# 说明书

# 1 项目简述

这是一个论文引用目的预测的项目，从论文引用网络、引用上下文人工提取特征后送入分类器进行预测。

目前项目的数据集是在ACL Anthology中融合了三个别人已经开源的数据集以及我和煜文学长2022年标注的一个额外数据集，目前采取的是二分类的设定，数据集中有1599个非重要引用和222个重要引用。之前在提取特征的时候有些数据可能对应的json或者pdf有数据缺失的情况，有些引用可能提不全特征，会有空值。空值的处理还是比较重要的，不要设置成同样的一个离群值，会对分类有提示。在代码里可以具体看一下。

### 1.1 关于数据：

大部分数据都在118和140的服务器上都可以找到（在xumufan的个人工作目录下，或是问季煜文学长也可），提取的所有特征等表都在mysql里直接有存好的版本

对于一些较大的文件，如68000篇ACL论文的pdf以及json文件，由于单个文件夹体积均为40G左右，请前往实验室服务器下载，不在这里提供

或是问一下夏蕾学长是否有新生成的版本。

同时部分feature的提取程序在罗元凯学长那里。这里提供的程序都是基于那个程序的特征提取结果。

# 2 文件介绍

### 2.1 annotated\_json文件夹

文件夹包含标注的文章json文件，其字典结构大致如下：

xxx-xxxx.json

|

|----------citation\_contexts

|

|----------1（第一个引用）

|-------------被引文标题等信息

|-------------引用上下文等信息

|-------------引用目的标签 “citation\_function”（部分）

|----------2（第二个引用）

|------------- ……

|----------3（第三个引用）

……

|-----------sections

|

|----------1（第一章）

|---------subsections

|----------1（第一节）

|------------sentences

|----------1（第一句）

|----------tokenize结果、原始文本

|----------语句其他信息等

……

|----------2（第二句）

|……

|……

|……

建议在浏览器中搜索online json viewer使用在线json查看器熟悉一下文件内容，注意只有被标注了的引用才会有名为“citation\_function”的元素。这种json格式是从parsCit引用提取工具中得到的xml转化而来，在获取对应文章的json文件前经历了pdf->word->txt->xml->json等几个步骤：

1. 从ACL Anthology官网上用python脚本下载所有可用的文章pdf，2022年1月大约下载了68000余篇论文
2. pdf-word使用wps稻壳办公工具转化（另一种用html的转换方法可以问孙烨学长），一批次可以转1000篇，最初版本是人工一次次转换好的
3. word-txt在wps使用宏脚本进行编辑自动转存word文档成平坦文本，请搜索wps的JS宏脚本使用方法，并参考下面的代码
4. txt-xml使用parsCit，注意在服务器上已经配好了perl环境，找到parsCit文件夹就可以运行

ACL Anthology所有文章的下载程序以及最后txt-xml-json流程可以请教煜文学长，前两步可以请教夏蕾学长。第（3）步骤代码参考如下，请修改路径至你的系统下所有word文档存放的位置：

/\*\*

\* Macro1 Macro

\* 宏由xmf录制，时间: 2022/08/05

\* 功能：把C:\\Users\\alienware\\Desktop\\citation\_function\\words\\目录下的word全部转换为C:\\Users\\alienware\\Desktop\\citation\_function\\words\_txt\\目录下的txt文件

\*/

function Macro1()

{

var file = Dir("C:\\Users\\alienware\\Desktop\\citation\_function\\words\\\*.docx");

while(true){

try{

file = Dir();

file = file.replace(".docx","")

Documents.Open("C:\\Users\\alienware\\Desktop\\citation\_function\\words\\"+file+".docx", false, false, true, "", "", false, "", "", 0, 0, true, false, 0, true);

Documents.Item("C:\\Users\\alienware\\Desktop\\citation\_function\\words\\"+file+".docx").SaveAs("C:\\Users\\alienware\\Desktop\\citation\_function\\words\_txt\\"+file+".txt",7, false, undefined, true, undefined, false, false, false, false, false, 65001, false, false, 0, false);

Documents.Close()

//ActiveDocument.SaveAs("C:\\Users\\alienware\\Desktop\\citation\_function\\words\_txt\\"+file+".txt", 7, false, undefined, true, undefined, false, false, false, false, false, 0, false, false, 0, false);

}catch(err){

alert(err);

break;

}

}

}

### 2.2 code文件夹

文件夹存放实验代码，环境为python 3.x sklearn（较新版本均可）等基础环境。

**i) all\_dataset\_link\_label\_with\_features.txt**

从数据库中提取好的数据文件，以加号+分隔的数据，其中\N是空值。每一列的特征表示内容见mysql数据库中的特征集合，在这里已经给出了一个版本的文件。该文件由元凯学长生成，生成程序可以问元凯学长

**ii）code/feature\_selection.py**

这个是特征抽取程序，读取上面的 all\_dataset\_link\_label\_with\_features.txt 从json文件中提取相应引用上下文等文本内容特征。

注意过程中需要从服务器下载ACL Anthology所有文章的json格式数据文件，并修改程序中的路径到对应位置，MAG\_id\_to\_xxx文件是针对匹配过程中存在id不一致情况而生成的补充对照表，无需修改。之后便可以进行特征提取。请正确修改代码中的路径，其他可以不改先试着跑一遍。程序会生成 all\_dataset\_link\_label\_with\_combined\_features.txt 文件，这里已经给了一版在外面。

**iii）code/new\_data\_model.py**

分类程序，依照上面的程序应该重新生成了 all\_dataset\_link\_label\_with\_combined\_features.txt 文件，当然这里已经提供了一版，可以直接运行。分类程序将在上面所说的这个数据文件上进行分类器训练，最终将会打印结果，目前是使用extratree分类器，可以替换为sklearn别的分类器。

**iv）code/annotate\_ARC.py**

人工标注程序，具体说明见下面。注意这个也需要下载好ACL所有文章的pdf以及json分别放在合适的目录下。.ftr文件在这里已提供，见prediction\_result文件夹。注意修改代码中的文件路径。

**v）code/bert\_classification.py**

用预训练模型直接对引用上下文句子进行分类预测引用目的，这里给出了一版使用的数据，见 sentences\_and\_labels.txt。其余py文件有些来自原始项目，在这里保留供参考。

### 2.3 prediction\_result文件夹

请看下面的描述，是存放弱分类器分类结果的文件目录，可以直接使用。

# 3 标注程序说明

### 3.1 依赖及需要的文件

程序运行环境 python3.x。依赖包都可以用conda install或者pip install安装

程序需要在机器上有以下文件：

1、季煜文同学20220208版本的所有json（命名不要动）

2、所有的ACL电子图书馆下载下来的pdf文件（命名不要动）

3、作为参考的弱分类器分类结果，转换为.ftr文件，内容格式见示例

**3.2 标注方法**

请在一开始输入自己的名字，方便后面把标注结果融合比较，标注结果生成的json中将会在字典中的‘annotator’一项中记录所输入的标注者名字。

标注时可以输入“1，2，0，exit”四个输入

1 —— 标注为Extend类别

2 —— 标注为非Extend类别

0 —— 如果显示的文段不足以展示这个引用的含义，可以输入0调出pdf文档进行查阅。输入0之后会提示该引用在文章中所在的大致位置。

exit —— 离开标注程序，当然直接关掉也可以

### 3.3 使用提示说明

1 这个程序会在开始的时候需要手动输入上述地址，所以放在同一个目录下用“../”进行简便输入会方便一些

2 程序在标注时，直到出现需要你输入东西的时候输入才有用，其他时候碰到键盘不会有影响

3 在标注时碰错键没事，直接回车会要求重新输入合法标注类别

4 每个引用会在标注过程中展示其所在文段上下大概三句话的位置，本身所在句子会标红

5 程序会在该程序所在的目录创建一个“annotation\_log.txt“，用来在多次标注时跳过已经标注过的文件，如果需要重新标注，可以直接删除或者进入文件删除某一个ID用来重新标注

6 在开始标注之前，程序会显示总的已经标注的extend标签有多少个，这个依赖于你的输出json文件目录，所以最好每次的输出都放在一起，并且不要挪走

# 4 其他说明

对于项目之前所做的工作以及一些未来可以进行的工作，可以参考答辩ppt中的内容。这里提供一些项目可以继续进行的研究点，仅供参考：

1. 预训练模型的使用：

实验表明预训练模型直接进行引用上下文表征的分类效果好于人工提取的特征，可以探究将预训练模型与引用网络特征结合进行分类的效果，引入先验知识后进行适当fine tune，或是有其他更好的手段？

2. 数据扩充手段：

虽然在前人的数据集上进行了人工标注，但是总体来说现在的数据量还是很小。针对数据获取困难的问题，是否可以尝试半监督学习、主动学习的方法进行数据扩充？

3. 分类器探究：

由于引用存在于引用网络中，可以看作引用图的一部分，那么是否可以用图神经网络作为分类器进行训练？