32	str	u32
23849	Q0	2647769	0.999556	DONT_CARE	1
23849	Q0	8010559	0.999528	DONT_CARE	2
23849	Q0	8010561	0.999473	DONT_CARE	3
23849	Q0	8010558	0.999406	DONT_CARE	4
...	...	...	...	...	...
1136962	Q0	80877	0.000278	DONT_CARE	997
1136962	Q0	8065423	0.000277	DONT_CARE	998
1136962	Q0	7101410	0.000274	DONT_CARE	999
1136962	Q0	1880431	0.000272	DONT_CARE	1000

4. Problemas e soluções no desenvolvimento

-

5. Resultados interessantes/inesperados

- meu reranker ficou pior que o BM25 sozinho
- bug do pyserini, salva docs raw (em formato json) ao invés do contents apenas
- reranker menor (mas certamente melhor treinado) ficou melhor que o meu modelo com finetune

6. Uma dúvida "básica" que você ou os colegas possam ter

- qual a melhor forma de “normalizar” scores para reranqueamento?

7. Um tópico "avançado" para discutirmos

- até que ponto bibliotecas com muitas abstrações valem a pena? exemplos:
  - loop manual de treino vs pytorch lightgning ou huggingface
  - reranking manual vs pygaggle (MonoBert)
  - estruturas de dados default Python (dict, lists, etc) vs bibliotecas de dataframe
- minha opinião:
  - pra quem está aprendendo ou nunca fez, melhor não abstrair tanto
  - às vezes tentamos economizar tempo, mas acabamos perdendo tempo com bugs de difícil
  - solução (ainda mais com bibliotecas mais “exóticas”)
  - problemas:
    - "premature optimization is the root of all evil."
    - “ter o martelo e sair procurando o prego”