LDA

——自然语言处理的深度学习报告(第三次)

崔潇雅 ZY2203502

1 摘要

本实验从 16 本金庸小说中随即均匀抽取 200 段落，每个段落标签即对应的小说名称。之后分别在字模式和词模式下进行 LDA 文本建模，计算出段落的主题分布与主题的词分布，借助困惑度与主题一致性对 LDA 模型进行性能分析。结果表明，主题的数目会对分类性能有所影响，不同的文本表示方式分类结果也有所不同，字模式在准确表达文本含义上更胜一筹。

2 理论基础

2.1 LDA 模型

LDA（Latent Dirichlet Allocation）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。所谓生成模型，就是说，我们认为一篇文章的每个词都是通过“以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语”这样一个过程得到。文档到主题服从多项式分布，主题到词服从多项式分布。

LDA 是一种非监督机器学习技术，可以用来识别大规模文档集（document collection）或语料库（corpus）中潜藏的主题信息。它采用了词袋（bag of words）的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，同时也为模型的改进提供了契机。每一篇文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。其生成过程如下：

对于语料库中的每篇文档，LDA 定义了如下生成过程：

(1)对每一篇文档，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;

(3)重复上述过程直至遍历文档中的每一个单词。

语料库中的每一篇文档与 T（通过反复试验等方法事先给定）个主题的一个多项分布 相对应，将该多项分布记为θ。每个主题又与词汇表中的 V 个单词的一个多项分布相对应，将这个多项分布记为φ。

2.2 困惑度（perplexity）、主题一致性（coherence）

困惑度表示模型对未见过的文本数据的预测性能。困惑度越小，模型的预测性能越好。主题一致性是通过计算每个主题中单词之间的相似度来评估主题的质量。主题一致性的值越大，主题之间的相似度越高，LDA 模型的质量越好。

3 实验过程

3.1 数据处理

包括（1）去除无意义的广告（2）加载停词表（先不去除停词，选取段落之后再去除停词）（3）去除出现频率较高的影响主题选取的词。

```
def content_deal(content):  # 语料预处理，进行断句，去除一些广告和无意义内容
    ad = '本书来自www.cr173.com免费txt小说下载站\n更多更新免费电子书请关注www.cr173.com'
    content = content.replace(ad, '')
    return content
```

```
#读取停词列表
stop_word_list = []
with open(stop_word_file,'r',encoding='utf-8') as f:
    for line in f:
        stop_word_list.append(line.strip())
stop_word_list.extend("\u3000")
stop_word_list.extend(['～',' ','没','听','一声','道', '见', '中', '便', '说', '一个','说道'])
```

3.2 段落生成

首先读取 16 篇文档，筛选出所有大于 500 词的段落，再使用 random 方法随机抽取 200 个段落，段落标签就是文章名称。

```
def read_novel(path):
    file_list = os.listdir(path)
    data_list = []
    data_label = []
    test_list = []
    test_label = []

    for file in file_list:
        novel_path = "E:/DLNL/--main/NLP-3/jinyong" + '/' + file
        with open (novel_path,'r',encoding='ANSI')as f:
            content = f.read()
            word_list0 = content_deal(content)

            #大于500词的段落
            for para in word_list0.split('\n'):
                if len(para)<500:
                    continue
                data_list.append(para)
                data_label.append(file)
                f.close()
    #随机200段落
    random_way = random.sample(range(len(data_list)), 200)
    test_list.extend([data_list[i] for i in random_way])
    test_label.extend([data_label[i] for i in random_way])
    #返回所有段落及其标签，选取的200段落及其标签
    return data_list,data_label,test_list,test_label
```

3.3 分词与构建词典、稀疏向量

```
#分词
#词模式
fenci_word= []
fenci_word_label=[]
fenci_char = []
fenci_char_label = []

for index,text in enumerate(test_list):
    fenci = [word for word in jieba.lcut(sentence=text) if word not in stop_word_list]
    fenci_word.append(fenci)
    fenci_word_label.append(test_label[index])
#字模式
    t = []
    for word1 in fenci:
        t.extend([char for char in word1])
    fenci_char.append(t)
    fenci_char_label.append(test_label[index])


#构建词典,形成稀疏向量
dic_word = corpora.Dictionary(fenci_word)
cor_word = [dic_word.doc2bow(i)for i in fenci_word]
dic_char = corpora.Dictionary(fenci_char)
cor_char = [dic_char.doc2bow(i)for i in fenci_char]
```

3.4 计算 para-topic 分布、topic-para 分布（词模式下）

```
#训练lda
num_topic = 6
lda_word = models.ldamodel.LdaModel(corpus=cor_word, id2word=dic_word, num_topics=num_topic)
print(lda_word)
# topic-word分布
for topic in lda_word.print_topics(num_words=10):
    print(topic)
#para-topic分布
    for e, values in enumerate(lda_word.inference(cor_word)[0]):
        print(test_list[e])
        for ee, value in enumerate(values):
            print('\t主题%d推断值%.2f' % (ee, value))

# 对于每个主题，所有词对应的概率，求和=1
print('概率总和', sum(i[1] for i in lda_word.show_topic(0, 9999)))
```

设置主题为 6 时的结果：

（1）topic-para

李沅芷心想，包袱一给阎世章拿到，他武功强，抢回来就不容易，灵机一动，在霍青桐耳边说了几句话，随即除下帽子，把长发披在面前，取出块手帕蒙住下半截脸，在地下拾起两块砖头，使劲向窗上掷去，碰破窗格，直打进房里。房里灯火骤灭，房门一开，窜出五六个人来。当先一人喝道：“甚么东西？胆子倒不小。”霍青桐胡哨一声，翻身出墙，众镖师纷纷追出。李沅芷待众镖师和那张大人追出墙去，直闯进房。童兆和被人点了大半天的穴，刚救治过来，手脚还不灵便，躺在炕上，见门外闯进一个披头散发、鬼不像鬼、人不像人的东西来，双脚进跳，口中吱吱直叫，登时吓得全身软瘫。那鬼跳将过来，在他手中将红包袱一把抢过去，吱吱吱的又跳出房去。众镖师追出数步，那张大人忽地住脚，道：“糟了，这是调虎离山之计，快回去！”阎世章等也即醒悟，回到店房，只见童兆和倒在炕上，呆了半晌，才把鬼抢包袱之事说了。张大人恨道：“甚么鬼？咱们阴沟里翻船，几十年的老江湖着了道儿。”李沅芷抢了包袱，躲在墙边，待众镖师都进了房，才翻墙出去。她轻轻吹了记口哨，对面树荫下有人应了一声，两个人影迎将上来，正是陆菲青和霍青桐。李沅芷得意非凡，笑道：“包袱抢回来了，可不怪我了吧……”一句话没说完，陆菲青叫道：“小心后面。”李沅芷正待回头，肩上已被人拍了一下，她反手急扣，却没扣住敌人手腕，心中一惊，知是来了强敌，此人悄没声的跟在后面，自己竟丝毫不觉，急忙转身，月光下只见一个身材魁梧的汉子站在面前。她万想不到敌人站得如此之近，惊得倒退两步，扬手将包袱向霍青桐掷去，叫道：“接着。”双手一错，护身迎敌。哪知敌人身法奇快，她包袱刚掷出，敌人已跟着纵起，一伸手，半路上截下了包袱，李沅芷又惊又怒，迎面一拳，同时霍青桐也从后攻到。那人左手拿住包袱，双手一分，使出的势子竟是武当长拳中的“高四平，气劲力足，把李沅芷和霍青桐同时震得倒退数步。李沅芷这时看清了敌人，正是那个张大人。武当长拳是武当派的入门功夫，她跟陆菲青学艺，学了练气的十段锦后，最先学的就是这套拳术，哪知平平常常一招“高四平”，在敌人手下使出来竟有如斯威力，不禁倒抽了口凉气，回头一望，师父却已不知去向。

主题0推断值306.41
主题1推断值0.17
主题2推断值0.17
主题3推断值0.17
主题4推断值0.17
主题5推断值0.17

郭靖道：“我拿了他一些治内伤的药，他大大生气，非杀了我不可。前辈可是受了伤？弟子这里有很多药，其中四味是田七、血竭、熊胆、没药，王道长也不需用这许多，前辈要是……”那女人怒道：“我受甚么伤，谁要你讨好？”郭靖碰了一个钉子，忙道：“是，是。”隔了片刻，听她不住喘气，心中不忍，又道：“前辈要是行走不便，晚辈负你老人家出去。”那女人骂道：“谁老啦？你这浑小子怎知我是老人家？”郭靖唯唯，不敢作声，要想舍她而去，总感不安，当下硬起头皮，又问：“您可要甚么应用物品，我去给您拿来。”那女人冷笑道：“你婆婆妈妈的，倒真好心。”左手伸出，搭在他肩头向里一拉，郭靖只觉肩上剧痛，身不由主的到了她面前，忽觉颈中一阵冰凉，那女人的右臂已拖住他头颈，只听她喝道：“背我出去。”郭靖心想：“我本来要背你出去。”于是转身弯腰，慢慢走出地道，那女人道：“是我逼着你背的，我可不受人卖好。”郭靖这才明白，这女人骄傲得紧，不肯受后辈的恩惠。走到洞口，举头上望，看到了天上的星星，不由得吁了口长气，心想：“刚才真是死里逃生，这黑洞之中，竟有人等着救我性命。我去说给蓉儿听，只怕她还不肯信呢。”他跟着马钰行走悬崖惯了的，那洞虽如深井，却也毫不费力的攀援了上去。出得洞来，那女子问道：“你这轻功是谁教的？快说！”手臂忽紧，郭靖喉头被扼，几乎喘不过气来。他心中惊慌，忙运内力抵御。那女人故意要试他功力，扼得更加紧了，过了一阵，才渐渐放松，喝道：“嘿，看你不出，浑小子还会玄门正宗的内功。你说王道长受了伤，王道长叫甚么名字？”郭靖心道：“你救了我性命，要问甚么，自然不会瞒你，何必动蛮？”当下答道：“王道长名叫王处一，人家称他为玉阳子。”突觉背上那女人身子一震，又听她气喘喘的道：“你是全真门下的弟子？那……那好得很。”语音中竟流露出情不自禁的欢愉之意，又问：“王处一是你甚么人？于么你叫他道长，不称他师父、师叔、师伯？”郭靖道：“弟子不是全真门下，不过丹阳子马钰马道长传过我一些呼吸吐纳的功夫。”那女人道：“嗯，你学过全真派内功，很好。”隔了一会，问道：“那么你师父是谁？”郭靖道：“弟子共有七位师尊，人称江南七侠。大师父飞天蝙蝠姓柯。”那女人剧烈的咳嗽了几下，声音甚是苦涩，说道：“那是柯镇恶！”郭靖道：“是。”那女人道：“你从蒙古来？”郭靖又道：“是。”心下奇怪：“她怎么知道我从蒙古来？”

主题0推断值0.17
主题1推断值0.17
主题2推断值0.17
主题3推断值0.17
主题4推断值286.96
主题5推断值0.17

(2) topic-word（每个主题由 10 个词描述）

```
In [135]: runfile('E:/DLNL/--main/NLP-3/test--.py', wdir='E:/DLNL/--main/NLP-3')
LdaModel(num_terms=21380, num_topics=6, decay=0.5, chunksize=2000)
(0, '0.003*"武功" + 0.002*"心中" + 0.002*"陈家洛" + 0.002*"韦小宝" + 0.002*"只见" + 0.002*"不知" + 0.002*"知道" + 0.002*"想" + 0.001*"心想" + 0.001*"走"')
(1, '0.002*"瞧" + 0.002*"郭靖" + 0.002*"杨过" + 0.002*"韦小宝" + 0.002*"武功" + 0.002*"不知" + 0.002*"死" + 0.002*"陈家洛" + 0.002*"只见" + 0.002*"出"')
(2, '0.003*"张无忌" + 0.002*"令狐冲" + 0.002*"走" + 0.002*"武功" + 0.002*"想" + 0.002*"韦小宝" + 0.002*"胡斐" + 0.002*"心中" + 0.002*"死" + 0.002*"只见"')
(3, '0.003*"武功" + 0.003*"笑" + 0.002*"令狐冲" + 0.002*"想" + 0.002*"师父" + 0.002*"心中" + 0.002*"不知" + 0.002*"只见" + 0.002*"黄蓉" + 0.002*"陈家洛"')
(4, '0.003*"韦小宝" + 0.003*"令狐冲" + 0.002*"想" + 0.002*"张无忌" + 0.002*"只见" + 0.002*"心想" + 0.002*"走" + 0.002*"两人" + 0.002*"弟子" + 0.002*"笑"')
(5, '0.002*"令狐冲" + 0.002*"武功" + 0.002*"想" + 0.002*"杨过" + 0.002*"笑" + 0.002*"只见" + 0.002*"心中" + 0.002*"不知" + 0.002*"走" + 0.002*"派"')
```

3.4 LDA 模型训练与评价

分别在词模式、字模式下主题数由 1-6 递增，计算困惑度与一致性。

```
x = [] # x轴
perplexity_values_word = [] # 困惑度
coherence_values_word = []   # 一致性
perplexity_values_char = [] # 困惑度
coherence_values_char = []   # 一致性
model_list = [] # 存储对应主题数量下的lda模型,便于生成可视化网页

for topic in range(num_topic):
    print("主题数量: ", topic+1)
    lda_word=models.ldamodel.LdaModel(corpus=cor_word, num_topics=topic+1, id2word =dic_word, chunksize = 2000, passes=20, iterations = 400)
    lda_char=models.ldamodel.LdaModel(corpus=cor_char, num_topics=topic+1, id2word =dic_char, chunksize = 2000, passes=20, iterations = 400)
    model_list.append(lda_word)
    x.append(topic+1)
    perplexity_values_word.append(-lda_word.log_perplexity(cor_word))
    coherencemodel_word = models.CoherenceModel(model=lda_word, texts=fenci_word, dictionary=dic_word, coherence='c_v')
    coherence_values_word.append(coherencemodel_word.get_coherence())

    perplexity_values_char.append(-lda_char.log_perplexity(cor_char))
    coherencemodel_char = models.CoherenceModel(model=lda_char, texts=fenci_char, dictionary=dic_char, coherence='c_v')
    coherence_values_char.append(coherencemodel_char.get_coherence())

    print("该主题评价完成\n")
```

4 实验结果

4.1 不同数量段落的主题分布

num_topic=5

```
In [140]: runfile('E:/DLNL/--main/NLP-3/test--.py', wdir='E:/DLNL/--main/NLP-3')
LdaModel(num_terms=22000, num_topics=5, decay=0.5, chunksize=2000)
(0, '0.002*"杨过" + 0.002*"想" + 0.002*"令狐冲" + 0.002*"走" + 0.002*"师父"')
(1, '0.003*"韦小宝" + 0.002*"令狐冲" + 0.002*"杨过" + 0.002*"心中" + 0.002*"想"')
(2, '0.003*"杨过" + 0.002*"心中" + 0.002*"师父" + 0.002*"不知" + 0.002*"想"')
(3, '0.002*"笑" + 0.002*"韦小宝" + 0.002*"陈家洛" + 0.002*"瞧" + 0.002*"不知"')
(4, '0.003*"杨过" + 0.002*"法王" + 0.002*"韦小宝" + 0.002*"想" + 0.002*"走"')
```

num_topic=10

```
In [141]: runfile('E:/DLNL/--main/NLP-3/test--.py', wdir='E:/DLNL/--main/NLP-3')
LdaModel(num_terms=21820, num_topics=10, decay=0.5, chunksize=2000)
(0, '0.003*"知道" + 0.002*"黄蓉" + 0.002*"不知" + 0.002*"武功" + 0.002*"笑"')
(1, '0.003*"笑" + 0.002*"郭" + 0.002*"陈家洛" + 0.002*"知道" + 0.002*"心中"')
(2, '0.003*"陈家洛" + 0.002*"韦小宝" + 0.002*"知道" + 0.002*"令狐冲" + 0.002*"死"')
(3, '0.002*"笑" + 0.002*"师父" + 0.002*"倒" + 0.002*"不知" + 0.002*"知道"')
(4, '0.002*"走" + 0.002*"笑" + 0.002*"韦小宝" + 0.002*"师父" + 0.002*"事"')
(5, '0.002*"想" + 0.002*"黄蓉" + 0.002*"芙" + 0.002*"瞧" + 0.002*"武功"')
(6, '0.006*"令狐冲" + 0.003*"韦小宝" + 0.003*"笑" + 0.002*"知道" + 0.002*"师父"')
(7, '0.003*"想" + 0.002*"瞧" + 0.002*"张无忌" + 0.002*"心中" + 0.002*"武功"')
(8, '0.003*"武功" + 0.002*"派" + 0.002*"韦小宝" + 0.002*"陈家洛" + 0.002*"笑"')
(9, '0.002*"韦小宝" + 0.002*"众人" + 0.002*"瞧" + 0.002*"便是" + 0.002*"想"')
```

num_topic=15

```
In [142]: runfile('E:/DLNL/--main/NLP-3/test--.py', wdir='E:/DLNL/--main/NLP-3')
LdaModel(num_terms=18340, num_topics=15, decay=0.5, chunksize=2000)
(0, '0.003*"郭靖" + 0.003*"想" + 0.002*"师父" + 0.002*"心中" + 0.002*"杨过"')
(1, '0.003*"师父" + 0.003*"武功" + 0.002*"令狐冲" + 0.002*"想" + 0.002*"郭靖"')
(2, '0.002*"心中" + 0.002*"笑" + 0.002*"武功" + 0.002*"只见" + 0.002*"令狐冲"')
(3, '0.002*"武功" + 0.002*"师父" + 0.002*"知道" + 0.002*"袁承志" + 0.002*"不知"')
(4, '0.005*"令狐冲" + 0.003*"想" + 0.002*"师父" + 0.002*"心中" + 0.002*"剑法"')
(5, '0.003*"笑" + 0.003*"想" + 0.002*"郭靖" + 0.002*"两人" + 0.002*"走"')
(6, '0.003*"韦小宝" + 0.003*"笑" + 0.003*"令狐冲" + 0.002*"张翠山" + 0.002*"想"')
(7, '0.003*"武功" + 0.002*"韦小宝" + 0.002*"想" + 0.002*"笑" + 0.002*"不知"')
(8, '0.003*"师父" + 0.003*"想" + 0.002*"不知" + 0.002*"武功" + 0.002*"事"')
(9, '0.003*"师父" + 0.002*"令狐冲" + 0.002*"倒" + 0.002*"袁承志" + 0.002*"武功"')
(10, '0.004*"袁承志" + 0.003*"想" + 0.002*"令狐冲" + 0.002*"请" + 0.002*"只见"')
(11, '0.003*"韦小宝" + 0.003*"笑" + 0.003*"心想" + 0.003*"袁承志" + 0.003*"瞧"')
(12, '0.008*"袁承志" + 0.002*"两人" + 0.002*"瞧" + 0.002*"武功" + 0.002*"心想"')
(13, '0.003*"袁承志" + 0.002*"想" + 0.002*"倒" + 0.002*"青青" + 0.002*"郭靖"')
(14, '0.003*"令狐冲" + 0.003*"袁承志" + 0.003*"师父" + 0.003*"瞧" + 0.003*"笑"')
```

4.1 不同基本单元下分类结果

表 1　LDA 模型不同文本单元主题一致性对比

Num_Topic	一致性（字）	一致性（词）
1	0.23673565802004623	0.22634753144245687
2	0.24963987914874763	0.25717952233314423
3	0.2595375486880317	0.25927352141982335
4	0.2710362793631204	0.3173249033780411
5	0.27820438150665533	0.3316718728611096
6	0.2729323601600798	0.3268288822422649
7	0.27311033669915813	0.34851889413289167
8	0.28242788544149466	0.3879758175773056
9	0.28302226261061186	0.3982226155401636
10	0.3116868867903648	0.3901327079806522
11	0.3203572815655646	0.4068072462671621
12	0.28136122145183134	0.40359711685147387
13	0.2979798707231454	0.38145546205514297
14	0.23673565802004623	0.39125871153022584
15	0.24963987914874763	0.4314041577365427

表 2　LDA 模型不同文本单元困惑度对比

Num_Topic	困惑度（字）	困惑度（词）
1	6.925008344470357	9.441383720940571
2	6.901675457654887	9.437746241868831
3	6.890123969217685	9.441455011603942
4	6.894513020298902	9.444191253229981
5	6.882705630629449	9.456336658951624
6	6.879127765352368	9.455950927835216
7	6.898067888252665	9.473685308429536
8	6.901540334448041	9.48544587205587
9	6.890803713461132	9.495749617363078
10	6.91246871616859	9.49270559984563
11	6.910186705954722	9.50623138115343
12	6.908691839570805	9.506911969343806
13	6.914991477075096	9.548588400236634
14	6.923166038773174	9.53248525766108
15	6.922545611553939	9.531621232066488

（1）在不同数量的主题个数下分类性能的变化

实验表明，文本单元不同、主题数目不同可能会影响LDA模型对文本建模的效果。首先以词模式为例观察在不同数量的主题个数下分类性能的变化，可以看到困惑度随着topic数目增加变大，主题一致性也变大；以字模式为例观察，也是呈现这样的趋势，说明主题数目增加，每个主题之间相似性更高，LDA模型效果越好。

（2）以“词”和以“字”为基本单元下分类结果差异

从困惑度来看，词模式整体要高于字模式，说明文本单元为“字”可以更好的表示文本性能，将原文义表达地更准确。

（3）实验总结

本次实验通过构建LDA模型，对金庸小说集主题进行分类，测试效果良好。本实验也加深了我对自然语言处理用途的理解，同时也存在不足之处，本次代码进行主题模型评价时运行时间较长，可以考虑对Topic进行精炼，选取重要的关键词进行处理，降低运行的时间。

（4）过拟合现象

文本单元选择字模式时虽然能更细致地进行文本建模，但当主题数目过高时会出现过拟合现象，本实验通过画出字模式下主题数-一致性曲线来确定最优主题数目为11。

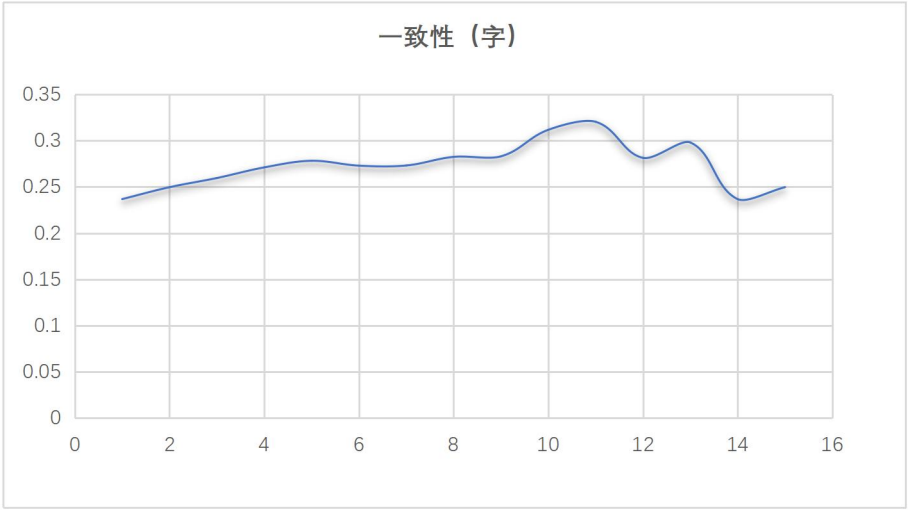

图 1　LDA 模型字模式过拟合现象