单位代码

学 号ZY2303803

分类号

****

**深度学习与自然语言处理（NLP）第二次课后作业**

|  |  |
| --- | --- |
| 院（系）名称 | 自动化科学与电气工程学院 |
| 专业名称 | 自动化 |
| 学生姓名 | 董晨辉 |

2024年05月

**深度学习与自然语言处理（NLP）第二次课后作业**

董晨辉

1127666815@qq.com

**Abstract**

从给定的语料库中均匀抽取1000个段落作为数据集（每个段落可以有 K 个 token, K 可以取20，100，500, 1000, 3000），每个段落的标签就是对应段落所属的小说。利用LDA模型在给定的语料库上进行文本建模，主题数量为 T，并把每个段落表示为主题分布后进行分类（分类器自由选择），分类结果使用 10 次交叉验证（i.e. 900 做训练，剩余100 做测试循环十次）。实现和讨论如下的方面：（1）在设定不同的主题个数T的情况下，分类性能是否有变化？；（2）以"词"和以"字"为基本单元下分类结果有什么差异？（3）不同的取值的K的短文本和长文本，主题模型性能上是否有差异？

**Introduction**

LDA是一种文档主题生成模型，也称为一个三层贝叶斯概率模型，包含词、主题和文档三层结构。LDA中文翻译为：潜在狄利克雷分布。LDA主题模型是一种文档生成模型，是一种非监督机器学习技术。它认为一篇文档是有多个主题的，而每个主题又对应着不同的词。一篇文档的构造过程，首先是以一定的概率选择某个主题，然后再在这个主题下以一定的概率选出某一个词，这样就生成了这篇文档的第一个词。不断重复这个过程，就生成了整篇文章(当然这里假定词与词之间是没有顺序的，即所有词无序的堆放在一个大袋子中，称之为词袋，这种方式可以使算法相对简化一些)。LDA的使用是上述文档生成过程的逆过程，即根据一篇得到的文档，去寻找出这篇文档的主题，以及这些主题所对应的词。LDA是NLP领域一个非常重要的非监督算法。

**Methodology**

**M1: 数据预处理**

1. 抽取每本小说的有效段落

要求段落大于500字，选择“倚天屠龙记”，“笑傲江湖”，“天龙八部”等小说作为数据样本。

1. 均匀抽取段落

一共抽取1000个段落，每本小说均匀随机抽取，其中900个段落为训练集，剩下100个段落为测试集，段落标签为对应的小说名。

1. 段落处理

去除空格；使用jieba进行分词；去除停用词；过滤词性，只保留名词词性；只保留中文字符。

**M2: LDA 模型和 SVM 模型实现**

1. LDA 模型实现

调用函数gensim.models.ldamodel.LdaModel()。该函数的重要参数设置如下：

corpus=train\_corpus：由操作[id2word.doc2bow(text)fortextintrain\_data]得到train\_corpus，词典转化为词袋，是一组向量，记录了train\_data中每个段落的词袋，每个向量的item为(词id，词频)。

id2word=id2word：由函数corpora.Dictionary(train\_data+test\_data)生成词典id2word，不重复地记录文本中的单词。

num\_topics=5：设置的主题分布的主题数，即主题分布向量维度。

1. SVM 模型实现

调用 sklearn 的包 from sklearn.svm import SVC

**M3: 训练**

1. 统计段落中的所有不重复单词得到的字典：
2. 得到段落的词袋向量：
3. 将1，2的数据放入 LDA 模型中训练得到五个主题的词分布：

**Experimental Studies**

**使用数据集进行试验结果如下所示：**

1. 在设定不同的主题个数T的情况下，分类性能是否有变化？

针对主题个数T分别设置为5，10，15，20，25，30，50，100，200，300，1000，3000进行实验。针对不同的k值，分别设定k为20、100、500、1000、3000进行试验，得到的结果如下图所示：

表 1 不同主题个数和token个数下的分类性能表现

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主题个数\token数 | 20 | 100 | 500 | 1000 | 3000 |
| 5 | 0.14 | 0.36 | 0.26 | 0.49 | 0.75 |
| 10 | 0.24 | 0.32 | 0.49 | 0.53 | 0.92 |
| 20 | 0.26 | 0.3 | 0.49 | 0.64 | 0.84 |
| 30 | 0.25 | 0.36 | 0.53 | 0.65 | 0.93 |
| 40 | 0.24 | 0.32 | 0.56 | 0.7 | 0.92 |
| 50 | 0.33 | 0.3 | 0.32 | 0.52 | 0.84 |
| 70 | 0.29 | 0.35 | 0.43 | 0.7 | 0.86 |
| 100 | 0.33 | 0.41 | 0.44 | 0.77 | 0.76 |
| 150 | 0.26 | 0.39 | 0.52 | 0.79 | 0.37 |
| 200 | 0.3 | 0.42 | 0.21 | 0.66 | 0.62 |
| 300 | 0.33 | 0.5 | 0.45 | 0.71 | 0.78 |
| 1000 | 0.19 | 0.54 | 0.45 | 0.65 | 0.71 |
| 3000 | 0.19 | 0.27 | 0.41 | 0.62 | 0.54 |

可以看出随着主题个数的增加，训练准确度和测试准确度在增高，针对不同的k值，主题数的增加所带来的准确率的提升不太一样，但都符合先增加后逐渐减小的特征。

1. 以"词"和以"字"为基本单元下分类结果有什么差异？

在主题数 T 为 50，k 为 1000，使用SVC分类器的情况下分类性能的变化如下表所示，其中分词使用jieba分词：

表 2 不同分词方法下的分类性能表现

|  |  |  |
| --- | --- | --- |
| 基本单元 | Token = 100准确度 | Token = 10准确度 |
| 词 | 0.55 | 0.22 |
| 字 | 0.87 | 0.14 |

从表 2可以看出以字为基本单元在token数较大的情况下准确度较高。这可能与先进行分词再进行去除停用词的操作有关，由于按字分割情况下产生的停用词较多，再满足相同token的情况下取的信息量更大导致准确度高。而token数较少的情况下反而按词分类的效果更好，这是有与按词分割保存的信息量较多。

1. 不同的取值的K的短文本和长文本，主题模型性能上是否有差异？

从表 1中的数据可以看出，随着token个数的增加，准确度不断的增加。

**Conclusions**

经过实验可以看出，随着主题数的增加，分类准确率先升高再下降这可能与真实的主题个数为16（一共有16本小说）有关，当主题数小于5时属于欠拟合状态，当主题数较大时则会出现过拟合。并且每次重复实验发现分类准确率有较大的误差，可能是SVM参数需要改进。 同时，在token数增加时，准确度能够较大地增加，这是由于随着token数增加，信息量更大，LDA的性能更好。

经过本次作业与课堂学习，我对LDA主题模型有了全局的认识，系统地学习了其原理，通过学习，对LDA主题模型生成文档的原理有了深刻的理解。此次作业应用主题分布和文本分类，加深了我们LDA如何进行训练并求得主题分布过程的熟悉程度。