2023-2024学年第2学期
《自然语言处理导论》
实验1：分词

专业：计算机科学与技术
学号：210340170
姓名：薛皓天
教师：卢敏

一、实验目的

1、学会使用爬虫进行文本的爬取。

2、学会使用jieba进行中文语料的分词，并将其转化为对应的字典。

3、使用正向最大匹配算法、反向最大匹配算法、HMM算法进行分词。

二、实验内容

1、网络爬虫

使用BeautifulSoup包进行网页的解析和数据的获取，使用re包对文字进行匹配，使用urllib包进行url的获取。

具体思路：第一步，从学校官网开始进行搜索，获取所有li标签下的网址，若其中有异常，则返回“####”，然后针对每一个网址，在重复上述操作，仅往下进行一轮（原因是，搜索速度过慢），最后存入文件当中。第二步，整理url，当然有一些网址和中国民航大学没有任何关系，因此只筛选出“cauc”的网址。第三步，使用正则表达式获取当中的文本，最后写入文件当中，如下述代码所示，仅展示核心代码。

Listing 1 获取li标签下的网址

```
def get_li(url):
    temp = []
    try:
        res = urllib.request.Request(url , headers=head)
        resp = urllib.request.urlopen(res)
        html = resp.read().decode('utf-8')
        soup = bs(html , "lxml")
        urls_li = soup.select("a")
        for i in urls_li:
            s = str(i)
            if "htt" in s and "target" in s:
                idx = s.index("htt")
                idx1 = s.index("target")
                s = s[idx:idx1 - 2]
                print(s)
                temp.append(s)
    finally:
            return "####" if not len(temp) else temp
```

2、jieba分词

具体思路：第一步：设置停用词，读取爬取文件，将停用词对应项替换成空即可。然后将文本以句号为分割，分割成不同的句子。第二步，调用jieba进行分词。第三步，从大到小排序选出即可得到词表。

如下述代码所示，仅给出关键算法。如图Figure 1所示，分词结果。

Listing 2 jieba分词

```
def jieba_spilt(corpus):
    split_words = []
    for i in range(len(corpus)):
        for x in jieba.cut(corpus[i]): # 分词
            split_words.append(x)

    point_split_words = []
    for i in range(len(corpus)):
        for x in pseg.cut(corpus[i]): # 词性标注
            point_split_words.append(x)

    return split_words , point_split_words
```

```
[('中国民航', 30609), ('学', 20185), ('机场', 6719), ('服务', 6454), ('学学', 5772), ('公开', 5117), ('分享', 5052), ('中航
', 3961), ('报', 3811), ('分析', 3616), ('学校', 3394), ('天津', 3328), ('影响', 3256), ('学生', 3162), ('建设', 3109), ('
平台', 3075), ('航空', 3015), ('基', 2985), ('系统', 2922), ('道', 2848), ('应力', 2840), ('混凝土', 2816), ('面板', 2676),
 ('校友', 2634), ('中', 2597), ('更', 2541), ('冻土', 2508), ('创新', 2502), ('理', 2439), ('发展', 2425), ('模型', 2412),
('推荐', 2369), ('天津市', 2330), ('民航', 2273), ('中国', 2271), ('试验', 2248), ('道面', 2204), ('学院', 2196), ('模拟',
2143), ('教育', 2046), ('科技', 2012), ('校区', 2003), ('微博', 1994), ('磋商', 1984), ('信息', 1961), ('飞机', 1951), ('技
术', 1894), ('文章', 1874), ('工作', 1861), ('剪切', 1804), ('号', 1746), ('企业', 1721), ('更正', 1712), ('优化', 1700),
('场', 1678), ('温度', 1644), ('数值', 1644), ('波速', 1636), ('学报', 1584), ('区', 1584), ('分号', 1548), ('文', 1531),
('路面', 1524), ('时', 1488), ('活动', 1480), ('国际', 1478), ('微信', 1476), ('全国', 1473), ('关', 1454), ('科学', 1451),
 ('评价', 1446), ('相关', 1439), ('碳', 1420), ('实验室', 1417), ('宋体', 1372), ('数字化', 1368), ('跑道', 1356), ('校园',
 1340), ('预压', 1332), ('力学', 1312), ('劣化', 1300), ('设备', 1276), ('碳纤维', 1264), ('北京', 1262), ('方法', 1233),
('走进', 1198), ('链接', 1171), ('请', 1168), ('超', 1155), ('高校', 1153), ('教师', 1153), ('新浪', 1152), ('寿命', 1148),
 ('路基', 1144), ('测试', 1142), ('提供', 1132), ('性', 1118), ('组织', 1115), ('会', 1111), ('天开', 1107), ('资源',
1098), ('目标', 1091), ('飞行', 1090), ('参考文献', 1080), ('设计', 1072), ('升温', 1072), ('科研', 1069), ('耦合', 1068),
```

Figure 1: 分词结果

3、正向最大匹配和反向最大匹配

具体思路：第一步，枚举字典，找到字典中词的最大长度。第二步，从左到右枚举每一个句子，从下标0开始，并以最大长度往下枚举，如果找到了符合的词，则截取出该词，然后下标向后偏移当前最大长度位即可，否则直到最大长度减为1，截取该长度为1的词，然后下标往后偏移一位即可。反向最大匹配，即与正向最大匹配枚举方向相反，其余一样。

如下述代码所示，仅给出关键算法。

Listing 3 正向最大匹配算法

```
def cut(self, text):
    result = [];index = 0;length = len(text)
    while index < length - 1:
        word = None
        for size in range(self.maximum , 0 , -1):
            piece = text[index: index + size]
            if piece in self.dictionary:
                word = piece;result.append(word);index += size
                break
        if word is None:
            result.append(text[index]);index += 1
    return result
```

Listing 4 逆向最大匹配算法

```
def cut(self, text):
    result = []
    while len(text) >= 1:
        word = None
        for size in range(self.maximum , 0 , -1): # 字典中的最大长度往下枚举
            piece = text[-size:]
            if piece in self.dictionary: # 在里面加入即可
                word = piece;result.append(word);text = text[:-size]
                break
        if word is None: # 字典中没有直接结束
            result.append(text[-1:]);text = text[:-1]
    return result
```

4、HMM分词

HMM 思想：生成式算法。根据标签Y，生成X。即由X(汉字串）生成Y(字标签，如B、M、E、S)基本思想可以由下述的公式(1)给出，公式(2)可由初始概率和状态转移概率给出,公式(3)可由发射概率给出

$$\begin{aligned} Y^* &= \arg\max_Y P(Y|X) \\ &= \arg\max_Y \frac{P(X|Y)P(Y)}{P(X)} \\ &= \arg\max_Y P(X|Y)P(Y) \\ &= \arg\max_{y_1 \cdots y_n} P(x_1 ... x_n | y_1 ... y_n) P(y_1 ... y_n) \end{aligned} \tag{1}$$

$$\mathrm{P}\left(\mathrm{y}_1, \mathrm{y}_2, \cdots, y_n\right) = P\left(y_1\right) \prod_{i=1}^{n-1} P\left(\mathrm{y}_{i+1} \mid y_i\right) \tag{2}$$

$$\mathrm{P}\left(x_1, x_2, \cdots, x_n \mid \mathrm{y}_1, \mathrm{y}_2, \cdots, y_n\right) = \prod_{i=1}^{n} P\left(\mathrm{x}_n \mid y_n\right) \tag{3}$$

具体需要计算以下参数:

1、状态集合：HMM包含一系列离散的、不可直接观测的隐藏状态。每个状态代表序列中某一时刻的潜在状态。

2、观察符号集合：对应于每个隐藏状态，会生成一个可观测的符号。这些符号构成了观测序列，是模型可以直接接触到的数据。

3、转移概率矩阵：描述隐藏状态之间的转移概率，即从一个状态转移到另一个状态的概率，反映了状态间的动态变化规律。

4、发射概率矩阵：定义了每个隐藏状态下生成特定观测符号的概率，体现了隐藏状态与可观测符号之间的联系。

使用HMM计算完成后使用viterbi算法进行状态的标注，最后得出概率最大的最优路径,如Listing5得到最优路径

Listing 5 viterbi算法

```
def viterbi(self , text , state_list , start_p , emit_p , trans_p ,smooth):
    text = list(text)
    dp = pd.DataFrame(index = state_list)
    # 初始化 dp 矩阵 (prop, last_state)
    dp[0] = [(start_p[s] * emit_p[s].get(text[0], smooth),
              '_start_') for s in state_list]
    # 动态规划地更新 dp 矩阵
    for i, ch in enumerate(text[1:]):  # 遍历句子中的每个字符 ch
        dp_ch = []
        for s in state_list:  # 遍历当前字符的所有可能状态
            emit = emit_p[s].get(ch, smooth)
            # 遍历上一个字符的所有可能状态，寻找经过当前状态的最优路径
            (prob, last_state) = max([
                (dp.loc[ls, i][0] * trans_p[ls].get(s, smooth) *
                 emit, ls) for ls in state_list
            ])
            dp_ch.append((prob, last_state))

        # frames = pd.DataFrame(pd.Series(dp_ch) , columns = i + 1)
        # dp = pd.concat([dp , frames], axis=1)
        dp[i + 1] = dp_ch.copy()
    # 回溯最优路径
    path = []
    end = list(dp[len(text) - 1])
    back_point = state_list[end.index(max(end))]
    path.append(back_point)
    for i in range(len(text) - 1, 0, -1):
        back_point = dp.loc[back_point, i][1]
        path.append(back_point)
    path.reverse()
    return path
```

三、实验结果和结论

如图Figure2为jieba分词后的前五十的结果，如图Figure3为爬虫获取的结果，如图Figure4为爬虫获取的网址，如图Figure5正向最大匹配分词结果，如图Figure6反向最大匹配分词结果，如图Figure7给出三种学习出的矩阵，如图Figure8viterbi算法给出的最优路径，如图Figure9给出当前句子的分词结果 实验结论：1. 爬虫部分：成功从学校官网获取了包含“cauc”的相关网址，并爬取了其中的文本内容，并把文本内容已整理并保存至文件中。但是存在问题，爬虫运行效率受限于网络速度和服务器响应速度，并没有增加异常处理机制，以提高爬虫的健壮性。也可以尝试使用多线程或异步IO来提高爬虫效率。2. jieba分词部分：jieba分词准确率高，且支持自定义词典和停用词，非常适用于中文文本处理。在处理大规模文本时，可以考虑使用jieba的并行模式来提高分词速度。3. 正向最大匹配与反向最大匹配部分：两种算法均简单易实现，但分词效果受限于词典的完备性和准确性。在实际应用中，可以结合多种分词算法（如基于统计的分词算法）来提高分词效果。

Figure 2: 词典

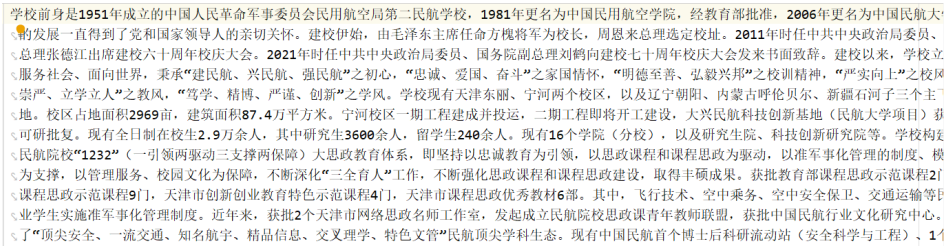
学校前身是1951年成立的中国人民革命军事委员会民用航空局第二民航学校，1981年更名为中国民用航空学院，经教育部批准，2006年更名为中国民航大
的发展一直得到了党和国家领导人的亲切关怀。建校伊始，由毛泽东主席任命方槐将军为校长，周恩来总理选定校址。2011年时任中共中央政治局委员、
总理张德江出席建校六十周年校庆大会。2021年时任中共中央政治局委员、国务院副总理刘鹤向建校七十周年校庆大会发来书面致辞。建校以来，学校立
服务社会、面向世界，秉承“建民航、兴民航、强民航”之初心，“忠诚、爱国、奋斗”之家国情怀，“明德至善、弘毅兴邦”之校训精神，“严实向上”之校风
崇严、立学立人”之教风，“笃学、精博、严谨、创新”之学风。学校现有天津东丽、宁河两个校区，以及辽宁朝阳、内蒙古呼伦贝尔、新疆石河子三个主
地。校区占地面积2969亩，建筑面积87.4万平方米。宁河校区一期工程建成并投运，二期工程即将开工建设，大兴民航科技创新基地（民航大学项目）获
可研批复。现有全日制在校生2.9万余人，其中研究生3600余人，留学生240余人。现有16个学院（分校），以及研究生院、科技创新研究院等。学校构
民航院校“1232”（一引领两驱动三支撑两保障）大思政教育体系，即坚持以思诚教育为引领，以思政课程和课程思政为驱动，以准军事化管理的制度、模
为支撑，以管理服务、校园文化为保障，不断深化“三全育人”工作，不断强化思政课程和课程思政建设，取得丰硕成果。获批教育部课程思政示范课程2门
课程思政示范课程9门，天津市创新创业教育特色示范课程4门，天津市课程思政优秀教材6部。其中，飞行技术、空中乘务、空中安全保卫、交通运输等
业学生实施准军事化管理制度。近年来，获批2个天津市网络思政名师工作室，发起成立民航院校思政课青年教师联盟，获批中国民航行业文化研究中心。
了“顶尖安全、一流交通、知名航宇、精品信息、交叉理学、特色文管”民航顶尖学科生态。现有中国民航首个博士后科研流动站（安全科学与工程）、1个

Figure 3: 文本

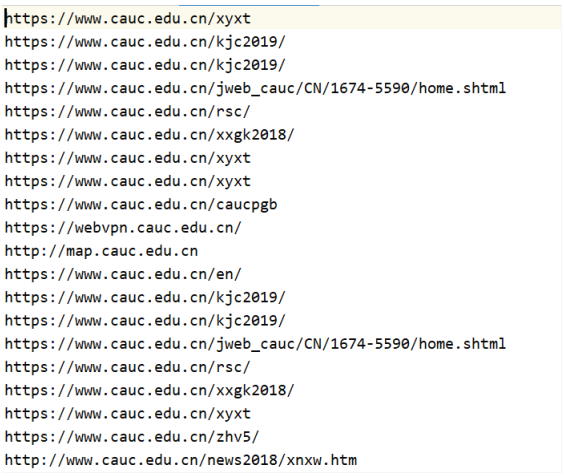
https://www.cauc.edu.cn/xyxt
https://www.cauc.edu.cn/kjc2019/
https://www.cauc.edu.cn/kjc2019/
https://www.cauc.edu.cn/jweb_cauc/CN/1674-5590/home.shtml
https://www.cauc.edu.cn/rsc/
https://www.cauc.edu.cn/xxgk2018/
https://www.cauc.edu.cn/xyxt
https://www.cauc.edu.cn/xyxt
https://www.cauc.edu.cn/caucpgb
https://webvpn.cauc.edu.cn/
http://map.cauc.edu.cn
https://www.cauc.edu.cn/en/
https://www.cauc.edu.cn/kjc2019/
https://www.cauc.edu.cn/kjc2019/
https://www.cauc.edu.cn/jweb_cauc/CN/1674-5590/home.shtml
https://www.cauc.edu.cn/rsc/
https://www.cauc.edu.cn/xxgk2018/
https://www.cauc.edu.cn/xyxt
https://www.cauc.edu.cn/zhv5/
http://www.cauc.edu.cn/news2018/xnxw.htm

Figure 4: url

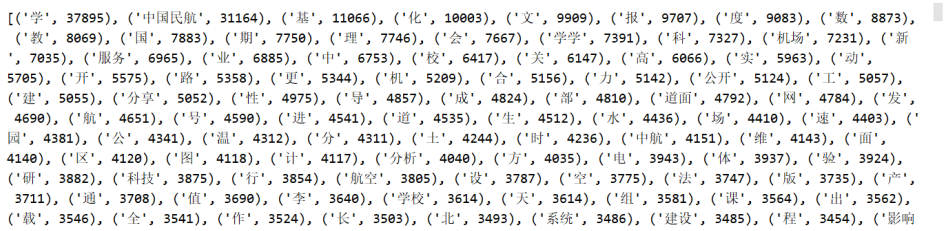
[('学', 37895), ('中国民航', 31164), ('基', 11066), ('化', 10003), ('文', 9909), ('报', 9707), ('度', 9083), ('数', 8873), ('教', 8069), ('国', 7883), ('期', 7750), ('理', 7746), ('会', 7667), ('学学', 7391), ('科', 7327), ('机场', 7231), ('新', 7035), ('服务', 6965), ('业', 6885), ('中', 6753), ('校', 6417), ('关', 6147), ('高', 6066), ('实', 5963), ('动', 5705), ('开', 5575), ('路', 5358), ('更', 5344), ('机', 5209), ('合', 5156), ('力', 5142), ('公开', 5124), ('工', 5057), ('建', 5055), ('分享', 5052), ('性', 4975), ('导', 4857), ('成', 4824), ('部', 4810), ('道面', 4792), ('网', 4784), ('发', 4690), ('航', 4651), ('号', 4590), ('进', 4541), ('道', 4535), ('生', 4512), ('水', 4436), ('场', 4410), ('速', 4403), ('园', 4381), ('公', 4341), ('温', 4312), ('分', 4311), ('土', 4244), ('时', 4236), ('中航', 4151), ('维', 4143), ('面', 4140), ('区', 4120), ('图', 4118), ('计', 4117), ('分析', 4040), ('方', 4035), ('电', 3943), ('体', 3937), ('验', 3924), ('研', 3882), ('科技', 3875), ('行', 3854), ('航空', 3805), ('设', 3787), ('空', 3775), ('法', 3747), ('版', 3735), ('产', 3711), ('通', 3708), ('值', 3690), ('李', 3640), ('学校', 3614), ('天', 3614), ('组', 3581), ('课', 3564), ('出', 3562), ('载', 3546), ('全', 3541), ('作', 3524), ('长', 3503), ('北', 3493), ('系统', 3486), ('建设', 3485), ('程', 3454), ('影响

Figure 5: 正向最大匹配分词

```
[('学', 38898), ('中国民航', 30538), ('基', 11160), ('化', 10166), ('文', 9909), ('报', 9783), ('度', 9094), ('数', 8873),
 ('教', 8069), ('国', 7883), ('理', 7842), ('会', 7777), ('期', 7762), ('科', 7327), ('机场', 7231), ('新', 7035), ('服务
', 6965), ('业', 6934), ('中', 6800), ('道', 6715), ('学学', 6523), ('校', 6424), ('关', 6147), ('高', 6066), ('动',
6010), ('实', 5963), ('开', 5688), ('路', 5358), ('更', 5344), ('力', 5234), ('机', 5209), ('合', 5156), ('公开', 5124),
('性', 5090), ('工', 5057), ('建', 5055), ('分享', 5052), ('导', 4857), ('成', 4824), ('部', 4817), ('网', 4784), ('发',
4734), ('航', 4663), ('航空', 4628), ('号', 4610), ('进', 4541), ('水', 4436), ('场', 4410), ('速', 4403), ('园', 4403),
('公', 4341), ('温', 4312), ('分', 4311), ('士', 4256), ('时', 4236), ('中航', 4151), ('维', 4143), ('面', 4140), ('图',
4122), ('区', 4120), ('计', 4117), ('分析', 4040), ('方', 4035), ('生', 4034), ('验', 3968), ('电', 3943), ('体', 3937),
('研', 3918), ('行', 3906), ('科技', 3875), ('设', 3787), ('值', 3786), ('法', 3755), ('版', 3735), ('通', 3715), ('产',
3711), ('学生', 3683), ('李', 3640), ('天', 3621), ('组', 3581), ('出', 3574), ('课', 3568), ('载', 3546), ('全', 3541),
('作', 3540), ('学校', 3526), ('长', 3510), ('程', 3510), ('北', 3493), ('中国', 3488), ('系统', 3486), ('建设', 3485), ('
校友', 3382), ('室', 3361), ('影响', 3332), ('天津', 3322), ('主', 3290), ('热', 3288), ('标', 3267), ('测', 3252), ('教育
', 3178), ('评', 3125), ('家', 3120), ('碳', 3116), ('平台', 3115), ('参', 3101), ('民', 3089), ('制', 3071), ('政',
```

Figure 6: 反向最大匹配分词

```
状态转移概率矩阵： {'B': {'B': 0.0, 'M': 0.2021874953016417, 'E': 0.7978125031398688,
'S': 0.0}, 'M': {'B': 0.0, 'M': 0.3097289084054968, 'E': 0.6902710862737977, 'S':
0.0}, 'E': {'B': 0.8062246257665844, 'M': 0.0, 'E': 0.0, 'S': 0.1937769327275986},
'S': {'B': 0.9409770239453815, 'M': 0.0, 'E': 0.0, 'S': 0.05902296848660107}}
发射概率矩阵： {'B': {'学': 0.039874995916678466, '前': 0.0005257762588685855, '成':
0.004374209585616753, '国': 0.00782753294268261, '委': 0.0011542188878121019, '航':
0.008421753448267816, '民': 0.005433539858662631, '更': 0.0042248766836895804, '中':
```

Figure 7: 各种矩阵

```
test = '学校前身成立中国民革命军事委员会民航空局二民航学校更名中国民航空学院教育部批准更名中国民航大
 学'
path = HMMMM.viterbi(test , HMMMM.state_list , HMMMM.Pi_dic , HMMMM.B_dic , HMMMM.A_dic ,
 1e-6)
print(path)
```

```
['B', 'E', 'B', 'E', 'B', 'E', 'B', 'M', 'M', 'E', 'B', 'E', 'S', 'B', 'E', 'B', 'M',
 'E', 'B', 'E', 'B', 'M', 'E', 'B', 'E', 'B', 'E', 'B', 'M', 'M', 'E', 'S', 'B', 'E',
 'B', 'M', 'E', 'B', 'E', 'B', 'E', 'B', 'M', 'M', 'E', 'B', 'E']
```

Figure 8: viterbi算法

```
res = HMMMM.cut(test)
print(res)
```

```
['学校', '前身', '成立', '中国民革', '命军', '事', '委员', '会民航', '空局', '二民航', '学
校', '更名', '中国民航', '空', '学院', '教育部', '批准', '更名', '中国民航', '大学']
```

Figure 9: 分词结果

四、实验过程分析与建议

在本次实验中，我们成功地利用爬虫技术从特定的网站获取了相关的文本数据。这些数据对于后续的文本分析和处理至关重要。在获取到这些原始文本数据后，我们进行了预处理步骤，包括使用jieba分词工具对中文文本进行分词处理，以及利用正向最大匹配算法和反向最大匹配算法进行分词对比实验。实验结果显示，jieba分词工具凭借其强大的分词能力和丰富的词典资源，展现出了极高的分词准确率。它能够有效地识别出文本中的词汇，并对其进行准确的切分，为后续的文本分析提供了坚实的基础。与此同时，我们也对比了正向最大匹配算法和反向最大匹配算法在分词效果上的差异。实验发现，虽然两种算法在大多数情况下都能实现分词功能，但在处理歧义词时，反向最大匹配算法的表现更为出色。它能够更准确地识别出文本中的歧义词，并进行正确的切分，从而提高了分词的准确性。基于本次实验的结果，我们建议在后续的研究中进一步探索和优化分词算法。可以通过引入更多的语言学知识和语料库资源，来提高分词算法的准确性和效率。同时，也可以结合其他先进的分词技术，如基于深度学习的分词方法，来进一步提高分词效果，以满足不同领域对文本处理的需求。