成 绩 _____

北京航空航天大學

BEIHANG UNIVERSITY

深度学习与自然语言处理

第三次作业

院（系）名称	自动化科学与电气工程学院
专 业 名 称	电子信息
学 生 学 号	ZY2103202
学 生 姓 名	黄君辉
指 导 教 师	秦曾昌

2022 年 5 月

三　　LDA 模型段落分类

一、问题描述

从给定的语料库中均匀抽取 200 个段落（每个段落大于 500 个词），每个段落的标签就是对应段落所属的小说。利用 LDA 模型对于文本建模，并把每个段落表示为主题分布后进行分类。验证与分析分类结果

二、方法介绍

2.1 LDA 模型

LDA 是自然语言处理中非常常用的一个主题模型，全称是隐含狄利克雷分布（Latent Dirichlet Allocation），简称 LDA。作用是将每篇文档的主题以概率分布的形式给出，然后通过分析归到同一主题下的文档来抽取其实际的主题（模型运行结果就是一个索引编号，通过分析，将这种编号赋予实际的意义，通常的分析方法就是通过分析每个 topic 下最重要的 term 来进行总结归纳），根据主题分布进行主题聚类或文本分类。

一篇文档可以包含多个主题，所以会有主题分布这个概率。可以认为文档中的每个词都由其中的一个主题生成。一篇文章的生成可以这么理解：先以一定的概率选取某个主题，然后再以一定的概率选取该主题下的某个词，不断重复这两步，直到完成整个文档。LDA 解决的问题就是，分析给定的一篇文章都有什么主题，每个主题出现的占比大小是多少（需要注意的是，输入模型的数据是由词袋构成的向量，没有考虑词与词的先后关系，对这个问题的改进可以用 bi-gram 二元模型来解决）。一般来说，LDA 模型对短文本的主题分类效果比较差。

整个模型的一个标准流程，整个模型可以抽象成以下几个步骤：

1. 对于语料库中的每篇文档，从主题分布中抽取一个主题
2. 从上述被抽到的主题所对应的单词分布中，抽取一个单词

重复上述过程，直至遍历文档中的每一个单词。

2.2 LDA 模型学习过程

给定文章合集 Doc 包含的 k 个主题，所有主题集合为 $Topic$，Doc 中每个文档看作一个单词序列$\{w_1,w_2,\cdots w_n\}$，其中w_i表第 i 个单词，一篇文章中共有 n 个单词，Doc 中涉及所有单词构成语料库 V,V 中共有 m 个单词。LDA 模型以文档集合 Doc 作为输入，希望训练得到两个结果：每个 $Topic$ 生成不同词的概率$\varphi_{t1}(p_{w1},p_{w2},\ldots,p_{wm})$和，每个文档对应到不同 $Topic$ 的概率$\theta_d(p_{t1},p_{t2},\ldots,p_{tk})$。

LDA 的核心公式如下所示：$P(w|d)=P(w|t)*P(t|d)$。学习过程如下：

1.随机给θ_d和φ_t赋值；

2.针对特定文档d_s中的第 i 个单词w_i,如果该单词对应的 $Topic$ 为t_j，则上述公式具体改写为$P_j(w_i|d_s)=P(w_i|t_j)*P(t_j|d_s)$

3.枚举 $Topic$ 中所有 topic，得到所有的$P_j(w_i|d_s)$，然后可以根据这些概率值结果为选择一个 topic 作为d_s的主题，一般选择令$P_j(w_i|d_s)$最大的主题t_j。

4.如果d_s中第 i 个单词w_i的 topic 与t_j不同，就需要改变θ_d和φ_t，把 D 中所有的 d 中所有的 w 进行计算并重新得到θ_d和φ_t称为一次迭代；经过 n 次循环迭代之后，就可以收敛到 LDA 需要的结果。

三、实验分析

3.1 实验设计

根据题目要求，本次实验主要分为以下几个步骤：

（1）对 16 篇金庸小说进行预处理，去除其中多余的符号和广告之类的无关内容，按行进行分词构成新的语料库；从其中随机均匀抽取 200 个段落（每个段落大于 500 词），每个段落的标签就是对应段落所属的小说。

（2）将文档合集作为输入训练 LDA 模型，本实验采用 python 中的 gensim 自然语言处理工具库实现 LDA 模型构建，设定主题个数，利用已经训练好的 LDA 模型得到抽取段落的主题分布。

（3）将每个段落表示为主题分布后，利用 SVM 进行分类，判断锻炼所属的文章，验证 LDA 模型的准确性。

3.2 步骤说明

首先进行数据预处理，去除小说文本多余的符号和广告之类的无关内容，按行进行分词构成新的文本库，作为 LDA 模型训练的输入；从其中随机均匀抽取 200 个段落（每个段落大于 500 词），每个段落的标签就是对应段落所属的小说，作为测试数据。

之后将文本合集输入 LDA 模型，考虑到 16 本小说均为武侠小说，主题比较类似，因此设定 LDA 模型的主题数为 12。LDA 模型训练采用的是 python 中的 gensim 自然语言处理工具库。构建部分代码如下：

```
dictionary = corpora.Dictionary(train)
# 将每个段落进行 ID 化
corpus = [dictionary.doc2bow(text) for text in train]
#构建 LDA 模型，设定主题数为 12
lda = models.LdaModel(corpus=corpus, id2word=dictionary, num_topics=12)
topic_list_lda = lda.print_topics(12)
#由 LDA 模型得到段落的主题分布
topics_test = lda.get_document_topics(corpus_test)
```

LDA 模型得到的结果如下（这里只列举前三个主题中前六个主要词的概率分布）：

主题一	我	了	你	道	的	是
	0.041	0.041	0.040	0.039	0.037	0.029
主题二	寻思	杀人	此言	恭恭敬敬	一出	当场
	0.027	0.022	0.017	0.013	0.011	0.009
主题三	一掌	张	相助	不但	山上	能够
	0.044	0.042	0.019	0.016	0.012	0.012

可以看出，主题一是常用的代词和助词，因此出现的频率均较高。除此之外，这些主题主要词之间似乎并没有太大的联系。

接下来从文本合集中随机均匀抽取 200 个段落，每个段落不少于五百词，但部分小说中超过 500 词的段落较少，因此最终测试集里面只有 155 个段落。将这些段落输入 LDA 模型中就可以获得每个段落的主题分布。例如：

段落部分内容：之极 蓦地 里 飞出 一腿 将 苏普 手中 的 长刀 踢飞 了 称 他 是 哈萨克族 的 第一 勇士 不论 竞力 比拳 赛马 他 从没 输过 给 人 这 两个 面貌 凶恶 的 强人 实 是 害怕 之极 若 能 不斗 能够 虚张声势 的 将 他们 吓 她 不再 拉 缰绳 任由 白马 在 沙漠 中 漫步 而行 也 不知 走 了 多少 时候 …
主题分布：(5, 0.5886692), (12, 0.32380807)
段落部分内容：袁承志 道 以后 你别 叫 我 甚么 英雄 不 英雄 了 洪胜海 道 是 我 叫 你 相公 心中 暗喜 只要 跟定 了 你 再也 不怕 归二娘 和 孙仲君 这 两个 女贼 来 杀 我 了 三个 月 后 伤势 发作 你 自然 也 不会 袖手旁观…
主题分布：[(5, 0.53380525), (9, 0.24206825), (10, 0.13254221), (14, 0.05406001)]

可以看出，不同段落对应的主题分布也不相同。有的段落可能只对应两个主题，有的段落可能对应多个主题。但每个段落中基本上均对应一个主要主题，其概率大于 0.5。

之后，将这 200 个段落和它们对应的标签（即小说名称）按照 7:3 的比例划分为训练集和测试集，利用 python 中 sklearn 机器学习库提供的模型构建 SVM 分类器，进行分类。

```
#划分数据集
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(data_train, label, test_size = 0.3,)
#构建 SVM
svm = svm.SVC(kernel='linear')  # 实例化
svm.fit(X_train, Y_train)  # 拟合
pred = svm.predict(X_test)
for i in range(len(X_test)):
    if pred[i] == Y_test[i]:
        count+1
acc = (count/len(pred))*100#计算准确率
print("准确率:", acc, "%")
```

最终得到分类器的准确率为：46%。可以看出，LDA 模型对于 500 词左右短文本主题的分类效果较差，导致最终 SVM 分类得到的结果准确率很低。

四、收获、体会及建议

通过此次作业，我对 LDA 模型有了更深刻的理解，学习了 sklearn 库的相关使用方法，对自然语言处理的相关内容有了更深入的认识。