自然语言处理
文本分类系统文档

December 20, 2019 at 20:51pm

教师：刘洋

小组：$Average - 3.5m^2$
组员：董超鹏　胡枭　宾浩宇
王荟昭　黄振洋

1. 背景

随着互联网的发展，网络中的数据和信息呈现指数级爆炸式增长。由于网络上电子文档越来越多，如何对这些信息和数据进行自动有效地组织和管理是一个巨大的挑战，基于内容的信息检索和文本数据挖掘渐渐成为了研究人员关注的领域。其中，文本分类技术近年来得到了广泛的关注和研究。文本分类是自然语言处理任务中的一项基础性工作，主要任务是在预先给定的类别标记 (label) 集合下，根据文本内容判定它的类别，目的是对文本资源进行整理和归类，同时文本分类也是解决文本信息过载问题的关键环节。随着技术的发展，文本分类已经在各个领域中落地和应用，比如：邮件分类、网页分类、文本索引、自动文摘、信息检索、信息推送、数字图书馆以及学习系统等。

本文实现了一个文本分类系统，通过深度学习，朴素贝叶斯、支持向量机等模型实现文本分类，实验表明基于深度学习的方法得到的效果最好。最后，通过 Web 的形式进行展示。

本文的组织结构如下：

1. 第 1 章介绍了本项目的背景；
2. 第 2 章介绍了小组成员信息以及人员分工；
3. 第 3 章介绍了文本分类的任务定义；
4. 第 4 章介绍了文本分类系统的系统架构，系统中相关模型、算法的设计与实现；
5. 第 5 章介绍了模型的实验，包括数据集、基准系统的介绍，并对实验的结果进行比较和分析；
6. 第 6 章介绍了文本分类系统的环境要求、系统部署以及使用的方法；
7. 第 7 章进行概括性的总结；

2. 小组人员分工

本次项目的小组人员分工如表1所示。项目的任务分配和管理主要由组长董超鹏同学负责，各位组员都负责了文本分类模型的搭建以及评估，包括 CNN 模型，LSTM 模型，朴素贝叶斯等，具体的内容见表1，王荟昭同学负责了本系统中的 Web 的搭建以及前端展示的设计，并且，文档各个部分由各位同学撰写，最终由黄振洋同学进行整合。

成员	分工
董超鹏	1. 任务分配，项目管理；2. 朴素贝叶斯等文本分类模型的实现；3. 文档编写
胡枭	1. 实现 CNN 文本分类模型的构建和评估； 2. 基准系统的调研 3. 文档编写
宾浩宇	1.LSTM 文本分类模型的构建与评估；2. 文档编写
王荟昭	1.Web 的搭建；2. 系统文档的编写
黄振洋	1. 数据预处理；2. 文档编写与整合；

表 1: 小组人员分工一览表

3. 文本分类定义

文本分类是自然语言处理 (Natural Language Processing, NLP) 的最经典的领域之一，其主要任务是基于文本内容，在给定的分类体系中，将文本自动分到预先设定的一个或多个类别中，这些类别可以属于不同领域，也可以具有层次结构，常见的类别包括政治、财经、体育、娱乐和艺术等。文本分类的对象包括短文本（如句子、标题和评论等）和长文本（如文章等），文本分类的方向包括二分类、多分

类和多标签分类，其应用场景包含情感分析、领域识别和个性化推荐等，文本分类的方法主要包括基于规则和专家系统的方法、传统机器学习方法和深度学习方法。

4. 模型算法

4.1 CNN

4.1.1 介绍

卷积神经网络 (Convolutional Neural Networks, CNN) 最初在计算机视觉和语音识别等领域取得了显著的效果。相较全连接网络，CNN 具有三大特点：局部感受野、权值共享和池化，其中局部感受野是最明显的区别，其依靠卷积运算，学习数据上近邻的特征，比如，图片中一个像素的跟周围像素联系更为紧密，其局部特征可以用卷积来学习，这非常符合直觉，类似的，在自然语言处理中，也可以用卷积神经网络解决众多问题，文本分类就是其中之一。

在文本分类问题中，使用卷积网络来学习文本序列的上下文信息，从直觉上可以理解为学习文本的 n-gram 信息，进而提取高维特征对文本类别进行判断。在本系统中，我们基于深度 Pytorch、卷积神经网络和中文文本分类数据集 THUCNews 实现了文本分类模型，准确率达到 91.15%，查准率、查全率和 F1 值等指标也达到这一水平，表现出相当不错的分类性能。

4.1.2 实现

本系统采用深度学习中的卷积神经网络作为文本分类的方法之一，旨在将原始文本归类到预先设定的分类体系中。本方法包括 1）数据集预处理，包括数据集格式化、分词、去停用词、文本截取和补齐、构建词汇表等；2）模型训练，包括神经网络构建，调参、优化和评估等；3）测试阶段，包括计算混淆矩阵、统计各类别分类情况等。CNN 文本分类模型架构如图1所示：

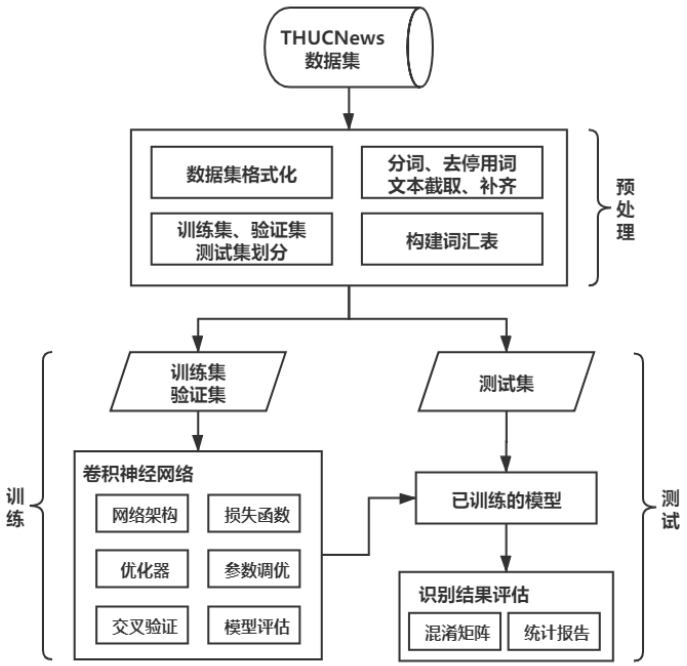

图 1: CNN 文本分类模型架构

本系统基于深度学习框架 Pytorch，使用 torchtext 加载数据集，对文本进行处理，具体的，将文本类别和文本内容放在一起，最后文本数据格式是：tsv 文件，包含序号，类别和内容三列，文本内容需要经过分词（使用 jieba 分词）、去停用词（维护一个停用词表）以及截取和补齐，使得输入 CNN 的文本长度为 100（直觉是，对于新闻文本，我们人阅读 100 个词足以判断其类别），然后使用训练集和验证集构建词汇表，我们设置的词汇表大小为 50000，再随机生成训练的小批量数据，batch size 为 128，到此数据预处理完成。

本系统借鉴 Kim 等人 [1] 提出的 CNN 句子分类网络结构，搭建的 CNN 网络结构如图2所示：

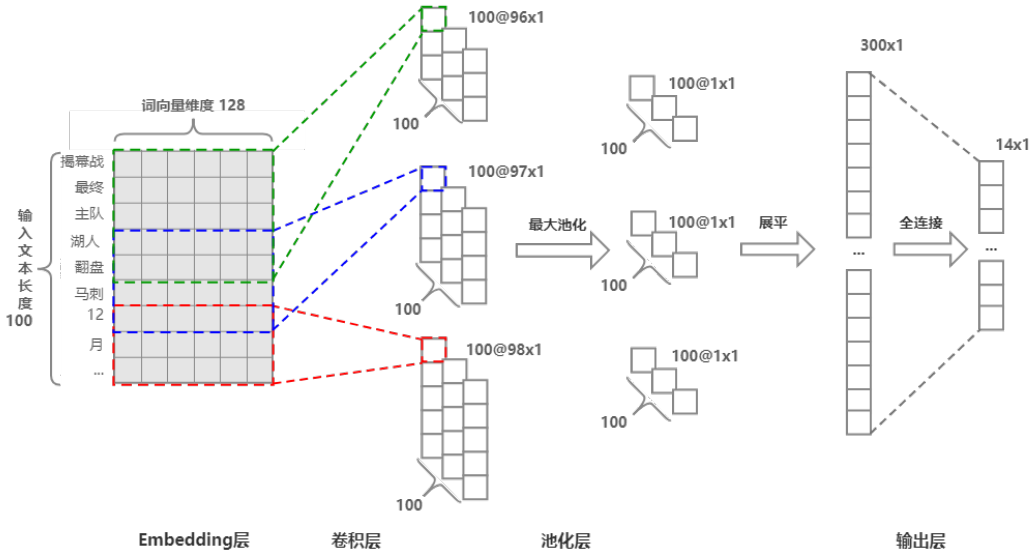

图 2: CNN 网络结构

网络涉及的各层输入输出的数据维度已在图中表出，具体的网络结构描述如下：

- 输入层：神经网络的输入是长度为 100 的文本词序列，经过一个 50000*128 维的 embedding 层（未使用预训练词向量，该层的权重会在训练过程中调整）
- 卷积层：我们使用 2 维卷积，卷积窗口长度分别为 3，4，5 的三种卷积核，宽度与词向量维度保持一致，为 128，每种卷积核数目为 100，步长为（1,1），未使用补 0 填充
- 池化层：在文本词序列方向进行最大池化，池化窗口长度为序列的长度
- 展平：将每类卷积核得到的特征展平，再将展平后的特征连接到一起
- Dropout 层：参数为 0.5
- 输出层：全连接层，输出 14 类属于每类的可能性大小，取最大的作为类别
- 损失函数：使用交叉熵
- 优化器：使用 Adam，学习率为 0.001

系统涉及到的超参如表2所示：

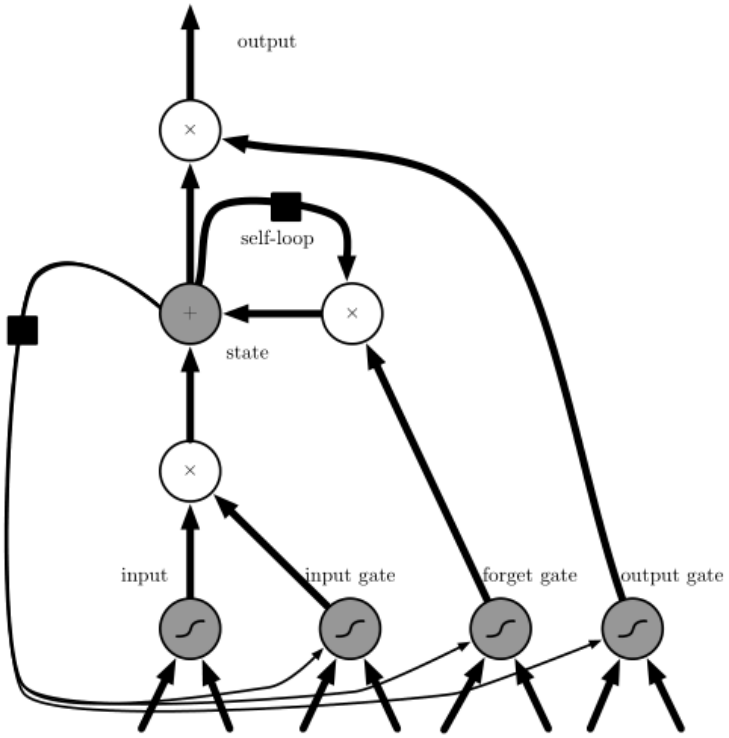

图 3: LSTM 网络结构

超参	值	超参	值
文本输入长度	100	卷积窗口长度	3，4，5
词汇表长度	50000	卷积核个数	100*3
词向量维度	128	Dropout	0.5
Batch Size	128	学习率	0.001
Epoch	20		

表 2: CNN 超参数列表

4.2 LSTM

4.2.1 介绍

LSTM(Long Short Term Memory) 长短期记忆网络，结构如图 3 所示，包含了多个门。不同的细胞之间循环连接，相当于循环网络中的隐藏单元。他的门包括了输入门、遗忘门、输出门。循环神经网络可能存在记忆过长的情况，一般认为当前的信息更可能和最近出现的内容相关，和距离较远的内容的关系更小。但是循环神经网络 RNN 的处理不够恰当，因此有了长短期记忆网络，他会选择性的遗忘一定距离之外的东西，使得学习到的内容和当前的上下文有更强的相关性。[?]

4.2.2 实现

本次实验构建使用 keras 构建了 LSTM 网络，结构如图。4 ，包含了一个输入层，一个嵌入层，一个 LSTM 层，两个全连接层和一个 Dropout 层。嵌入层对输入的词向量进行词嵌入操作，将词语映射到向量空间中。全连接层 (fully connected layers)，将学到的特征表示映射到了样本的标记空间之中。

Layer (type)	Output Shape	Param #
inputs (InputLayer)	(None, 300)	0
embedding_1 (Embedding)	(None, 300, 128)	384128
lstm_1 (LSTM)	(None, 128)	131584
Dense1 (Dense)	(None, 128)	16512
dropout_1 (Dropout)	(None, 128)	0
Dense2 (Dense)	(None, 14)	1806

Total params: 534,030
Trainable params: 534,030
Non-trainable params: 0

图 4: 本实验采用的 LSTM 网络架构参数

LSTM 层将嵌入层输出的数据，通过输入门遗忘门输出门对输入的词向量进行学习。Dropout 层主要为了防止过拟合。在训练过程中，很容易出现模型在训练集合上损失函数很低，准确率比较高，但是在验证集上准确率急剧下降的情况，这就是出现了过拟合。过拟合的模型错误地将训练数据的无关信息也加入到了预测中，使得模型的泛化性能变差。Dropout 的加入会按照一定概率去抑制某些神经元，防止模型过度依赖某些特征。

4.3 朴素贝叶斯

4.3.1 介绍

朴素贝叶斯 (Naive Bayes, NB) 是贝叶斯理论的一种特殊情况，它假设所有特征项之间都是独立的，只与文档本身有关。这种强假设能够迅速降低计算的成本，但同时也忽略了特征之间的关联信息，在独立分布的数据集中会有很好的效果，但不太适合对于上下文关联性较强，语义信息对分类影响较大的数据。在实际运用中，NB 虽然会丢失语义信息，但大部分情况根据各个特征项本身已经能够确定数据的分类结果。因此，NB 仍然凭借着不错的分类效果、快速的训练过程和对大数据量的低依赖性在分类问题中经久不衰。

训练朴素贝叶斯分类器的过程主要是根据训练集计算各个分类的先验概率和所有特征项在已知分类下的似然概率（又称条件概率）。基础贝叶斯公式如下所示：

$$P(C_i|d) = \frac{P(d|C_i) \times P(C_i)}{P(d)} \tag{1}$$

其中，C_i 表示第 i 个分类，d 表示输入的文档内容，相当于给定文档后预测其属于某个分类的概率。考虑 $P(d|C_i)$，文档 d 可以视为由各个词构成，且依据朴素贝叶斯原理，各个词之间相互独立，因此得到计算公式如下：

$$P(d|C_i) = P(w_1, w_2, ..., w_n|C_i) = \prod_{j=1}^{n} P(w_j|C_i) = \prod_{k=1}^{m} P(w_k|C_i)^{N(w_k)} \tag{2}$$

其中，$w_i, i = 1, 2...n$ 是组成文档 d 的所有词，$N(w_k)$ 表示该词出现的次数，但是此计算方法考虑了文

档中所有的词，很多词其实对文本分类结果没有影响。实际上，在预处理，特征提取过后，我们只需找出文档中属于语料库的特征词，计算它们在分类下的似然概率即可，因此将公式化简如下：

$$P(d|C_i) = P(w_1, w_2, ..., w_{n'}|C_i) = \prod_{j=1}^{n'} P(w_j|C_i) \tag{3}$$

其中，n' 表示文档 d 含语料库中特征词的数量，$w_k, k = 1, 2, .., n'$ 都是文档 d 在语料库中的特征词。而由于 $P(d)$ 在各个分类中是固定不变的，因此只需考虑使得 $P(d|C_i) \times P(C_i)$ 的计算结果最大的分类 C_{max}，计算公式如下：

$$c^* = \arg\max_{i=1}^{C} P(C_i) \prod_{j=1}^{n'} P(w_j|C_i) \tag{4}$$

4.3.2 实现

本系统中采用朴素贝叶斯作为文本分类的方法之一，最终实现对于给定的数据文本，将其分类到对应的类别中去。数据的选取上与前面的分类方法相同，由于贝叶斯分类器不需要大量的训练数据，因此从原数据集中筛选出 7000 条训练数据（每个分类 500 条），测试数据则与前面方法相同，方便对比。用贝叶斯分类器进行文本分类的过程可以分为三个过程：

(1) **数据预处理**
包含格式化数据，分词，去停用词，特征提取，构建特征语料库。其中分词、去停用词）均与前面的方法统一。在特征提取时，对训练集中每个词，计算其与各个类别的互信息 (MI) 大小，再取每个类别互信息最大的前 50 个特征词组成语料库，部分语料库如图 3 所示：

```
['比赛', '前', '球队', '国家队', '之后', '球员', '体育讯', '新', '足球', '足协', '联赛', '参加', '最后', '队', '对手', '中国足
协', '俱乐部', '曾', '之前', '主教练', '来说', '队员', '肯定', '主任', '教练', '男篮', '一次', '一场', '原因', '透露', '无法',
'决定', '主帅', '谢亚龙', '训练', '世界杯', '球迷', '亚运会', '女足', '期间', '这位', '热身赛', '赛季', '主席', '亚洲杯', '南
勇', '曾经', '刚刚', '回到', '备战', '讯', '新浪', '12', '观众', '音乐', '演唱会', '这次', '2010', '歌曲', '电视剧', '歌手',
'电影', '拍摄', '导演', '唱', '表演', '举行', '故事', '节目', '经典', '此次', '演绎', '文', '卫视', '演唱', '专辑', '角色', '采
访', '邀请', '歌迷', '该剧', '一部', '首次', '组合', '举办', '笑', '艺人', '主持人', '人物', '亮相', '听', '第一次', '播出',
'剧中', '上演', '舞台', '饰演', '戏', '歌', '剧', '地板', '家居', '木地板', '装饰', '环保', '实', '材料', '实木', '装修', '厨
房', '家装', '木材', '绿色', '开启', '电器', '营销', '温馨', '宅', '森林', '红', '线条', '德意', '安装', '酒吧', '面', '混搭小
窝', '成就', '盛大', '调节', '式', '餐厅', '奢华', '简约', '铺装', '复合地板', '一块', '诞生', '清洁', '色彩', '背景墙', '客厅',
'纹理', '质感', '厨电', '装', '复古', '卧室', '厨卫', '木', '之旅', '10', '期', '14', '本期', '奖金', '双色球', '注', '彩票',
```

图 5: 部分特征语料库

(2) **模型训练**
模型的训练过程本质上是通过遍历训练集中所有文档，计算各个分类在训练集中先验概率 $P(C_i)$ 和语料库中各个特征词在分类下的似然概率 $P(w_j|C_i)$。最后将训练好的贝叶斯分类器保存到本地，等待测试或预测的时候导入使用。

(3) **模型测试**
模型测试主要测试模型的分类效果，包括分类精度，召回率，$F_1 - scoure$ 等评估指标，以及混淆矩阵用于比较错误分类。

5. 实验分析

本部分首先介绍我们实验所采用的数据集以及基准系统，然后对各个模型算法在相同数据集上的测试结果进行分析。

5.1 数据集介绍

本系统中采用的数据集是 THUCNews，是由清华大学自然语言处理实验室在原始新浪新闻 RSS 订阅频道 2005 2011 年间的历史数据筛选过滤得到的 74 万篇新闻文档（2.19 GB）基础上，重新整合划分出了 14 个分类类别：财经、彩票、房产、股票、家居、教育、科技、社会、时尚、时政、体育、星座、游戏、娱乐。在实际使用中，由于设备条件限制，我们选取了该数据集的部分子集作为训练，训练集的大小根据不同模型算法的要求调整，测试集则统一每个分类分类 150 条，共 7000 条测试数据。

5.2 基准系统介绍

由于我们使用中文文本分类数据集 THUCNews，所以我们使用的基准系统是清华大学自然语言处理实验室推出的中文文本分类工具包THUCTC（THU Chinese Text Classification），它能够自动高效地实现用户自定义的文本分类语料的训练、评测、分类功能。在 THUCTC 中选取二字串 bigram 作为特征单元，特征降维方法为 Chi-square，权重计算方法为 tfidf，分类模型使用的是 LibSVM 或 LibLinear。THUCTC 对于开放领域的长文本具有良好的普适性，不依赖于任何中文分词工具的性能，具有准确率高、测试速度快的优点。

由于时间关系，我们没有开发一个功能与 THUCTC 具有相同或类似功能的系统，主要是在文本分类性能上和该系统进行对比，以更好的评估我们的分类系统。THUCTC 的分类性能如表6所示：

类别	正确率	召回率	F-measure	类别	正确率	召回率	F-measure
体育	0.979	0.986	0.983	星座	0.974	0.618	0.756
娱乐	0.936	0.966	0.951	游戏	0.922	0.536	0.678
家居	0.871	0.883	0.877	社会	0.796	0.802	0.799
彩票	0.967	0.862	0.911	科技	0.845	0.882	0.863
房产	0.957	0.953	0.955	股票	0.858	0.854	0.856
教育	0.887	0.850	0.868	财经	0.779	0.656	0.713
时尚	0.868	0.881	0.875	宏平均	0.886	0.829	0.856
时政	0.764	0.868	0.813	微平均	0.875		

表 3: THUCTC 分类性能

5.3 实验结果

5.3.1 CNN

训练 CNN 时，我们使用验证集进行模型评估，记录了模型准确率和损失值的变化情况，如图6所示：

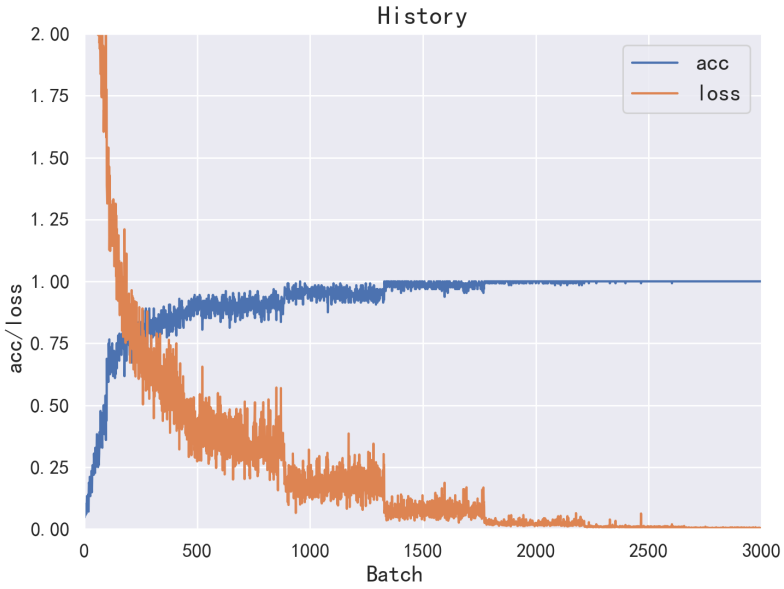

图 6: 训练过程准确率的变化情况

可以看到，模型在经过 2500 个 batch 后，模型收敛，准确率达到 91.15%。

在经过训练和适当的参数调优后，我们使用测试集对训练好的模型进行评估，评估的指标主要是准确率，查准率，查全率和 F1 值，测试集的混淆矩阵如图7所示。

由图7可知，每一类的分类精度都达到了很高的水平，但仍有部分错误分类的样本，通过观察，容易错分的类别主要是：“财经”和“股票”、“社会”和“时政”、“财经”和“房产”，值得一提的是，这些类别之间本身就具有一定的包含性和相关性，对于本系统实现的多分类问题（而非多标签分类问题）具有潜在的歧义。

除此之外，我们还对每个类的分类情况进行统计，结果如表4所示。

由表4可知，CNN 文本分类模型的宏平均查准率、查全率和 F1 值均高于 91%，较 THUCTC 的 F1 值提高了 5.6%，这说明本模型具有很好的分类准确率和鲁棒性。从每个分类的角度来看，绝大多数类的 F1 值高于 THUCTC，在我们的模型中，分类性能最低的类别“社会”，查准率为 80.1%，查全率为 80.7%，F1 值为 80.4%，正如之前的分析一样，这可能是由于其与“时政”的类别本身含义所决定的，其他类似的情况也发生在“财经”和“股票”上，它们的分类性能在平均水平下。

类别	准确率	召回率	F-measure	数量	类别	准确率	召回率	F-measure	数量
体育	0.928	0.947	0.937	150	时政	0.828	0.933	0.878	150
娱乐	0.911	0.953	0.932	150	星座	0.993	0.920	0.955	150
家居	0.920	0.920	0.920	150	游戏	0.938	0.900	0.918	150
彩票	0.973	0.973	0.973	150	社会	0.801	0.807	0.804	150
房产	0.889	0.960	0.923	150	科技	0.911	0.887	0.899	150
教育	0.910	0.940	0.925	150	股票	0.871	0.987	0.925	150
时尚	0.959	0.947	0.953	150	财经	0.991	0.700	0.820	150
宏平均	0.916	0.912	0.912	2100					

表 4: CNN 文本分类模型性能

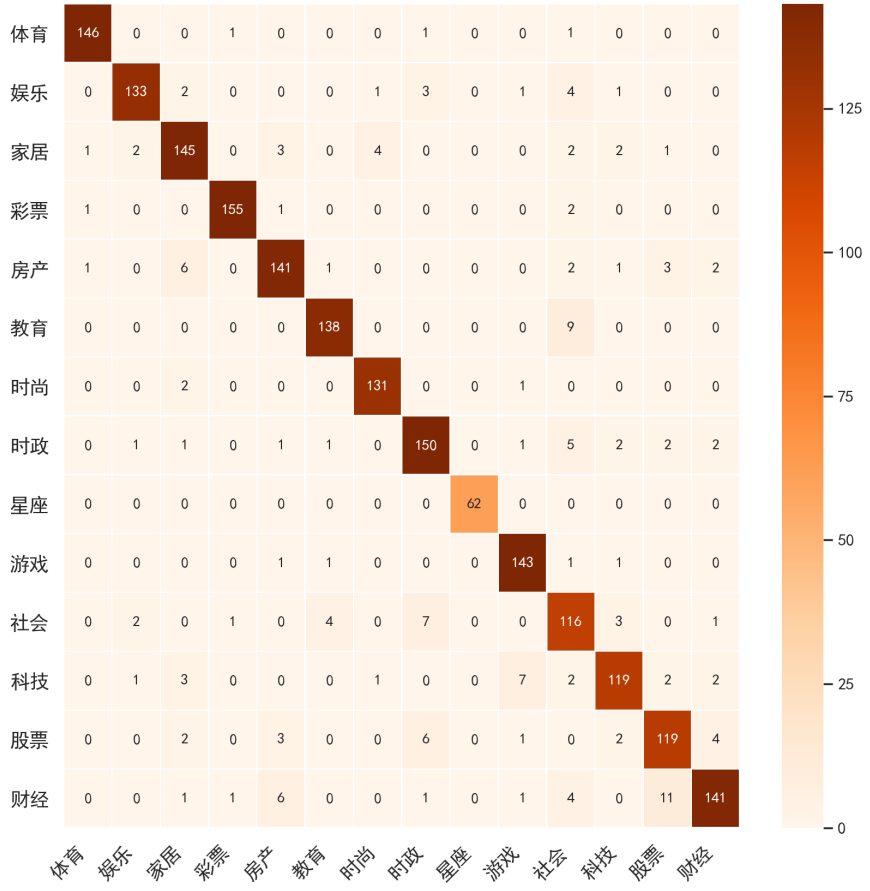

图 7: 测试集混淆矩阵

对于本系统所设计的 CNN 文本分类模型，我们认为还可以从以下几个方面改进：1）使用预训练词向量、使用预训练向量微调、使用多通道词向量；2）更大的数据集，由于硬件条件限制，我们从原数据集中随机选择了部分文本作为数据集使用；3）更细致的神经网络结构设计和参数调优，我们可以设计多个网络结构进行对比实验，可以加入正则尝试解决过拟合问题，各个超参也可以设计超参集合进行组合对比实验。

5.3.2 LSTM

在经过训练和适当的参数调优后，我们使用测试集对训练好的模型进行评估，评估的指标主要是准确率，查准率，查全率和 F1 值，测试集的混淆矩阵如图7所示：

从混淆矩阵可以看出，模型总体较好，但是存在误报。分析之后克制这些分类之间存在一定的相关性，并且有交集，部分的测试内容可能可以归类为多个类别。

除此之外，我们还对每个类的分类情况进行统计，结果如图5所示

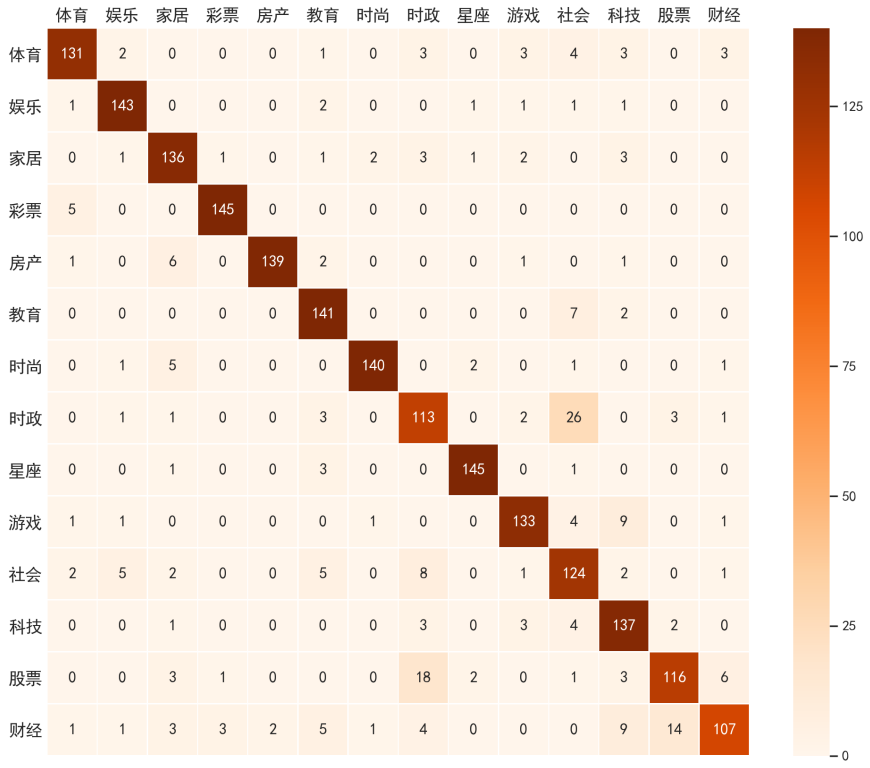

图 8: 测试集混淆矩阵

	类别	准确率	召回率	F-measure	数量
0	体育	0.922535	0.873333	0.897260	150
1	娱乐	0.922581	0.953333	0.937705	150
2	家居	0.860759	0.906667	0.883117	150
3	彩票	0.966667	0.966667	0.966667	150
4	房产	0.985816	0.926667	0.955326	150
5	教育	0.865031	0.940000	0.900958	150
6	时尚	0.972222	0.933333	0.952381	150
7	时政	0.743421	0.753333	0.748344	150
8	星座	0.960265	0.966667	0.963455	150
9	游戏	0.910959	0.886667	0.898649	150
10	社会	0.716763	0.826667	0.767802	150
11	科技	0.805882	0.913333	0.856250	150
12	股票	0.859259	0.773333	0.814035	150
13	财经	0.891667	0.713333	0.792593	150
14	总体	0.884559	0.880952	0.881039	2100

表 5: lstm 分类性能

由上图可知，LSTM 文本分类模型的总体准确率为 88.5%，召回率为 88.1% F1 值为 88.1%。

5.3.3 朴素贝叶斯

采用朴素贝叶斯在测试集上测试的结果如下表所示：

类别	正确率	召回率	F1-score	样本数
体育	0.893	0.946	0.919	150
娱乐	0.905	0.960	0.932	150
家居	0.951	0.840	0.913	150
彩票	1.000	0.840	0.913	150
房产	0.941	0.853	0.895	150
教育	0.984	0.853	0.914	150
时尚	0.970	0.866	0.915	150
时政	0.668	0.846	0.747	150
星座	0.933	0.933	0.933	150
游戏	0.932	0.920	0.926	150
社会	0.675	0.846	0.751	150
科技	0.994	0.966	0.923	150
股票	0.756	0.786	0.771	150
财经	0.829	0.646	0.726	150
总体	0.880	0.869	0.871	2100

表 6: 朴素贝叶斯分类性能

从实验结果来看，相比于深度学习 CNN，LSTM 等方法，朴素贝叶斯在分类的精度和召回率上会略有不足，这是因为朴素贝叶斯忽略文本的语义信息，认为所有特征相互独立导致的。在所有类别当中，社会和时政类是精度最低的，而股票和财经类是召回率最低的。这说明经常有类别被错误分类为社会和时政，而股票和财经则经常被错分为其他类，我们基于类别的混淆矩阵进一步分析，如图9所示。

其中 0-13 分别对应 [‘体育’,’ 娱乐’,’ 家居’,’ 彩票’,’ 房产’,’ 教育’,’ 时尚’,’ 时政’,’ 星座’,’ 游戏’,’ 社会’,’ 科技’,’ 股票’,’ 财经’] 这 14 个分类，横坐标代表真实的分类，纵坐标代表预测的分类。

从精度方面来看（按列看），7(时政) 和 10(社会) 是精度最差的，与我们的评测指标一致，4(房产)，5(教育)，10(社会)，12(股票)，13(财经) 是相对容易被错分成时政类的，而 4(房产)，7(时政) 是相对容易被错分成社会的，根据我们个人的直观映像，的确这几类被弄混的可能性较高。除此之外，最后两列错分的数量也相对较多，可以看出 12(股票),13(财经)，两个话题也时常搞混。与深度学习的方法相比，贝叶斯出现错分的情况明显要更多一些，特别是在股票和财经两个分类上，在缺少语义信息的情况下，很难将两者进行区分。

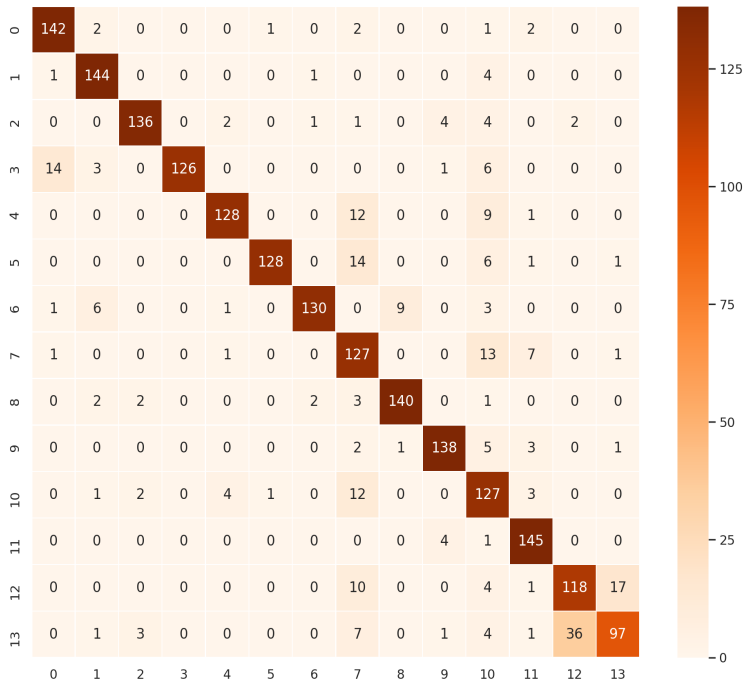

图 9: 朴素贝叶斯混淆矩阵

本系统中的贝叶斯分类器，其分类效果已经可以说是很好了，相比于深度学习的方法，虽然精度和召回率上都要逊色，但优势在于训练所需数据量少，训练速度快。在实际生产中对于缺乏数据样本的分类任务，特别是数据分布独立的任务，采用贝叶斯分类器会更为合适。本系统的贝叶斯分类器仍有以下几点方向可以进行改进：

(1) 特征的选择在特征的选取上，我们选择了每个分类互信息排在前 50 的特征，构成了语料库，这个数量可以减小或增大，来看分类的效果是否改善，减小则是对互信息要求进一步提高，可能会过滤掉更多的不重要特征，增大则放松了对互信息的要求，会添加更多的特征信息，也可能提高分类效果。

(2) TF-IDF 阈值设计贝叶斯分类器的时候，可以对特征进行 TF-IDF 值计算，设定阈值，筛去 TF-IDF 值较低，对文本区分影响不大的特征。

(3) 多个特征与分类关联指标目前仅采用了特征与分类互信息这一种计算两者之间关联性的指标，实际上可以将信息增益 (IG)，χ^2 统计量，文档频率 (DF) 等多个指标都进行计算，仿照决策森林的方式选择出对于区分类别贡献最大的特征构成语料库。

6. 部署与使用

6.1 环境要求

本项目的环境要求为：

- 操作系统：Windows 10, Mac OS, Linux
- Python: 3.6+
- Django: 3.0

- 机器学习库：
  - Pytorch: 1.3.1
  - tensorflow: 1.14
  - keras 2.2.5

如果想要连接上 Apache 或者 Nginx 服务器，需要设置 wsgi.py 文件。这里是测试环境，只需要执行 python manage.py runserver 8080，就会在 8080 端口开启一个轻量级 web 服务。

6.2 安装与部署

具体的步骤如下：

1. 在服务器上，从 github 克隆项目：git@github.com:GentleCP/UCAS-NLP.git
2. 通过 pip install -r requirements.txt 可以安装全部需要的依赖库
3. 使用 python manage.py runserver 8080 开启 web 服务

6.3 使用介绍

现在一共有三个模型，CNN，NativeBayes，LSTM 可以使用，能看到三个模型共同判断的结果，以此能够获得更加准确的结果。

在页面上能够看到一个文本输入框，里面放入需要分类的文本信息，点击提交按钮，我们的文本信息就会提交至后台分析，之后会在右边呈现分类结果。

比如，在这里输入文本：

```

北京时间12月22日，篮网主场逆转老鹰的比赛中，湖人名宿科比-布莱恩特来到现场观战，比赛中，科比还专门给二女儿讲解战术。 科比今天带二女儿Gianna来到现场观战，他们坐在篮网替补席旁边，比赛中这一幕简直不要太有爱：在镜头抓拍到的画面中，科比耐心地为女儿讲解场上的战术，女儿也认真地在听着，让科比频频点头。二女儿可以说是几乎完美地继承了科比的篮球基因，在此前传出的一段视频中，她面对三人包夹，曾从容地命中中投，让无数球迷感慨，不愧是老科的女儿！
```

结果如图10所示，可以看到该文本的分类结果为“体育”，符合实际情况。

图 10: 运行测试效果

7. 总结

本文介绍了文本分类系统的设计与实现，并介绍了系统中用于文本分类的模型与算法，包括：Text-CNN，LSTM 以及朴素贝叶斯算法，然后测试了它们在同一数据集上的效果，与现有的基准系统进行了比较，实验结果表明，使用 Text-CNN 模型得到的结果优于基准系统。同时，本文将训练好的模型与 Web 应用结合，实现一个较为完善的文本分类系统，对输入的文本可以得到三种不同方法的分类结果，方便用户使用。

参考文献

[1] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.