学　号 20376310

北京航空航天大學
BEIHANG UNIVERSITY

深度学习与自然语言处理

大作业 2

文本 LDA 建模与主题分类

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	杨佳木
学 号	**20376310**

2024 年 5 月

1.摘要

本作业基于 LDA 主题模型和深度学习分类模型，完成给定语料库段落分类任务。首先，基于 LDA 主题模型的构建方法，得到训练文本的主题特征向量的分布。其次，将训练集所得 LDA 模型的状态特征向量输入通过至深度学习分类模型，标签为各个段落对应的小说，获得训练完成的模型。最后将测试集通过 LDA 模型所得的特征向量喂入训练完成的分类器，得到测试集分类结果，计算分类正确率，并通过 K 折交叉验证消除过拟合造成的影响。

本文基于上述的设计流程，通过设计三个实验改变不同主题数量 T，选择“字”或“词”作为基本单元或选择不同取值长度 K，并验证短文本和长文本对分类效果的影响。本文最终得到在其他情况不变的时候，主题数 T 越多，训练准确性越高，但是相同 epoch 下，更易出现过拟合现象。token 数越多，训练准确性越高，且能够一定程度上减弱过拟合的现象，深度学习方法相较于传统方法分类有优势。token 数超过 1000，如取 token 数为 3000，以词为基本单元的分类效果好，低于 1000，以字为基本单元的分类效果好，token 为 1000，二者效果差不多。深度学习方法相较于传统方法分类有优势。

2.问题描述

本文基于 LDA 模型在给定的语料库中抽取 1000 个段落最为数据集（每个段落可以选取 K 个 token，K 可取 20，100，500，1000，3000），标签为对应段落所属的小说，利用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（900 组数据做训练，剩余 100 组做测试循环十次）。研究问题如下：

（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？

3.问题的解决方案

LDA 在主题模型中占有非常重要的地位，常用来文本分类。该方法由 Blei, David M.、Ng, Andrew Y.Jordan 于 2003 年提出，用来推测文档的主题分布。它可以将文档集中每篇文档的主题以概率分布的形式给出，从而通过分析一些文档抽取出它们的主题分布后，便可以根据主题分布进行主题聚类或文本分类。

为了解决所提出的三个问题，解决方案流程如下：

1.从给定的 16 本金庸小说数据集中，随机、均匀地抽取 1000 个段落，每个段落的标签为对应小说的小说名，每个段落包含n个字（$n \geqslant 500$），每个段落作为一个样本；将随机抽取的k个段落（即k个样本）中的 80%作为训练样本，剩余 20%作为测试样本。对样本进行 Jieba 中文分词，将文本转化为一系列词汇，并根据停用词词表过滤无意义词汇和标点符号，保留用用的词汇。构建词袋模型，进行 LDA 建模，设定段落主题数目 T，得到每个段落的主题特征向量分布。

2.构建基于深度学习的分类模型，对不同段落特征进行分类训练，并使用验证集验证分类效果。使用 K 折交叉验证避免过拟合问题。

针对问题 1，设计实验 1，通过改变主题数 T 以改变分类效果，得到主题数与分类效果间的关系。

针对问题 2，设计实验 2，通过改变基本单元的类型，得到基本单元的类型与分类效果间的关系。

针对问题 3，设计实验 3，通过改变取值长度 K，得到取值长度与分类效果间的关系。

4.基于 LDA 的主题分类模型的建立

基于语料库，使用 LDA 模型进行主题模型建模，具体建模方法如下：

1．对于每一篇文档m，选择一个文档-主题分布

$$\theta_m \sim \text{Dirichlet}(\alpha) \tag{1}$$

对于文档m中的每一个词n, 选择一个主题

$$z_{mn} \sim \text{Multinomial}(\theta_m) \tag{2}$$

并选择一个词，有词的分布先验满足

$$w_{mn} \sim \text{Multinomial}(\beta_{z_{mn}}) \tag{3}$$

其中，α是文档-主题分布的先验参数，β是主题-词分布的先验参数。

2. 参数估计

LDA 的目标是估计出(θ)和(β)，使得整个文档集的概率最大化。

基于这一理论思想，以字为基本单位，假设文章主题数为 10，构建语料库主题模型如下：

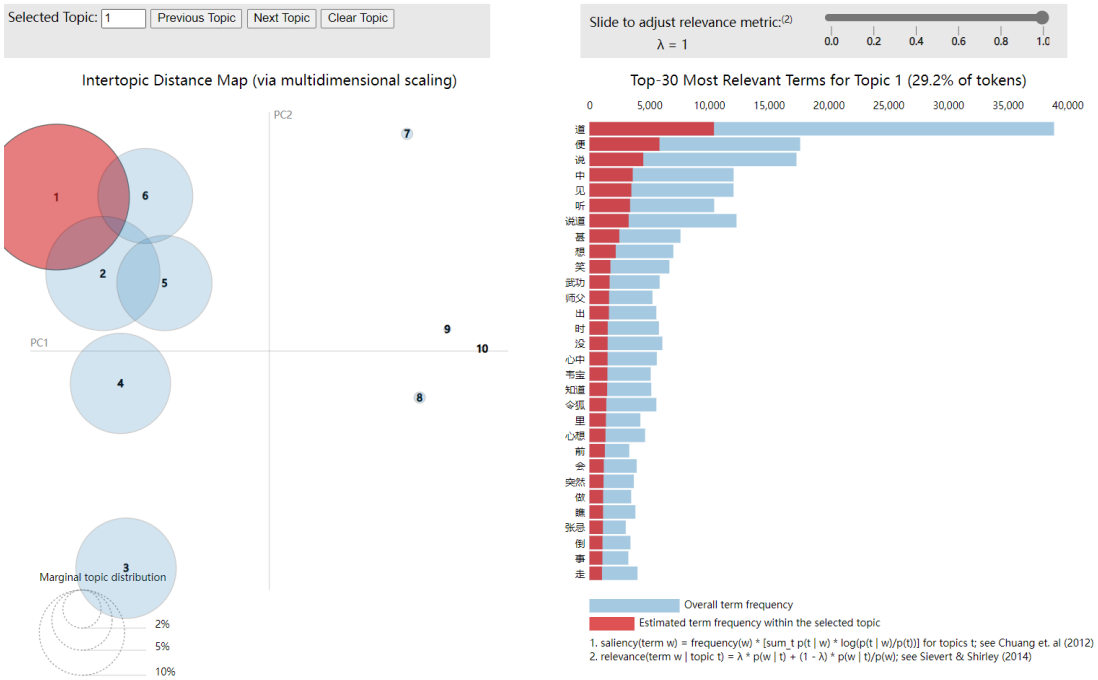


图 1 语料库 LDA 主题模型构建

如图左侧所示为主题数目，右侧所示为每个主题相关词（字）（列举 top-30）。

基于这一主题模型，构建测试集 LDA 主题分布特征矩阵，从每篇小说中抽取的所有段落组合到一起作为训练集，进行训练。首先为每篇文章中的词随机分配一个 topic，然后统计每个段落的 topic 频率。结果如下：

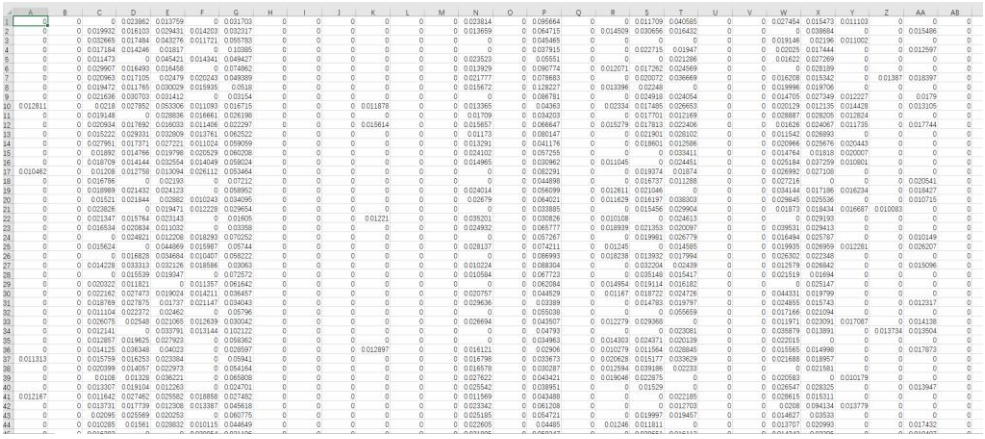


图 2 段落主题频率

将段落名作为标签，构建训练集。同理，选取 10%的数据作为验证集。基于这一样本设计分类算法。

采用基于自注意力机制优化的多层感知机模型进行训练，该模型属于深度学习网络，网络结构如下：

```
class SelfAttrntion(nn.Module):
    def __init__(self, embed_dim):
        super(SelfAttrntion, self).__init__()
        self.query = nn.Linear(embed_dim, embed_dim)
        self.key = nn.Linear(embed_dim, embed_dim)
        self.value = nn.Linear(embed_dim, embed_dim)

    def forward(self, x):
```

```
        q = self.query(x)
        k = self.key(x)
        v = self.value(x)
        attn_weight = torch.matmul(q, k.transpose(1, 2))
        attn_weight = nn.functional.softmax(attn_weight, dim=-1)
        attned_values = torch.matmul(attn_weight, v)
        return attned_values
class Net(nn.Module):
    def __init__(self, in_dim, n_hidden_1, n_hidden_2, out_dim):
        super().__init__()
        self.attention = SelfAttrntion(in_dim)
        self.layer1 = nn.Sequential(
            nn.Linear(in_dim, n_hidden_1), nn.ReLU(True), nn.Dropout(0.3))
        self.layer2 = nn.Sequential(
            nn.Linear(n_hidden_1, n_hidden_2), nn.ReLU(True), nn.Dropout(0.3))
        self.layer4 = nn.Sequential(nn.Linear(n_hidden_2, out_dim))

    def forward(self, x):
        x = self.attention(x)
        x = self.layer1(x)
        x = self.layer2(x)
        x = self.layer4(x)
        return x
```

基于这一网络结构，使用交叉熵损失函数进行训练，得到训练结果。并和基于 SVM 的训练结果进行对比。

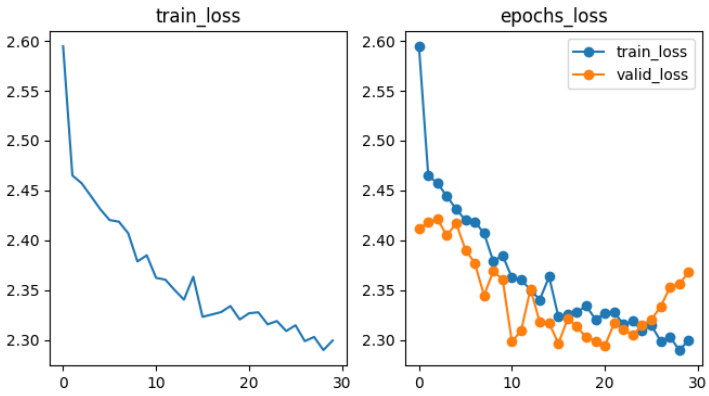
```
100%|██████████| 90/90 [00:00<00:00, 402.55it/s]
100%|██████████| 10/10 [00:00<00:00, 1666.46it/s]
train acc = 39.222%, loss = 1.831393297513326
epoch = 29, valid acc = 29.00%, loss = 2.188531219959259
```

实验 1 不同主题数与分类结果间的关系

设置主题分类的数目分别为 T=10、20、50、100、200，固定 token 数为 1000，字为单位，分类结果如下

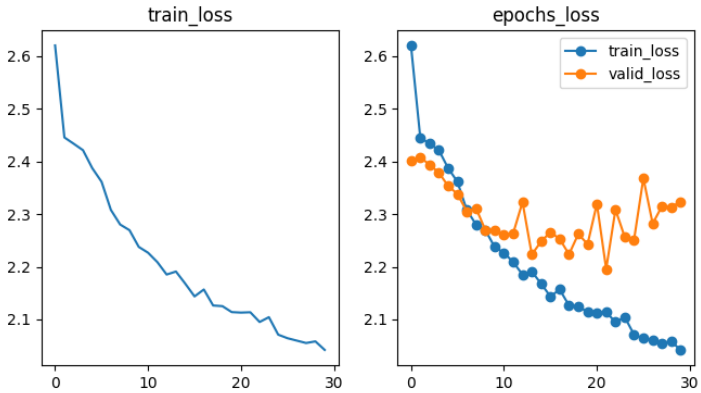
1. 分类主题数 T=10

训练集准确率 21.333%，测试集准确率 22.000%，损失函数如下：



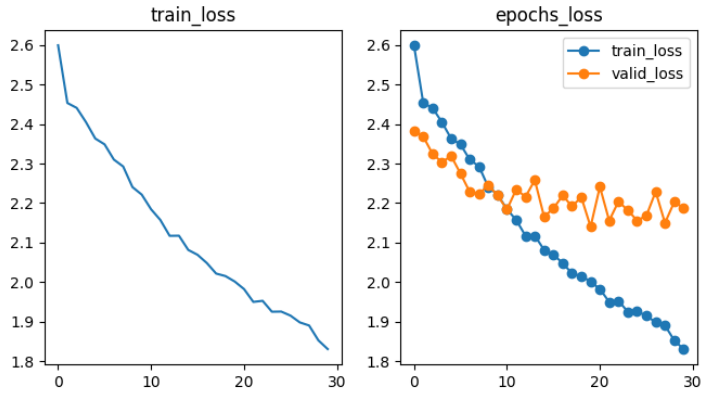
2. 分类主题数 T=20

训练集准确率 30.566%，测试集准确率 23.000%，损失函数如下：



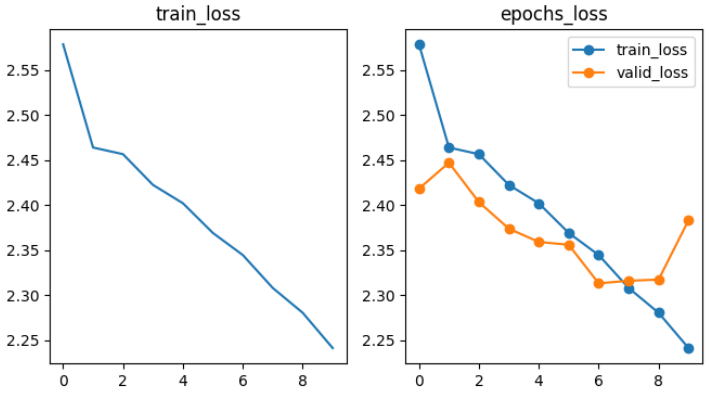
3. 分类主题数 T=50

训练集准确率 39.226%，测试集准确率 29.000%，损失函数如下：



4. 分类主题数 T=100

训练集准确率 45.245%，测试集准确率 39.000%，损失函数如下：

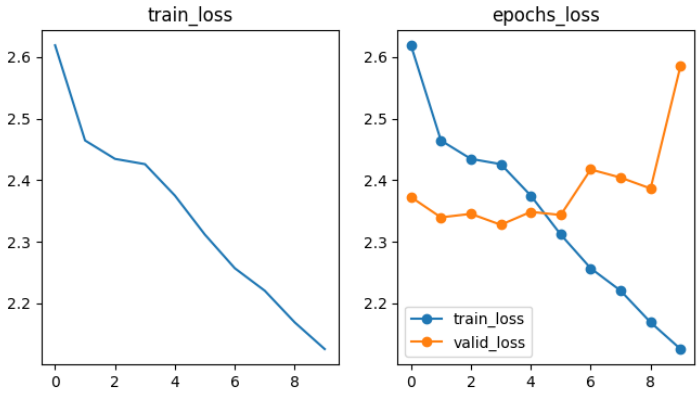


（epochs 过高，会出现过拟合）

5. 分类主题数 T=200

训练集准确率 65.937%，测试集准确率 32.000%，损失函数如下：

（epoch 为 10 的情况下，出现过拟合）



准确率 主题数	Self Attention-MLP 训练集准确率/%	Self Attention-MLP 验证集准确率/%	SVM 验证集准确率/%	SVM 验证集准确率/%
10	21.333	22.000	12.564	11.423
20	30.566	23.000	18.342	15.342
50	39.226	29.000	23.231	19.258
100	45.245	39.000	27.543	25.379
200	65.397	32.000	35.437	32.456

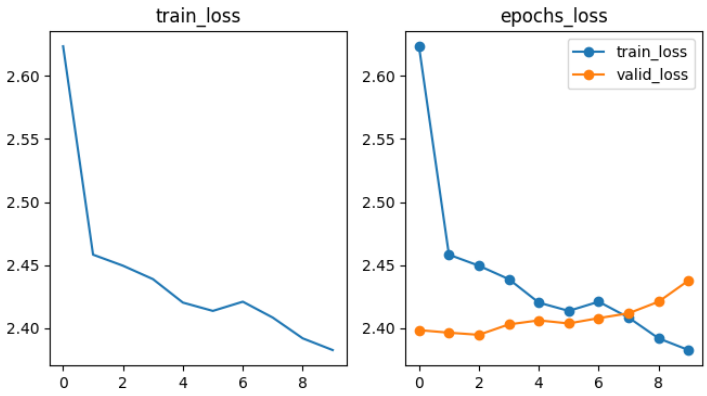
实验结论：主题数 T 越多，训练准确性越高，但是相同 epoch 下，更易出现过拟合现象，深度学习方法相较于传统方法分类有优势。

实验 2 不同 token 与分类结果间的关系

选取 token 数为 20，100，500，1000，3000，以字为单位，K=100，进行分类，结果如下：

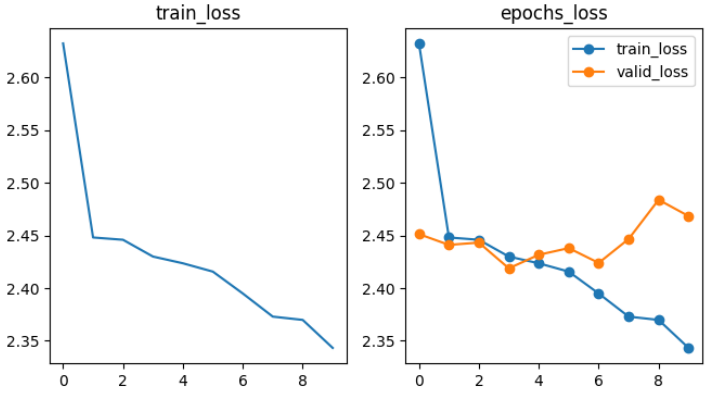
1. token 数 T=20

训练集准确率 17.333%，测试集准确率 13.000%，损失函数如下：



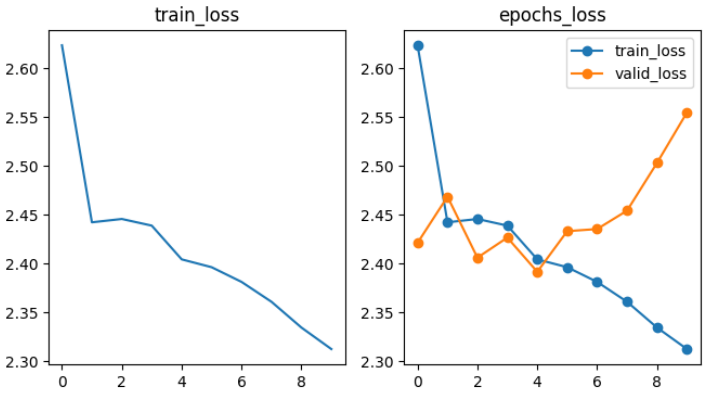
2. token 数 T=50

训练集准确率 19.667%，测试集准确率 17.000%，损失函数如下：



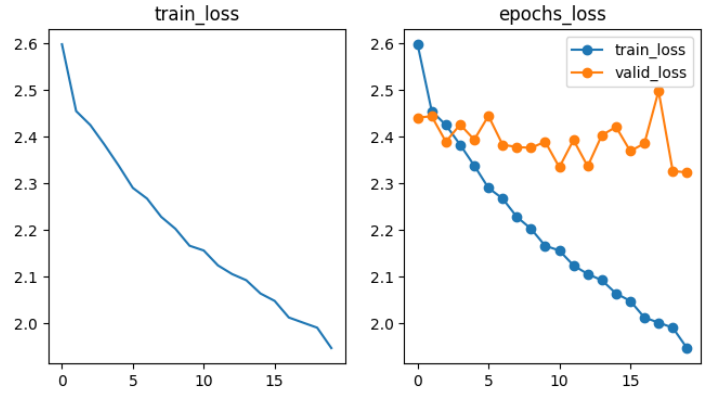
3. token 数 T=100

训练集准确率 25.556%，测试集准确率 19.000%，损失函数如下：



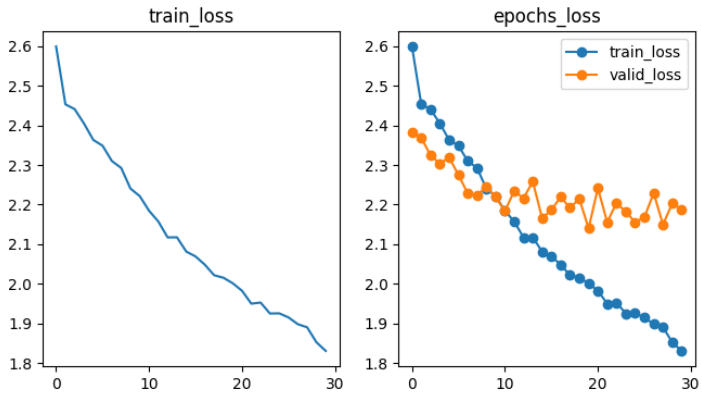
4. token 数 T=500

训练集准确率 34.123%，测试集准确率 31.000%，损失函数如下：



6. token 数 1000

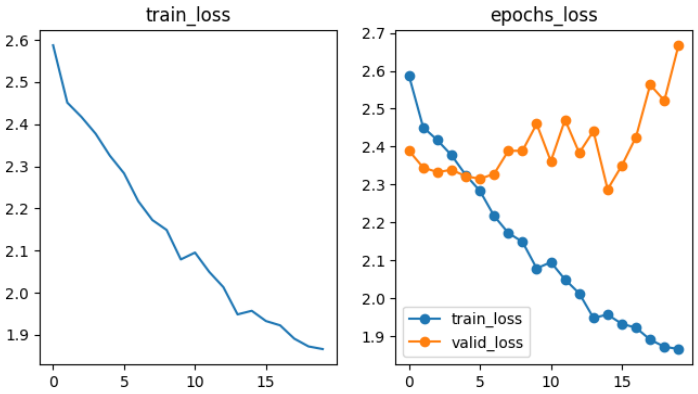
训练集准确率 41.582%，测试集准确率 36.000%，损失函数如下：



7. token 数 3000

训练集准确率 47.942%，测试集准确率 46.000%，损失函数如下：

8.



准确率 / token 数	Self Attention-MLP 训练集准确率/%	Self Attention-MLP 验证集准确率/%	SVM 验证集准确率/%	SVM 验证集准确率/%
20	17.333	13.000	13.663	12.426
50	19.667	17.000	15.462	15.331
100	25.556	19.000	21.233	19.258
500	34.123	31.000	27.543	25.379
1000	41.582	36.000	36.433	32.856
3000	47.942	46.000	40.231	37.538

实验总结：token 数越多，训练准确性越高，且能够一定程度上减弱过拟合的现象，深度学习方法相较于传统方法分类有优势。

实验 3 字与词与分类结果间的关系

选取 token 数为 1000，以字或词为单位，K=100，进行分类，结果如下：

准确率 字/词	Self Attention-MLP 训练集准确率/%	Self Attention-MLP 验证集准确率/%	SVM 验证集准确率/%	SVM 验证集准确率/%
字	41.582	36.000	36.433	32.856
词	41.268	37.000	35.462	31.331

选取 token 数为 3000，以字或词为单位，K=100，进行分类，结果如下：

准确率 字/词	Self Attention-MLP 训练集准确率/%	Self Attention-MLP 验证集准确率/%	SVM 验证集准确率/%	SVM 验证集准确率/%
字	47.942	46.000	40.231	37.538
词	48.387	42.000	37.462	32.331

选取 token 数为 100，以字或词为单位，K=100，进行分类，结果如下：

准确率 字/词	Self Attention-MLP 训练集准确率/%	Self Attention-MLP 验证集准确率/%	SVM 验证集准确率/%	SVM 验证集准确率/%
字	25.556	19.000	21.233	19.258
词	22.345	17.000	17.462	15.331

实验结论：token 数超过 1000，如取 token 数为 3000，以词为基本单元的分类效果好，低于 1000，以字为基本单元的分类效果好，token 为 1000，二者效果差不多。

总结

1. 主题数 T 越多，训练准确性越高，但是相同 epoch 下，更易出现过拟合现象，深度学习方法相较于传统方法分类有优势。

2. token 数越多，训练准确性越高，且能够一定程度上减弱过拟合的现象，深度学习方法相较于传统方法分类有优势。

3. token 数超过 1000，如取 token 数为 3000，以词为基本单元的分类效果好，低于 1000，以字为基本单元的分类效果好，token 为 1000，二者效果差不多。