Vector Space Model 和 KNN

一、 实验目的

通过实验，加深对 Vector Space Model 和 KNN 的理解，锻炼编程能力。

二、 实验要求

1、预处理文本数据集，并且得到每个文本的 VSM 表示。

2、实现 KNN 分类器，测试其在 20Newsgroups 上的效果。

3、20%作为测试数据集，保证测试数据中各个类的文档均匀分布

三、 实验环境

windows 10

python 3.6.5

四、 实验过程

1、预处理数据。遍历所有文档，对数据利用 NLTK 工具进行预处理，包括 tokenization，stemming，stopwords，lower 全部转换成小写，并且把 a-z 外的用空格替换，并保存到 preprocess 文件下对应的相同路径。

2、划分数据集。遍历 preprocess 文件夹，随机选取预处理之后的每个类里的 80%的数据作为训练集，剩下的 20%作为测试集，并分别保存到 split/split_train/和 split/split_test/文件夹下，同时把训练集和测试集的文档编号和类别分别保存到 splitTrain.txt splitTest.txt，划分之后，训练集有 15056 个文档，测试集有 3772 的文档。

3、提取特征词，创建词典，过滤低频次。分别对训练集和测试集进行此操作。遍历数据集文件，统计每个单词的词频，暂时放到词典中，同时统计得到训练集有 79268 的单词，测试集有 40471 个单词。然后遍历词典，根据词频进行筛选，去掉词频小于等于 4 的词，并对过滤后的词典进行排序，过滤后，训练集有 24202 个，测试集有 11155 个，并把过滤后的词典分别保存到 dictionary 文件夹下。然后根据过滤后的词典，对每个文档遍历去掉低频词，再依次保存到 featureWord 文件夹下。

4、计算 tf-idf。首先计算每个单词的 idf，保存到 IDF_perWord 文件夹下，然后计算 tf,并同时读取 idf，计算 tf-idf 值，得到每个文档的向量空间表示。格式：类别名 文档名 word1 tfidf1 word2 tfidf2……。

5、KNN 分类。计算测试集中的每一个文档与所有训练集的文档的距离，使用 consine 来计算相似度，然后取前 k 个与它相似度最大的文档，然后在计算每个类的的距离和，最大的那个就是预测的类。将测试的类名与预测的类名进行比较，计算准确率。

五、 出现的问题及解决

1、刚开始做的时候，就遇到了文件读写时的编码问题。以 r 的方式打开文件，报错：UnicodeDecodeError: 'gbk' codec can't decode byte 0xff in position 31810: illegal multibyte sequence。解决办法：rb 打开。但是后面利用工具分词，会报错：TypeError: cannot use a string pattern on a bytes-like object。解决办法：读取文件时加上“.decode('utf-8', 'ignore')”

2、分词工具的选择。最先使用的时 Textblob 工具进行分词，但他处理过的结果是列表的形式，在后面进行读取遍历文本时会报错，忘记记录此错误，没选择改格式，选择了 nltk。

3、划分数据之后一系列操作的路径问题。随机划分数据集之后，训练集和测试集都要选取特征词，计算 tfidf，路径问题有时候会迷糊，所以选择手动修改路径。还要注意一点，随机划分数据集之后，不要再运行 splitData.py，这样又会重新选择数据集，添加到路径里，会变多。所有划分数据之后就不要再运行，如果运行要先把之前保存的全部删除掉。

六、 实验结果

K	1	5	10	15	20	25
acc	0.7622	0.8118	0.8245	0.8309	0.8372	0.8367

七、 心得体会

通过本次实验，对文本分类的步骤、原理和方法有了进一步的认识，学会了使用 github。同时，发现上课觉得自己听懂的知识，并不代表真正理解和掌握，实际运用的时候还是会遇到这样或那样的问题，发现对知识点理解的并不透彻，写代码过程中也发现自己想得不够全面，需要一点点改进、完善。在本科阶段，没有学过 python，在暑假的时候自学过 python 基础，但没有真正的

这样实践过，对代码还不够熟练，一些知识点还得现搜现学现用，不断向同学请教，虽然遇到各种问题，但是看到成果心里还是有点小开心。在今后的学习中，一定要提高自己写代码的能力，多动手，只有实践出来，才能对知识有一个更深的理解。