

自然语言处理大作业

环境配置 —— OS: Window11

注意!!! 项目目录不能有中文路径!!!

Sentence-bert

python=3.9.18 torch=2.0

首先使用conda创建虚拟环境py39pt20并激活,然后下载主目录下的requirements中的包

```
1 conda create -n py39pt20 python==3.9
2 conda activate py39pt20
3 pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

SimCSE

python=3.9.18 torch=2.0

```
1 conda install pytorch==2.0.1 torchvision==0.15.2 torchaudio==2.0.2 pytorch-
  cuda=11.8 -c pytorch -c nvidia
2 pip install jsonlines -i https://pypi.tuna.tsinghua.edu.cn/simple
3 pip install loguru -i https://pypi.tuna.tsinghua.edu.cn/simple
4 pip install scipy -i https://pypi.tuna.tsinghua.edu.cn/simple
5 pip install tqdm -i https://pypi.tuna.tsinghua.edu.cn/simple
6 pip install transformers -i https://pypi.tuna.tsinghua.edu.cn/simple
```

STT&TTS

python=3.9.18 torch=2.0

```
1 pip install pyaudio -i https://pypi.tuna.tsinghua.edu.cn/simple
2 pip install baidu-aip -i https://pypi.tuna.tsinghua.edu.cn/simple
3 pip install chardet -i https://pypi.tuna.tsinghua.edu.cn/simple
4 pip install pygame -i https://pypi.tuna.tsinghua.edu.cn/simple
5 #注意使用时关闭代理
```

(应该是所有的包了, 如有缺少, 自行conda/pip install即可)

项目说明

- exp: 实验结果记录文件, 保存了实验过程截图, 同时录制了STT&TTS的demo视频, 以及配置环境运行项目的视频
- data: 处理后的数据
 - jieba_data_exp_10_17: 10月17日进行最后一次修改的数据, 为数据清洗后的jieba分词的结果, 各个csv文件的含义见命名。
 - data_final_exp_10_29: 10月29日进行最后一次修改的数据, 为最终实验中使用的数据。
 - label_exp_train_10_29: 清洗后的训练集

- label_exp_dev_10_29:清洗后的评估集
 - label_exp_test_10_29:清洗后的测试集
 - dev_data_F1:用于得到F1得分的数据，其中test_final.csv为本次任务预测的结果
 - result:最终提交结果
- origin_data: 原始数据
- process:数据预处理的相关python脚本文件
 - csv_to_json:csv文件转为json文件
 - clean: 数据清洗函数
 - process_json: 清洗json文件保存为csv文件
 - process_xlsx:清洗xlsx文件保存为csv文件
 - (-) dev&test_label_generate:生成dev和test负样例
 - (-) train_label_generate:生成train的负样例
- origin_data: 原始数据
- data_pre_process_10_17: 数据预处理，对比了一下是否去除停用词的效果
- pytorch: 学习pytorch的框架，做了一些练习，记录了一下
- lab01-04: 借鉴的部分开源项目代码
- STT&TTS: 语音模块
 - STT: 语音转文字
 - TTS: 文字转语音
 - predict: 运行训练好的simcse模型，得到预测的结果并返回TTS模块
- test01:SimCSE模块
 - chinese_roberta_wwm_ext_pytorch: 存放预训练的chinese_roberta_wwm_ext_pytorch模型
 - clean:数据清洗模块
 - config.yaml:超参数配置文件，可以配置文件路径
 - data_load:数据读取以及装入到dataset
 - eval:评估函数，数据来自data_process构建的验证集
 - main:训练主函数
 - model:定义SimCSE模型
 - predict:进行test数据集的预测
 - run.sh:脚本文件
- test02 : sentence-bert模块
 - s-bert-finetuning:微调s-bert模型
 - s-bert: 运行s-bert模型进行评估
 - result: 将预测结果写入test.csv文件
- test03 : word2vec模块（最后舍弃了）
- demo: 语音展示、环境配置展示、项目运行展示

demo命令

- STT&TTS

```
1 conda activate py39pt20
2 cd STT&TTS
3 python TTS.py
```

运行后，您可以等待提示回车录制您想要标准化的文本输入，稍等片刻，会得到标准化匹配后的语音回复。

- Sentence-bert

```

1 conda activate py39pt20
2 cd test02
3 python s-bert-finetuning.py #开始微调
4 python s-bert.py #加载微调模型dev数据集上的预测并给出F1得分
5 python result.py #生成test.csv
6 cd ..
7 cd data
8 cd process
9 python csv_to_json.py#记得修改路径

```

```

Iteration: 99%|██████████| 450/456 [02:18<00:01, 3.231t/s]
Iteration: 99%|██████████| 451/456 [02:18<00:01, 3.221t/s]
Iteration: 99%|██████████| 452/456 [02:18<00:01, 3.261t/s]
Iteration: 99%|██████████| 453/456 [02:18<00:00, 3.261t/s]
Iteration: 100%|██████████| 454/456 [02:19<00:00, 3.261t/s]
Iteration: 100%|██████████| 455/456 [02:19<00:00, 3.231t/s]
Iteration: 100%|██████████| 456/456 [02:19<00:00, 3.261t/s]
Epoch: 93%|██████████| 14/15 [36:07<02:25, 145.29s/it]2023-10-29 16:41:51 - EmbeddingSimilarityEvaluator: Evaluating the model on sts-dev dataset after epoch 14:
Epoch: 100%|██████████| 15/15 [36:12<00:00, 144.84s/it]
2023-10-29 16:41:55 - Cosine-Similarity : Pearson: 0.9403 Spearman: 0.8559
2023-10-29 16:41:55 - Manhattan-Distance: Pearson: 0.9289 Spearman: 0.8568
2023-10-29 16:41:55 - Euclidean-Distance: Pearson: 0.9282 Spearman: 0.8564
2023-10-29 16:41:55 - Dot-Product-Similarity: Pearson: 0.9319 Spearman: 0.8529
2023-10-29 16:41:55 - Load pretrained SentenceTransformer: test_output
2023-10-29 16:41:56 - Use pytorch device: cuda
2023-10-29 16:41:56 - EmbeddingSimilarityEvaluator: Evaluating the model on sts-test dataset:
2023-10-29 16:42:01 - Cosine-Similarity : Pearson: 0.9446 Spearman: 0.8584
2023-10-29 16:42:01 - Manhattan-Distance: Pearson: 0.9311 Spearman: 0.8587
2023-10-29 16:42:01 - Euclidean-Distance: Pearson: 0.9307 Spearman: 0.8587
2023-10-29 16:42:01 - Dot-Product-Similarity: Pearson: 0.9354 Spearman: 0.8562

Process finished with exit code 0

```

图示为训练过程

```

慢性淋巴细胞白血病 (Score: 0.9535)
Micro-F1得分: 0.3940985246311578
=====
Number: 1999
Query: 右侧海绵窦癌
Result:Top 1 most similar sentences in corpus:
小梁性癌 (Score: 0.9117)
小梁性癌 (Score: 0.9117)
小梁性腺癌 (Score: 0.9079)
小梁性腺癌 (Score: 0.9079)
圆柱性腺癌 (Score: 0.8867)
Micro-F1得分: 0.394
=====
Process finished with exit code 0

```

图示为评估过程

- SimCSE

```

1 conda activate py39pt20
2 cd test01
3 bash run.sh

```

```

2023-10-31 13:48:28.988 | INFO | __main__:<module>:94 - epoch: 0
4%|███████|
2023-10-31 13:48:39.382 | INFO | __main__:train:59 - loss: 1.9853 | 9/250 [00:09<02:52, 1.39it/s]
0.9345318860244234
2023-10-31 14:00:39.617 | INFO | __main__:train:66 - higher corrcoeff: 0.9345 in batch: 10, save model
8%|███████|
2023-10-31 14:00:47.092 | INFO | __main__:train:59 - loss: 1.9692 | 19/250 [12:17<36:22, 9.45s/it]
0.9382632293080054
2023-10-31 14:04:47.674 | INFO | __main__:train:66 - higher corrcoeff: 0.9383 in batch: 20, save model
12%|███████|
2023-10-31 14:04:54.660 | INFO | __main__:train:59 - loss: 1.6311 | 29/250 [16:24<14:12, 3.86s/it]
1

```

图示为训练过程，评估过程没来得及做完

- word2vec (这个并没有用到，是一开始想用的，后来发现无监督的效果不好，舍弃了，不过自己写了一个shell脚本，学了一下shell脚本的编写，学习了一下yaml文件使用，日志文件在output目录下，感兴趣可以运行)

```
1 | cd test03
2 | cd model_exp_word2vec_10_20
3 | bash run.sh
```

参考开源仓库：

<https://github.com/princeton-nlp/SimCSE.git>

<https://github.com/vdogmcgee/SimCSE-Chinese-Pytorch>

<https://github.com/DataArk/CHIP2021-Task3-Top3.git>

https://github.com/Lisennlp/two_sentences_classifier.git