Homework 1: VSM and KNN 实验报告

姓名：孙畅畅
学号：201814832

（一）实验任务

（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。

（二）实验数据集

（1）数据集种类：20 个种类　数据集数目：18828
种类：alt.atheism（799）comp.graphics（973）comp.os.ms-windows.misc（985）comp.sys.ibm.pc.hardware（982）comp.sys.mac.hardware（961）comp.windows.x（980）misc.forsale（972）rec.autos（990）rec.motorcycles（994）rec.sport.baseball（994）rec.sport.hockey（999）sci.crypt（991）sci.electronics（981）sci.med（990）sci.space（987）soc.religion.christian（997）talk.politics.guns（910）talk.politics.mideast（940）talk.politics.misc（775）talk.religion.misc（628）（括号内表示各个类别的样本数）
（2）在利用 KNN 对文本进行分类时，考虑到在训练集和测试集中各个类别的数目比例，随机从上诉各个类别中供选择一定数目的样本作为训练集与测试集，各个类别数目依次为：
训练集：
[700,800,800,800,800,800,800,800,800,800,800,800,800,800,800,800,800,800,700,450]
测试集：
[99,173,185,182,161,180,172,190,194,194,199,191,181,190,187,197,110,140,175,78]
训练集与测试集比例约为：4:1

（三）实验过程

（1）向量空间模型

（1-1）读入训练集文本，切词，词汇预处理。
切词：在此阶段中，使用基于 python 的自然语言处理工具集 NLTK 对训练集文本进行切分，得到文本中出现的词汇。
词汇预处理：使用文本处理工具，对上一步得到的词 term 进行复数变单数、动词过去式变原型、term 变成小写形式。
（1-2）建立词典，获得每个词在训练集文本中的词频 tf。
使用 python 中的容器“词典”与“列表”建立词频矩阵，词典的键值作为 term，词典的 value 值用列表表示，列表中对应索引位置表示该词在对应文档中的词频 tf。得到每个 term 在每个文档中出现的频率。
（1-3）计算文档频率 df
对上一步得到的词频矩阵进行遍历计算，得到每一个词在全局中的文档频率 df。
（1-4）词典处理
在得到的词典中，并不是每一个词 term 都对向量空间模型的建立有帮助，所以根

据文档频率 df 对上面所得的词典中的词汇进行部分的删除。

删除条件：（a）term 中含有数字（b）term 的长度为 1 或 term 的长度大于 25

（c）term 在 NLTK 工具包中的英文停用词中（d）文档频率小于 5

（e）文档频率大于 400

（1-5）词典大小

最终词典中有 20937 个词，在 VSM 中用一个长度为 20937 的向量表示每一个文档。

（1-6）VSM 中向量权重

在实验过程中使用 TF-IDF 方法来计算每个文档向量中各个维度的权重。

$$\mathrm{Weight(t)} = \mathrm{IDF(t)} * \mathrm{TF(t,d)}$$

$$IDF(t) = \log(\frac{N}{df(t)})$$

$$TF(t,d)\begin{cases} 1+\log c(t,d), ifc(t,d)>0 \\ 0, otherwise \end{cases}$$

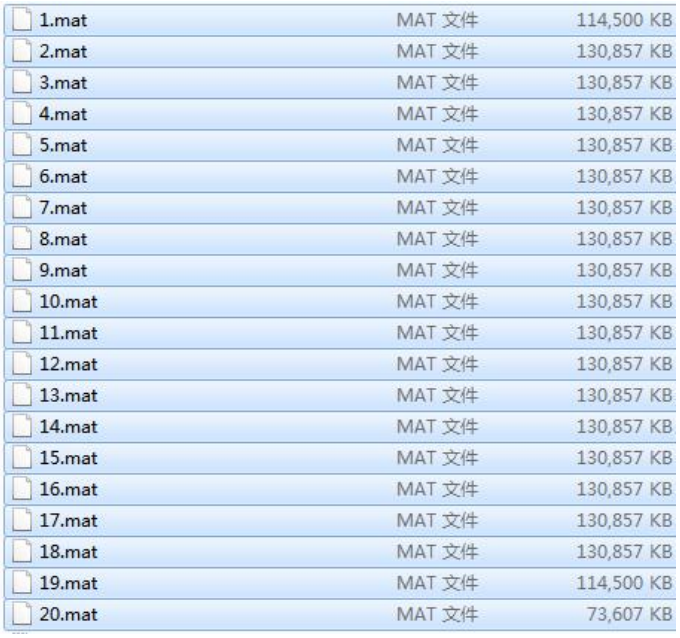
（1-7）测试集处理

在利用训练集建立好词典之后，分析测试集文本，得到各个测试集文本中词典中对应词汇的词频，利用在训练阶段得到的词典中词汇的全局文档频率，得到测试集中每个文档的向量空间表示。

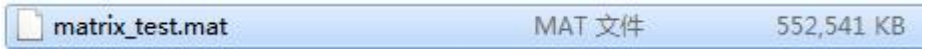
（1-8）数据保存

由于训练数据和测试数据较多，为了更好的数据保存，我将每个类别的文本对应的向量表示结果分别存放在 MAT 文件中，用于后续的分类过程。

训练集：



测试集：



（2）KNN 分类

（2-1）文本相似度计算

利用两个向量的余弦值代表两个向量的相似度，余弦值越大，两个文本越相似，越有可能属于统一个类别。

$$\cos(v_1, v_2) = \frac{v_1 * v_2}{|v_1||v_2|}$$ （v_1， v_2 表示文本的向量表示）

（2-2）K 值设置

在经过多次测试后，发现当 K=10 时，在测试阶段可以取得较高的准确率。准确率达到 70.72%。

（2-3）分类过程

在上述阶段完成后，每一个测试集与训练集求余弦，求出前 50 个最相似的测试集，选择其中个数最多的文本所对应的类别作为测试集的种类，最后求得所有测试集的平均准确率。$ACC = \frac{\text{分类正确的测试样本数}}{\text{测试样本数}}$

（四）实验结果

在经过一定时间的训练之后，选取不同的 K 值会有不同的准确率，如下表所示。

经过对比发现，当 K 为 10 时，能够去的较大的准确率。

（五）实验总结

在本次实验中，通过完成文本的分类工作，对向量空间模型 VSM 和 KNN 的分类方法有了更加清晰的认识，通过实际的动手操作，完成了对文本的分类，并取得了不错的准确率。

在代码的实现过程中，也遇到了许多实际的问题，如开始编写代码进行切词时，直接选择 TextBlob 工具包对文本中的句子进行切词，但实验过程中，出现了许多奇怪的切法，并且有些应该切开的 term 并没有切开。因此构建的词典中的词汇并不完美，最后得到的结果并不好。所以最后经过改进，选用 NLTK 工具包进行切词，发现得到的词项还是挺满足要求的，并且最后在对词汇进行变单数和过去式变原型时，也使用到了 TextBlob 中的方法，最终得到了比较理想的结果。

在本次实验中，选择余弦值来判定文本之间的相似度，而不是两个向量之间的欧几里得距离，考虑到了向量的长度，分类的效果更好。

无论是在词典过滤中文档频率 DF 值的选择，还是最终分类时 K 值的选择，都要经过多次的摸索，试验，选择最佳值。在本次试验中，词典的建立是个关键阶段，词典建立的好坏直接决定最终结果的好坏。

通过本次作业的完成，对于文本分类的过程与方法有了更加深刻的认识，遇到了很多问题，也积极的思考解决了相应的问题，收获很大。