



中南大学  
CENTRAL SOUTH UNIVERSITY

# 人民日报内容浅析

课程： Matlab 及其应用

姓名： 李维

班级： 信息管理与信息系统 1802

学号： 8102180608

学期： 2020 秋季

老师： 施 文

## 目录

<b>0</b>	<b>摘要 .....</b>	<b>- 1 -</b>
<b>1</b>	<b>数据爬取 .....</b>	<b>- 1 -</b>
1.1	相关代码 (PYTHON) .....	- 1 -
1.2	执行过程示例 .....	- 5 -
1.3	爬取结果 .....	- 6 -
<b>2</b>	<b>分词 .....</b>	<b>- 6 -</b>
2.1	主程序 .....	- 6 -
2.2	分词函数 .....	- 7 -
2.3	分词结果 .....	- 8 -
<b>3</b>	<b>文本清洗 .....</b>	<b>- 8 -</b>
3.1	初步清洗 .....	- 8 -
3.2	进一步清洗 .....	- 8 -
<b>4</b>	<b>分析 .....</b>	<b>11</b>
4.1	LDA 模型分析——主题数目选择 .....	11
4.2	LDA 模型分析数据 .....	13
4.3	每月报道量分析 .....	18
4.4	时序分析 .....	19
<b>5</b>	<b>分析结果 .....</b>	<b>22</b>
	<b>附录：代码说明 .....</b>	<b>23</b>

## 0 摘要

本次作业通过编写爬虫代码，爬取了人民日报 2020 年 1 月 1 号到 2021 年 1 月 13 号的所有新闻内容，并将其分词，使用 python 及 matlab 编写或自带的文本清洗函数进行清洗文本。最后，使用 LDA 模型等方法分析清洗过后的文本，最终得出对 2020 年人民日报报道新闻时事的回顾总结。

## 1 数据爬取

### 1.1 相关代码（python）

- 导入包

```
import requests
import bs4
import os
import datetime
import time

from gne import GeneralNewsExtractor
```

- 访问 url 网页，返回网页内容

```
def fetchUrl(url):

    headers = {
        'accept':
            'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8',
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36',
    }

    r = requests.get(url, headers=headers)
    r.raise_for_status()
    r.encoding = r.apparent_encoding
    return r.text
```

- 获取当天报纸各版面链接列表

```
def getPageList(year, month, day):  
    '''  
    参数: 年, 月, 日  
    '''  
    url = 'http://paper.people.com.cn/rmrb/html/' + year + '-' + month + '/'  
        + day + '/nbs.D110000renmrb_01.htm'  
    html = fetchUrl(url)  
    bsobj = bs4.BeautifulSoup(html, 'html.parser')  
  
    temp = bsobj.find('div', attrs = {'id': 'pageList'})  
    if temp:  
        pageList = temp.ul.find_all('div', attrs = {'class': 'right_title-  
            name'})  
    else:  
        pageList = bsobj.find('div', attrs = {'class': 'swiper-  
            container'}).find_all('div', attrs = {'class': 'swiper-slide'})  
  
    linkList = []  
  
    for page in pageList:  
        link = page.a["href"]  
        url = 'http://paper.people.com.cn/rmrb/html/' + year + '-' + month +  
            '/' + day + '/' + link  
        linkList.append(url)  
  
    return linkList
```

- 获取报纸某一版面的文章链接列表

```
def getTitleList(year, month, day, pageUrl):  
    '''  
    参数: 年, 月, 日, 该版面的链接  
    '''  
    html = fetchUrl(pageUrl)  
    bsobj = bs4.BeautifulSoup(html, 'html.parser')  
  
    temp = bsobj.find('div', attrs = {'id': 'titleList'})  
    if temp:  
        titleList = temp.ul.find_all('li')  
    else:  
        titleList = bsobj.find('ul', attrs = {'class': 'news-  
            list'}).find_all('li')
```

```
linkList = []

for title in titleList:
    tempList = title.find_all('a')
    for temp in tempList:
        link = temp["href"]
        if 'nw.D110000renmrb' in link:
            url = 'http://paper.people.com.cn/rmrb/html/' + year + '-' +
                month + '/' + day + '/' + link
            linkList.append(url)

return linkList
```

- 获取文章内容

```
def getContent(html):
    """
    参数: html 网页内容
    """
    bsobj = bs4.BeautifulSoup(html, 'html.parser')

    # 获取文章 标题
    title = bsobj.h3.text + '\n' + bsobj.h1.text + '\n' + bsobj.h2.text +
        '\n'
    # print(title)

    # 获取文章 内容
    pList = bsobj.find('div', attrs={'id': 'ozoom'}).find_all('p')
    content = ''
    for p in pList:
        content += p.text + '\n'
        # print(content)

    # 返回结果 标题+内容
    resp = title + content
    return resp
```

- 将文章内容保存在本地

```
def saveFile(content, path, filename):
    """
    参数: 要保存的内容, 路径, 文件名
    """
```

```
# 如果没有该文件夹, 则自动生成
if not os.path.exists(path):
    os.makedirs(path)

# 保存文件
with open(path + filename, 'w', encoding='utf-8') as f:
    f.write(content)
```

- 爬取《人民日报》网站 某年 某月 某日 的新闻内容并保存

```
def download_rmrh(year, month, day, destdir):
    """
    参数: 年, 月, 日, 文件保存的根目录
    """
    pageList = getPageList(year, month, day)
    for page in pageList:
        titleList = getTitleList(year, month, day, page)
        for url in titleList:
            # 获取新闻文章内容
            html = fetchUrl(url)
            content = getContent(html)

            # 生成保存的文件路径及文件名
            temp = url.split('_')[2].split('.')[0].split('-')
            pageNo = temp[1]
            titleNo = temp[0] if int(temp[0]) >= 10 else '0' + temp[0]
            # path = destdir + '/' + year + month + day + '/'
            path = destdir + '/'
            fileName = year + month + day + '-' + pageNo + '-' + titleNo +
                '.txt'

            # 保存文件
            saveFile(content, path, fileName)
```

- 获取日期列表

```
def gen_dates(b_date, days):
    day = datetime.timedelta(days=1)
    for i in range(days):
        yield b_date + day * i

def get_date_list(beginDate, endDate):
    """
    :param start: 开始日期
```

```
:param end: 结束日期
:return: 开始日期和结束日期之间的日期列表
"""

start = datetime.datetime.strptime(beginDate, "%Y%m%d")
end = datetime.datetime.strptime(endDate, "%Y%m%d")

data = []
for d in gen_dates(start, (end - start).days):
    data.append(d)

return data
```

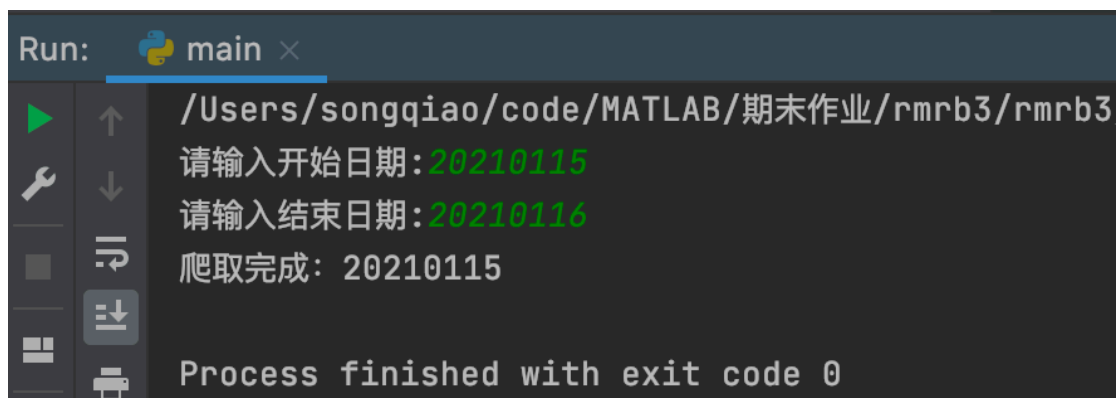
## • 爬虫执行

```
if __name__ == '__main__':
    """
    主函数：程序入口
    """

    # 输入起止日期，爬取之间的新闻
    beginDate = input('请输入开始日期:')
    endDate = input('请输入结束日期:')
    data = get_date_list(beginDate, endDate)

    for d in data:
        year = str(d.year)
        month = str(d.month) if d.month >= 10 else '0' + str(d.month)
        day = str(d.day) if d.day >= 10 else '0' + str(d.day)
        download_rmrh(year, month, day, 'data')
        print("爬取完成: " + year + month + day)
        # time.sleep(3)          # 怕被封 IP 爬一爬缓一缓，爬的少的话可以注释掉
```

## 1.2 执行过程示例



```
Run: main ×
/Users/songqiao/code/MATLAB/期末作业/rmrh3/rmrh3
请输入开始日期:20210115
请输入结束日期:20210116
爬取完成: 20210115
Process finished with exit code 0
```

图 1

## 1.3 爬取结果

共爬取从 2020-01-01 到 2021-01-13 共 27784 篇新闻报道。



图 2

## 2 分词

### 2.1 主程序

```
import os
import pandas as pd
from word_cut import *

root = "data/" # 数据集目录
name_list = os.listdir(root) # root 下文件
no_list = []
month_list = []
day_list = []
raw_list = []
'''
按年 月 日 内容 创建列表
'''
for name in name_list:
    no_list.append(name[:-4])
    month_list.append(name[4:6])
    day_list.append(name[6:8])
    tmp_list = [line.strip() for line in open(root + name, encoding="UTF-8")]
    tmp_str = "".join(tmp_list)
    raw_list.append(tmp_str)
'''
调用分词函数
'''
proc_list = cut_words(raw_list)
'''
```



将分词后的内容存储至列表

```
'''
no_df = pd.DataFrame(no_list, columns=['no'])
month_df = pd.DataFrame(month_list, columns=['month'])
day_df = pd.DataFrame(day_list, columns=['day'])
data_df = pd.DataFrame(proc_list, columns=['content'])
df = no_df.join(month_df)
df = df.join(day_df)
df = df.join(data_df)
print(df)
'''
```

将完成分词的内容存储至文件

```
'''
df.to_csv('proc_data.csv')
```

## 2.2 分词函数

```
import os
import jieba

def cut_words(raw_dataset: list) -> list:
    '''
    创建停用词列表
    '''
    stopwords_dict = os.listdir("stopwords/")
    stop_list = []
    for dic in stopwords_dict:
        stop_list.append([line.strip() for line in open('stopwords/' + dic,
            'r', encoding='utf-8').readlines())])
    '''
    分词
    '''
    cut_dataset = []
    for data in raw_dataset:
        cut_dataset.append(jieba.lcut(data))
    '''
    去除停用词
    '''
    clean_dataset = []
    for data in cut_dataset:
        clean_data = []
        for word in data:
            if word not in stop_list:
                clean_data.append(word)
```

```

clean_dataset.append(clean_data)
dataset = []
for data in clean_dataset:
    str_data = ' '.join(data)
    dataset.append(str_data)
return dataset

```

## 2.3 分词结果



```

Run: preprocess
/Users/songqiao/code/MATLAB/期末作业/rmrb3/rmrb3/venv/bin/python /Users/songqiao/code/MATLAB/期末作业
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/hw/3_mmdlkx1lq3gpm5fxsycs_w0000gn/T/jieba.cache
Loading model cost 0.709 seconds.
Prefix dict has been built successfully.
[['习近平', '主席', '新年贺词', '引发', '海外', '华侨', '华人', '热烈', '反响', '-', '-', '温暖', '人心', '
no ... content
0 20200101-02-06 ... 习近平 主席 新年贺词 引发 海外 华侨 华人 热烈 反响 -- 温暖 人心 振奋人心...
1 20200101-02-05 ... 习近平 主席 新年贺词 在 各地 干部群众 中 引发 热烈 反响 -- 只争朝夕 不负...
2 20200101-02-04 ... 用 汗水 浇灌 收获 以 实干 笃定 前行 -- 习近平 主席 二〇二〇年 新...
3 20200101-02-01 ... 在 全国政协 新年 茶话会 上 的 讲话 ( 2019 年 12 月 31 日 ) 同志 们...
4 20200101-02-03 ... 在 全国政协 新年 茶话会 上 的 讲话 ( 二〇一九年 十二月 三十一日 ) 同志 ...
5 20200101-02-02 ... 习近平 《 在 庆祝 澳门 回归祖国 二十周年 大会 暨 澳门特别行政区 第五届 政府 就职...
6 20200101-01-01 ... 新年 戏曲 晚会 在京举行 习近平 李克强 栗 战书 汪洋 王沪宁 赵乐际 韩正 王岐山 出...
7 20200101-01-02 ... 国家 主席 习近平 发表 二〇二〇年 新年贺词 2019 年 , 我们 用 汗水 浇...
8 20200101-01-03 ... 中 俄 两国 元首 互致 新年 贺电 中 俄 两国 总理 互致 新年 贺电 新华社 北京 1...
9 20200101-01-04 ... 全国政协 举行 新年 茶话会 习近平 发表 重要讲话 李克强 栗 战书 王沪宁 赵乐际 韩正...

[10 rows x 4 columns]

Process finished with exit code 0

```

图 3

## 3 文本清洗

### 3.1 初步清洗

在分词函数中使用停用词进行初步清洗。

### 3.2 进一步清洗

由于中文文本清洗不比英文文本清洗便利，停用词列表归根结底也是人工添置。故在清洗过程中无法彻底去除不需要文本。因此在停用词列表清洗过后，在 Matlab 中使用 processRMRB 函数进行清洗。最后制成云图观察，手动清洗无关内容。

- processRMRB.m

```

function [documents] = preprocessRMRB(textData)
% Convert the text data to lowercase.
cleanTextData = lower(textData);
% Tokenize the text.
documents = tokenizedDocument(cleanTextData);

```

```

% Erase punctuation.
documents = erasePunctuation(documents);
% Remove a list of stop words.
documents = removeStopWords(documents);
% Remove words with 1 or fewer characters,
% and words with 15 or greater characters.
documents = removeShortWords(documents,1);
documents = removeLongWords(documents,15);
% Lemmatize the words.
documents = addPartOfSpeechDetails(documents); % No definition
documents = normalizeWords(documents,'Style','lemma');
end

```

## • 制作云图并分析

```

% 加载示例数据。
data =
readtable("proc_data.csv",'Text
Type','string');
head(data)
% 从 content 字段中提取文本数据。
textData = data.content;
textData(1:10)
% 文本清洗
documents =
preprocessRMRB(textData);
documents(1:5)
% 绘制云图观察
bag = bagOfWords(documents)
figure
wordcloud(bag)

```

		month	day	content
1	25-...	5	525	"扶一把老...
2	19-...	8	819	"推动 社会主...
3	10-...	8	810	"用 高质量 发...
4	08-...	6	608	"图片 报道 俄...
5	14-...	10	1014	"生鲜 不 " 鲜 ...
6	16-...	11	1116	"健身房, 要...
7	16-...	1	116	"中国共产党 ...
8	04-...	11	1104	"第二届 中法 ...

```

ans = 10x1 string
"扶一把老百姓" (人民论...
"推动社会主义核心价值...
"用高质量发展吸引高素...
"图片报道俄罗斯多地日...
"生鲜不"鲜"也得能退换...
"健身房,要用服务留住...
"中国共产党第十九届中...
"第二届中法二轨高级别...
"民法典推动经济高质量...
"频"报平安"你在那边...

ans =
5x1 tokenizedDocument:

6 tokens: 2019 1109 108 900 5000 8000
2 tokens: 青少年 四中全会
1 tokens: 100
0 tokens:
0 tokens:

```

图 4



- 去除此类词后的结果



图 6

## 4 分析

#### 4.1 LDA 模型分析——主题数目选择

- 代码

## 1.Extract and Preprocess Text Data

```
% 加载示例数据。
```

```
% 从 event narrative 字段中提取文本数据。
```

```
filename = "proc_data.csv";
```

```
data = readtable(filename, 'TextType', 'string');
```

```
textData = data.content;
```

% 文本清洗

```
documents = preprocessRMRB(textData);
```

```
documents(1:5)
```

% 随机留出 10% 的 test 文档。

```
numDocuments = numel(documents):
```

```
cvp = cvpartition(numDocuments, 'HoldOut', 0.1);
```

```
documentsTrain = documents(cvp.training);
documentsValidation = documents(cvp.test);
% 从 training 中创建一个词袋。
% 删除总共出现不超过两次的单词。
% 删除所有不包含任何单词的文档。
bag = bagOfWords(documentsTrain);
bag = removeInfrequentWords(bag,2);
bag = removeEmptyDocuments(bag);
```

## 2.Choose Number of Topics

我们的目的是选择一定数量的主题，使模型困惑度最低。

但这不是唯一的考虑因素：适合大量主题的模型可能需要更长的时间才能收敛。综合二者的影响，需计算拟合优度和拟合时间。如果最佳主题数很高，那么您可能想要选择一个较低的值以加快拟合过程。

```
numTopicsRange = [5 10 15 20 40];
for i = 1:numel(numTopicsRange)
    numTopics = numTopicsRange(i);
    mdl = fitlda(bag,numTopics, ...
        'Solver','savb', ...
        'DataPassLimit',10,...
        'Verbose',0);
    [~,validationPerplexity(i)] = logp(mdl,documentsValidation);
    timeElapsed(i) = mdl.FitInfo.History.TimeSinceStart(end);
end
% 下图显示不同主题数量的困惑度和耗时。
% 在左轴上绘制困惑度，在右轴上绘制时间。
figure
yyaxis left
plot(numTopicsRange,validationPerplexity,'+-')
ylabel("Validation Perplexity")
yyaxis right
plot(numTopicsRange,timeElapsed,'o-')
ylabel("Time Elapsed (s)")
legend(["Validation Perplexity" "Time Elapsed (s)"],'Location','southeast')
xlabel("Number of Topics")
```

## • 结果

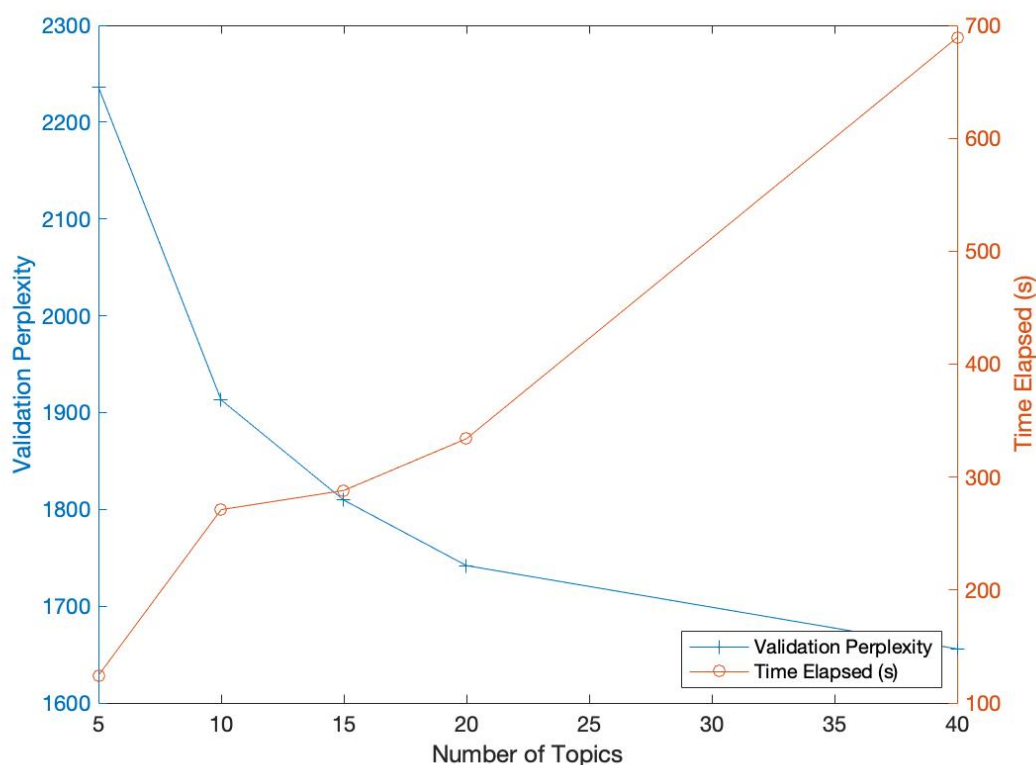


图 7

如图所示，设置 15 个左右的主题较为合适。其困惑度较低，且耗时合理。使用不同的主题数，会发现增加主题数量可以更好地拟合，但是拟合模型需要更长的时间才能收敛。

## 4.2 LDA 模型分析数据

### • 载入文件及文本清洗

#### 1. Load and Extract Text Data

```
% Load the example data
data = readtable("proc_data.csv", 'TextType', 'string');
head(data)

% Extract the text data from the field event_narrative
textData = data.content;
textData(1:10)
```

#### 2. Prepare Text Data for Analysis

```
% Use the example preprocessing function
% preprocessWeatherNarratives
% to prepare the text data.
```

```
documents = preprocessRMRB(textData);
documents(1:5)
```

ans = 8x5 table

	no	month	day	content
1	200525-...	5	525	"扶一把老
2	200819-...	8	819	"推动社会主
3	200810-...	8	810	"用高质量发
4	200608-...	6	608	"图片报道
5	201014-...	10	1014	"生鲜不“鲜
6	201116-...	11	1116	"健身房，要
7	200116-...	1	116	"中国共产党
8	201104-...	11	1104	"第二届中法

图 8

ans = 10x1 string

```
"“扶一把老百姓”（人民论...
"推动社会主义核心价值观在家...
"用高质量发展吸引高素质人才...
"图片报道俄罗斯多地日增新冠...
"生鲜不“鲜”也得能退换...
"健身房，要用服务留住消费...
"中国共产党第十九届中央纪律检查...
"第二届中法二轨高级别对话视...
"民法典推动经济高质量发展（...
"“频”报平安“你在那边怎...
```

图 9

## • LDA 模型分析

### 3. Fit LDA Model

```
% Create a bag-of-words model from the tokenized documents.
```

```
bag = bagOfWords(documents)
```

```
% 从词袋中删除出现次数不超过两次的词。
```

```
% 从词袋中删除所有不包含单词的文档。
```

```
bag = removeInfrequentWords(bag,2);
```

```
bag = removeEmptyDocuments(bag)
```

```
% Fit an LDA model with 60 topics.
```

```
numTopics = 15;
```

```
mdl = fitlda(bag,numTopics);
```

bag =

```
bagOfWords - 属性:
```

```
Counts: [27739x98237 double]
```

```
Vocabulary: [1x98237 string]
```

```
NumWords: 98237
```

```
NumDocuments: 27739
```

bag =

```
bagOfWords - 属性:
```

```
Counts: [27731x53436 double]
```

```
Vocabulary: [1x53436 string]
```

```
NumWords: 53436
```

```
NumDocuments: 27731
```

图 11

lative nge in og(L)	Training perplexity	Topic concentration	Topic concentratio iterations
348e-02	3.396e+03	3.750	
520e-03	1.829e+03	3.750	
642e-04	1.788e+03	3.750	
121e-04	1.782e+03	3.750	
149e-04	1.777e+03	3.750	
085e-04	1.773e+03	3.750	
248e-04	1.769e+03	3.750	
749e-04	1.765e+03	3.750	
604e-04	1.762e+03	3.750	
799e-04	1.758e+03	3.750	
175e-04	1.755e+03	3.750	
881e-04	1.752e+03	4.234	
	1.749e+03	4.335	

1

图 10



## • 查看出现频率最高词(前 4 个主题)

### 4. Visualize Topics Using Word Clouds

% 利用云图查看出现率最高的词。

```
figure;
for topicIdx = 1:4
    subplot(2,2,topicIdx)
    wordcloud mdl,topicIdx);
    title("Topic: " + topicIdx)
end
```

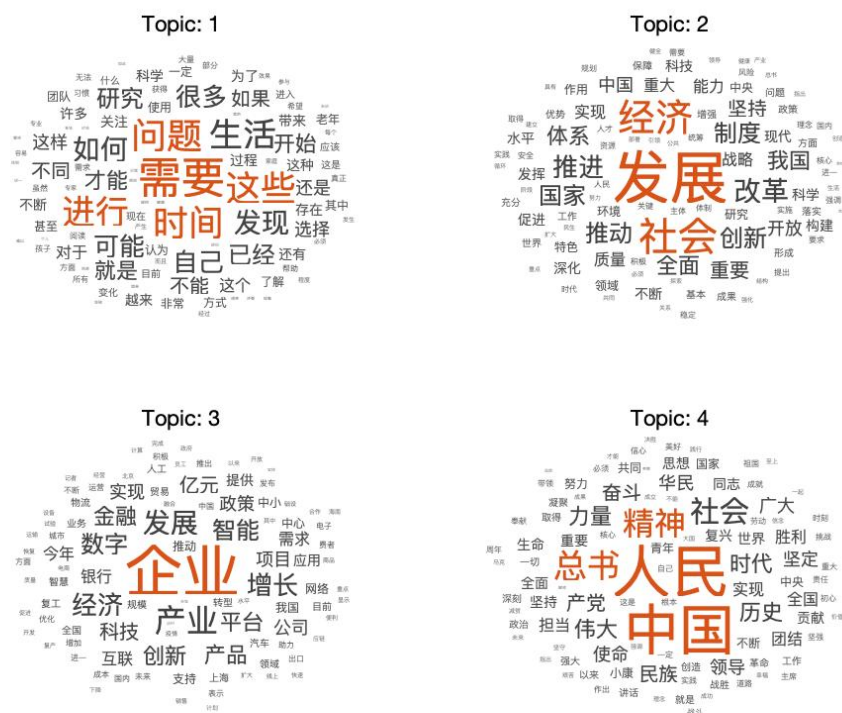


图 12

## • 查看出现频率最高词(15 个主题)

% 利用云图查看出现率最高的词。

```
figure;
for topicIdx = 1:15
    subplot(4,4,topicIdx)
    wordcloud mdl,topicIdx);
    title("Topic: " + topicIdx)
end
```

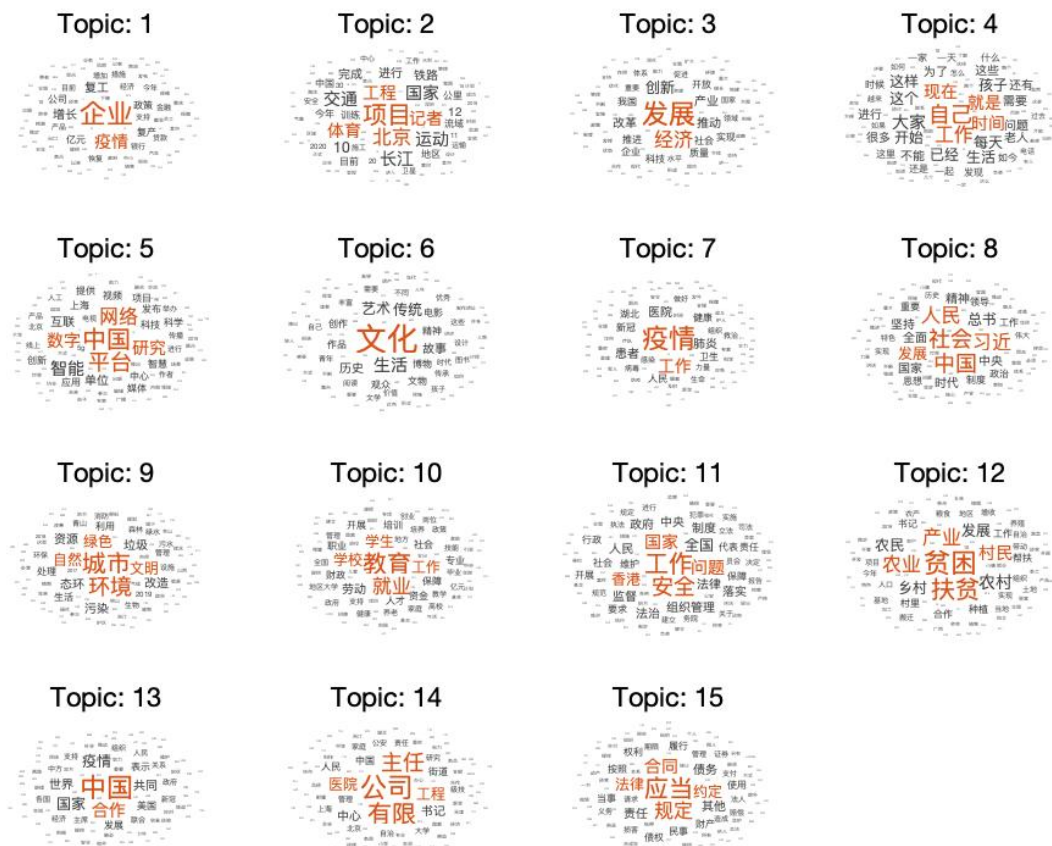


图 13

## • 绘制出现概率直方图

### 5. View Mixtures of Topics in Documents

% 绘制出现概率直方图

```
newDocument = tokenizedDocument("A tree is downed outside Apple Hill Drive,  
Natick");
```

```
topicMixture = transform mdl, newDocument);
```

```
figure
```

```
bar(topicMixture)
```

```
xlabel("Topic Index")
```

```
ylabel("Probabilitiy")
```

```
title("Document Topic Probabilities")
```

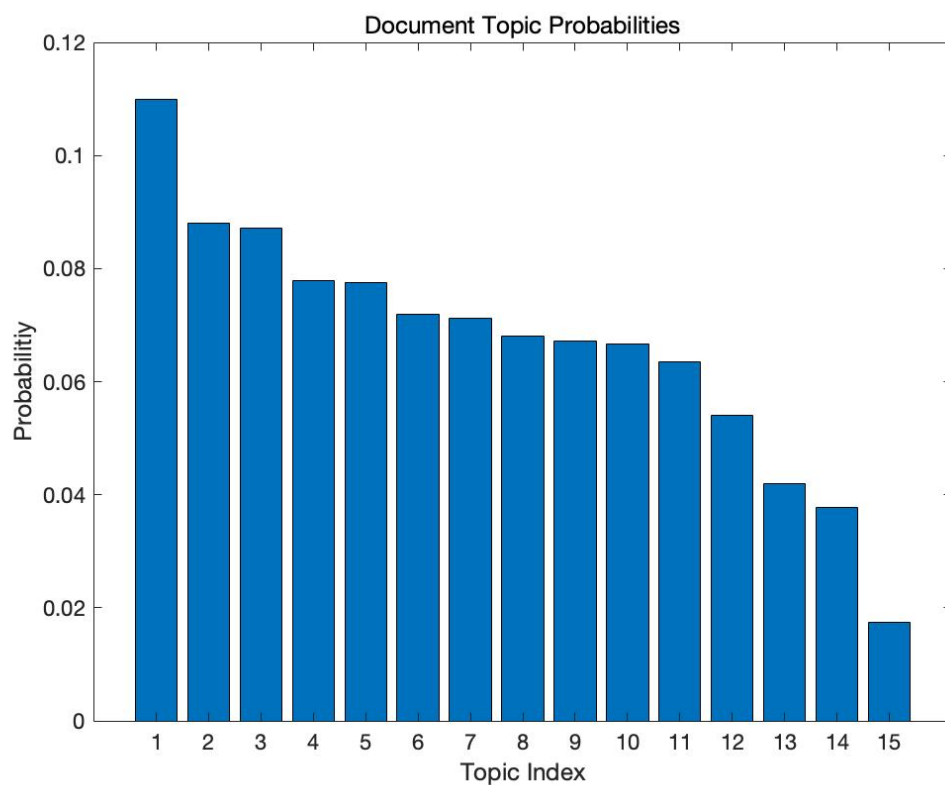


图 14

- 前 10 篇报道各主题出现概率

% 前十个 documents 各个 topics 出现概率

```
figure
topicMixtures = transform mdl, documents(1:10));
barh(topicMixtures(1:10,:), 'stacked')
xlim([0 1])
title("Topic Mixtures")
xlabel("Topic Probability")
ylabel("Document")
```

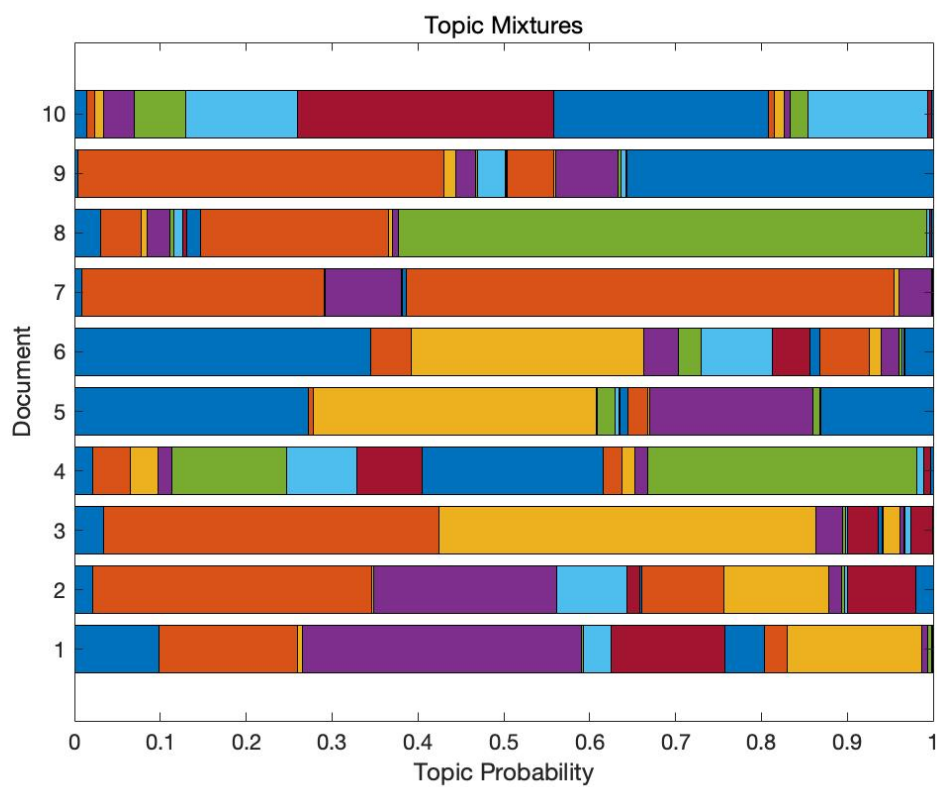


图 15

### 4.3 每月报道量分析

% 使用直方图显示。

```
data.month = categorical(data.month);  
figure  
h = histogram(data.month);  
xlabel("Class")  
ylabel("Frequency")  
title("Class Distribution")
```

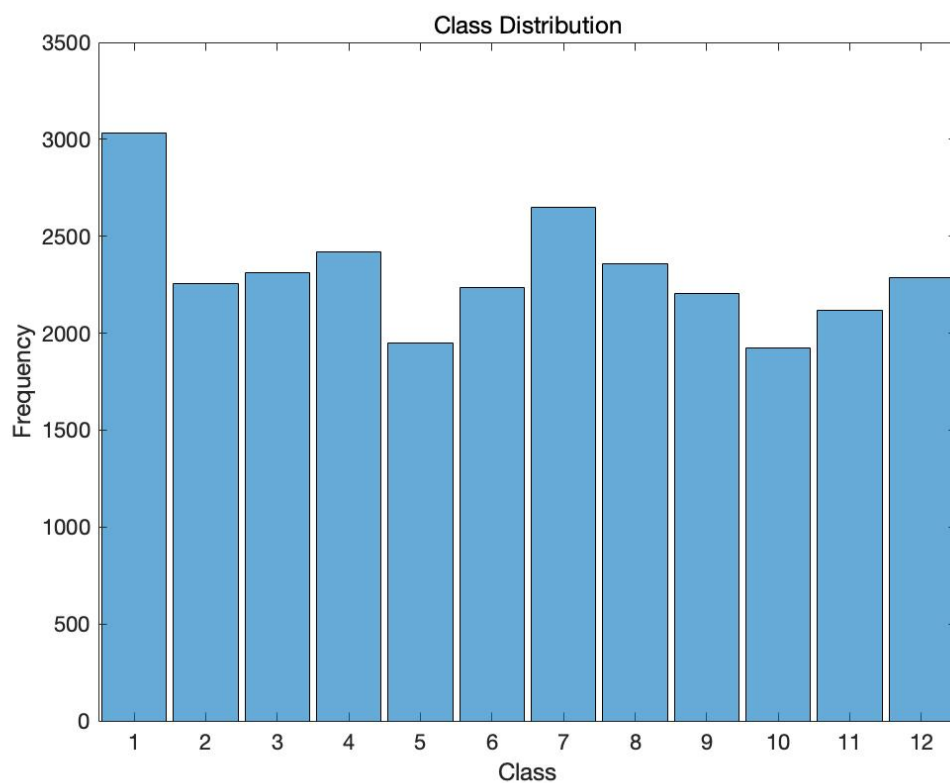


图 16

## 4.4 时序分析

### • 部分代码

#### 1. Load and Extract Text Data

```
% Load the example data
data = readtable("proc_data_row.csv", 'TextType', 'string');
head(data)
% Extract the text data from the field event_narrative
textData = data.content;
textData(1:10)
```

#### 2. Prepare Text Data for Analysis

```
% Use the example preprocessing function
% preprocessWeatherNarratives
% to prepare the text data.
documents = preprocessRMRB(textData);
documents(1:5)
% 将报告中每月关键字绘制成云图进行比较。
figure
labels = data.month;
subplot(1,4,1)
```

```

idx = labels == 1;
wordcloud(documents(idx), 'Color', 'blue', 'Shape', 'rectangle', 'Box', 'on');
title("一月")
subplot(1,4,2)
idx = labels == 2;
wordcloud(documents(idx), 'Color', 'red', 'Shape', 'rectangle', 'Box', 'on');
title("二月")
subplot(1,4,3)
idx = labels == 3;
wordcloud(documents(idx), 'Color', 'magenta', 'Shape', 'rectangle', 'Box', 'on');
title("三月")
subplot(1,4,4)
idx = labels == 4;
wordcloud(documents(idx), 'Shape', 'rectangle', 'Box', 'on');
title("四月")

```

## • 结果

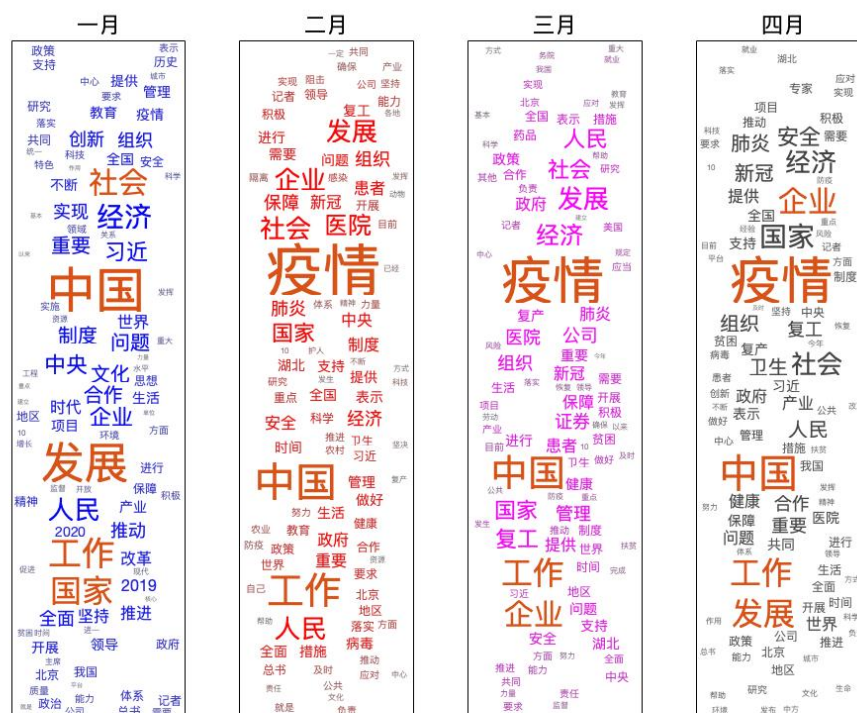


图 17





图 18

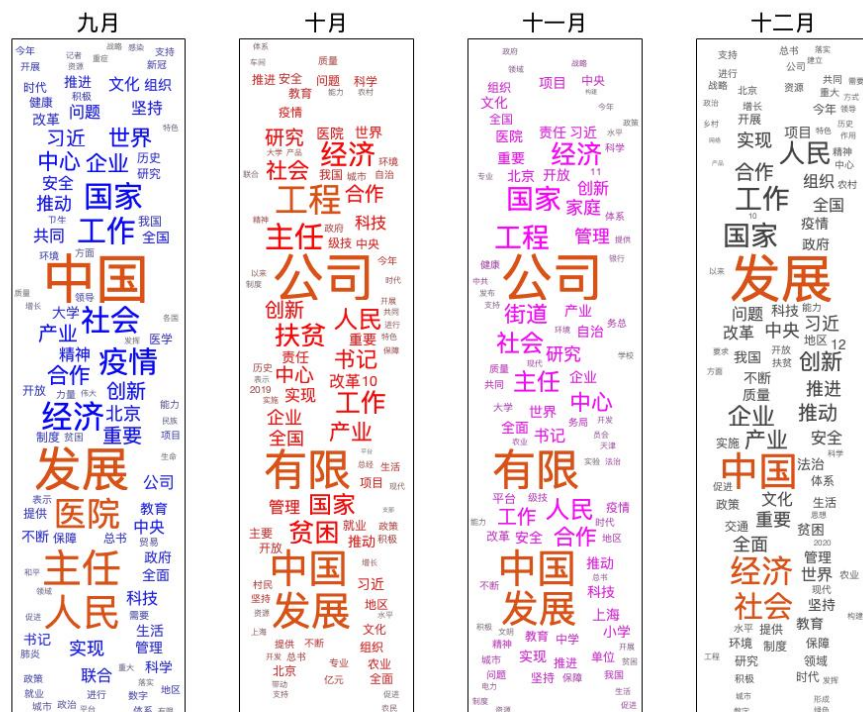


图 19

## 5 分析结果

- 1) 从图 13 可以看出, 在 LDA 模型训练下得出的 15 个主题大致包括: 1. 疫情影响企业的运作; 2. 各类项目的推动; 3. 经济的发展与创新; 4. 人民生活; 5. 互联网发展; 6. 文化发展; 7. 疫情情况; 8. 社会发展; 9. 城市建设; 10. 教育问题; 11. 国家安全问题; 12. 扶贫; 13. 疫情对国际社会影响; 14. 公司发展; 15. 法律问题。这些主题映射了 2020 年我国重点社会时事, 较全面地概括了社会生活的各个方面。根据图 14, 15 个主题出现频次依次递减, 媒体对社会时政报道数量一定程度上可反映该问题的重要性。
- 2) 从图 17、图 18、图 19 可以分析出每月新闻的主题。一月是一年的起始, 报道重心在国家经济、社会、文化各方面的发展以及全年工作布置、展望。而二月到五月, 报道内容大多围绕疫情展开。不同的是, 二月主要关注的是新冠肺炎的治疗、各个单位关于防控新冠开展的工作, 三月四月出现对于复工、经济发展等的报道, 五月经济发展复苏占据了更重要的位置。六月开始, 疫情相关报道减少。与其他月份不同, 七月有关香港、涉及国家安全问题的报道增多。八月、九月新闻主题仍聚焦于疫情防控以及经济复苏、发展, 其中经济发展占较多内容。十月到十一月, 疫情相关报道基本消失, 社会、管理、公司、发展等为主要关键词。其中, 十月报道了较多扶贫工作。
- 3) 从图 16 可以看出, 每月报道量变化较为平缓(1 月显著较多是由于多出 2021 年 13 天, 约多出 1000 条)。其中, 5 月、10 月报道量相对较少, 7 月报道较多。可见, 结合各月主题, 5 月为复工早期阶段, 疫情依旧需要小心控制, 逐步复工, 以保证人民复工后的防护意识。7 月出现较多香港问题报道, 由此可见, 国家危害国家安全行为的抵制以及极高重视。



## 附录：代码说明

- 数据爬取代码文件 rmrB/main.py
- 爬取新闻报道存储位置 rmrB/data/
- 分词主程序代码文件 rmrB/preprocess.py
- 分词函数代码文件 rmrB/word\_cut.py
- 分词后文件位置 rmrB/proc\_data.csv
- 停用词文件位置 rmrB/stopwords/
- 进一步文本清洗\_python 部分代码文件 /rmrB/deep\_preprocess.py
- 进一步文本清洗观察/rmrB\_wordswashing.mlx
- 进一步文本清洗\_matlab 文本清洗函数 /preprocessRMRB.m
- LDA 模型主题数确定 /rmrB\_LDA\_NUM.mlx
- LDA 模型分析数据 /rmrB\_LDA.mlx
- 文本时序分析 /rmrB\_time\_content.mlx
- 其他分析 /rmrB.mlx