

题目: English Text Data Processing

胡求 MG1633031 huqiu00@163.com 188-5182-3759

(南京大学 计算机科学与技术系, 南京 210023)

1 实现细节

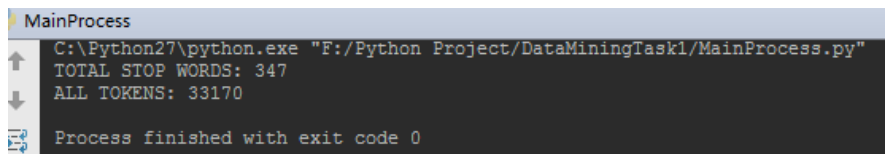
1. 首先集合所有文章，得到他们的路径
2. 然后将所有文章分词，将无关或者非法字符用空格替换，然后采用 NLTK 的 `word_tokenize` 方法进行分词，然后用 NLTK 的 `SnowballStemmer` 进行词干提取，然后将单词进行小写化，通过字典去重（存成字典 `key`），然后去除一些无关词，数字以及停用词，停用词选用了网上一个版本的停用词，约 900 个停用词。
3. 然后将所有词排序，编号，后续输出时以编号代替词，而不需写出词。
4. 然后循环处理每个类（每种论文的目录），对这个目录下的所有文章，对每篇文章的词算出他们的 TF 和 IDF 值，得到 TF-IDF 值，存储到这个类这篇文章的结果数据中。
5. TF 词频采用的计算方法为： $TF = \text{词在文章中出现次数} / \text{文章总词数}$
6. IDF 计算方法为： $IDF = \log(\text{总文章数} / \text{此词出现的文章数目})$
7. 处理完每个类，结束。

2 结果

2.1 实验设置

1. 数据来源为从作业网站提供的作业数据集 ICML
2. 采用 Python 2.7.10 和 Windows10 系统 作为编程环境
3. 输出的结果名为 “类名_RESULT.txt”，位于与类目录同级的目录下。
4. 结果数据包含 N 条，N 为此类中文文章数目，在每条输出结果中，第一行为文章名字，然后紧跟着的是词的 TF-IDF 值向量，由于值向量稀疏，所以做了压缩，即以 “序号：TF-IDF 值” 的表示方法。

2.2 实验结果



```
MainProcess
C:\Python27\python.exe "F:/Python Project/DataMiningTask1/MainProcess.py"
TOTAL STOP WORDS: 347
ALL TOKENS: 33170
Process finished with exit code 0
```

由图可知，总的词数为

33170 个，其中检出停用词 347 个。

此电脑 > 文档 (F:) > Python Project > DataMiningTask1				
名称	修改日期	类型	大小	
.idea	2016/9/15 16:13	文件夹		
ICML	2016/9/15 15:27	文件夹		
MainProcess	2016/9/15 15:28	JetBrains PyChar...	6 KB	
stopwords	2016/9/14 21:53	文本文档	7 KB	

此为运行时必须的文件。

1. Active Learning	2016/9/15 15:11
2. Applications	2016/9/15 15:11
3. Bayesian Learning and Graphical ...	2016/9/15 15:11
4. Deep Learning	2016/9/15 15:11
5. Ensemble and Crowdsourcing	2016/9/15 15:11
6. Feature Learning	2016/9/15 15:12
7. Kernel Methods	2016/9/15 15:12
8. Online Learning	2016/9/15 15:12
9. Optimization	2016/9/15 15:12
10. Ranking	2016/9/15 15:12
11. Reinforcement Learning	2016/9/15 15:12
12. Supervised Learning	2016/9/15 15:12
13. Theory	2016/9/15 15:13
14. Unsupervised and Semi-Supervis...	2016/9/15 15:13
15. Others	2016/9/15 15:13
1. Active Learning_RESULT	2016/9/15 15:29
2. Applications_RESULT	2016/9/15 15:29
3. Bayesian Learning and Graphical ...	2016/9/15 15:29
4. Deep Learning_RESULT	2016/9/15 15:29
5. Ensemble and Crowdsourcing_RES...	2016/9/15 15:29
6. Feature Learning_RESULT	2016/9/15 15:29
7. Kernel Methods_RESULT	2016/9/15 15:29
8. Online Learning_RESULT	2016/9/15 15:29

```

7. Kernel Methods_RESULT.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

A Divide-and-Conquer Solver for Kernel Support Vector Machines.txt
['73: 0.00044', '83: 0.00038', '100: 0.00094', '106: 0.00000', '121: 0.00546', '130: 0.00030', '137: 0.00059', '138: 0.00010', '145: 0.00055', '141: 0.00096', '2685: 0.00032', '2884: 0.00027', '2920: 0.00034', '3025: 0.00208', '3085: 0.00239', '3135: 0.00100', '3136: 0.00091', '3139: 0.00113', '5259: 0.00006', '5280: 0.00009', '5284: 0.00104', '5343: 0.00016', '5355: 0.00079', '5381: 0.00181', '5401: 0.00001', '5415: 0.00058', '7579: 0.00021', '7696: 0.00874', '7697: 0.00036', '7870: 0.00053', '7956: 0.00034', '8042: 0.00060', '8046: 0.00027', '8074: 0.00039', '8105: 0.00133', '10643: 0.00118', '10672: 0.00013', '10795: 0.00075', '10851: 0.00013', '10862: 0.00474', '10894: 0.00024', '10952: 0.00068', '10990: 0.00030', '13239: 0.00069', '13269: 0.00037', '13270: 0.00015', '13309: 0.00000', '13364: 0.00001', '13378: 0.00064', '13604: 0.00025', '13656: 0.00083', '16444: 0.00011', '16446: 0.00032', '16474: 0.00090', '16534: 0.00080', '16536: 0.00266', '16555: 0.00012', '16563: 0.00044', '16587: 0.00070', '20169: 0.00133', '20240: 0.00306', '20275: 0.00115', '20293: 0.00021', '20361: 0.00099', '20500: 0.00033', '20565: 0.00013', '20587: 0.00108', '23448: 0.00133', '23544: 0.00101', '23591: 0.00018', '23607: 0.00016', '23615: 0.00059', '23687: 0.00018', '23726: 0.00052', '23739: 0.00013', '26616: 0.00010', '26631: 0.00038', '26669: 0.00080', '26707: 0.00023', '26828: 0.00036', '26888: 0.00114', '26902: 0.00069', '26975: 0.00027', '28981: 0.00009', '29009: 0.00069', '29152: 0.00035', '29177: 0.00091', '29205: 0.00074', '29251: 0.00066', '29321: 0.00021', '32236: 0.00030', '32364: 0.00072', '32481: 0.00034', '32597: 0.00044', '32961: 0.00143']

A Kernel Independence Test for Random Processes.txt
['73: 0.00017', '83: 0.00035', '100: 0.00292', '106: 0.00000', '127: 0.00095', '138: 0.00035', '140: 0.00026', '175: 0.00011', '208: 0.00024', '251: 0.00486', '2587: 0.00056', '2639: 0.00072', '2724: 0.00046', '2731: 0.00061', '2795: 0.00093', '2997: 0.00053', '3077: 0.01404', '3093: 0.00088', '3113: 0.00000', '5163: 0.00040', '5176: 0.00001', '5226: 0.00020', '5249: 0.00082', '5255: 0.00006', '5259: 0.00068', '5280: 0.00055', '5302: 0.00033', '5311: 0.00001', '6808: 0.00021', '6829: 0.00464', '6849: 0.00015', '6851: 0.00008', '6901: 0.00005', '6952: 0.00124', '7014: 0.00094', '7025: 0.00102', '7026: 0.00000', '773: 0.00244', '8781: 0.00050', '8801: 0.00005', '8815: 0.00050', '8817: 0.00040', '8829: 0.00051', '8839: 0.00052', '8841: 0.00097', '8843: 0.00000', '891: 0.00828', '11894: 0.00121', '11945: 0.00055', '11969: 0.00466', '12055: 0.00074', '12082: 0.00173', '12139: 0.00300', '12233: 0.00013', '12238: 0.00000', '14043: 0.00055', '14050: 0.00038', '14103: 0.00035', '14114: 0.01285', '14245: 0.00155', '14251: 0.00310', '14370: 0.00112', '14522: 0.00466', '14574: 0.00000', '17784: 0.00291', '17855: 0.00001', '17864: 0.00102', '18125: 0.00021', '18177: 0.00168', '18220: 0.00091', '18245: 0.00095', '18250: 0.00223', '18257: 0.00155', '22334: 0.00155', '22351: 0.00139', '22364: 0.00006', '22415: 0.00020', '22469: 0.00018', '22473: 0.00072', '22538: 0.00010', '22616: 0.00020', '22617: 0.00000', '24257: 0.00160', '24276: 0.00016', '24307: 0.00235', '24323: 0.00023', '24346: 0.00012', '24351: 0.00162', '24365: 0.00089', '24373: 0.00193', '24381: 0.00000', '26961: 0.00033', '27057: 0.00140', '27062: 0.00301', '27129: 0.00054', '27140: 0.00043', '27194: 0.00102', '27205: 0.00014', '27206: 0.00000', '29009: 0.00040', '29062: 0.00117', '29094: 0.00057', '29132: 0.00095', '29177: 0.00086', '29208: 0.00155', '29226: 0.00038', '29306: 0.00000', '32919: 0.00138', '32961: 0.00048', '33042: 0.00243', '33139: 0.00128']

A low variance consistent test of relative dependency.txt
['72: 0.00030', '73: 0.00069', '106: 0.00000', '116: 0.00059', '162: 0.00027', '175: 0.00011', '208: 0.00024', '235: 0.00015', '255: 0.00011', '251: 0.00000', '2317: 0.00129', '2327: 0.00049', '2335: 0.00053', '2405: 0.00156', '2448: 0.00117', '2724: 0.00046', '2731: 0.00061', '2851: 0.00225', '2884: 0.00000', '4949: 0.00062', '4993: 0.00029', '5024: 0.00122', '5028: 0.00037', '5034: 0.00079', '5045: 0.00027', '5057: 0.00049', '5075: 0.00014', '5089: 0.00000']

```

最后的工作空间如下：

名称	修改日期	类型	大小
.idea	2016/9/26 17:30	文件夹	
ICML	2016/9/15 15:27	文件夹	
PC MainProcess.py	2016/9/27 8:17	JetBrains PyChar...	7 KB
ReadMe.txt	2016/9/21 19:04	文本文档	1 KB
result.txt	2016/9/27 8:20	文本文档	297 KB
stopwords.txt	2016/9/14 21:53	文本文档	7 KB
word_vector.txt	2016/9/27 8:20	文本文档	270 KB

其中 result.txt 是从 ICML 文件夹取出的第 7 类的结果。

ReadMe.txt 是程序的执行说明。

word_vector.txt 中是单词的集合，即排序后的词的排列，输出结果中的序号即为单词在此文件中的序号。

（实验结果后可简述对当前实验的思考）

TF, IDF 的计算方法多种多样，本文采用的方法中，最终结果的 TF-IDF 数值上偏小，在保证含义的情况下可以采用其他方法，使 TF-IDF 大致为一个正常值，以便更好地比较。

用 NLTK 包的 stemmer 提取词干的时候难免会有一些提取的错误，这是由于词干提取并非完美的缘故，由于这个缘故，也可以不进行词干提取，这样可以保持所有的信息，但是缺点就是会有多种变形词，单词数会增加 1/4 到 1/3，导致矩阵增大。

注意：

1. 最终提交的报告最好保存为 pdf 格式
2. 压缩格式为 zip 格式，请勿使用需要安装特定软件才能打开的压缩方式
3. 作业的文件夹目录请按照网页要求，代码、结果放在不同子文件夹中。作业网页上给出的数据不需要再次提交

