**文件夹：**

1. attention

Beer\_train：Beer错分样本的最近邻的注意力矩阵

Beer:Beer测试集所有样本的注意力矩阵

attentionX.csv：第1行-第102行是layer1，head1的attention矩阵；第103行-第204行是layer1，head2的attention矩阵；以此类推.......总共是14688（102\*12\*12）行，102列。

1. attention\_pictures：基于注意力机制的可解释性的相关图片。/.csv是作为注意力矩阵可解释性可视化所选取的样本对attention\_weight的筛选结果
2. best\_order：各数据集的最优排序，即元数据集
3. CLS\_Roberta：各个数据集测试集错分样本、训练集的CLS，使用的模型是基于Roberta的
4. CLStest：各个数据集测试集的CLS，使用的模型是基于Roberta的
5. ditto\_Distilbert：存取各数据集的test\_result.txt，使用的模型是基于Distilbert的
6. ditto\_Roberta：存取各数据集的test\_result.txt，使用的模型是基于Roberta的；还有json格式的匹配结果；log（time）.txt记录匹配时间（得到CLS后）
7. ditto\_Roberta\_summarize：存取各数据集的test\_result.txt，使用的模型是基于Roberta的，引入了summarize模块
8. EMBDI：EMBDI算法，计算属性相似度，实现属性对齐。

/Similarity：存储两个数据集间的属性相似度，可用于后续实现属性对齐

1. knn(confidence)：对测试集在匹配过程中的置信度进行判断，对低于某一阈值（threshold）的样本进行knn算法，记录在不同阈值、不同k值下的模型精确率、召回率、f1、时间。
2. knn（错分样本）：knn测试结果可视化
3. knn（所有样本）：knn测试结果可视化
4. Similarity：存取了数据集属性间的相似度，方便进行后续属性对齐
5. vis4rjxb：参数测试可视化

**python文件：**

1. 20nntest.py：knn模型(k=20)，找出错分样本的5个近似样本
2. changedata.py:

changeOrder，changeOrder2：交换属性顺序

extractCLS，adjustCLS：提取CLS、调整CLS

save\_errorCLS：保存错分样本的CLS，错分样本从test\_result.txt中对比读取

labelCLS：为CLS向量标注label

deal\_json：处理匹配结果的output\_small.jsonl，键值包括left，right，match，match\_confidence，生成test\_result.txt

1. confidence\_group\_result.py :测试不同阈值期间样本的预测效果
2. deal attention.py：计算注意力矩阵的miu（定值）和sigma
3. deal\_meta\_feature\_similarity.py:基于元特征计算数据集间的相似度
4. dealresult.py：参数影响测试，测试阈值和k值发生变化对预测结果的影响
5. find\_weight.py：对所有数据集的注意力矩阵进行属性值关联筛选
6. find\_weight\_single.py对单个数据集的注意力矩阵进行属性值关联筛选
7. FindBestOrder.py：算法1最大化属性顺序约束算法
8. knn\_alldata.py：对测试集所有实体对做knn预测，与原效果对比
9. knn\_test.py：对测试集错分样本做knn预测，正确预测结果+knn预测结果与原效果对比
10. seaborn\_single.py：某一层某一头的注意力矩阵热力图
11. seaborn\_single\_column.py：某一层某一头某一token的注意力矩阵热力图
12. seaborn\_test.py：某一实体对针对某俩token的注意力矩阵热力图
13. 12、13、14+“\_train”：针对训练集
14. search\_top2\_weight.py：对所有数据集的注意力矩阵进行属性值关联筛选，保留top2
15. search\_top2\_weight\_single.py：对单个数据集的注意力矩阵进行属性值关联筛选，保留top2
16. testOrder.py：对数据集的属性换序