上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

推特文本情感分析

第五组：张露雨　高鹏轩　邹佳俊

2022年12月30日

目 录

CONTENTS

01

任务简介

- 任务介绍
- 数据探索

1.1任务简介

任务介绍

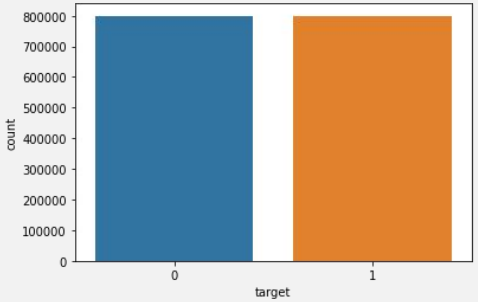
情感分析一般指判断一段文本所表达的情绪状态，属于文本分类问题。主要任务是对文本中的主观信息（如观点、情感、评价、态度、情绪等）进行提取、分析、处理、归纳和推理。

本次任务中，我们需要通过模型对一段文本进行情绪的正负判断，可以作为一个简单的二分类任务。

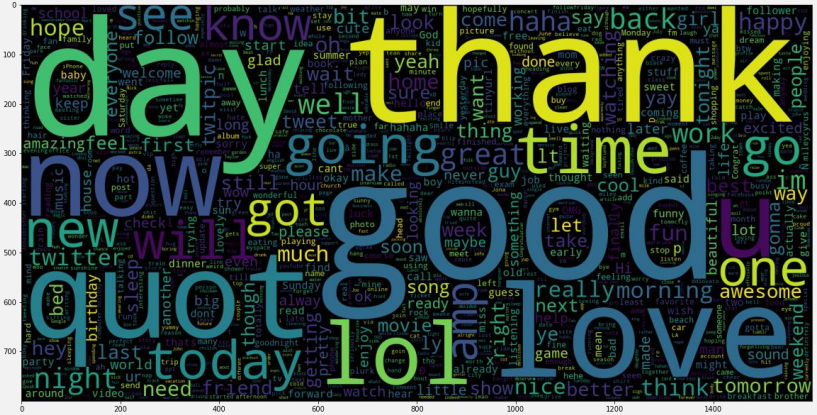
数据集介绍

训练集160万条；
六个维度：标签、ID、日期、话题、用户及推文内容；
无缺失数据；
正负样本平衡。

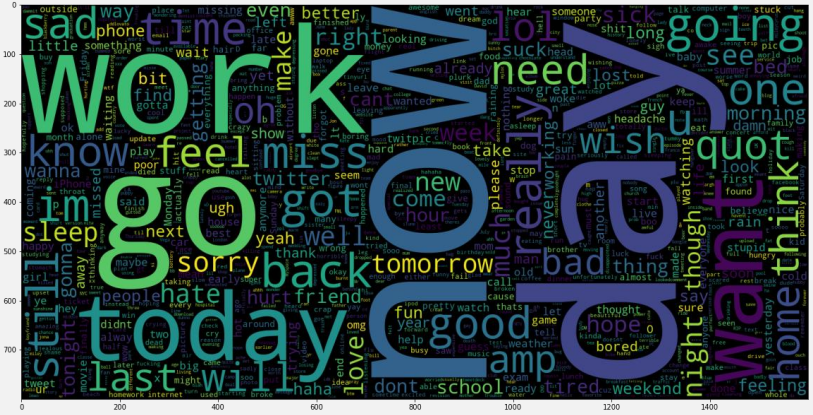
1.2 数据探索



正负样本比例



正样本词云图



负样本词云图

02

实验思路

• 机器学习 • BERT • GPT2

2 实验思路

多种尝试

1. 传统分类
   - 逻辑回归：LogisticRegression()
   - 支持向量机：LinearSVC()
   - 随机梯度下降：SGDClassifier()
   - 集成学习：VotingClassifier()
2. BERT微调
   - 基于预训练大模型BERT，并在本次数据集上进行训练微调
   - from_pretrained('bert-base-uncased')
3. GPT2微调
   - 基于预训练大模型GPT2，并在本次数据集上进行训练微调
   - model_name = "gpt2"

03

过程及结果

• 传统分类 • BERT • GPT2

3.1 传统分类

预处理: 转小写-->去停用词-->去标点-->去链接-->去数字-->TFIDF

逻辑回归

模型超参：L2正则化
正则化系数λ的倒数C=2
优化算法：solver=sag

支持向量机

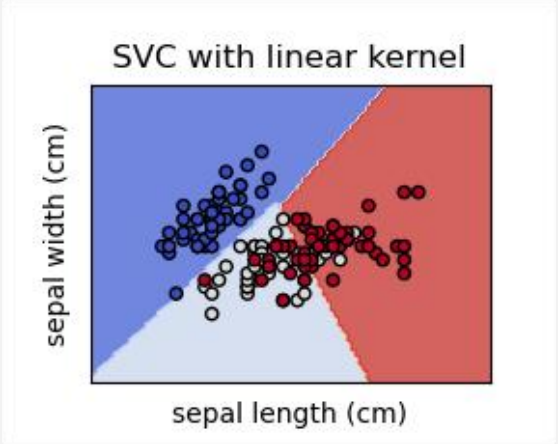
模型超参：L2正则化
损失函数：loss=squared_hinge
最大迭代次数：max_iter=1000

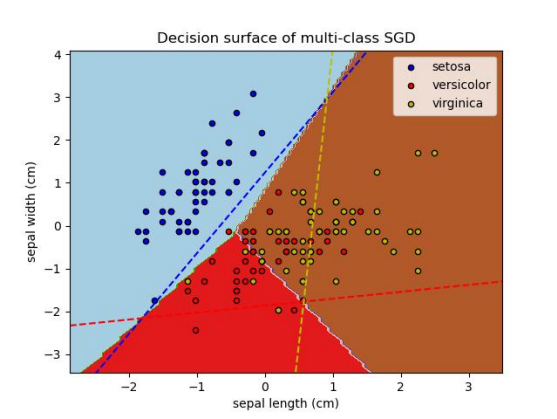
随机梯度下降

模型超参：L2正则化
损失函数：loss='log'
最大迭代次数：max_iter=1000

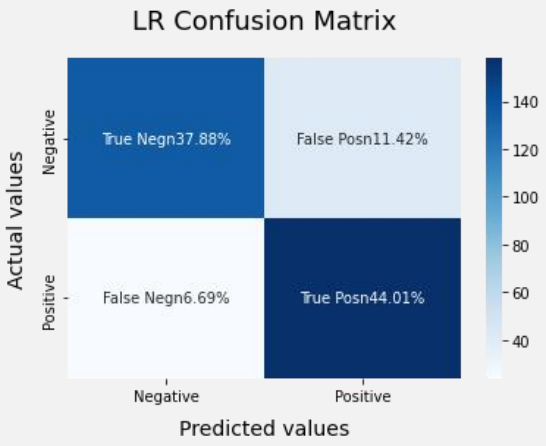
集成学习

模型超参：投票方法：voting='hard'

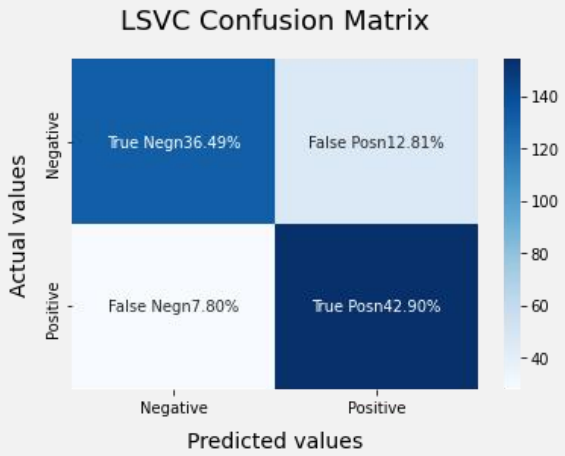




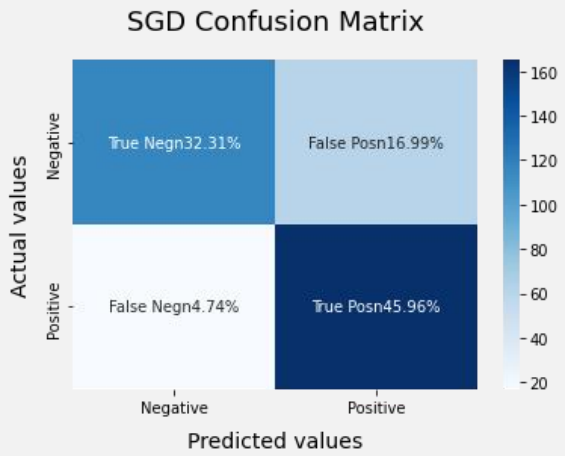
3.1 传统分类



accuracy=0.82
time: 20.6s

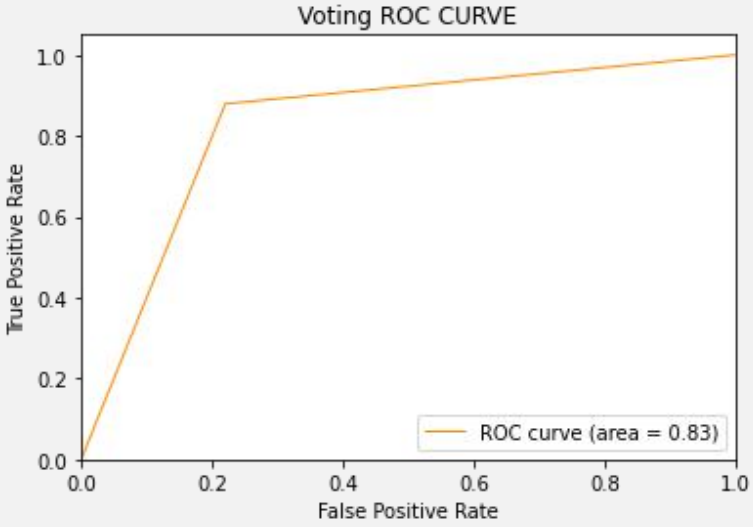


accuracy=0.79
time: 15.3s



accuracy=0.78
time: 4.1s

3.1传统分类

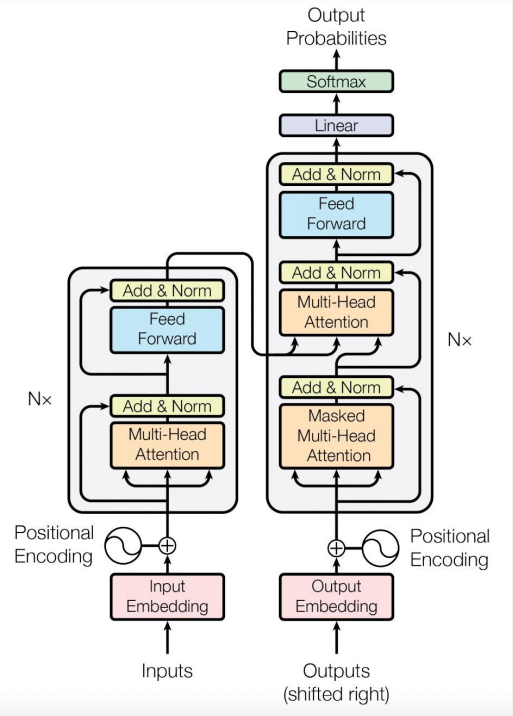


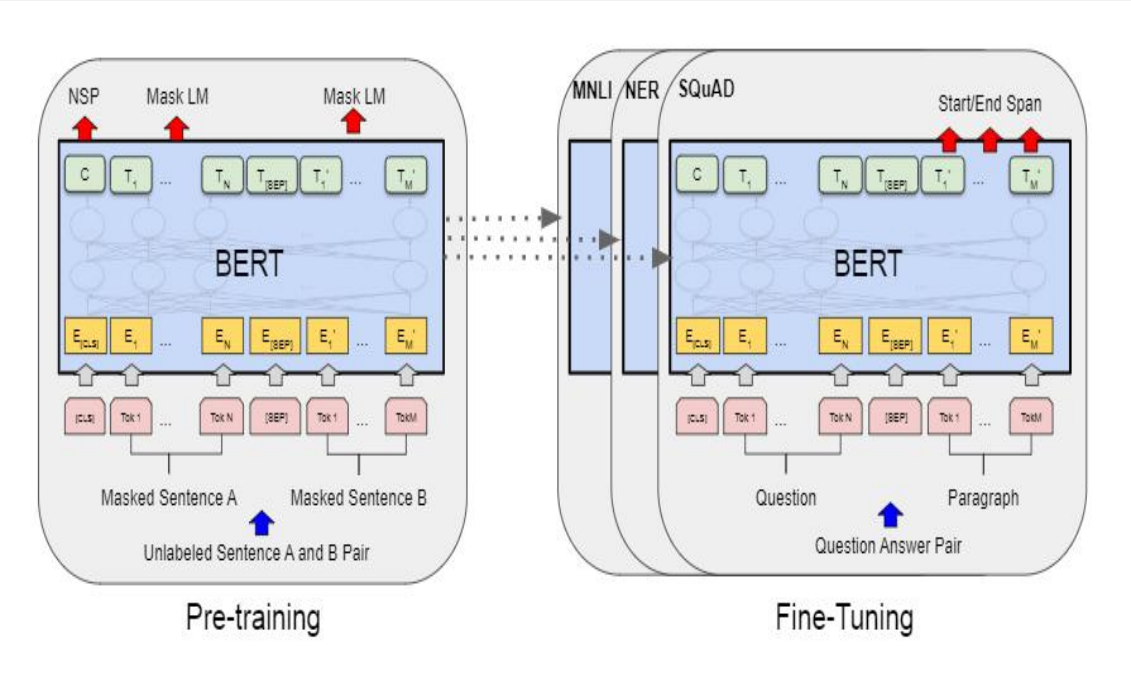
accuracy=0.83
time：15.2s

	precision	recall	f1-score	support
0	0.86	0.78	0.82	177
1	0.80	0.88	0.84	182
accuracy			0.83	359
macro avg	0.83	0.83	0.83	359
weighted avg	0.83	0.83	0.83	359

集成学习提高了准确率，对于时间的影响不大

3. 2 BERT微调

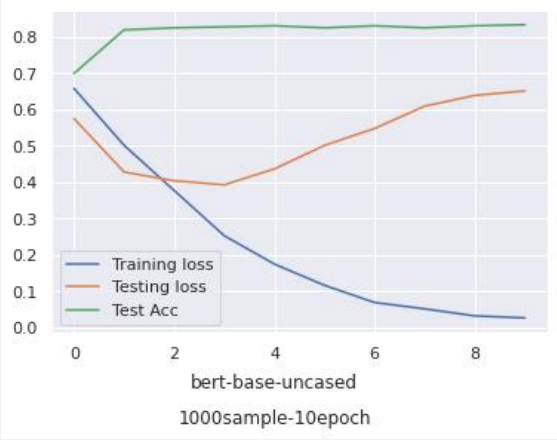




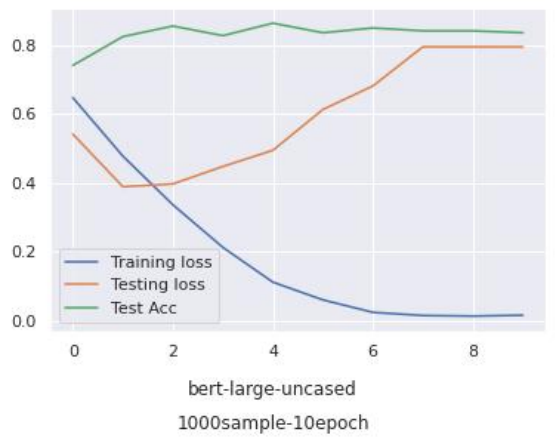
arXiv:1810.04805v2

3.2 BERT微调

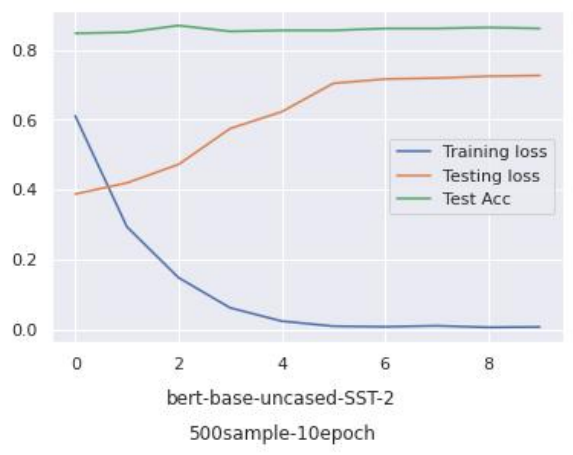
预处理:转小写-->去停用词-->去标点-->去链接-->去数字



Accuracy=0.832

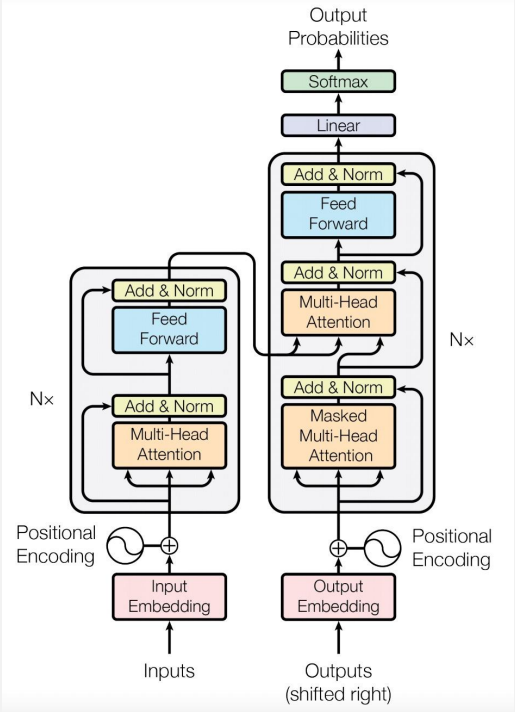


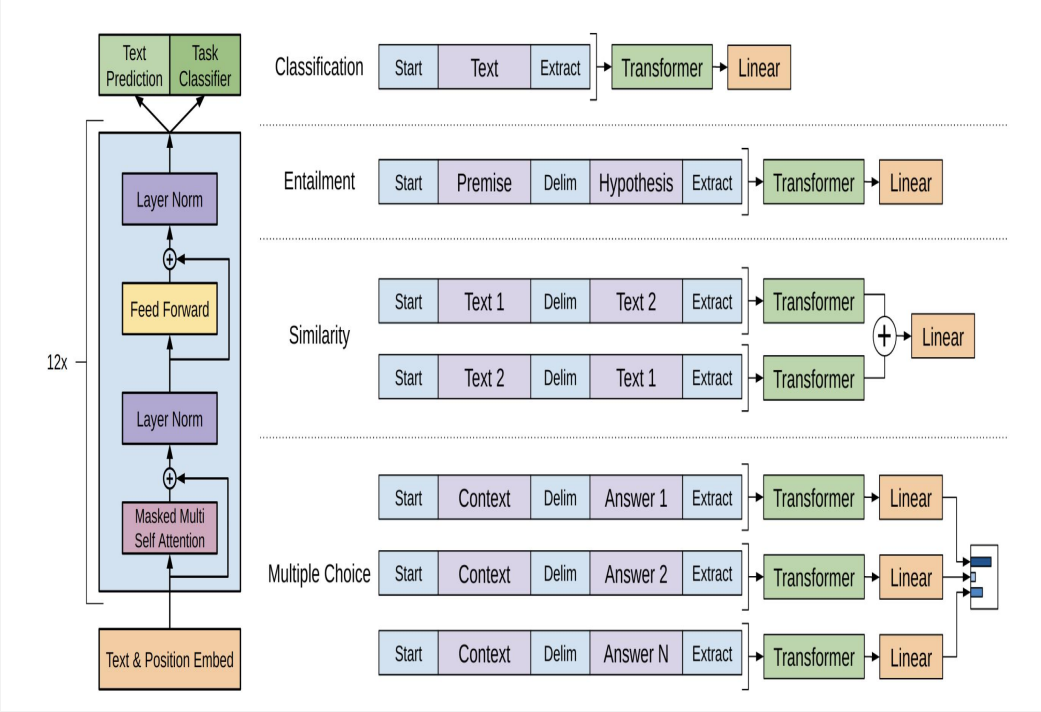
Accuracy=0.8635



Accuracy=0.869

3.3 GPT2微调


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

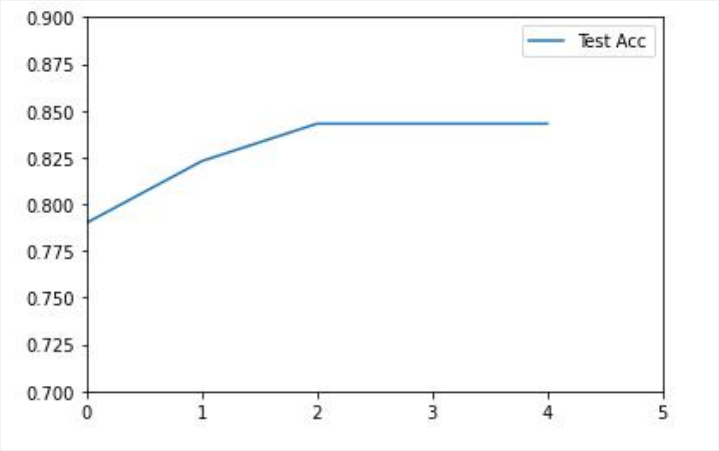

Text Prediction
Task Classifier
12x
Layer Norm
Feed Forward
Layer Norm
Masked Multi Self Attention
Text & Position Embed
Classification
Start
Text
Extract
Transformer
Linear
Entailment
Start
Premise
Delim
Hypothesis
Extract
Transformer
Linear
Similarity
Start
Text 1
Delim
Text 2
Extract
Transformer
Start
Text 2
Delim
Text 1
Extract
Transformer
Linear
Multiple Choice
Start
Context
Delim
Answer 1
Extract
Transformer
Linear
Start
Context
Delim
Answer 2
Extract
Transformer
Linear
Start
Context
Delim
Answer N
Extract
Transformer
Linear

3.3 GPT2微调

数据： df = df.sample(1000)

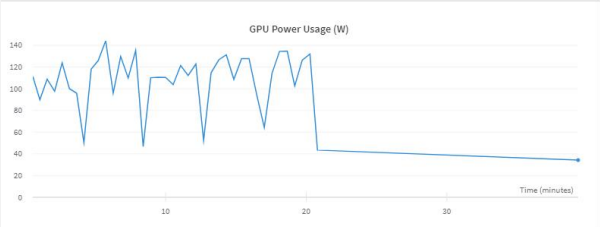
```
for trial_no in range(5):

    print("Loading model...")
    # load tokenizer and model
    tokenizer = GPT2Tokenizer.from_pretrained(model_name, bos_token='<|startoftext|>',
                                              eos_token='<|endoftext|>', pad_token='<|pad|>')
    model = GPT2LMHeadModel.from_pretrained(model_name).cuda()
    model.resize_token_embeddings(len(tokenizer))

    print("Loading dataset...")
    train_dataset, test_dataset = load_sentiment_dataset(tokenizer, trial_no)

    print("Start training...")
    training_args = TrainingArguments(output_dir='results', num_train_epochs=2,
                                      logging_steps=200, load_best_model_at_end=True,
                                        save_strategy="epoch", per_device_train_batch_size=2, per_device_eval_batch_size=2,
                                      warmup_steps=100, weight_decay=0.01, logging_dir='logs')

    Trainer(model=model, args=training_args, train_dataset=train_dataset,
            eval_dataset=test_dataset, data_collator=lambda data: {'input_ids': torch.stack([f[0] for f in data]),
                                                                   'attention_mask': torch.stack([f[1] for f in data]),
                                                                   'labels': torch.stack([f[0] for f in data])}).train()

    # test
    print("Start testing...")
    # eval mode on model
    _ = model.eval()
```

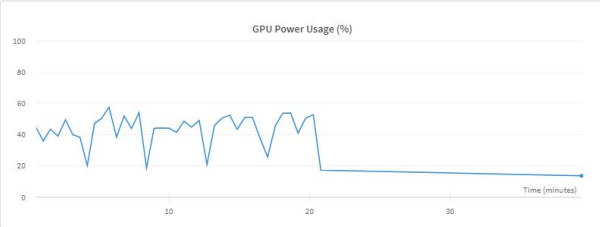
参数设置

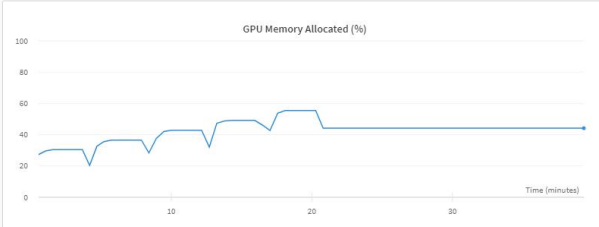


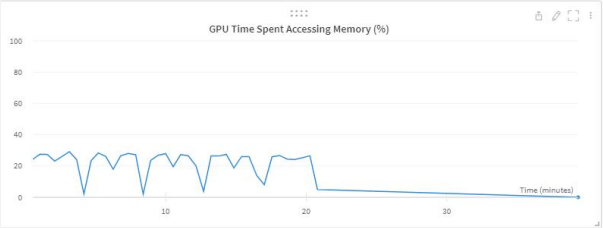
Accuracy=0.843

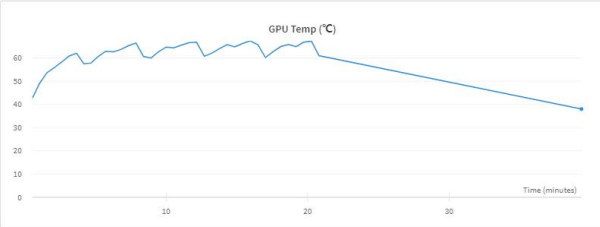
3.3 GPT2微调

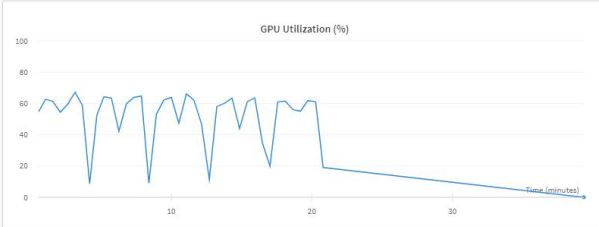

GPU Power Usage (W)
140
120
100
80
60
40
20
0
10
20
30
Time (minutes)


GPU Power Usage (%)
100
80
60
40
20
0
10
20
30
Time (minutes)


GPU Memory Allocated (%)
100
80
60
40
20
0
10
20
30
Time (minutes)


GPU Time Spent Accessing Memory (%)
100
80
60
40
20
0
10
20
30
Time (minutes)


GPU Temp (℃)
60
50
40
30
20
10
0
10
20
30
Time (minutes)


GPU Utilization (%)
100
80
60
40
20
0
10
20
30
Time (minutes)

04

思考与展望


CREATIVE
TEAM
Sample text sample
text sample text sample text
sample text sample text
sample text sample text.

4 思考与展望

模型融合

1. 我们主要使用了机器学习、BERT微调、GPT2微调三种方法进行了尝试。模型最高准确率0.869。

2. 未来可以考考虑加入日期、话题等特征进行训练。

3. 未来可以使用RoBERT、MacBERT等模型进行尝试，并进行模型融合。

感谢！


CREATIVE
TEAM
Sample text sample
text sample text sample text
sample text sample text
sample text sample text.