Задание

взять данные из
https://www.kaggle.com/datasets/mrapplexz/bashim-quotes
(https://www.kaggle.com/datasets/mrapplexz/bashim-quotes)
обучить модель GPT для генерации своих цитат

взять новостные данные из
https://github.com/natasha/corus (https://github.com/natasha/corus)
load_lenta2
нам понадобиться сам текст и заголовок
обучить модель T5/ или GPT для генерации заголовков для статей

загрузим данные

https://www.kaggle.com/datasets/mrapplexz/bashim-quotes
(https://www.kaggle.com/datasets/mrapplexz/bashim-quotes)

In [35]:

```
# model_name = "bankholdup/rugpt3_song_writer"
model_name = "sberbank-ai/rugpt3small_based_on_gpt2"
```

In [36]:

```
import numpy as np
import pandas as pd
```

In [37]:

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [38]:

```
import logging
from transformers.trainer import logger as noisy_logger
noisy_logger.setLevel(logging.WARNING)
```

In [39]:

```
df_rec = pd.read_json('dataset.jsonl', lines=True).set_index('id')
```

In [40]:

```
df_rec.shape
```

Out[40]:

```
(81497, 3)
```

In [41]:

```
df_rec = df_rec.sample(10000)
```

In [42]:

```
import re

def clear_text(text):
    clr_text = re.sub(r"<.*?>", " ", text).lower()
    clr_text = summary = re.sub(r"\s", " ", clr_text)
    return clr_text
```

In [43]:

```
df_rec["clear_text"] = df_rec["text"].apply(lambda x: clear_text(x))
df_rec.head()
```

Out[43]:

id	date	rating	text	clear_text
451487	2018-07-27 07:12:00+00:00	2262.0	Кккротов: опечатка "зов передков" точнее отра...	кккротов: опечатка "зов передков" точнее отра...
398613	2008-08-22 10:54:00+00:00	38421.0	Анна**ЗАКОНЧИЛА ШКОЛУ!!!"Викторова вступила в ...	анна**закончила школу!!!"викторова вступила в ...
456044	2019-05-29 07:13:00+00:00	2362.0	stupidchemist: На моё место "проще химиков раз...	stupidchemist: на моё место "проще химиков раз...
431520	2014-12-16 08:12:00+00:00	14257.0	XXX: Мне нравилось в 90-е, я бы с удовольствие...	xxx: мне нравилось в 90-е, я бы с удовольствие...
397561	2008-06-30 13:54:00+00:00	28256.0	Восторженный возглас нашего препода по логике:...	восторженный возглас нашего препода по логике:...

In [44]:

```
data = df_rec.loc[:, 'clear_text']
```

In [45]:

```
data
```

Out[45]:

```
id
451487    ккккротов: опечатка "зов передков" точнее отра...
398613    анна**закончила школу!!!"викторова вступила в ...
456044    stupidchemist: на моё место "проще химиков раз...
431520    xxx: мне нравилось в 90-е, я бы с удовольствие...
397561    восторженный возглас нашего препода по логике:...
                                ...
417334    na-ta: срочно! затопили соседей! хелп! после 8...
430027    xxx: нашёл в ленте крик души какого-то школьни...
442396    xxx: калифорния устраивает референдум по отсое...
399826    xxx хочу 201розу  xxx не выйду за тебя,пока ты...
414686    xxx: сидим в одной комнате с начальником отдел...
Name: clear_text, Length: 10000, dtype: object
```

In [46]:

```
import re
from sklearn.model_selection import train_test_split

def build_text_files(data_json, dest_path):
#     f = open(dest_path, 'w')
    with open(dest_path, "w", encoding="utf-8") as f:
        data = ''
        for texts in data_json:
            summary = str(texts).strip()
    #         summary = re.sub(r"<.*?>", " ", summary)
    #         summary = re.sub(r"\s", " ", summary)
            data += summary + "  "
    #     with open(fname, "w", encoding="utf-8") as f:
    #     f.write(html)
        f.write(data)
```

In [47]:

```
train, test = train_test_split(data, test_size=0.15)
```

In [48]:

```
build_text_files(train,'train_dataset.txt')
build_text_files(test,'test_dataset.txt')
```

In [49]:

```
print("Train dataset length: "+ str(len(train)))
print("Test dataset length: "+ str(len(test)))
```

```
Train dataset length: 8500
Test dataset length: 1500
```

In [50]:

```
train[:5]
```

Out[50]:

```
id
415125    эоп: допустим, я хочу телепортироваться из точ...
392895    xxx: я сегодня вычислил, за сколько пылесос вы...
431520    xxx: мне нравилось в 90-е, я бы с удовольствие...
226142    nansee: вот крашусь и думаю: а ведь в фотошопе...
399450    катерина >>  мих, а что такое рапидшара?  tera...
Name: clear_text, dtype: object
```

In [51]:

```
from transformers import AutoTokenizer
#sberbank-ai/rugpt3large_based_on_gpt2
#sberbank-ai/rugpt3medium_based_on_gpt2
#sberbank-ai/rugpt3small_based_on_gpt2

tokenizer = AutoTokenizer.from_pretrained(model_name)

train_path = 'train_dataset.txt'
test_path = 'test_dataset.txt'
```

```
Special tokens have been added in the vocabulary, make sure the associated w
ord embeddings are fine-tuned or trained.
```

In [52]:

```
from transformers import TextDataset, DataCollatorForLanguageModeling

def load_dataset(train_path, test_path, tokenizer):
    train_dataset = TextDataset(
          tokenizer=tokenizer,
          file_path=train_path,
          block_size=128)

    test_dataset = TextDataset(
          tokenizer=tokenizer,
          file_path=test_path,
          block_size=128)

    data_collator = DataCollatorForLanguageModeling(
        tokenizer=tokenizer, mlm=False,
    )
    return train_dataset, test_dataset, data_collator

train_dataset, test_dataset, data_collator = load_dataset(train_path, test_path, tokenizer)
```

```
C:\Users\voron\AppData\Roaming\Python\Python37\site-packages\transformers\da
ta\datasets\language_modeling.py:58: FutureWarning: This dataset will be rem
oved from the library soon, preprocessing should be handled with the 🤗 Data
sets library. You can have a look at this example script for pointers: http
s://github.com/huggingface/transformers/blob/main/examples/pytorch/language-
modeling/run_mlm.py (https://github.com/huggingface/transformers/blob/main/e
xamples/pytorch/language-modeling/run_mlm.py)
  FutureWarning,
```

Fine-tuning the model

In [53]:

```
from transformers import Trainer, TrainingArguments, AutoModelForCausalLM

model = AutoModelForCausalLM.from_pretrained(model_name)
```

```
Downloading pytorch_model.bin:   0%|          | 0.00/526M [00:00<?, ?B/s]
```

In [54]:

```
training_args = TrainingArguments(

    "phrase",
    evaluation_strategy = "epoch",
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    num_train_epochs=2,
    learning_rate=1e-5,
    weight_decay=0.01,
    save_strategy='no',
    report_to='none',

    )
```

```
C:\Users\voron\AppData\Roaming\Python\Python37\site-packages\torch\cuda\__in
it__.py:80: UserWarning: CUDA initialization: The NVIDIA driver on your syst
em is too old (found version 10010). Please update your GPU driver by downlo
ading and installing a new version from the URL: http://www.nvidia.com/Downl
oad/index.aspx (http://www.nvidia.com/Download/index.aspx) Alternatively, go
to: https://pytorch.org (https://pytorch.org) to install a PyTorch version t
hat has been compiled with your version of the CUDA driver. (Triggered inter
nally at  ..\c10\cuda\CUDAFunctions.cpp:112.)
  return torch._C._cuda_getDeviceCount() > 0
```

In [55]:

```
trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=train_dataset,
    eval_dataset=test_dataset
)
```

In [56]:

```
trainer.train()
```

```
C:\Users\voron\AppData\Roaming\Python\Python37\site-packages\transformers\op
timization.py:310: FutureWarning: This implementation of AdamW is deprecated
and will be removed in a future version. Use the PyTorch implementation torc
h.optim.AdamW instead, or set `no_deprecation_warning=True` to disable this
warning
  FutureWarning,
```

[2190/2190 7:02:43, Epoch 2/2]

Epoch	Training Loss	Validation Loss
1	4.091000	3.931326
2	3.965500	3.919243

Out[56]:

```
TrainOutput(global_step=2190, training_loss=4.033529481931364, metrics={'tra
in_runtime': 25376.7375, 'train_samples_per_second': 0.345, 'train_steps_per
_second': 0.086, 'total_flos': 572098904064000.0, 'train_loss': 4.0335294819
31364, 'epoch': 2.0})
```

generate text

In [59]:

```
def generate_text(prefix):
    tokens = tokenizer(prefix, return_tensors='pt')
    size = tokens['input_ids'].shape[1]

    output = model.generate(
        **tokens,
        #end_token=end_token_id,
        do_sample=False,
        max_length=size+50,
        early_stopping=True,
        length_penalty=2.0,
        repetition_penalty=8.,
        temperature=0.5,
        num_beams=3,
        no_repeat_ngram_size=5
    )

    decoded = tokenizer.decode(output[0])
    result = decoded[len(prefix):]
    return prefix + result
```

In [60]:

```
print(generate_text("ну ты собираешься идти?"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

ну ты собираешься идти?  xxx: у нас в городе есть один магазин, где можно ку
пить все что душе угодно. yyy: а я вот не могу себе позволить такую роскошь
- покупать всякую фигню на развес и ходить с ней по магазину...
```

In [61]:

```
print(generate_text("заводи, поехали"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

заводи, поехали.  xxx: а у нас в городе есть такой магазинчик - "сувениры".
там можно купить что-нибудь на память о детстве и юности... yyy: ну так вот
я тебе сейчас расскажу историю про то, как
```

In [62]:

```
print(generate_text("захвати в магазине что-нибудь к чаю"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

захвати в магазине что-нибудь к чаю.  xxx: у меня есть знакомый, который раб
отает на заводе по переработке нефти и газа (входит в холдинг "нефтегазовая
компания") - он очень любит свою работу! yyy: а я вот не люблю
```

In [ ]:

```

```