单位代码_____

学　　号 SY2303801

分 类 号_____

北京航空航天大學

BEIHANG UNIVERSITY

基于金庸小说数据集的 LDA 主题提取与分类

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　　成城

2024 年 04 月

1 内容介绍

从下面链接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？

2 实验原理

2.1 主题模型

2.1.1 主题模型介绍

一篇文档应该有多个主题，每个主题的比例不同，每一个主题下面也应该有很多词语，每个词语的比例也不同。主题模型就是用数学框架来体现出文档的这种特点，主题模型自动分析每篇文档，统计文档内的词语，根据统计的信息来断定当前文档含有哪些主题，以及每个主题所占的比例各为多少。从上面的定义可以看出，主题模型其实主要在学习两个分布，文档-主题分布（doc-topic）和主题-词分布（topic-word）。既然是分布就要满足两个条件，第一是非负性，第二是积分或者求和为 1。也就是 doc-topic 矩阵或 topic-word 矩阵中，任意一行元素均为非负数且元素和为 1 。Topic models 主要可以分为四大类：

1.无监督无层次结构， 主要有： PLSA(Hofmann 1999), LDA, Correlated Topic Model CTM 主要是为了克服标准 LDA 模型不能建模话题在文档中出现的 相关性的缺点，将 LDA 中文档话题分布服从的 Dirichlet 分布改为 Logistic 正态 分 布。例如 CTM 论文中举的一个例子是在 Science 杂志语料中，一篇遗传学文 章很可能也跟健康和疾病有关，但是却不大可能跟射线天文学有关。因为Logistic 正态分布不再是 Multinomial 分布的共轭分布，因此模型的解

变得更加复杂。对 此，作者使用的方法是，在变分推理的过程中，继续使用 Taylor 展开式以简化似 然函数下界的复杂性。

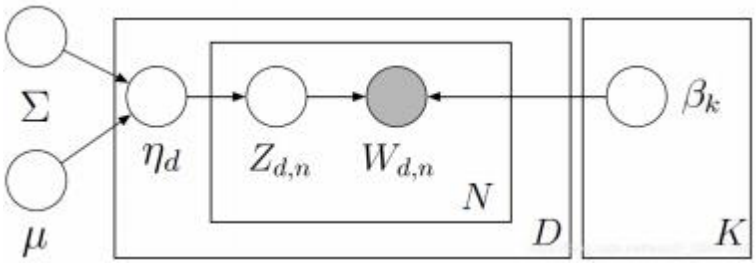


图 1 Model 主题模型结构

2. 无监督有层次结构, 主要有：HLDA，HDP：标准 LDA 模型中话题的个数 K 需要已知，然而很多时候确定 K 的大小是一件困难的事情。HDP 能够根据数 据自动确定 K 的大小。

3. 有监督无层次结构, 主要有：S-LDA ， Disc-LDA, MM-LDA, Author-Model, Labeled LDA, PLDA 等。

4. 有监督有层次结构, 主要有： hLLDA, HSLDA。

除上述集中类型的话题模型外， 还有一些半监督的话题模型， 主要有：Semi- LDA, SSHLDA。

2.1.2 LDA 模型

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。

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或语料库中潜藏的主题信息。它采用了词袋的方法，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。

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。

先定义一些字母的含义：文档集合 D，主题（topic)集合 T

D 中每个文档 d 看作一个单词序列$<w1,w2, \ldots ,wn>$，wi表示第i个单词，设 d 有 n 个单词。D 中涉及的所有不同单词组成一个大集合，LDA 以文档集合 D 作为输入，希望训练出的两个结果向量：对每个 D 中的文档 d，对应到不同 Topic 的概率$\theta d = <pt1, \ldots , ptk>$，其中，$pti$表示 d 对应 T 中第 i 个 topic 的概率。计算 方法是直观的，$pti = nti/n$，其中nti表示 d 中对应第 i 个 topic 的词的数目，n 是 d 中所有词的总数。 对每个 T 中的 topict ，生成不同单词的概率$\varphi t = <pw1, \ldots , pwm>$，其中，$pwi$表示 t 生成 VOC 中第 i 个单词的概率。计算方法同 样很直观，$pwi = Nwi/N$，其中Nwi表示对应到 topict 的 VOC 中第 i 个单词的数

目，N 表示所有对应到topict 的单词总数。LDA 的核心公式如下：

$$p(w|d) = p(w|t) * p(t|d)$$

直观的看这个公式，就是以 Topic 作为中间层，可以通过当前的θd和φt给出 了文档 d 中出现单词 w 的概率。其中$p(t|d)$利用θd计算得到，$p(w|t)$利用φt计 算得到。

实际上，利用当前的θd和φt，我们可以为一个文档中的一个单词计算它对应任意一个 Topic 时的$p(w|d)$，然后根据这些结果来更新这个词应该对应的 topic 。然后，如果这个更新改变了这个单词所对应的 Topic，就会反过来影响θd和φt 。

LDA 算法开始时， 先随机地给θd和φt赋值（对所有的d 和 t）。然后上述过 程不断重复，最终收敛到的结果就是 LDA 的输出。再详细说一下这个迭代的学 习过程：

1.针对一个特定的文档 ds 中的第 i 单词 wi，如果令该单词对应的 topic 为 tj，可以把上述公式改写为：

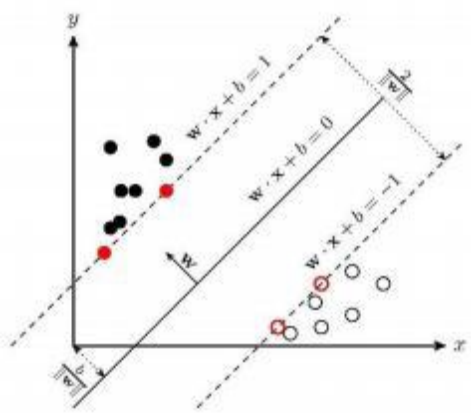
$$pj(wi|ds) = p(wi|tj) * p(tj|ds)$$

2.现在我们可以枚举 T 中的 topic，得到所有的$pj(wi|ds)$，其中 j 取值 1~k。然后可以根据这些概率值结果为 ds 中的第 i 个单词 wi 选择一个 topic。最简单的想法是取令$pj(wi|ds)$最大的 tj （注意，这个式子里只有 j 是变量）。

3.然后，如果 ds 中的第 i 个单词wi 在这里选择了一个与原先不同的 topic ，就会对θd和φt有影响了（根据前面提到过的这两个向量的计算公式可以很容易知道）。它们的影响又会反过来影响对上面提到的$p(w|d)$的计算。对 D 中所有 的 d 中的所有 w 进行一次$p(w|d)$的计算并重新选择 topic 看作一次迭代。这样进 行 n 次循环迭代之后，就会收敛到 LDA 所需要的结果了。

2.1.3 SVM分类器

SVM 学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如下图所示， wx+b=0 即为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个（即感知机），但是几何间隔最大的分离超平面却是唯一的。



SVM 方法建立在统计学 VC 维和结构风险最小化原则上，既可以用于分类（二/多分类）、也可用于回归和异常值检测。 SVM 具有良好的鲁棒性，对未知数据拥有很强的泛化能力，特别是在数据量较少的情况下，相较其他传统机器学习算法具有更优的性能。

使用 SVM 作为模型时，通常采用如下流程：

对样本数据进行归一化；应用核函数对样本进行映射（最常采用和核函数是 RBF 和 Linear，在样本线性可分时，Linear 效果要比 RBF 好）；用 cross-validation 和 grid-search 对超参数进行优选；用最优参数训练得到模型；测试。

sklearn 中支持向量分类主要有三种方法：SVC 、NuSVC 、LinearSVC。本次实验采用 SVC 分类器。

3 实验过程

3.1 流程

1. 数据预处理

（1）抽取每本小说的有效段落

要求均匀抽取1000个段落,选择 “倚天屠龙记”，“笑傲江湖”，“天龙八部”，“射雕英雄传”，“神雕侠侣”五本书，每本抽取200个段落，以这五本小说作为数据样本。

（2）均匀抽取段落

一共抽取 1000 个段落，每本小说均匀随机抽取 200 个段落,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。

（3）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。

2. 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) for text in train_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。

（2）SVM 模型实现

调用 sklearn 的包 from sklearn.svm import SVC，该函数的重要参数设置如下：

kernel='poly'：算法中采用的核函数类型，核函数是用来将非线性问题转化为 线性问题的一种方法。参数选择有 RBF, Linear, Poly, Sigmoid ， precomputed。这 里选择 Poly 指的是多项式核。

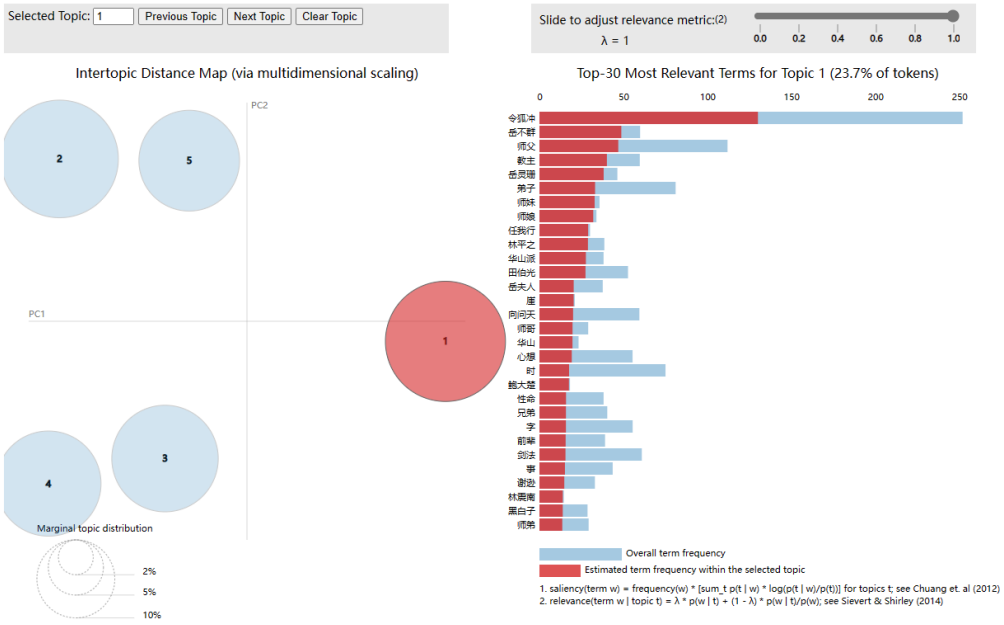
degree=3 ：当指定 kernel 为'poly'时，表示选择的多项式的最高次数。

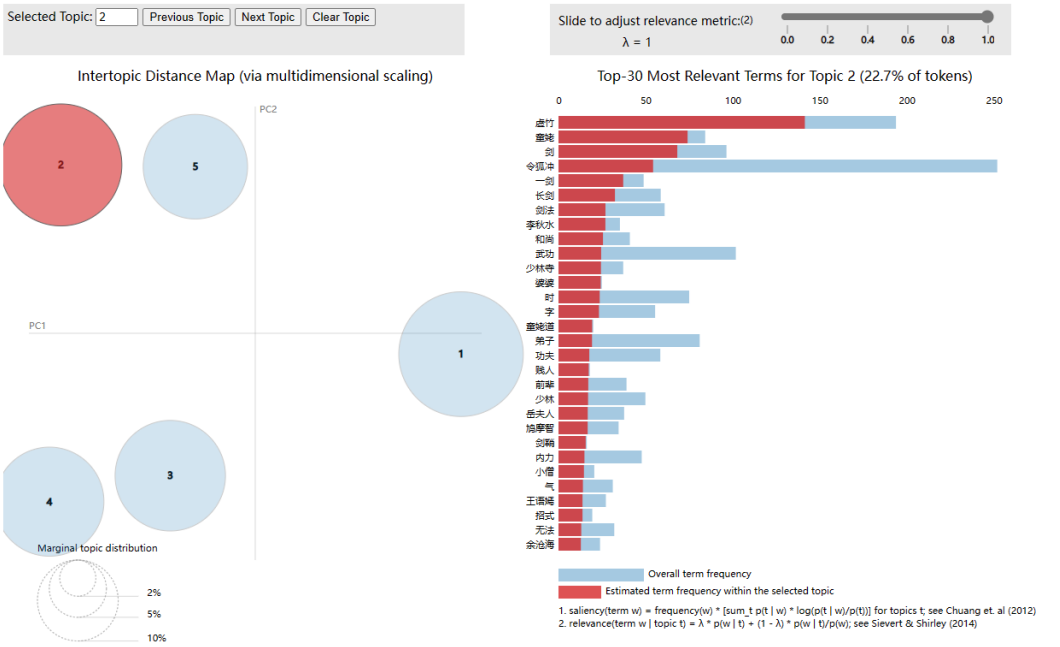
C=1.0 ：惩罚系数，用来控制损失函数的惩罚系数。 C 越大， 相当于惩罚松 弛变量，希望松弛变量接近 0，即对误分类的惩罚增大，趋向于对训练集全分对 的情况，这样会出现训练集测试时准确率很高，但泛化能力弱，容易导致过拟合。C 值小，对误分类的惩罚减小，容错能力增强，泛化能力较强，但也可能欠拟合。

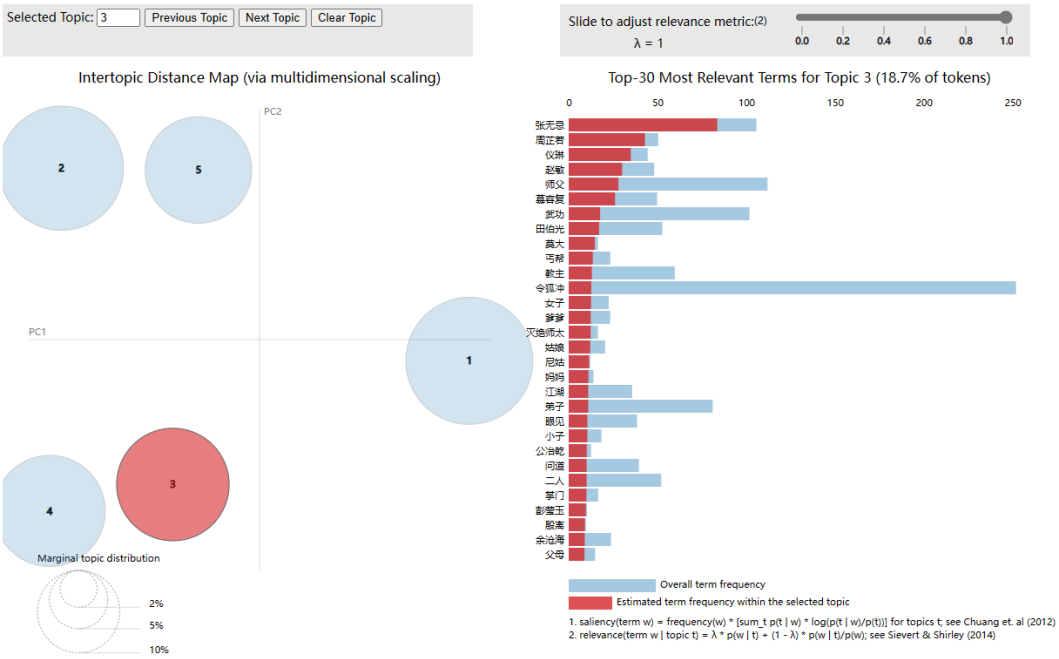
tol=0.0001 ：残差收敛条件，默认是 0.0001，即容忍 1000 分类里出现 一个错误，与 LR 中的一致；误差项达到指定值时则停止训练。

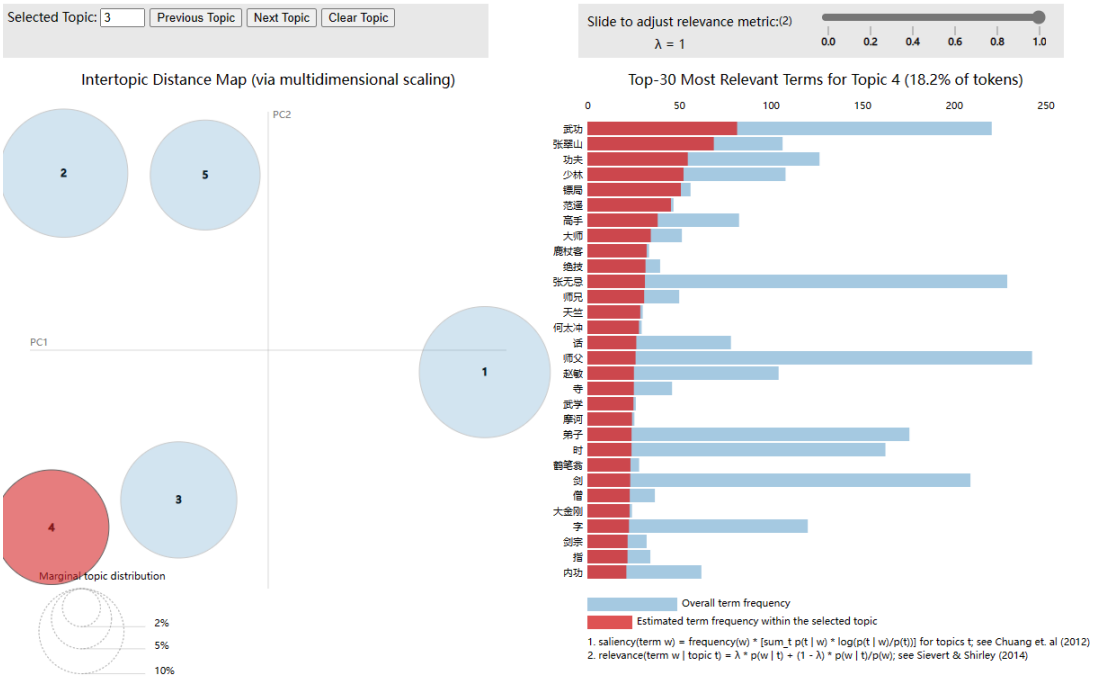
3.2 训练

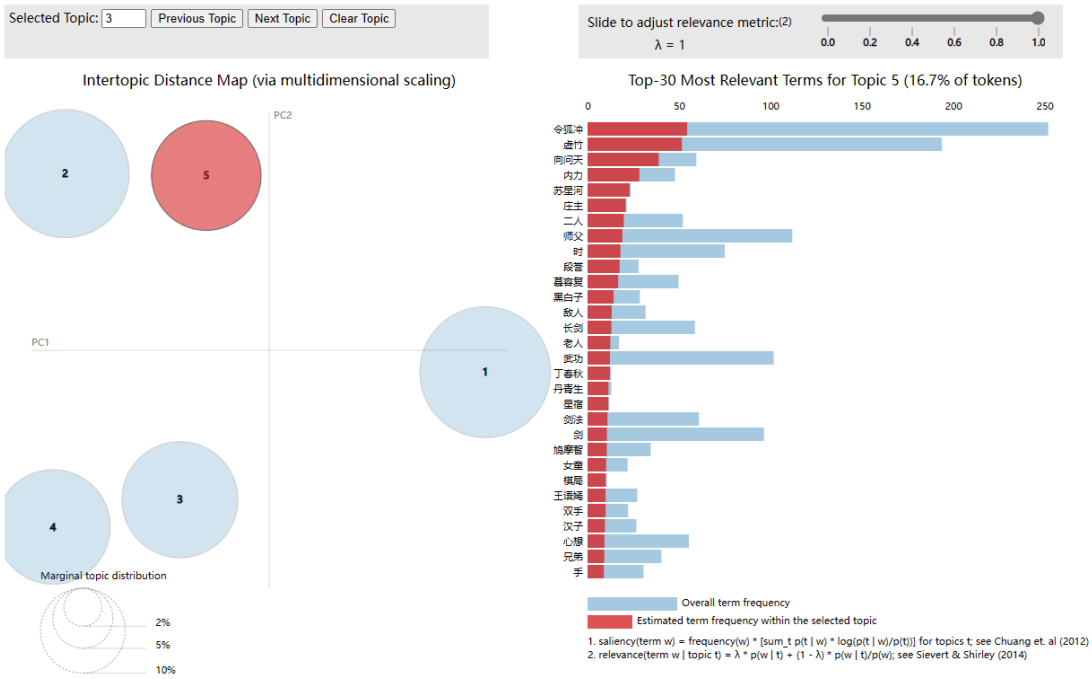
1. 统计段落中的所有不重复单词得到的字典
2. 得到段落的词袋向量（某一个段落的词袋向量如下）
3. 将 1 ，2 的数据放入 LDA 模型中训练得到五个主题的词分布（只列出分布概率 top10 的词）
4. 训练集得到的对五个主题的词分布可视化：


Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (23.7% of tokens)
0
50
100
150
200
250
令狐冲
岳不群
师父
教主
岳灵珊
弟子
师妹
师娘
任我行
林平之
华山派
田伯光
岳夫人
崖
向问天
师哥
华山
心狠
时
鲍大楚
性命
兄弟
字
前辈
剑法
掌
谢逊
林震南
黑白子
师弟
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 2
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
2
5
1
4
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (22.7% of tokens)
0 50 100 150 200 250
虚竹
童姥
剑
令狐冲
一剑
长剑
剑法
李秋水
和尚
武功
少林寺
婆婆
时
字
童姥道
弟子
功夫
贱人
前辈
少林
岳夫人
旭摩智
剑鞘
内力
小僧
气
王语嫣
招式
无法
余沧海
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 3
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
2
5
1
4
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 3 (18.7% of tokens)
0 50 100 150 200 250
张无忌
周芷若
仪琳
赵敏
师父
慕容复
武功
田伯光
黄大
丐帮
教主
令狐冲
女子
爹爹
灭绝师太
姑娘
尼姑
妈妈
江湖
弟子
眼见
小子
公冶乾
问道
二人
掌门
彭莹玉
服毒
余沧海
父母
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)





5. 本实验对该 LDA 采样模型的分类效果进行评估，计算其相应的困惑度（Perplexity）和相关性得分（Coherence Score）：

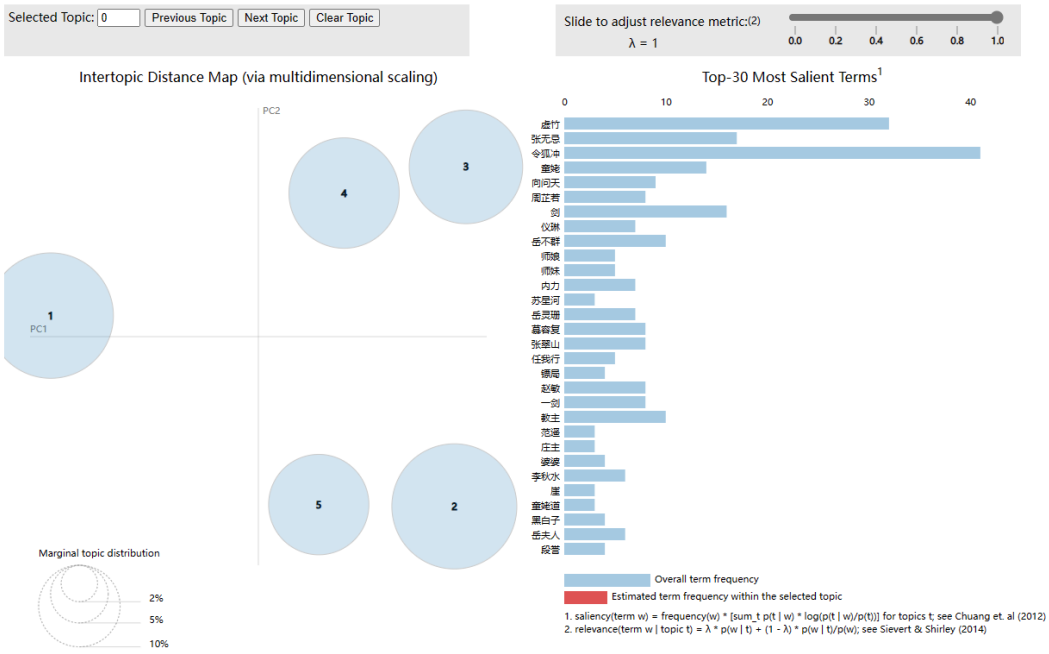
```
倚天屠龙记.txt
天龙八部.txt
射雕英雄传.txt
神雕侠侣.txt
笑傲江湖.txt
len(train_data): 900
len(test_data): 100
[(0, '0.021*"手" + 0.016*"杨" + 0.016*"剑" + 0.015*"出" + 0.014*"声" + 0.014*"身" + 0.012*"中" + 0.011*"时" + 0.011*"法" + 0.010*"郭"'), (1, '0.020*"中" +
0.015*"盈" + 0.013*"时" + 0.010*"头" + 0.009*"气" + 0.008*"口" + 0.007*"渐" + 0.007*"水" + 0.007*"条" + 0.007*"面"'), (2, '0.037*"道" + 0.020*"令" + 0.018*
"子" + 0.017*"狐" + 0.013*"说" + 0.012*"老" + 0.010*"中" + 0.009*"笑" + 0.009*"见" + 0.009*"山"'), (3, '0.036*"岳" + 0.022*"剑" + 0.022*"武" + 0.018*"派" +
 0.017*"功" + 0.014*"三" + 0.012*"练" + 0.010*"中" + 0.010*"十" + 0.010*"法"'), (4, '0.036*"道" + 0.017*"师" + 0.016*"说" + 0.014*"心" + 0.010*"子" + 0.009
*"想" + 0.008*"见" + 0.008*"中" + 0.007*"知" + 0.007*"女"')]
Perplexity:  -6.8203727017183375

Coherence Score:  0.31223745722784646
```

3.3 预测

与上述流程相似利用 SVC 分类得到的准确率为：

0.57

测试集得到的对五个主题分布的词分布可视化：



3.4对比试验

（1）改变主题数量

主题数	分类准确率
5	0.4
20	0.42
50	0.43
100	0.52
1000	0.53

可以看出随着主题数的增加，分类准确率升高，但达到100之后提升不明显。

（2）改变分词方式（5个主题数）

分词方式	分类准确率
字	0.33
词	0.4
保留名词	0.57

（3）改变token数量(主题数为50)

Token数	分类准确率
5	0.51

20	0.45
50	0.49
100	0.55
200	0.61

可见token200时准确率最高，并且除token=5时除外，主题模型分类准确率随token数量上升而上升

4 总结

本文在给定的金庸小说数据库上利用 LDA 模型做无监督学习，学习得主题的分布。在数据库中抽出段落作为训练数据，和测试测数据，利用SVM 分类 器对给定的段落属于哪一本小说进行分类。

自然语言是一种上下文相关的信息表达和传递的方式，让计算机处理自然语言，一个基本的问题就是为自然语言这种上下文相关的特性建立数学模型，即统计语言模型。在LDA主题模型中，一篇文档可以包含好几个主题，每个主题可以生成一系列词。LDA运作方式为从主题分布中，为每篇文档选定一个主题。从上述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词汇。

经过实验可以看出，随着主题数的增加，分类准确率在上升，同时分类效果按词分割>按字分割，说明按词分割更加适合LDA主题提取，样本有更多的信息保留。

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。