

自然语言处理实验 工程报告

课程：自然语言处理

姓名：付炎平

学号：2019217819

班级：物联网工程 19-2 班

日期：2021.1.16

目录

一、实验一 语料库的收集与整理.....	3
1.1. 研究背景.....	3
1.2. 模型方法.....	3
1.3. 系统设计.....	3
1.4. 系统演示与分析.....	3
1.5 实验分析.....	7
二、实验二 词汇知识库使用技术.....	7
2.1. 研究背景.....	7
2.2. 模型和方法.....	7
2.3. 系统设计.....	8
2.4. 系统设计演示.....	8
2.5 实验分析.....	9
三、实验三 中文分词技术应用.....	10
3.1 研究背景.....	10
3.2 实验的模型和方法.....	10
3.3 系统设计.....	11
3.4 系统设计演示.....	12
3.5 实验分析.....	13
四、实验四 文本分类技术应用.....	14
4.2 模型和方法.....	14
4.3 系统设计.....	15
4.4 系统设计演示.....	15
4.5 实验分析.....	20
五、对本门课的感想和建议.....	20
六、自然语言处理会议总结感悟.....	21

一、实验一 语料库的收集与整理

1.1. 研究背景

词频统计是一种词汇分析研究方法,它通过统计一定长度的语言材料中每个词出现的次数,分析统计结果,以便描绘词汇规律。这种方法也有助于评价著作,确定某种语言或某学科的基本词汇。在情报科学领域,词频统计方法逐渐引起人们的重视,现已成为一种新的情报分析研究方法和情报语言、情报检索研究的手段鉴于词频分析在情报学中的重要作用,有必要对它作全面的了解。

词频统计最初的作用,仅仅是作为一种确定一门语言的基本词汇的手段,为语言教学,尤其是外语教学服务。随着语言学和科学、社会的发展,语言学的应用范围不断扩大,已逐渐成为一门社会科学同自然科学学科。语言学在情报科学领域也得到了应用。词频统计分析既是确定基本词汇的手段,也是测定词汇使用度的手段。

在本工程中,我将利用 Unigrams 和 Bigrams 方法对新闻语料库进行统计。

1.2. 模型方法

本实验使用 Unigrams 和 Bigrams 方法对语料库进行统计词频,并且把统计的词频保存到本地中,同时把新闻语料库中分好的词也保存在本地中,便于后面进行中文分词,其中 Unigrams 生成的词典中全是为单个字的词频, Bigrams 生成的词典全是为两个字组成一个词的词频。

1.3. 系统设计

- (1) 打开开发环境, 根据自己熟悉的语言, 确定开发环境。
- (2) 下载语料库(新闻语料库)到特定目录下
- (3) 根据文本编码, 加载语料库文本
- (4) 分别统计 n-gram (n=1, 2) 的词频, 存储到相应的数据结构, 该数据结构包括词(词本身)和词的频度(出现次数)
- (5) 将内存中的数据结构存储到文本中, 方便后面随时加载。

1.4. 系统演示与分析

1.4.1 下载新闻语料库

19980101-01-001-001/m 迈向/vt 充满/vt 希望/n 的/ud 新/a 世纪/n —/wp 一九九八年/t 新年/t 讲
19980101-01-001-002/m 中共中央/nt 总书记/n 、/wu 国家/n 主席/n 江/nrf 泽民/nrg
19980101-01-001-003/m (/wkz 一九九七年/t 十二月/t 三十一日/t) /wky
19980101-01-001-004/m 1 2月/t 3 1日/t , /wd 中共中央/nt 总书记/n 、/wu 国家/n 主席/n 江/nr
19980101-01-001-005/m 同胞/n 们/k 、/wu 朋友/n 们/k 、/wu 女士/n 们/k 、/wu 先生/n 们/k
19980101-01-001-006/m 在/p 1 9 9 8年/t 来临/vi 之际/f , /wd 我/rr 十分/dc 高兴/a 地/ui 通过
19980101-01-001-007/m 1 9 9 7年/t , /wd 是/vl 中国/ns 发展/vn 历史/n 上{shang5}/f 非常/dc
19980101-01-001-008/m 在/p 这/rz 一/m 年/qt 中/f , /wd 中国/ns 的/ud 改革/vn 开放/vn 和/c
19980101-01-001-009/m 在/p 这/rz 一/m 年/qt 中/f , /wd 中国/ns 的/ud 外交/n 工作/vn 取得/
19980101-01-001-010/m 1 9 9 8年/t , /wd 中国/ns 人民/n 将/d 满怀信心/iv 地/ui 开创/vt 新/a
19980101-01-001-011/m 实现/vt 祖国/n 的/ud 完全/ad 统一/vt , /wd 是/vl 海内外/s 全体/n 中国
19980101-01-001-012/m 台湾/ns 是/vl 中国/ns 领土/n 不可分割/lv 的/ud 一/m 部分/n 。/wj 完成
19980101-01-001-013/m 环顾/vt 全球/n , /wd 日益/d 密切/a 的/ud 世界/n 经济/n 联系/vn , /wd
19980101-01-001-014/m [中国/ns 政府/n]nt 将/d 继续/vt 坚持/vt 奉行/vt 独立自主/lv 的/ud 和平
19980101-01-001-015/m 在/p 这/rz 辞旧迎新/lv 的/ud 美好/a 时刻/n , /wd 我/rr 祝/vt 大家/rr
19980101-01-001-016/m 谢谢/vt ! /wt (/wkz 新华社/nt 北京/ns 1 2月/t 3 1日/t 电/n) /wky

图 1.4.1 新闻语料库

这是下载的新闻语料库的内容，可以看出这个语料库里面已经分好词啦。

1.4.2 进入自然语言处理 Web 界面

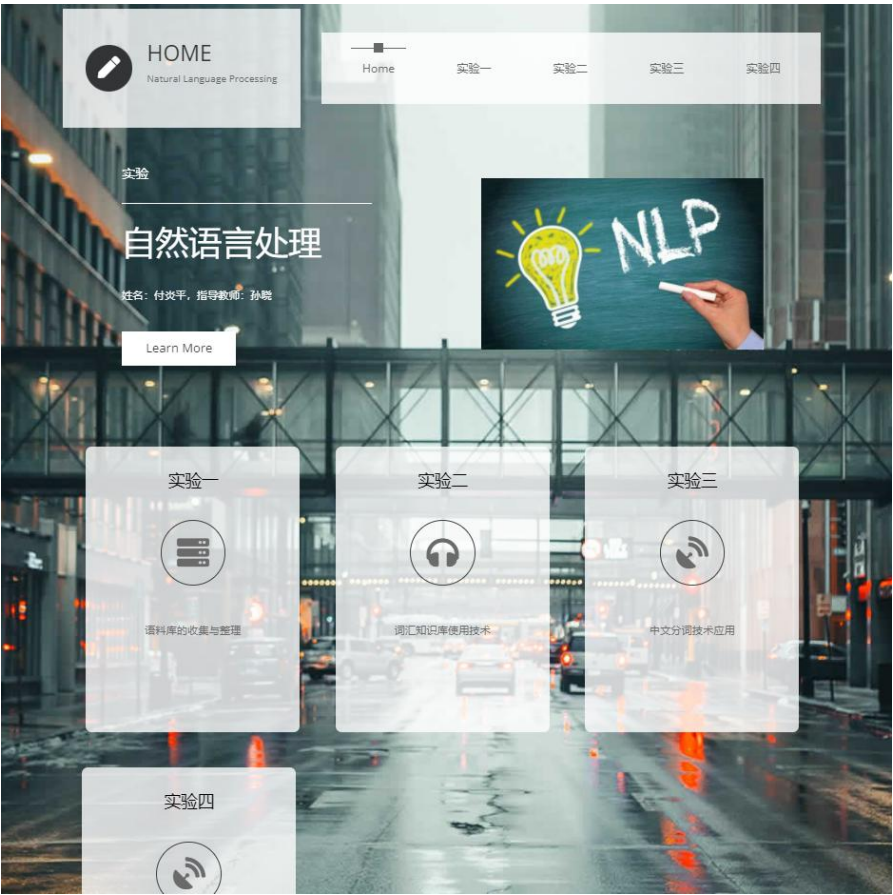


图 1.4.2 自然语言处理 Web 主页面

这里我通过 Web 网页的方式来简单呈现一下自然语言处理的实验。

1.4.3 进入实验一的界面



图 1.4.3 实验一页面

在这个页面中，我简单的把实验一的步骤和内容进行了一个介绍，当点击最下方这个“开始统计词频”按钮时，将开始统计 Unigrams、Bigrams 和已经分好词的词频，并生成相应的词典到本地。

1.4.4 Unigrams 词典

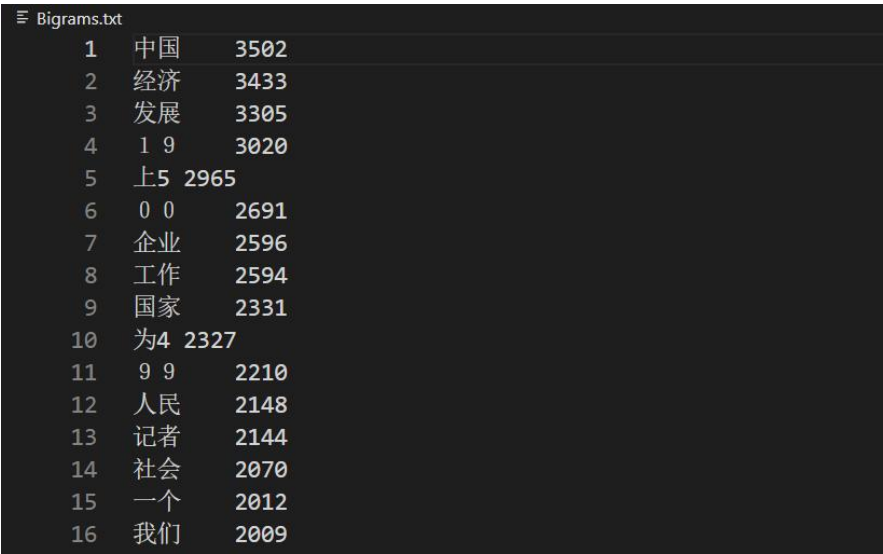
1	,	74496
2	的	54898
3	。	35771
4	国	17793
5	一	17530
6	在	13627
7	中	12875
8	人	12545
9	了	12382
10	1	12373
11	和	11890
12	是	11541
13	有	11113
14	年	11030
15	大	10907
16	不	9216

图 1.4.4 Unigrams 词典

可以看出，Unigrams 词典中仅为单个字以及一些常用的标点符号和数字的频

率，并且是按照频率从大到小的顺序排列的。

1.4.5 Bigrams 词典

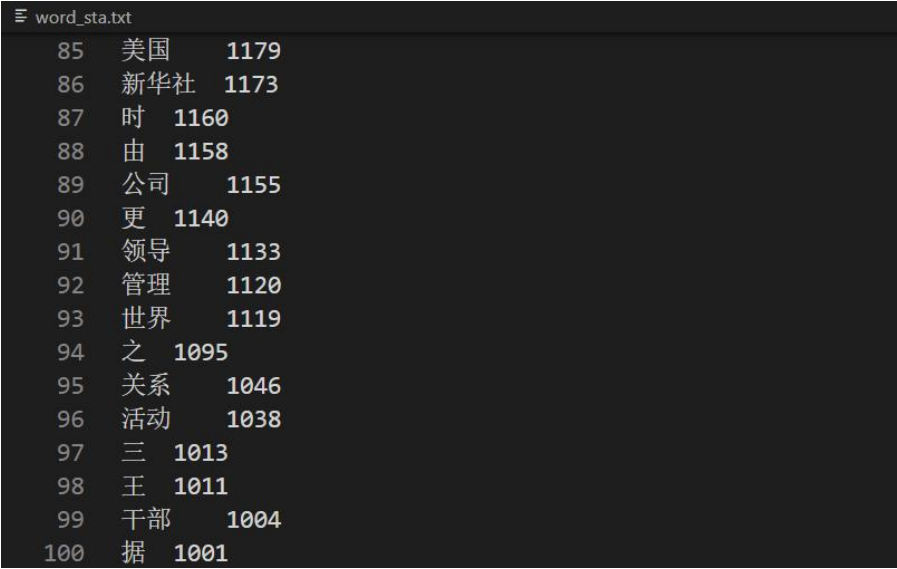


1	中国	3502
2	经济	3433
3	发展	3305
4	1 9	3020
5	上5	2965
6	0 0	2691
7	企业	2596
8	工作	2594
9	国家	2331
10	为4	2327
11	9 9	2210
12	人民	2148
13	记者	2144
14	社会	2070
15	一个	2012
16	我们	2009

图 1.4.5 Bigrams 词典

可以看出，Bigrams 词典是连续两个字或者标点符号或者数字组成的词的频率，也是按照频率从大到小的顺序排列的。

1.4.6 已经分好词的词典



85	美国	1179
86	新华社	1173
87	时	1160
88	由	1158
89	公司	1155
90	更	1140
91	领导	1133
92	管理	1120
93	世界	1119
94	之	1095
95	关系	1046
96	活动	1038
97	三	1013
98	王	1011
99	干部	1004
100	据	1001

图 1.4.6 已经分好词的词典

这是根据语料库中对已经分好的词进行统计，计这个的目的主要是为了后面的中文分词实验。

1.5 实验分析

我在统计词频的时候只统计了一些常用的标点符号和数字，而一些特殊的标点符号没有统计，没有统计的原因是在语料库中出现了不规则的标点符号，如果统计这些不规则的标点符号，将会对后面生成文本产生不良影响。

二、实验二 词汇知识库使用技术

2.1. 研究背景

文本自动生成是自然语言处理领域的一个重要研究方向，实现文本自动生成也是人工智能走向成熟的一个重要标志。简单来说，我们期待未来有一天计算机能够像人类一样会写作，能够撰写出高质量的自然语言文本。文本自动生成技术极具应用前景。例如，文本自动生成技术可以应用于智能问答与对话、机器翻译等系统，实现更加智能和自然的人机交互；我们也可以通过文本自动生成系统替代编辑实现新闻的自动撰写与发布，最终将有可能颠覆新闻出版行业；该项技术甚至可以用来帮助学者进行学术论文撰写，进而改变科研创作模式。

在本次工程中，我将采用 **n_gram** 最大概率匹配和随机匹配的方式来生成一个句子，并对这两种方法进行比较。

2.2. 模型和方法

在生成句子中，首先要确定生成的句子的首个字，在这里我采用随机选取实验一中生成的 **Unigrams** 词典中的一个字作为句子中的首字，然后输入要生成句子的长度，后面采取两种方法进行生成句子。

方法一：采用最大概率匹配的方式生成句子，根据实验一中的 **Bigrams** 词典，寻找和第一个字匹配的第二个字，并把这两个字组成的词的词频最大的字作为下一个字，然后依次查找，直到生成的句子长度达到输入的句子长度。

方法二：采用随机匹配的方式生成句子，根据实验一中的 **Bigrams** 词典，寻找所有和第一个字匹配的第二个字，并随机在这所有的字中选择一个字作它的下一个字，然后依次生成，直到生成的句子长度达到输入的句子长度。

2.3. 系统设计

- (1) 打开开发环境，根据自己熟悉的语言，确定开发环境。
- (2) 将实验一中生成的词典到特定目录下
- (3) 将词典加载到内存中，主要包括词和词频
- (4) 采用随机生成，或者 n -gram 等算法，生成文本序列
- (5) 构建前端，演示文本生成。根据人民日报的词典，自动生成文本内容。

2.4. 系统设计演示

2.4.1 打开实验二 Web 页面



图 2.4.1 实验二页面

这是使用 Web 网页简单介绍实验二的内容和一些基本的步骤。

2.4.2 句子生成



图 2.4.2 句子生成页面

在右侧的输入框中输入要生成句子的长度，这里我输入生成的句子长度为13，然后点击开始生成按钮即可生成句子。

2.4.3 生成句子结果



图 2.4.3 句子生成结果页面

2.5 实验分析

分别使用最大概率匹配和随机匹配这两种方法生成的句子，可以看出，使用最大概率匹配生成的句子容易出现循环，使用随机匹配生成的句子能有效解决循环的问题，但是生成的句子并不是很通顺。

三、实验三 中文分词技术应用

3.1 研究背景

词是自然语言中能够独立运用的最小单位，是自然语言处理的基本单位。自动词法分析就是利用计算机对自然语言的形态进行分析，判断词的结构和类别等。词性或称词类是词汇最重要的特性是连接词汇到句法的桥梁。

中文信息处理是指自然语言处理的分支，是指用计算机对中文进行处理。和大部分西方语言不同，书面汉语的词语之间没有明显的空格标记，句子是以字串的形式出现。因此对中文进行处理的第一步就是进行自动分词，即将字串转变成词串。

自动分词的重要前提是以什么标准作为词的分界。词是最小的能够独立运用的语言单位。词的定义非常抽象且不可计算。给定某文本，按照不同的标准的分词结果往往不同。词的标准成为分词问题一个很大的难点，没有一种标准是被公认的。但是，换个思路思考，若在同一标准下，分词便具有了可比较性。因此，只要保证了每个语料库内部的分词标准是一致的，基于该语料库的分词技术便可一较高下。

目前中文分词的方式有很多，主要的有最大匹配法，复杂最大匹配法，基于字标注的分词法。我们在上课主要介绍了 n -gram 模型，前后向最长匹配，隐马尔科夫，条件随机场等模型。

3.2 实验的模型和方法

3.2.1 前向最大匹配方法

前向最大匹配 FMM 的原理是：假设词典中最大词条所含的汉字个数为 n 个，取待处理字符串的前 n 个字作为匹配字段，查找分词词典。若词典中含有该词，则匹配成功，分出该词，然后从被比较字符串的 $n+1$ 处开始再取 n 个字组成的字段重新在词典中匹配；如果没有匹配成功，则将这 n 个字组成的字段的最后一位剔除，用剩下的 $n-1$ 个字组成的字段在词典中进行匹配，如此进行下去，直到切分成功为止。

3.2.2 后向最大匹配方法

后向最大匹配 RMM 的分词原理和过程与正向最大匹配相似，区别在于前者从文章或者句子(字串)的末尾开始切分，若不成功则减去最前面的一个字。比如对于字符串“处理机器发生的故障”，第一步，从字串的右边取长度以步长为单位的字段“发生的故障”在词典中进行匹配，匹配不成功，再取字段“生的故障”进行匹配，依次匹配，直到分出“故障”一词，最终使用 RMM 方法切分的结果为：故障、发生、机器、处理。该方法要求配备逆序词典。

3.2.3 双向最大匹配方法

双向最大匹配分词是综合了前两种算法。先根据标点对文档进行粗切分，把文档分解成若干个句子，然后再对这些句子用正向最大匹配法和逆向最大匹配法进行扫描切分。如果两种分词方法得到的匹配结果相同，则认为分词正确，否则，按最小集处理。准确的结果往往是词数切分较少的那种。

3.2.4 使用 n_gram 最大概率匹配方法进行中文分词

n_gram 最大概率匹配方法是根据句子的首字，查找首字后面所有能组成的词在词典中词频最高的那一个，将它作为一个完整的词进行切分，然后将该词后面的第一个字作为首字继续查找词典进行切分，直到将句子切分完为止。

3.3 系统设计

- (1) 打开开发环境，根据自己熟悉的语言，确定开发环境。
- (2) 导入相应的分词词典，如果基于模型，则导入相应的模型。
- (3) 实现 FMM 和 BMM，或者基于 n-gram 的分词方法
- (4) 实现界面，输入一句话，输出分词结果
- (5) 存储分词结果到文件中。

3.4 系统设计演示

3.4.1 打开实验三 Web 界面



图 3.4.1 实验三页面

这是简单介绍实验三内容和步骤的 Web 界面。

3.4.2 开始分词



图 3.4.2 开始分词页面

这里我输入的句子为：国家旅游局局长何光江苏省政府领导等以及数千名中外宾朋参加了这一隆重而特别的仪式。点击按钮，开始分词。

3.4.3 分词结果



图 3.4.3 分词结果页面

3.5 实验分析

上面使用了四种分词方法分别对输入的句子进行分词，从这四种分词结果中可以看出，前向最大匹配算法和后向最大匹配算法分词的结果相似但有一定的区别，双向最大匹配算法是对前两种算法中进行选择，n_gram 最大概率匹配算法的分词结果中词比较短、个数比较多，跟它算法中最大概率匹配有关。

在这几种算法中，前向最大匹配、后向最大匹配、双向最大匹配是基于词典的分词方法，n_gram 最大概率匹配是基于统计的分词方法。

四、实验四 文本分类技术应用

研究文本分类是自然语言处理中最基本的任务。由于深度学习的空前成功，过去十年中该领域的研究激增。已有的文献提出了许多方法，数据集和评估指标，从而需要对这些内容进行全面的总结。本文回顾 1961 年至 2020 年的文本分类方法，重点是从浅层学习到深度学习的模型。根据所涉及的文本以及用于特征提取和分类的模型创建用于文本分类的分类法。然后，详细讨论这些类别中的每一个类别，涉及支持预测测试的技术发展和基准数据集。并提供了不同技术之间的全面比较，确定了各种评估指标的优缺点。最后，通过总结关键含义，未来的研究方向以及研究领域面临的挑战进行总结。

在许多 NLP 应用中，文本分类-为文本指定预定义标签的过程-是一项基础而重要的任务。文本分类的主要流程：首先是预处理模型的文本数据。浅层学习模型通常需要通过人工方法获得良好的样本特征，然后使用经典的机器学习算法对其进行分类。因此，该方法的有效性在很大程度上受到特征提取的限制。但是，与浅层模型不同，深度学习通过学习一组非线性变换将特征工程直接集成到输出中，从而将特征工程集成到模型拟合过程中。

在本工程中，我采用基于 Pytorch 框架的 LSTM 神经网络模型对今日头条文本进行文本分类。

4.2 模型和方法

4.2.1 读取数据集

使用 Pytorch 自定义 dataset 读取今日头条数据集，并对数据集进行预处理，得出样本和标签，并把数据集分为训练集和验证集。

4.2.2 中文分词

使用 jieba 分词库，对样本进行中文分词，并统计每个词的词频并保存到本地以便于后面特征向量化。

4.2.3 特征向量化

读取词频文件，先把每个词用它在词频文件中的索引位置来表示，由于对不同句子分词后得到的词的个数不一致，词转换成数字后数字的个数也不一致，需要把每个句子转换成同一数量的数字来表示，在本工程中我把这个数量设置为 30，并把词个数小于 30 的句子用 0 来补充，把词个数大于 30 的句子多余部分进行裁剪，最后采用 `torch.nn.Embedding()` 中的预训练模型把词转换成词向量，每个词用 100 维的向量来表示，实现 Word2Vec 的过程。

4.2.4 定义模型

使用 Pytorch 搭建 LSTM 神经网络模型，在 LSTM 后面使用线性层来进行分类。

4.2.5 训练模型

总共训练 100 个 epoch，每 5 个 epoch 用验证集进行验证，训练完成之后保存模型到本地。

4.2.6 测试模型

读取保存的模型，输入一个句子，对它进行分类。

4.3 系统设计

- (1) 打开开发环境，根据自己熟悉的语言，确定开发环境。
- (2) 下载语料库（今日头条分类语料）到特定目录下
- (3) 根据不同的文本特征提取方法，提取文本特征，实现文本向量化，即将文本转换成向量。
- (4) 将向量转换为 LSTM 神经网络模型的格式，使用 LSTM 进行训练，获得模型。
- (5) 搭建演示系统，输入一段文本，输出分类结果。

4.4 系统设计演示

4.4.1 下载今日头条数据集


```

6552263394420850951 !_101!_news_culture!_夕阳无语燕归愁，如何接下句? !_
6552290314294395139 !_101!_news_culture!_上联：山水醉人何须酒。如何对下联? !_
6552462208314376462 !_101!_news_culture!_上联：上班为下班，如何对下联? !_
6552311866947797262 !_101!_news_culture!_下联：夕陽西下已黄昏。上联是什麼? !_
6552466638304707079 !_101!_news_culture!_初夏方好，小荷初露，一起来读那些美不胜收的咏荷经典诗词 !_荷花
6552431613437805063 !_102!_news_entertainment!_谢娜为李浩菲澄清网络谣言，之后她的两个行为给自己加分 !_
6552320560913711629 !_102!_news_entertainment!_谢娜曾为他与主办方撕破脸！谢娜复出快本！他第一个到场支
6552390546051039747 !_102!_news_entertainment!_中国网红竟红到美国？不多说了，连小编都心动了 !_飞纱,新
6552150358678831624 !_102!_news_entertainment!_赵丽颖很久没有登上微博热搜了，但你们别急，她只是在憋大
6552408585177924099 !_102!_news_entertainment!_因戴一个眼镜更改变气质的6大娱乐圈明星，你最喜欢哪一个
6552147830184608263 !_102!_news_entertainment!_后来的我们，抢先看 !_电影院,前任3,刘若英,张一白,田壮壮
6552472345548685837 !_102!_news_entertainment!_超级英雄演员颜值身材排名，钢铁侠进不了前5，第一名很意外
6552385284682547716 !_102!_news_entertainment!_《无限歌谣季》热播 张绍刚毛不易组合似“父子” !_张绍刚,新
6552364464715334151 !_102!_news_entertainment!_张靓颖透露右耳已经间歇性失聪10年，这些年她都是怎么过来
6552310157706002702 !_102!_news_entertainment!_成龙改口决定不裸捐了，20亿财产给儿子一半，你怎么看? !_

```

图 4.4.1 今日头条数据集页面

上图是今日头条数据集中的一部分，这个数据集每一行为一段文本，文本的左侧包括它的类别，这个数据集中一共有 15 个类别，分别是 news_tech, stock, news_car, news_entertainment, news_military, news_sports, news_finance, news_culture, news_game, news_world, news_travel, news_house, news_edu, news_agriculture。

4.4.2 中文分词

我使用了 jieba 分词库对读入的数据集文本进行分词，分词之后统计词频并把词频文件保存到本地，便于进行特征向量化，词频文件如图所示：

```

exp4_word_sta.txt
1  的 8243
2  了 2353
3  你 2293
4  是 2154
5  吗 1846
6  有 1760
7  在 1729
8  怎么 1531
9  如何 1408
10 什么 1385
11 为什么 1297
12 中国 1215
13 被 1074
14 和 1047
15 看 1038
16 会 1028
17 人 1003
18 都 938
19 不 936
20 美国 922

```

图 4.4.2 分词后的词典

4.4.3 开始训练模型

```
The 0 loss of 1 epoch is 2.7134523391723633
The 200 loss of 1 epoch is 2.307008981704712
The 400 loss of 1 epoch is 2.7141430377960205
The 600 loss of 1 epoch is 2.267549753189087
The 800 loss of 1 epoch is 2.3296048641204834
The 1000 loss of 1 epoch is 2.340648651123047
The 1200 loss of 1 epoch is 2.250553846359253
The 1400 loss of 1 epoch is 2.1946632862091064
The 1600 loss of 1 epoch is 2.1612355709075928
The 1800 loss of 1 epoch is 2.3263354301452637
The 0 loss of 2 epoch is 1.5668582916259766
The 200 loss of 2 epoch is 1.7012187242507935
The 400 loss of 2 epoch is 1.2095680236816406
The 600 loss of 2 epoch is 1.4883058071136475
The 800 loss of 2 epoch is 1.7081981897354126
The 1000 loss of 2 epoch is 1.3931409120559692
The 1200 loss of 2 epoch is 1.2833027839660645
The 1400 loss of 2 epoch is 1.6002333164215088
The 1600 loss of 2 epoch is 1.9471378326416016
The 1800 loss of 2 epoch is 1.304559350013733
```

图 4.4.3-1 开始训练模型

上图是刚开始训练的效果，在训练时一共训练了 100 个 epoch，每训练 5 个 epoch 进行了验证，从上图中可以看出，训练的 loss 是呈现出下降的趋势，说明我们的训练是正常的。

```
The 0 loss of 99 epoch is 0.008194957859814167
The 200 loss of 99 epoch is 0.4310005009174347
The 400 loss of 99 epoch is 0.2060798704624176
The 600 loss of 99 epoch is 0.6142268180847168
The 800 loss of 99 epoch is 0.1953771412372589
The 1000 loss of 99 epoch is 0.8198233842849731
The 1200 loss of 99 epoch is 0.11469068378210068
The 1400 loss of 99 epoch is 0.5763441920280457
The 1600 loss of 99 epoch is 0.127468079328537
The 1800 loss of 99 epoch is 1.288743495941162
The 0 loss of 100 epoch is 0.17520444095134735
The 200 loss of 100 epoch is 0.03894542530179024
The 400 loss of 100 epoch is 0.1357024610042572
The 600 loss of 100 epoch is 0.047504521906375885
The 800 loss of 100 epoch is 0.26823723316192627
The 1000 loss of 100 epoch is 0.8766639828681946
The 1200 loss of 100 epoch is 0.30103039741516113
The 1400 loss of 100 epoch is 0.8716837167739868
The 1600 loss of 100 epoch is 0.983893871307373
The 1800 loss of 100 epoch is 0.7514621615409851
```

图 4.4.3-2 训练结束

当 epoch 为 99 和 100 时，发现模型已经收敛，loss 已经很小了。

```
Predicted: tensor([11]) label: tensor([11])
Predicted: tensor([7]) label: tensor([7])
Predicted: tensor([4]) label: tensor([4])
Predicted: tensor([10]) label: tensor([10])
Predicted: tensor([12]) label: tensor([12])
Predicted: tensor([1]) label: tensor([3])
Predicted: tensor([10]) label: tensor([10])
Predicted: tensor([9]) label: tensor([12])
Predicted: tensor([4]) label: tensor([4])
Predicted: tensor([7]) label: tensor([7])
Predicted: tensor([1]) label: tensor([1])
Predicted: tensor([4]) label: tensor([4])
Predicted: tensor([11]) label: tensor([11])
Predicted: tensor([5]) label: tensor([5])
Predicted: tensor([9]) label: tensor([9])
Predicted: tensor([11]) label: tensor([11])
```

图 4.4.3-3 验证

上图是训练中验证的过程，Predicted 是模型预测的结果，label 是该文本的标签，可以看出模型在验证过程中的准确率还是挺高的。

训练完成之后，使用 Pytorch 把模型参数保存到本地并命名为 model.pt，以便后面在 Web 网页上进行测试。

4.4.4 打开实验四 Web 页面



图 4.4.4 实验四页面

这个 Web 页面简单对实验四的内容和步骤进行了介绍,在页面的输入框我输入了一段文本:画家杨象乔以作品《蚩尤》讴歌民族精神蚩尤,战神蚩尤,少数民族,中国神话,战神,中华民族。

点击开始分类按钮,加载训练好的模型,开始分类。

4.4.5 文本分类结果



图 4.4.5 文本分类结果

从上图可以看出，文本分类的结果是 News_culture，与文本涵义一致。

4.5 实验分析

本实验采用的是深度学习方法 LSTM 模型进行文本分类，最后分类的准确率在 80%左右，造成这个准确率不是特别高的原因可能是有些文本的内容较短，而本工程设置所有的输入词的数量均为 30，对于那些缺少的词采用的是补 0 法，导致一个较短文本的有效特征较少，最后可能影响准确率，改进方法是采用更好的特征提取方法，实现 Word2Vec 的过程。

五、对本门课的感想和建议

自然语言处理的课程，让我学到了很多知识，也锻炼了我的动手能力，代码编写能力，让我对自然语言处理这个领域有了一定的认识，从词频统计到文本生成再到中文分词最后到文本分类，每一步都很重要，并且用不同的算法会产生不同的结果，把这些算法的结果进行不断比对，发现每个算法都有各自的特点，让我对老师上课讲的相关内容进行了进一步的巩固。

这次课程，提高了我的工程能力，让我把一些理论使用代码进行实现，并且了解自然语言处理具有很大的意义，同时我也感受到了处理自然语言的困难，因为中文语义的复杂性，使得我们的处理遇到了很大的问题，通过本次课程，也加深了我对自然语言处理的兴趣。自然语言处理这门课让我收获很多，我希望这门课能够发展的越来越好。

六、自然语言处理会议总结感悟

6.1 会议截图

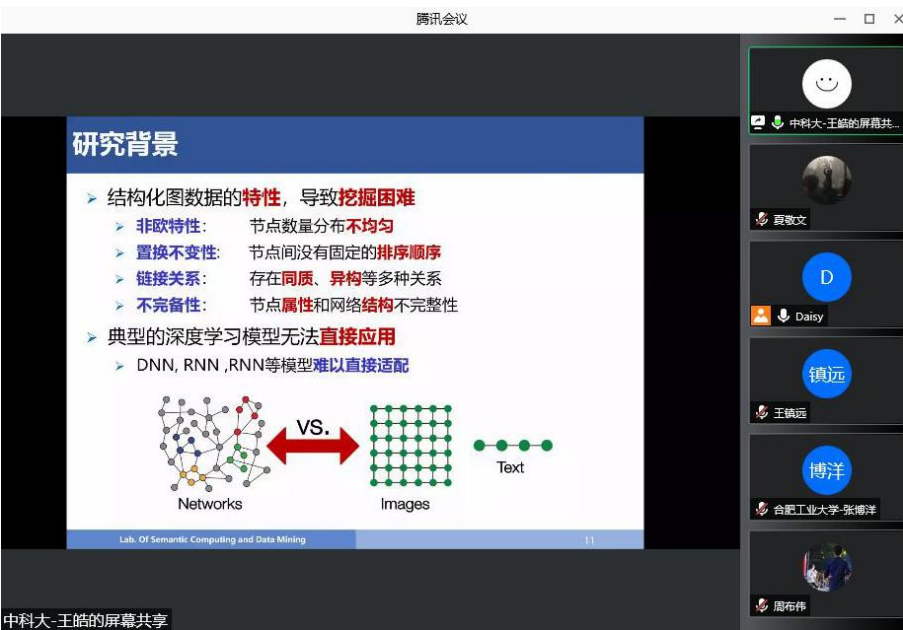


图 6.1.1 会议截图 1

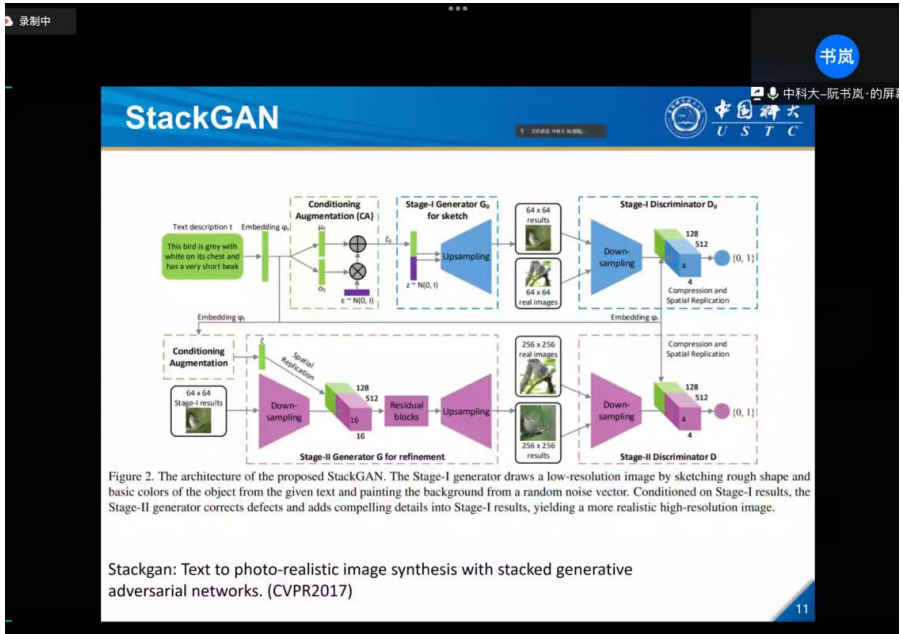


图 6.1.2 会议截图 2

听了学长们的总结报告，让我对深度学习有了一个更加深刻的理解，让我印象比较深刻的有那个介绍图神经网络的王皓学长，听完学长的讲座之后我在网上查了一些关于图神经网络的资料，图神经网络简称 GNN，主要处理图相关的节点和边之间的表征。在我们之前学习的课程中，对神经网络的理解都是处理一般

数据的，没有想到还能来处理图数据，在听学长后面的讲解，关于处理图数据的神经网络模型有：GCN，GraphSage 等，然后我上网查了关于这两个模型的一些资料，发现他跟我们传统神经网络的最大的区别是在特征提取的部分，图神经网络在特征提取的时候把一个节点的周围的节点特征进行了聚合，然后把聚合后的特征再送入我们传统的神经网络中，进行分类和回归的任务，在图神经网络中比较重要的环节是特征聚合，从而也有不同的算法来处理这个问题，像学长介绍的 deepWalk，而 GraphSage 采用的特征聚合方式是抽样聚合，听了学长在图神经网络的讲解，让我又认识了一个新的神经网络结果，并且让我对神经网络也有了一个更加深入的理解。

其次让我印象比较深刻的是 StackGAN 的讲解，我之前是对 GAN 有过一定的了解，知道 GAN 是用来生成数据的，在图像合成这个领域，GAN 应用广泛，并且有很多的变体，GAN 中最经典的它的生成模型和判别模型，两者相互对抗，生成模型需要学习的是生成数据来欺骗判别模型，而判别模型需要学习的是尽可能的区分生成的数据和真实数据，两者相互对抗，而 StackGAN 具有两个 GAN 堆叠在一起形成了一个能够生成高分辨率图像的网络。它分为两个阶段，Stage-I 和 Stage-II。Stage-I 网络生成具有基本颜色和粗略草图的低分辨率图像，并以文本嵌入为条件，而 Stage-II 网络获取由 Stage-I 网络生成的图像并生成以文本嵌入为条件的高分辨率图像。基本上，第二个网络可以纠正缺陷并添加细节，产生更逼真的高分辨率图像，StackGAN 让我又认识了一个新的 GAN 的变体，拓宽了我的视野，并且让我对 GAN 有了一个更加深入的理解。

这次会议让我收获很大，学到了很多前沿的新的知识，非常感谢学长学姐们能够给我们讲解这些前沿的知识。