Kaggle - LLM Science Exam

赛后讲解

主讲人：H老师

赛题背景

竞赛受OpenBookQA数据集的启发，**要求参与者使用大型语言模型（LLM）回答一些科学上的困难问题，从ABCDE五个选项中选出正确答案**。通过这项工作，研究者们希望更好地理解LLM测试自身的能力，以及在资源受限环境中运行LLM的潜力。

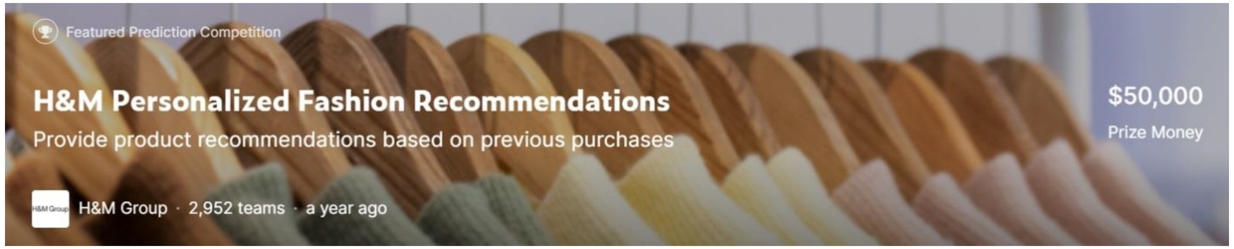
随着大型语言模型（LLM）能力范围的扩大，越来越多的研究领域正在使用LLM来表征。由于许多现有的 NLP 基准已被证明对于最先进的模型来说是微不足道的，因此也有一些有趣的工作表明 LLM 可用于创建更具挑战性的任务来测试更强大的模型。与此同时，量化和知识蒸馏等方法被用来有效地缩小语言模型并在更普通的硬件上运行它们。 Kaggle 环境提供了一个独特的视角来研究这一问题，提交内容受到 GPU 和时间限制。

此挑战的数据集是通过提供从维基百科提取的一系列科学主题的 gpt3.5 文本片段，并要求其编写多项选择题（带有已知答案），然后过滤掉简单的问题来生成的。目前，**我们估计 Kaggle 上运行的最大模型约有 100 亿个参数**，而 gpt3.5 的参数为 1750 亿个。如果一个问答模型能够在由比其规模大 10 倍的问题编写模型编写的测试中表现出色，这将是一个真正有趣的结果；另一方面，如果一个较大的模型能够有效地击败较小的模型，这对LLM自我基准表现和测试的能力具有引人注目的影响。

赛题背景

评价指标：MAP@3 (Mean Average Precision @ 3)

$$MAP@3 = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{\min(n,3)} P(k) \times \mathrm{rel}(k)$$



Featured Prediction Competition

H&M Personalized Fashion Recommendations

Provide product recommendations based on previous purchases

$50,000

Prize Money

H&M Group · 2,952 teams · a year ago

Research Prediction Competition

Happywhale - Whale and Dolphin Identification

Identify whales and dolphins by unique characteristics

$25,000

Prize Money

Happywhale · 1,588 teams · 2 years ago

- GT

> A

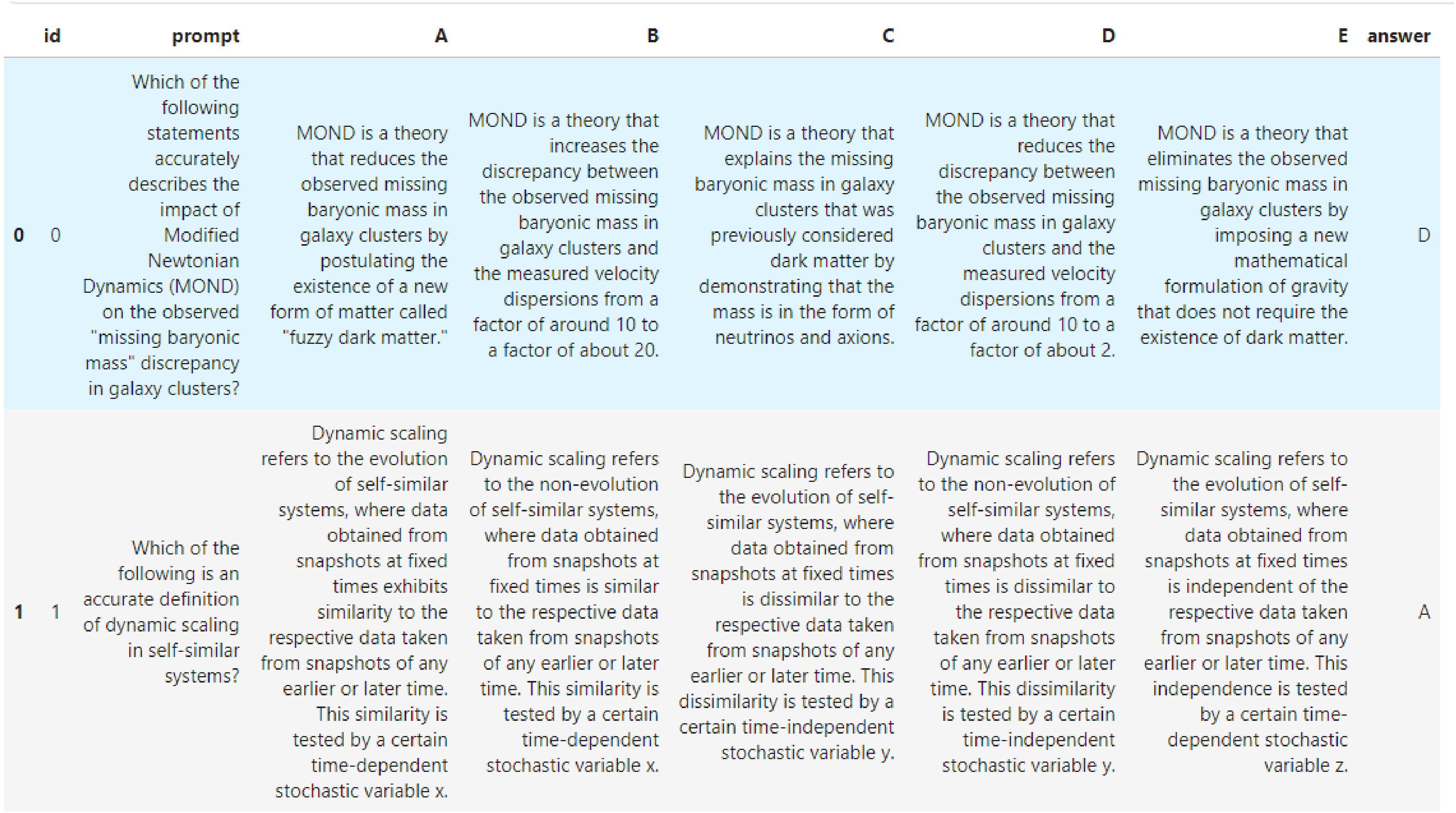
- prediction

> [A, B, C] map=1.0
> [B, A, C] map=0.5
> [B, C, A] map=0.333
> [B, C, D] map=0.0

数据概览

数据简介

竞赛官方只提供了200个问题样本，以下是前2个样本范例：



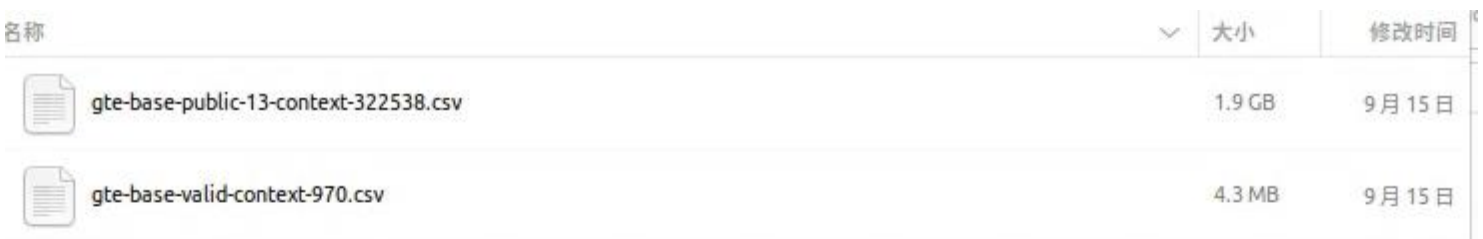
	id	prompt	A	B	C	D	E	answer
0	0	Which of the following statements accurately describes the impact of Modified Newtonian Dynamics (MOND) on the observed "missing baryonic mass" discrepancy in galaxy clusters?	MOND is a theory that reduces the observed missing baryonic mass in galaxy clusters by postulating the existence of a new form of matter called "fuzzy dark matter."	MOND is a theory that increases the discrepancy between the observed missing baryonic mass in galaxy clusters and the measured velocity dispersions from a factor of around 10 to a factor of about 20.	MOND is a theory that explains the missing baryonic mass in galaxy clusters that was previously considered dark matter by demonstrating that the mass is in the form of neutrinos and axions.	MOND is a theory that reduces the discrepancy between the observed missing baryonic mass in galaxy clusters and the measured velocity dispersions from a factor of around 10 to a factor of about 2.	MOND is a theory that eliminates the observed missing baryonic mass in galaxy clusters by imposing a new mathematical formulation of gravity that does not require the existence of dark matter.	D
1	1	Which of the following is an accurate definition of dynamic scaling in self-similar systems?	Dynamic scaling refers to the evolution of self-similar systems, where data obtained from snapshots at fixed times exhibits similarity to the respective data taken from snapshots of any earlier or later time. This similarity is tested by a certain time-dependent stochastic variable x.	Dynamic scaling refers to the non-evolution of self-similar systems, where data obtained from snapshots at fixed times is similar to the respective data taken from snapshots of any earlier or later time. This similarity is tested by a certain time-dependent stochastic variable x.	Dynamic scaling refers to the evolution of self-similar systems, where data obtained from snapshots at fixed times is dissimilar to the respective data taken from snapshots of any earlier or later time. This dissimilarity is tested by a certain time-independent stochastic variable y.	Dynamic scaling refers to the non-evolution of self-similar systems, where data obtained from snapshots at fixed times is dissimilar to the respective data taken from snapshots of any earlier or later time. This dissimilarity is tested by a certain time-independent stochastic variable y.	Dynamic scaling refers to the evolution of self-similar systems, where data obtained from snapshots at fixed times is independent of the respective data taken from snapshots of any earlier or later time. This independence is tested by a certain time-dependent stochastic variable z.	A

数据概览

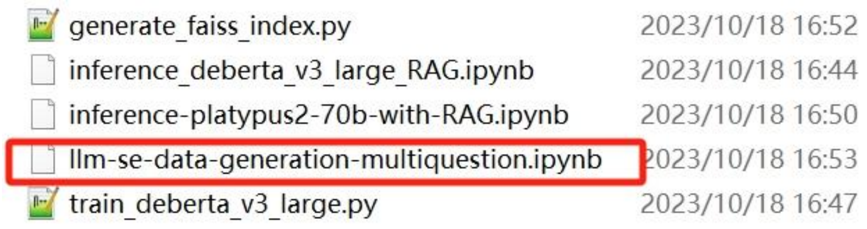
外部数据

STEM 文本语料库1：https://www.kaggle.com/datasets/mbanaei/all-paraphs-parsed-expanded
STEM 文本语料库2：https://www.kaggle.com/datasets/mbanaei/stem-wiki-cohere-no-emb
Wikipedia完整语料库：https://www.kaggle.com/datasets/jjinho/wikipedia-20230701

https://www.kaggle.com/datasets/cdeotte/60k-data-with-context-v2
https://www.kaggle.com/datasets/cdeotte/40k-data-with-context-v2
https://www.kaggle.com/datasets/cdeotte/99k-data-with-context-v2



名称	大小	修改时间
gte-base-public-13-context-322538.csv	1.9 GB	9月15日
gte-base-valid-context-970.csv	4.3 MB	9月15日



generate_faiss_index.py	2023/10/18 16:52
inference_deberta_v3_large_RAG.ipynb	2023/10/18 16:44
inference-platypus2-70b-with-RAG.ipynb	2023/10/18 16:50
llm-se-data-generation-multiquestion.ipynb	2023/10/18 16:53
train_deberta_v3_large.py	2023/10/18 16:47

算法思路

算法流程：基于RAG检索增强生成架构的自然语言问答模型

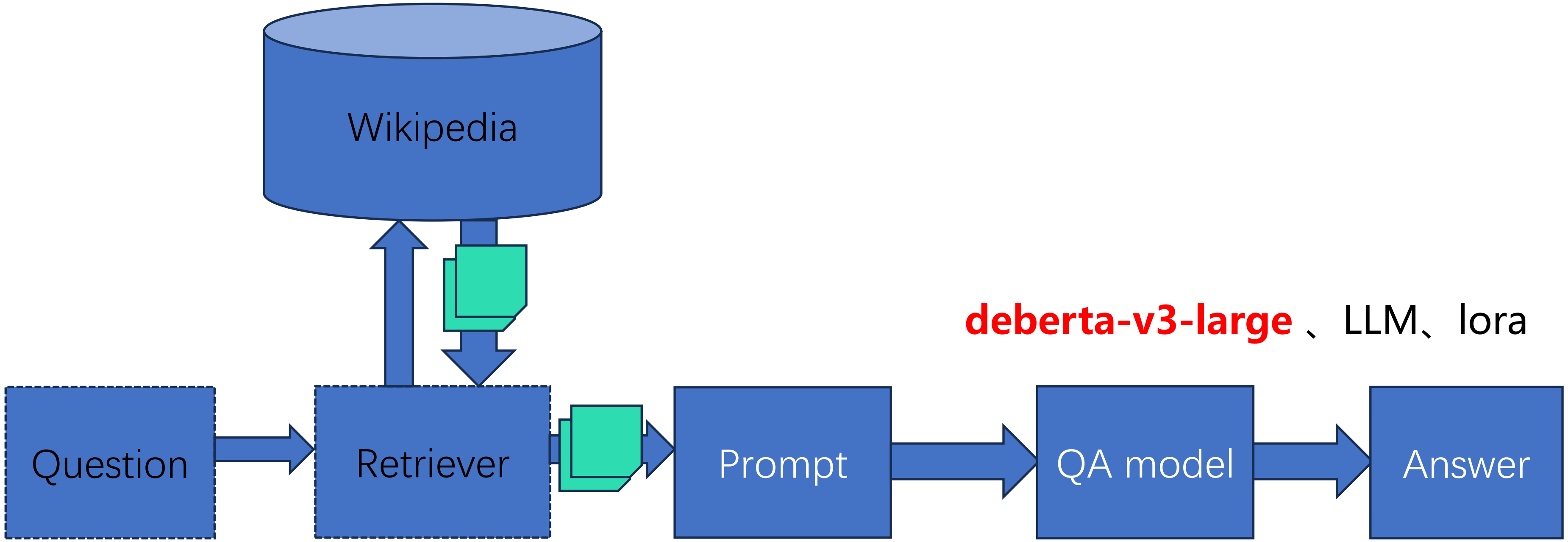
- **数据提取**：分别对样本提示语prompt和外部大型知识语料库进行清洗和整理，prompt重复三次与问题选项拼接构成检索文本（query），将语料库进行整理和区分后形成三种：完整的所有文本（full text）、STEM相关文本片段1、STEM相关文本片段2。
- **向量化**Embedding：分别对检索文本与三种文本语料库使用embedding模型gte-base生成embedding（特征向量）。
- **创建索引**（Index）：使用Faiss对三种文本语料库的embedding创建Index，以便后续查询。
- **检索**（Retrieval）：使用向量内积（Inner Product, IP）相似度进行语义相似搜索，使用query文本embedding逐一在三种文本语料库Index中查询，获取top10文本作为最终检索结果。
- **生成**（Generation）：将基于完整文本（full text）语料检索结果作为训练集文本外部知识库（context），由context（知识库）+ prompt（问题）+ option（选项）构成完整样本。
- **模型微调**（Finetune）：将所得完整样本输入deberta-v3-large模型进行五分类训练，基于下述形式进行RAG和风格行为词汇适配：

  五分类形式：外部知识文本{context}，请回答题目{question prompt }，选项{ option [1-5]}中最正确的是ABCDE中的哪一个?
- **推理答案**（Inference）：将在三种语料知识库得到检索文本分别输入微调后的deberta-v3-large模型预测每个选项概率，将预测结果按照加权平均得到最终的前三答案选项。

算法思路

算法流程图

Retrieval-augmented generation是一种针对知识密集型NLP任务的生成方法。 它通过在生成过程中引入检索组件，从已知的知识库中检索相关信息，并将这些信息与生成模型中的自然语言生成能力结合，从而提高生成的准确性和可靠性。在这里则是根据问题的信息**从Wikipedia知识库中检索相关的文本片段**，将这些文本片段与问题所有文本结合输入到问答模型里，让问答模型输出正确的选项。



Wikipedia

Question

Retriever

Prompt

QA model

Answer

deberta-v3-large 、LLM、lora

算法思路

Embedding 向量化

语义相似搜索是将文本片段进行比较，以找出包含最相似含义的文本的过程。虽然这对普通人来说似乎很容易，但语言是相当复杂的。将非结构化文本数据提炼成机器学习模型可以理解的格式一直是许多自然语言处理研究人员的研究主题。

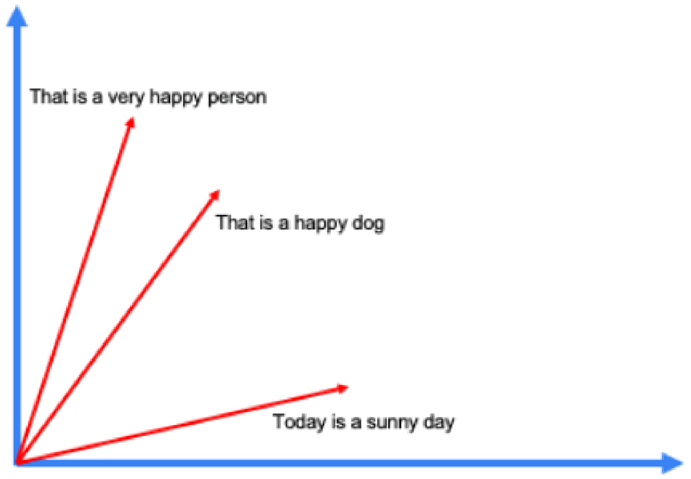
向量Embeddings为任何人提供了一种执行语义相似搜索的方法，而不仅仅是NLP研究人员或数据科学家。它们提供了一种有意义的、计算效率高的数字表示，可以通过预先训练的模型“开箱即用”来创建。下面是一个语义相似度的例子，它概述了用上面所示的sentence_transformers库创建的向量Embedding。

让我们看看下面的句子:

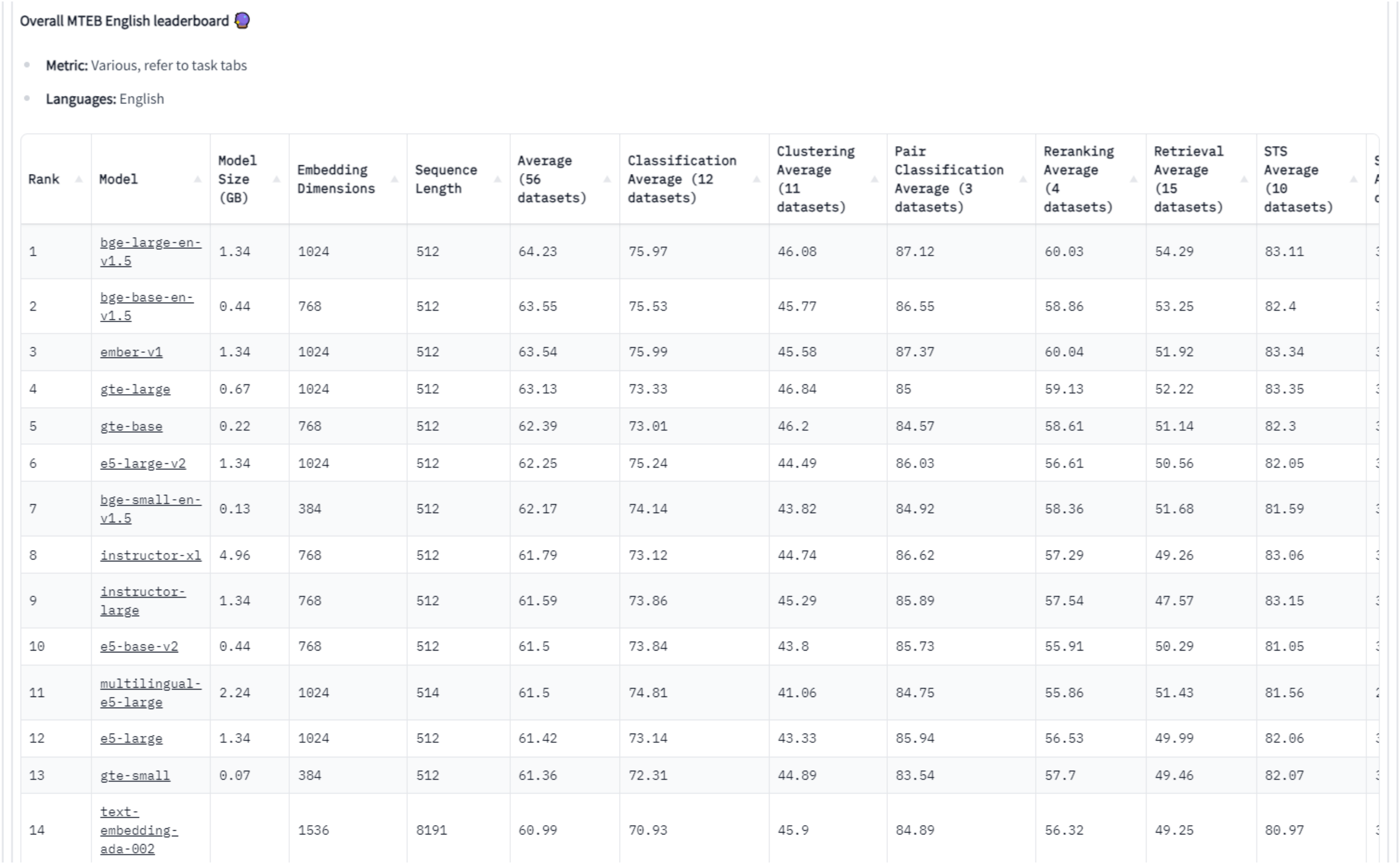
“That is a happy dog（那是一只快乐的狗）”

“That is a very happy person（那是一个非常幸福的人）”

“Today is a sunny day（今天是个晴天）”



Embedding 模型： gte-base



Overall MTEB English leaderboard 🔮

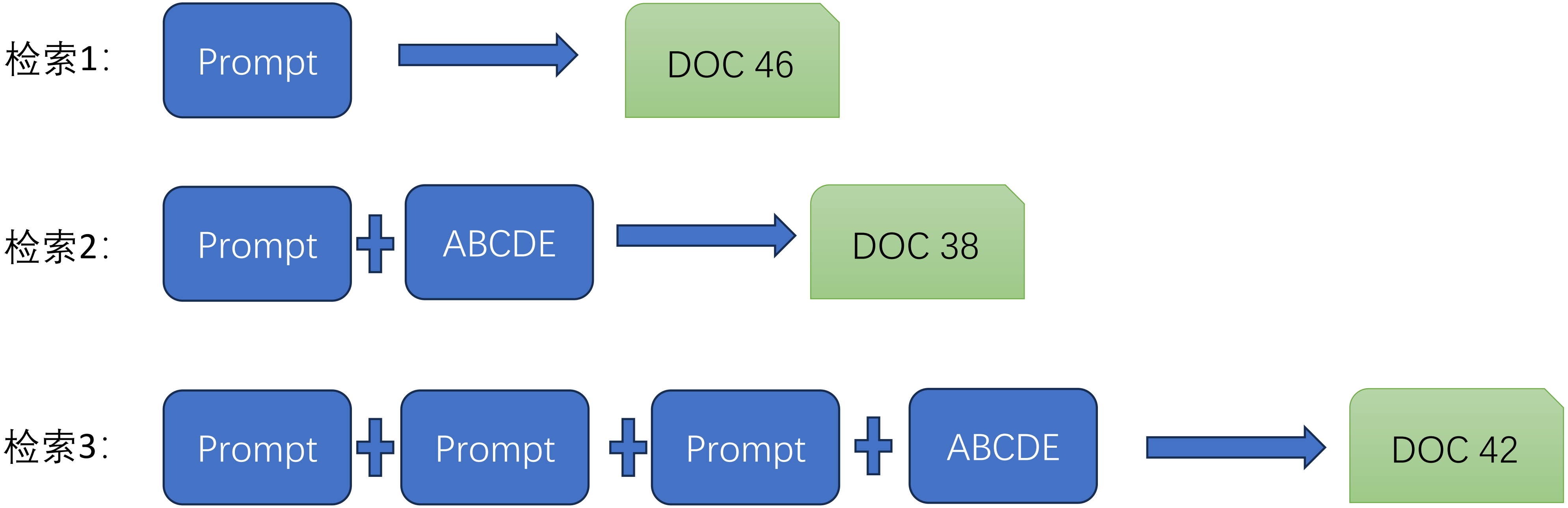
- **Metric:** Various, refer to task tabs
- **Languages:** English

Rank	Model	Model Size (GB)	Embedding Dimensions	Sequence Length	Average (56 datasets)	Classification Average (12 datasets)	Clustering Average (11 datasets)	Pair Classification Average (3 datasets)	Reranking Average (4 datasets)	Retrieval Average (15 datasets)	STS Average (10 datasets)
1	bge-large-en-v1.5	1.34	1024	512	64.23	75.97	46.08	87.12	60.03	54.29	83.11
2	bge-base-en-v1.5	0.44	768	512	63.55	75.53	45.77	86.55	58.86	53.25	82.4
3	ember-v1	1.34	1024	512	63.54	75.99	45.58	87.37	60.04	51.92	83.34
4	gte-large	0.67	1024	512	63.13	73.33	46.84	85	59.13	52.22	83.35
5	gte-base	0.22	768	512	62.39	73.01	46.2	84.57	58.61	51.14	82.3
6	e5-large-v2	1.34	1024	512	62.25	75.24	44.49	86.03	56.61	50.56	82.05
7	bge-small-en-v1.5	0.13	384	512	62.17	74.14	43.82	84.92	58.36	51.68	81.59
8	instructor-xl	4.96	768	512	61.79	73.12	44.74	86.62	57.29	49.26	83.06
9	instructor-large	1.34	768	512	61.59	73.86	45.29	85.89	57.54	47.57	83.15
10	e5-base-v2	0.44	768	512	61.5	73.84	43.8	85.73	55.91	50.29	81.05
11	multilingual-e5-large	2.24	1024	514	61.5	74.81	41.06	84.75	55.86	51.43	81.56
12	e5-large	1.34	1024	512	61.42	73.14	43.33	85.94	56.53	49.99	82.06
13	gte-small	0.07	384	512	61.36	72.31	44.89	83.54	57.7	49.46	82.07
14	text-embedding-ada-002		1536	8191	60.99	70.93	45.9	84.89	56.32	49.25	80.97

算法思路

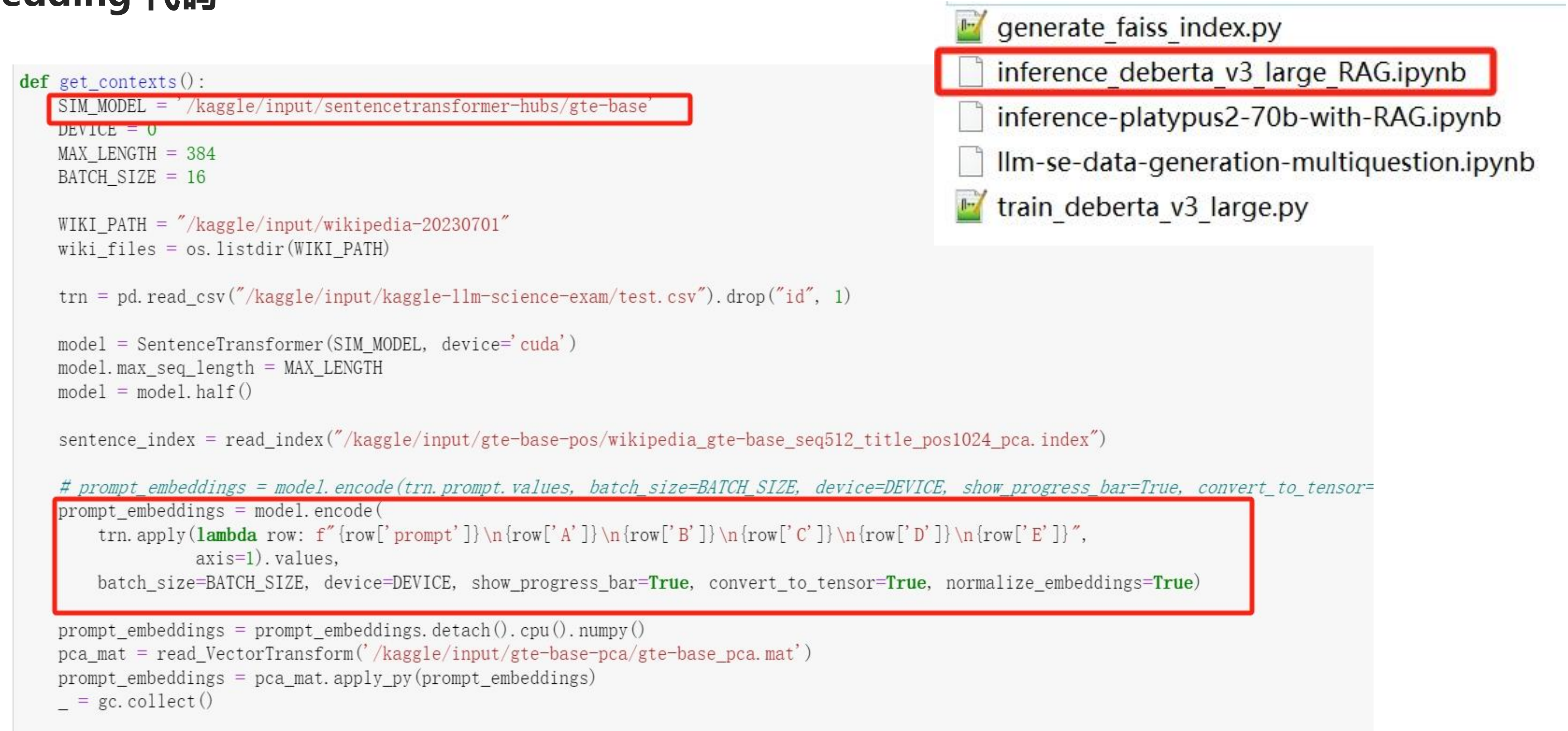
文本数据优化

RAG 将问答模型（QA model）的能力与特定数据联系起来。如果数据质量不高，那么整个系统将会受到影响。例如，使用的数据包含冲突或冗余信息，那么检索的过程将很难找到正确的上下文。当这种情况发生时，问答模型预测答案时则会产生误差。



算法思路

Embedding 代码

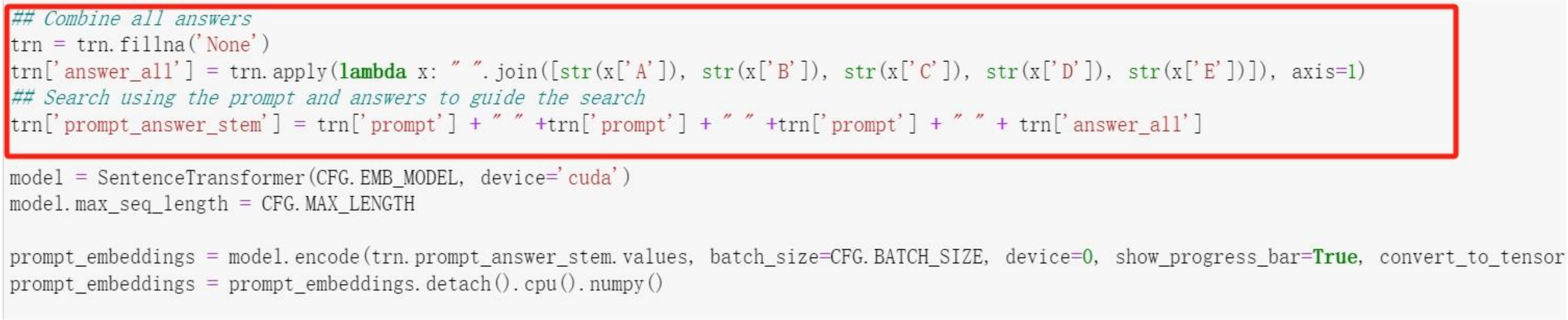


```
generate_faiss_index.py
inference_deberta_v3_large_RAG.ipynb
inference-platypus2-70b-with-RAG.ipynb
llm-se-data-generation-multiquestion.ipynb
train_deberta_v3_large.py
```

```python
def get_contexts():
    SIM_MODEL = '/kaggle/input/sentencetransformer-hubs/gte-base'
    DEVICE = 0
    MAX_LENGTH = 384
    BATCH_SIZE = 16

    WIKI_PATH = "/kaggle/input/wikipedia-20230701"
    wiki_files = os.listdir(WIKI_PATH)

    trn = pd.read_csv("/kaggle/input/kaggle-llm-science-exam/test.csv").drop("id", 1)

    model = SentenceTransformer(SIM_MODEL, device='cuda')
    model.max_seq_length = MAX_LENGTH
    model = model.half()

    sentence_index = read_index("/kaggle/input/gte-base-pos/wikipedia_gte-base_seq512_title_pos1024_pca.index")

    # prompt_embeddings = model.encode(trn.prompt.values, batch_size=BATCH_SIZE, device=DEVICE, show_progress_bar=True, convert_to_tensor=
    prompt_embeddings = model.encode(
        trn.apply(lambda row: f"{row['prompt']}\n{row['A']}\n{row['B']}\n{row['C']}\n{row['D']}\n{row['E']}",
                  axis=1).values,
        batch_size=BATCH_SIZE, device=DEVICE, show_progress_bar=True, convert_to_tensor=True, normalize_embeddings=True)

    prompt_embeddings = prompt_embeddings.detach().cpu().numpy()
    pca_mat = read_VectorTransform('/kaggle/input/gte-base-pca/gte-base_pca.mat')
    prompt_embeddings = pca_mat.apply_py(prompt_embeddings)
    _ = gc.collect()
```

算法思路

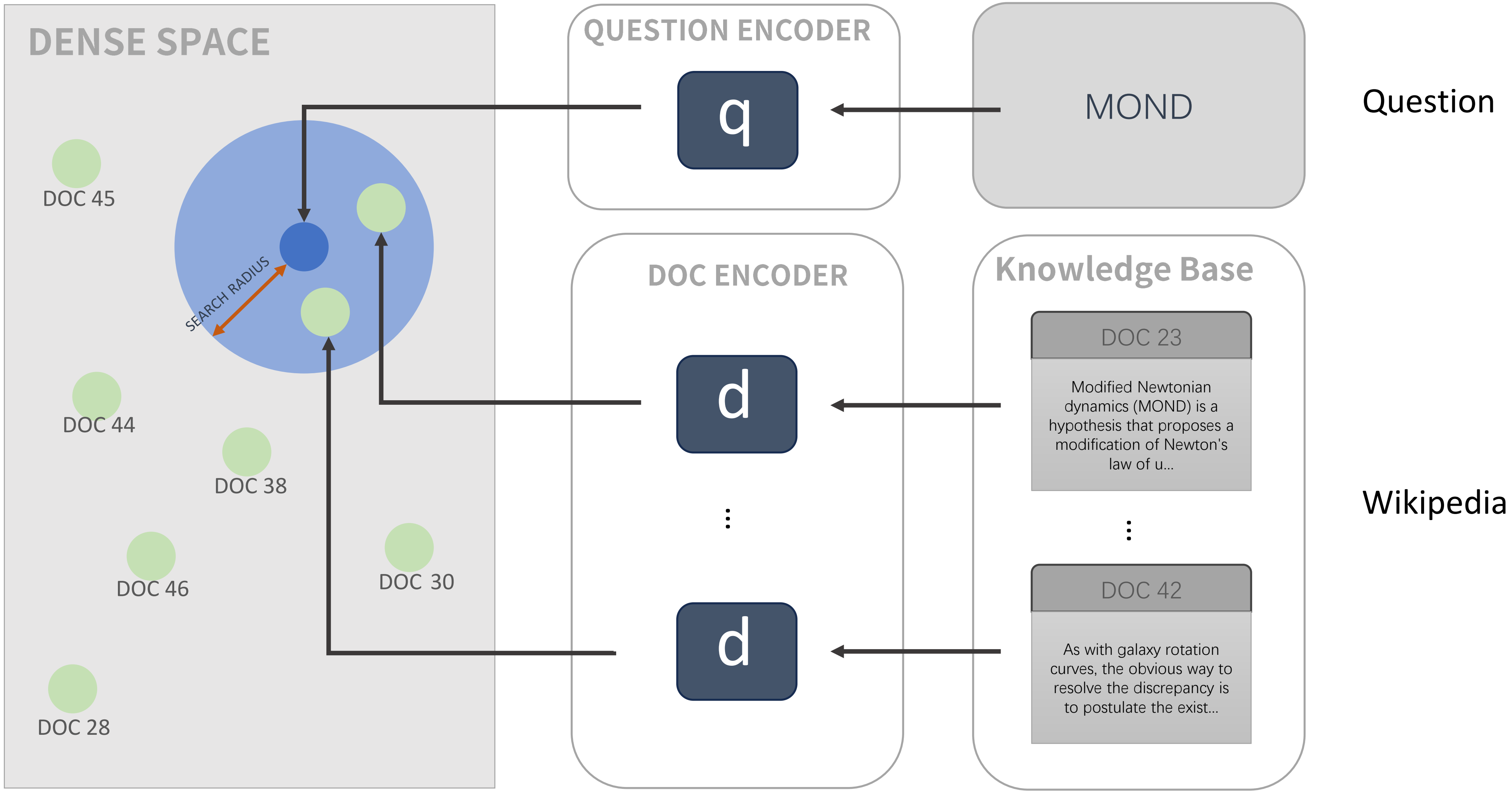
Embedding 代码



```
## Combine all answers
trn = trn.fillna('None')
trn['answer_all'] = trn.apply(lambda x: " ".join([str(x['A']), str(x['B']), str(x['C']), str(x['D']), str(x['E'])]), axis=1)
## Search using the prompt and answers to guide the search
trn['prompt_answer_stem'] = trn['prompt'] + " " +trn['prompt'] + " " +trn['prompt'] + " " + trn['answer_all']

model = SentenceTransformer(CFG.EMB_MODEL, device='cuda')
model.max_seq_length = CFG.MAX_LENGTH

prompt_embeddings = model.encode(trn.prompt_answer_stem.values, batch_size=CFG.BATCH_SIZE, device=0, show_progress_bar=True, convert_to_tensor
prompt_embeddings = prompt_embeddings.detach().cpu().numpy()
```

算法思路

检索过程

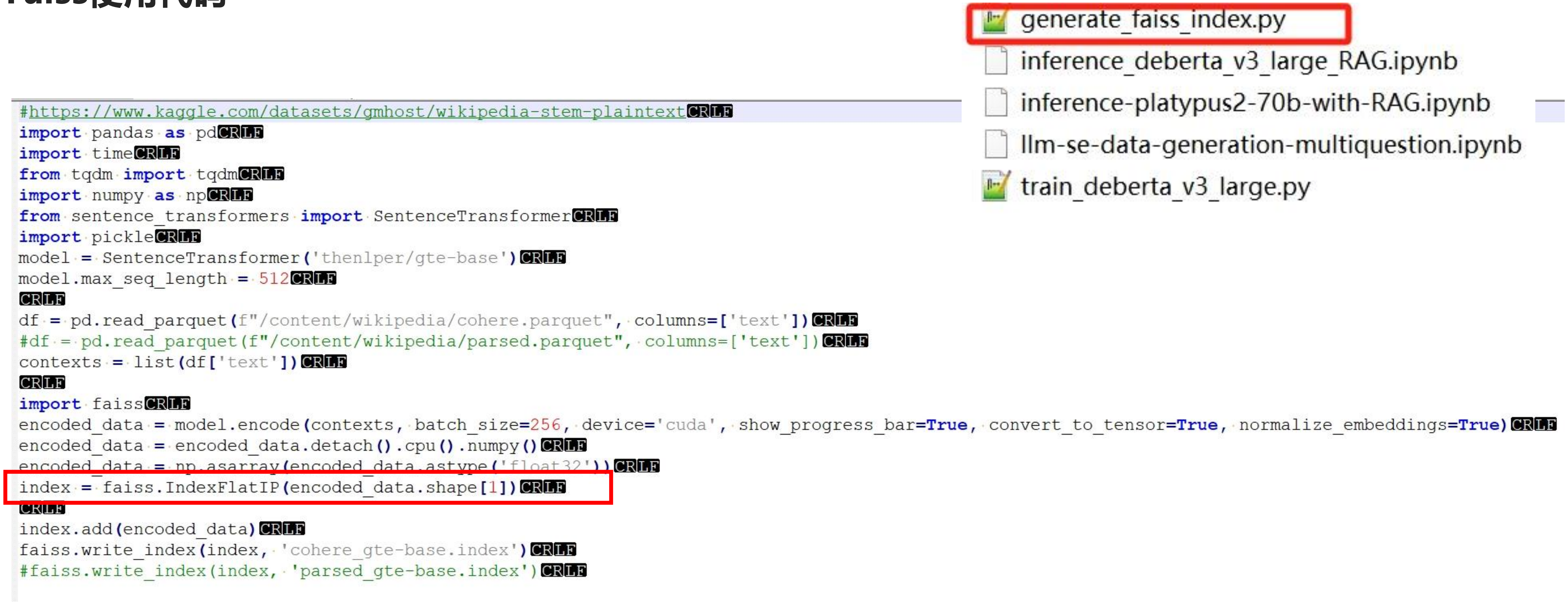

DENSE SPACE
DOC 45
SEARCH RADIUS
DOC 44
DOC 38
DOC 46
DOC 30
DOC 28
QUESTION ENCODER
q
MOND
Question
DOC ENCODER
d
⋮
d
Knowledge Base
DOC 23
Modified Newtonian dynamics (MOND) is a hypothesis that proposes a modification of Newton's law of u...
⋮
DOC 42
As with galaxy rotation curves, the obvious way to resolve the discrepancy is to postulate the exist...
Wikipedia

Faiss:Inner Product

GTE-base:Embedding

算法思路

Faiss使用代码

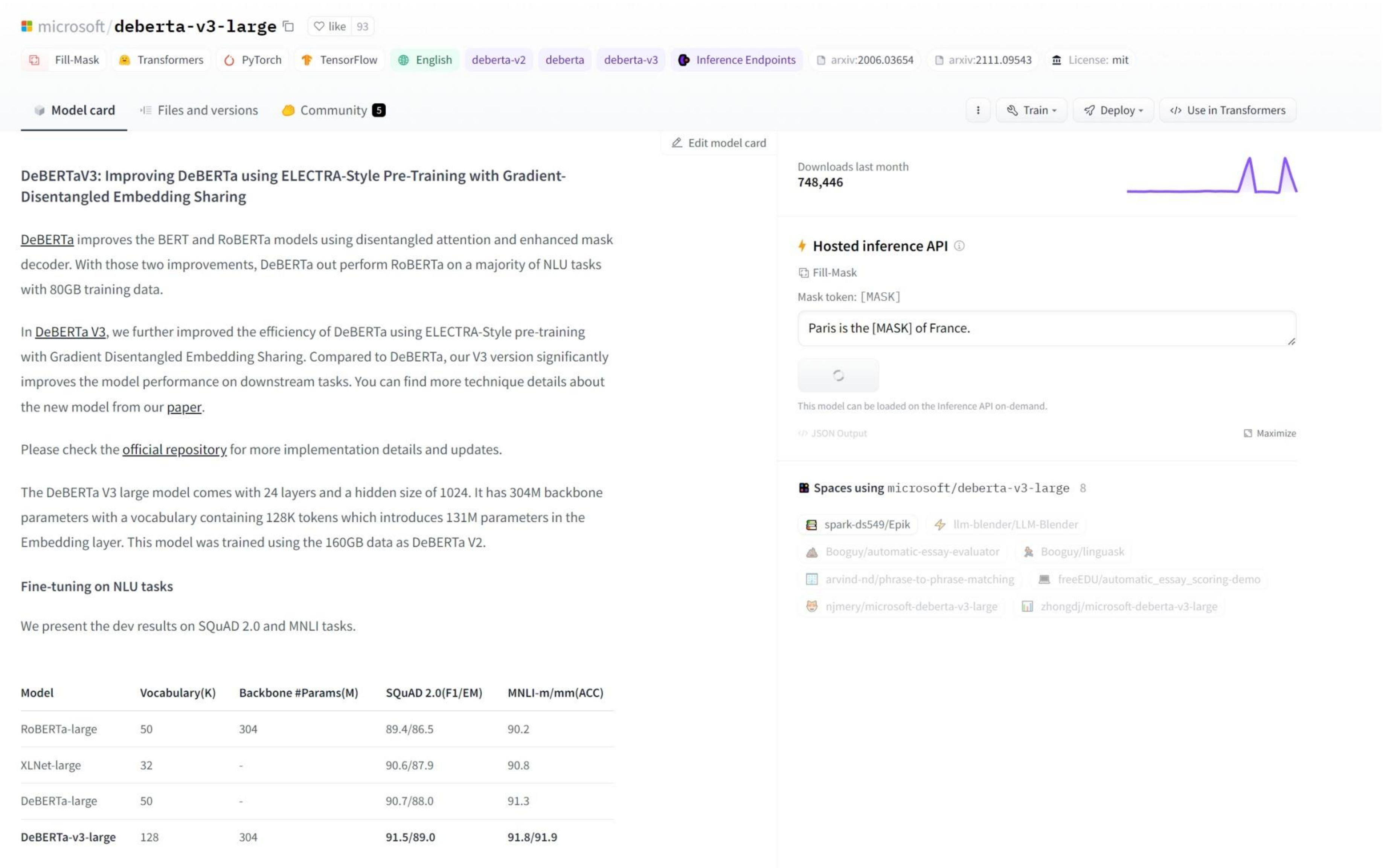


generate_faiss_index.py
inference_deberta_v3_large_RAG.ipynb
inference-platypus2-70b-with-RAG.ipynb
llm-se-data-generation-multiquestion.ipynb
train_deberta_v3_large.py

```
#https://www.kaggle.com/datasets/gmhost/wikipedia-stem-plaintext
import pandas as pd
import time
from tqdm import tqdm
import numpy as np
from sentence_transformers import SentenceTransformer
import pickle
model = SentenceTransformer('thenlper/gte-base')
model.max_seq_length = 512

df = pd.read_parquet(f"/content/wikipedia/cohere.parquet", columns=['text'])
#df = pd.read_parquet(f"/content/wikipedia/parsed.parquet", columns=['text'])
contexts = list(df['text'])

import faiss
encoded_data = model.encode(contexts, batch_size=256, device='cuda', show_progress_bar=True, convert_to_tensor=True, normalize_embeddings=True)
encoded_data = encoded_data.detach().cpu().numpy()
encoded_data = np.asarray(encoded_data.astype('float32'))
index = faiss.IndexFlatIP(encoded_data.shape[1])

index.add(encoded_data)
faiss.write_index(index, 'cohere_gte-base.index')
#faiss.write_index(index, 'parsed_gte-base.index')
```

算法思路

QA Model



DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing

DeBERTa improves the BERT and RoBERTa models using disentangled attention and enhanced mask decoder. With those two improvements, DeBERTa out perform RoBERTa on a majority of NLU tasks with 80GB training data.

In DeBERTa V3, we further improved the efficiency of DeBERTa using ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing. Compared to DeBERTa, our V3 version significantly improves the model performance on downstream tasks. You can find more technique details about the new model from our paper.

Please check the official repository for more implementation details and updates.

The DeBERTa V3 large model comes with 24 layers and a hidden size of 1024. It has 304M backbone parameters with a vocabulary containing 128K tokens which introduces 131M parameters in the Embedding layer. This model was trained using the 160GB data as DeBERTa V2.

Fine-tuning on NLU tasks

We present the dev results on SQuAD 2.0 and MNLI tasks.

Model	Vocabulary(K)	Backbone #Params(M)	SQuAD 2.0(F1/EM)	MNLI-m/mm(ACC)
RoBERTa-large	50	304	89.4/86.5	90.2
XLNet-large	32	-	90.6/87.9	90.8
DeBERTa-large	50	-	90.7/88.0	91.3
DeBERTa-v3-large	128	304	**91.5/89.0**	**91.8/91.9**

Paris is the [MASK] of France.

This model can be loaded on the Inference API on-demand.

算法思路

总结

- RAG（检索增强） 通过为LLM提供回答查询时使用的事实背景，使LLM变得更加实用。

- 在实际项目中，提示词是非常脆弱和敏感的，当前的大模型对提示具有非常高的依赖性，这种依赖性与模型的能力成反比，也就是模型的能力越弱，对提示的依赖越强。选择不同的模型、不同的数据，甚至不同的索引，都需要调整提示来得到一个比较优秀的结果。

- 基于向量化的相似性是 RAG 的标准检索机制，提高Embedding model将使得LLM性能更高。