作业报告

学　号：201814841　　姓　名：徐　强

10月8日收到此次作业后，经过近一个月的学习和实践，基本完成了作业要求的内容。报告内容主要分为作业完成情况、学习收获和几点思考三个部分。

一、作业完成情况

（一）学习 python

对于一个仅在十年前学过C语言且基本没有编程基础的学生，学习 python 还是比较困难的，我利用两周时间结合老师教学课件给出的教程及《零基础入门学习 python》一书，了解了 python 的基础知识，根据老师教学课件的相关内容搭建了 python 编程环境和注册 GitHub 账号，掌握了一些基本操作。因 deadline 临近，只能先开始慢慢写程序，边写边学。

（二）程序实现

按照 TF-IDF 和 KNN 文档处理流程，将程序划分为五个步骤编写，具体实现过程如下：

1.文档读取

编写文件读取时比较顺利，基本能够利用 OS 模块及前期所学编程知识解决，以嵌套列表的结构存储了所有文档。

2.预处理

使用 String 模块替换了文档中其它字符内容，使用老师 PPT 所给的 Textblob 模块进行了 tokenization，使用 NLTK 模

块进行了 lemmatize、steming 和去掉 stopwords，使语料库里的每个文档均成为一个由一系列单词组成的列表。

3.统计词频生成词典

使用 collections 模块的 Counter 类统计了所有单词在所有文档中的词频，并根据统计出的词频，设置了高频词和低频词上下限（9 ≤ wordfrequency ≤ 1500）去除了 non-informative words and rare words，生成了一组由 23211 个单词组成的词典。

4.计算 TF、IDF 生成 VSM

使用 collections 模块的 Counter 类分别计算每个文档对于词典的词频 N1、语料库中文本的总数 N2、语料库中包含词 x 的文本数 N(x)，利用 TF 公式 a+（1-a）N1/max N1（a=0.1）和 TDF 公式 log((N2+1)/(N(x)+1)+1)（该公式主要解决在一些特殊的情况下会出现的小问题，对 IDF 做了一些平滑），最后利用 numpy 模块完成 VSM 的生成。

5.利用 KNN 完成 test 文档测试分类并统计正确率

使用 sklearn.model_selection 模块对语料库和标签进行了随机选取，按照作业要求取出 80%文档做训练，剩余 20%做测试，经过计算测试文档和训练文档向量夹角的余弦值来确定向量所表示文档的相似性，设置的 K 值为 3，取出 cos 值最大的 K 个，统计划分出测试文档的分类，对比测试文档初始的分类，经过五次计算平均的正确率为 0.709506。

二、学习收获

1.通过编写 TF-IDF 和 KNN 程序，初步掌握了一些程序编写的基本方法和 python 的列表、字典和数组等结构的处理及计算的基本方法，因编程经验不足造成了语句产生的错误较多，通过请教同学也掌握一些程序调试的方法。

2.对于 TF-IDF 及 KNN 主要通过老师所发上课课件进行理解和巩固，处理语料库及词频统计中所使用的模块主要通过《NLTK 基础教程》和 CSDN 网站学习使用方法，通过此次作业基本掌握了上述内容。

3.通过调整设置词典生成中 wordfrequency 上下限和 KNN 中的 K 值（具体记录数据见表格），选取的了正确率较高参数，进一步理解了 TF-IDF 及 KNN 原理。

三、几点思考

1.在上述的程序中，训练集和测试集是在生成 VSM 后划分的，这样测试集在词典选取时已经在训练集中，影响了词典的生成，从而影响了后面 KNN 的正确率。后面编写了在读取文档时按 80%随机选取的函数，但是影响后续处理的数据结构，故未进行测试，争取在下次作业实现。

2.在整个处理过程中基本默认列表中文档的顺序没有发生改变，这样才能和储存标签的列表一一对应，如何同步处理标签和文档，需要一个合适的数据存储结构才能实现，但未能实现，还需要继续学习。