

“智慧政务”中的文本挖掘应用

摘要

摘要内容：近年来，网络问政平台逐步成为政府了解社情民意、解决社民难题的重要渠道，但由于各类社情民意相关的文本数据量不断攀升，以往以人工处理数据为主的相关部门也逐渐力不从心。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。为此，我们通过对与社情民意相关的文本的数据进行评估和分析，并由此建立相关模型，获取相关重要信息以提高政府了解并解决民难的效率。

针对问题一，在基于 Python 的 Anaconda 软件下，为提高分类模型处理数据的效率

和分类的准确度，运用 pandas 的 series.str.contains 函数将附件一的数据按照“一

级分类”的类别分为七大数据类。利用 jieba 中文分词工具分别对七大数据类中的“留言主题”列的数据进行分词并运用 value_counts() 函数统计各类数据关键词。选取合适的关键词作为分类的依据，然后分别在七大数据类中随机抽取合适的数据量作为测试集（七大测试集），将分类数据与相应的原数据集进行比较，利用 pandas 的 DataFrame 中 drop_duplicates() 函数对数据进行去重，最后统计测试集和训练集，计算出七组查准率和查全率，并根据 F-Score 计算出七个 F1，取其平均值作为对分类模型的评价，其平均值约为 80%（数据量越大，F1 越高，分类模型越好）。

针对问题二，在基于 Python 的 Anaconda 软件下，运用 pandas.DataFrame.sort_values 函数对数据进行快速和综合排序，运用 pandas.DataFrame.diff 函数中 SequenceMatcher() 方法对 arr1 中的数据进行相似度计算，并选取合适的相似度对数据进行归类，建立字典将数据数值化并存入，其中相似度高的数据映射为相同值，然后运用 value_counts() 函数对 data4 中“问题 ID”数据列进行统计，并计算热度指数，比较热度指数，筛选出排名前 5 的热点问题，最后运用 to_csv() 方法将数据导出，另存为获得“热点问题表.xls”和“热点问题留言明细表.xls”文件。

针对问题三，在基于 Python 的 Anaconda 软件下，首先用 pandas 对数据进行清洗，删除因空格而导入的“/n”“/t”字符，然后利用 pandas 的 corr() 方法计算相关系数矩阵，分析数据的相关性；然后运用 pandas.DataFrame.diff 函数中 SequenceMatcher() 方法对 arr1 中的数据进行相似度计算，分析数据的可解释性；通过比较“留言时间”和“答复时间”分析数据的时效性；最后通过遍历方法查找数据中是否存在空值分析数据的完整性

关键词：pandas jieba 中文分词工具 drop_duplicates() 函数 SequenceMatcher() 方法
数据清洗

Application of Text Mining in "Smart Government"

Abstract

In recent years, the online political platform has gradually become an important channel for the government to understand the social situation and public opinion and solve the social problems. However, due to the increasing amount of text data related to all kinds of social situation and public opinion, the previous relevant departments mainly dealing with data manually are gradually unable to cope with it. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of smart government system based on natural language processing technology has become a new trend of social governance innovation and development, which has a great role in promoting the management level and efficiency of the government. For this reason, we evaluate and analyze the text data related to social situation and public opinion, and then establish relevant models to obtain relevant important information to improve the efficiency of government understanding and solving the people's difficulties.

To solve the problem 1, in order to improve the efficiency of data processing and classification accuracy of the classification model under the Python based Anaconda software, we use the `series.str.contains` function of pandas to divide the data in Annex 1 into seven data categories according to the category of "first level classification". Use the Chinese word segmentation tool of Jieba to segment the data in the "message subject" column of the seven data categories respectively, and use the `value_counts()` function to count all kinds of data keywords. Select the appropriate key words as the basis of classification, and then randomly select the appropriate amount of data in the seven data classes as the test set (seven test sets), compare the classification data with the corresponding original data set, use the `drop_duplicates()` function in the dataframe of pandas to de duplicate the data, and finally count the test set and training set, and calculate the seven sets of precision and check According to F-score, seven F1 are calculated, and the average value is taken as the evaluation of the classification model. The average value is about 80% (the larger the amount of data, the higher F1, the better the classification model).

In order to solve the second problem, under the Python based Anaconda software, we use the `pandas.dataframe.sort_values` function to sort the data quickly and comprehensively. We use the `sequencematcher()` method in the `pandas.dataframe.diff` function to calculate the similarity of the data in the Arr1, and select the appropriate similarity to classify the data, and build a dictionary to store the data, which is similar The data with high degree is mapped to the same value, and then the data column of "问题 ID" in Data4 is counted by using the `value_counts()` function, and the heat index is calculated, compared with the heat index, and the top 5 hot issues are screened out. Finally, the data is exported by using the `to_csv()` method, and saved as the "热点问题表.xls" and "热点问题留言明细表.xls" files.

In order to solve the third problem, in the Python based Anaconda software, first use pandas to clean the data, delete the "/" "n" character imported due to the space, then use `pandas.Corr()` method to calculate the correlation coefficient matrix,

analyze the data correlation; then use the `sequencematcher ()` method in `pandas.DataFrame`. `DiffLib` function to calculate the similarity of the data in `Arr1`, analyze the interpretability of the data; analyze the timeliness of the data by comparing the "留言时间" and "答复时间"; finally, find out whether there is the integrity of the null value analysis data through the traversal method

Keywords: `pandas` `jieba` 中文分词工具 `drop_duplicates ()` 函数 `SequenceMatcher ()` 方法
数据清洗

目录

1. 问题重述

1.1 问题背景

1.2 问题概述

2. 问题的分析方法与过程

2.1 问题一分析方法与过程

2.2 问题二分析方法与过程

2.3 问题三分析方法与过程

2.3.1 对相关性、完整性、时效性、可解释性的理解

2.3.2 如何用程序实现对数据相关性、完整性、时效性、可解释性的分析

2.3.2.1 实现对数据相关性的分析

2.3.2.2 实现对数据完整性的分析

2.3.2.3 实现对数据时效性的分析

2.3.2.4 实现对数据可解释性的分析

2.3.3 算法实现

3. 结果分析

4. 结论

5. 参考文献

1. 问题重述

1.1 问题背景：

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

1.2 问题概述：

1.2.1 问题一：参考附件 1 提供的内容分类三级标签体系，对留言进行分类，并根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。

1.2.2 问题二：根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评价指标，并给出排名前 5 的热点问题的文件“热点问题表.xls”和相应热点问题对应的留言信息的文件“热点问题留言明细表.xls”。

1.2.3 问题三：针对附件 4 相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出一套评价方案。

2. 问题的分析方法与过程

2.1 问题一的分析方法与过程

2.1.1 将附件 1 和附件 2 另存为 annex1.csv 和 annex2.csv 文件

2.1.2 基于 Python 的 Anaconda 软件下，通过 pandas 的 .read_csv() 方法将 annex2.csv 文件导入，并分别用 .drop() 和 .dropna() 函数删除文件中的原标题行和任意含有 Nan 值的行。设定“message_number”“message_users”“message_subject”“message_time”“message_details”“first_classification”为新标题行。通过 pandas 的 .unique() 函数统计发现“一级分类”列中有七类。生成数据集 data2。

2.1.3 为提高分类模型处理数据的效率和分类的准确度，运用 pandas 的 series.str.contains 函数将附件一的数据按照“一级分类”的类别分为“城乡建设”“环境保护”“交通运输”“教育文体”“劳动和社会保障”“商贸旅游”“卫生计生”七大数据类。并将七大数据类的数据通过 .to_csv() 方法分别导入“data_cj.csv”“data_hb.csv”“data_jy.csv”“data_jw.csv”“data_ls.csv”“data_sl.csv”“data_wj.csv”七个文件中

2.1.4 以下以“城乡建设”数据类为例，运用 pandas 的 .unique() 函数将 data_cj.csv 文件中的“message_subject”列数据以数组形式输出，存储在数组 arr_ms。利用 jieba 中文分词工具进行分词断句，并将断句结果存放在数组 arr1 中，并运用 value_counts() 函数统计各类数据关键词及其出现的次数，通过 pandas 的 .to_csv() 方法存入到 keys_cj.csv 文件中

2.1.5 在 keys_cj.csv 文件中选取合适的关键词作为分类依据，并将这些关键词设置到数组 arr2 中，建立分类模型。

2.1.6 运用 pandas 的 .sample(n=*,axis=0) 函数在 data_cj 数据类中随机抽取 90 列作为测试集 data_test。

2.1.7 将 data_test["message_subject"] 中数据以数组方式输出并存入数组 arr3

中。将 arr2 中分类数据与 arr3 中的原数据进行比较，输出重复（正确）的数据并存入数组 arr4 中。

2.1.8 将 arr4 中数据通过 pandas 的 DataFrame() 函数生成地方，再利用 pandas 的 DataFrame 中 drop_duplicates() 方法对 arr4 中数据进行去重处理，获得新 DataFrame 数据 df1。

2.1.9 通过对 data_test 和 data_cj 数据集使用 len() 函数分别求出测试集数据总数 data_test_sum 和训练集数据总数 data_train_sum，再次使用 len() 函数求出测试集中预测正确的数据个数 data_test_zheng（因为训练集选取的是 data_cj 数据，因此训练集中正确的数据个数 data_train_zheng 与训练集数据总数相等）。

2.1.10 计算一级分类为“城乡建设”的查准率 P_cj，查全率 R_cj 和 F-score: F1_cj, 其中 R （查全率）=（检出的正确数据数量/检索的数据总量） $\times 100\%$ ， P （查准率）=（检出的正确数据数量/检出的数据总量） $\times 100\%$ 。因此，以“城乡建设”数据为基础的查准率 $P_{cj} = \text{data_test_zheng} / \text{data_test_sum} \approx 0.688889$ ，以“城乡建设”数据为基础的查全率 $R_{cj} = \text{data_test_zheng} / \text{data_train_sum} \approx 0.613861$ 。

2.1.11 求出查全率和查准率之后，根据公式 $F1 = (2 * P * R) / (P + R)$ 求出以“城乡建设”数据为基础的 $F1_{cj} = (2 * P_{cj} * R_{cj}) / (P_{cj} + R_{cj})$ ， $F1_{cj} \approx 0.649215$ 。

2.1.12 同理按照“城乡建设”数据类求值的方法，分别求出另外六大数据类的查全率、查准率和 F1:

以“环境保护”数据类为基础的查准率 $P_{hb} \approx 0.900000$ ，

以“环境保护”数据类为基础的查全率 $R_{hb} \approx 0.545455$ ，

以“环境保护”数据类为基础的 $F1_{hb} \approx 0.679245$ ；

以“交通运输”数据类为基础的查准率 $P_{jy} \approx 0.975000$ ，

以“交通运输”数据类为基础的查全率 $R_{jy} \approx 0.709091$ ，

以“交通运输”数据类为基础的 $F1_{jy} \approx 0.821053$ ；

以“教育文体”数据类为基础的查准率 $P_{jw} \approx 0.937500$ ，

以“教育文体”数据类为基础的查全率 $R_{jw} \approx 0.781250$ ，

以“教育文体”数据类为基础的 $F1_{jw} \approx 0.852273$ ；

以“劳动和社会保障”数据类为基础的查准率 $P_{ls} \approx 1.000000$ ，

以“劳动和社会保障”数据类为基础的查全率 $R_{ls} \approx 0.865385$ ，

以“劳动和社会保障”数据类为基础的 $F1_{ls} \approx 0.927835$ ；

以“商贸旅游”数据类为基础的查准率 $P_{sl} \approx 1.000000$ ，

以“商贸旅游”数据类为基础的查全率 $R_{sl} \approx 0.833333$ ，

以“商贸旅游”数据类为基础的 $F1_{sl} \approx 0.909091$ ；

以“卫生计生”数据类为基础的查准率 $P_{wj} \approx 0.875000$ ，

以“卫生计生”数据类为基础的查全率 $R_{wj} \approx 0.603448$ ，

以“卫生计生”数据类为基础的 $F1_{wj} \approx 0.714286$ ；

2.1.13 计算七大数据类的平均 F1: $F1_{pj} \approx 0.8$ （数据量越大，F1 越高，分类模型越好）

2.2 问题二的分析方法与过程

2.2.1 将附件 3 另存为 annex3.csv 文件

2.2.2 基于 Python 的 Anaconda 软件下，通过 pandas 的 read_csv() 方法将 annex3.csv 文件导入，并分别用 .drop() 和 .dropna() 函数删除文件中的原标题行和任意含有 Nan 值的行。设定“message_number”“message_users”“message_subject”

“message_time” “message_details” “agree_num” , “disagree_num” 为新标题行。生成数据集 data2。

2.2.3 运用 pandas.DataFrame.sort_values 函数对“message_subject”列数据进行快速排序, 生成数据集 data3, 对数据的分析更加直观。

2.2.4 将 data3 数据集中“message_subject”类的的数据以数组形式输出, 存储到 arr1 中

2.2.5 运用 pandas.DataFrame.diff 函数中 SequenceMatcher() 方法对 arr1 中的数据进行相似度计算, 并选取合适的相似度将类似的数据存储到数组 arri 中, 本例中选择数据相似度大于 0.2 的。

2.2.6 根据数据集 arri 中数据, 建立字典 dict, 并将 data2 中数据数值化存入, 相似度高的数据映射为同一数值, 并使用 map() 函数在 data2 数据中新建“问题 ID”列用以表示对应的数值

2.2.7 运用 pandas.DataFrame.sort_values 函数对 data2 数据中“问题 ID”列数据进行快速排序

2.2.8 利用.rename() 函数更改各类数据列名, 对应如下:

```
"message_number": "留言编号",  
"message_users": "留言用户",  
"message_subject": "留言主题",  
"message_time": "留言时间",  
"message_details": "留言详情",  
"agree_num": "点赞数",  
"disagree_num": "反对数"。
```

2.2.9 运用 to_csv() 方法将排序好的 data4 数据导出, 导出的文件名为“热点问题留言明细表.csv”。

2.2.10 进入“热点问题留言明细表.csv”文件, 删除没有列名的首列, 并将“问题 ID”列移动到最前面, 保存文件, 另存为“热点问题留言明细表.xls”文件。

2.2.11 运用 value_counts() 函数对 data4 中“问题 ID”数据列进行统计, 并计算热度指数(统计数/数据总数)。

2.2.12 运用 pandas.DataFrame.sort_values 函数对 data4 数据的“问题 ID”和“留言时间”列数据进行综合排序, 生成数据 data5, 以便于统计热点问题的时间发生范围

2.2.12 运用 pandas.DataFrame 函数, 结合 data5 数据, 使用包含列表的字典 dict 创建 df, 筛选出排名前 5 的热点问题并存入 df 中

字典 dict 内容如下:

```
'热度排名': ['1', '2', '3', '4', '5'],  
'问题 ID': [1, 2, 8, 3, 4],  
'热度指数': [0.333333, 0.166667, 0.033333, 0.033333, 0.033333],  
'时间范围': ['2019-07-21 至 2019-09-25 ', '2017-06-08 至 2019-11-22', '2019-05-14 ', '2018-11-15 ', '2019-09-08 '],  
'地点/人群': ['A5 市 A5 区魅力之城小区', 'A 市经济学院学生', 'A 市', 'A 市人才 app', '西地省'],  
'问题描述': ['小区临街餐饮店油烟噪音扰民', '学校强制学生去定点企业实习', '请问 A 市什么时候能普及 5G 网络?', '在 A 市人才 app 上申请购房补贴为什么通不过', '希望西地省把抗癌药品纳入医保范围']
```


2.2.13 运用 `to_csv()` 方法将 `df` 数据导出，文件名为“热点问题表.csv”。

2.2.14 将“热点问题表.csv”文件另存为“热点问题表.xls”文件

2.3 问题三的分析方法与过程

2.3.1 对相关性、完整性、时效性、可解释性的理解

2.3.1.1 相关性：数据的相关性是指数据与特定的应用和领域有关。与数据相关的应用场景一般有，比如进行数据挖掘或构造模型预测时，需要采集相关的数据。例如考虑构造一个模型，预测交通事故发生率。如果忽略了驾驶员的年龄和性别信息，那么除非这些信息可以间接地通过其他属性得到，否则模型的精度可能是有限的。这种情况下，我们就需要尽量采集全面的相关的数据信息。

另外一个相关性的质量问题表现在相同的数据，在不同的应用领域中，相关性也是不一样的。例如，对于某个公司的大型客户数据库，由于时间和统计的原因，顾客地址列表的正确性为 80%，其他地址可能过时或不正确。当市场分析人员访问公司的数据库，获取顾客地址列表时，基于目标市场营销考虑，市场分析人员对于该数据库的准确性满意度较高。而当销售经理访问该数据库时，由于地址的缺失和过时，对该数据库的满意度较低。

2.3.1.2 完整性：完整性是指信息具有一个实体描述的所有必需的部分。在传统关系型数据库中，完整性通常与空值（NULL）有关。空值是指缺失或不知道具体的值，可能是一条记录中的某个属性缺失，也可能是整条记录都丢失。

2.3.1.3 时效性：数据的时效性是指有些数据会随时间而变化的，这些数据收集后就开始老化，使用老化后的数据进行数据分析、数据挖掘，将会产生不同的分析结果。如商品推荐。顾客的当时购买行为或 Web 浏览行为称为快照，它只代表有限时间内的真实情况。如果数据已经过时，则基于它的模型和模式也就已经过时，所以进行商品推荐需要采集当前的数据进行分析 and 推荐。在这种情况下，我们需要考虑重新采集数据信息，及时对数据进行更新。

2.3.1.3 可解释性：数据的可解释性，也称为可读性，是指数据被人理解的难易程度，如果数据具有解释性或包含有注释性信息，而且数据书写规范，则数据的可解释性越高。相反如果数据晦涩难懂就根本不具备分析的条件。

2.3.2 如何用程序实现对数据相关性、完整性、时效性、可解释性的分析

2.3.2.1 实现对数据相关性的分析

(1) 协方差：如果协方差大于 0，则表明正相关，若协方差小于 0，则为负相关，协方差为 0 时，X 和 Y 不相关；

$$\text{公式: } cov(X, Y) = \frac{\sum_{i=1}^n (X^i - \bar{X})(Y^i - \bar{Y})}{n-1}$$

(2) pearson 相关系数：取值范围为 $[-1, 1]$ ，其绝对值在 0.8-1 之间极强相关，0.6-0.8 强相关，0.4-0.6 中等强度相关，0.2-0.4 弱相关，0-0.2 极弱相关或无关。

$$\text{公式: } r = \frac{\sum_{i=1}^n (x^i - \bar{x})(y^i - \bar{y})}{\sqrt{\sum_{i=1}^n (x^i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y^i - \bar{y})^2}}$$

2.3.2.2 实现对数据完整性的分析

通过遍历查找空值，并计算空值所占所有数据比例。

2.3.2.3 实现对数据时效性的分析

通过比较“留言时间”和“答复时间”的先后分析数据的时效性。

2.3.2.4 实现对数据可解释性的分析

运用 `pandas.DataFrame.diff` 函数中 `SequenceMatcher()` 方法对 `arr1` 中的数据进行相似度计算，分析数据的可解释性。

2.3.3 算法实现

2.3.3.1 将附件4另存为 `annex4.csv` 文件

2.3.3.2 基于 Python 的 Anaconda 软件下，通过 `pandas` 的 `.read_csv()` 方法将 `annex3.csv` 文件导入，并分别用 `.drop()` 和 `.dropna()` 函数删除文件中的原标题行和任意含有 `Nan` 值的行。设定“`message_number`”“`message_users`”“`message_subject`”“`message_time`”“`message_details`”“`agree_num`”，“`disagree_num`”为新标题行。生成数据集 `data2`。

2.3.3.3 用 `pandas` 对数据进行清洗，删除因空格而导入的“`/n`”“`/t`”字符，然后利用 `pandas` 的 `.corr()` 方法计算相关系数矩阵

2.3.3.4 通过 `for` 循环遍历查找空值，并计算空值所占所有数据比例。

2.3.3.5 通过比较“留言时间”和“答复时间”的先后分析数据的时效性。

2.3.3.6 运用 `pandas.DataFrame.diff` 函数中 `SequenceMatcher()` 方法对 `arr1` 中的数据进行相似度计算，分析数据的可解释性。

2.3.3.7 总结数据，给出评价方案

3. 结果分析（行政人员如何提高工作效率）

3.1 首先要具备认真的工作态度，作为行政人员，认真的工作态度是必不可少的，如果不够认真，不够细心，很多事情都无法完成，甚至会耽误很多事情，导致自己不停返工等等。

3.2 要养成良好的工作习惯，要提高自己的工作效率，习惯是很重要的，如果平时自己就注意保持良好的工作习惯，很多事情都能够按部就班完成，那么行政人员的工作效率会大大提高。

3.3 保证自己能够处理好家庭和工作的关系，很多行政人员都是女性，一方面要忙于工作，一方面要照顾自己的家庭，所以工作效率的提高和自己是否可以很好处理家庭和工作的关系也有直接或者间接的关系。

3.4 找到适合自己的高效工作的方法，每一个人都有自己的特长，都有自己擅长的领域，和熟悉的工作方法，要让自己工作效率提高，就需要找到最适合自己的工作方式和方法。

3.5 懂得求助别人，身边的领导或者同事都是自己的好帮手，当一个人无法在规定的时间内完成某项工作的时候，适当的求助同事和领导也是很明智的做法和选择。

3.6 坚持学习，努力进步。读书也好，参加培训也罢，即便自己的薪酬不高，作为新时代的职场人，也需要在自己学习方面投资，增加自己的工作能力，才会提高工作效率。

.....

4. 结论

充分运用好基于自然语言处理技术的智慧政务系统,可以有效提升政府的管理水平和施政效率。

5. 参考文献

- [1] <https://github.com/aaronlou/DataScienceProject/> [OL]
- [2] https://blog.csdn.net/roger_royer/article/details/81125128[OL]
- [3] <https://blog.csdn.net/songyunlil1111/article/details/79306639>[OL]
- [4] https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html [OL]
- [5] https://blog.csdn.net/weixin_37801695/article/details/86481361[OL]
- [6] https://blog.csdn.net/Nicole_Li1095/article/details/101595818[OL]
- [7] <https://blog.csdn.net/qql449301756/article/details/39207293>[OL]
- [8] <https://blog.csdn.net/qql36523839/article/details/80639022>[OL]
- [9] <https://blog.csdn.net/qql41228218/article/details/86765042>[OL]
- [10] <https://blog.csdn.net/qql41081716/article/details/104983186>[OL]
- [11] <https://blog.csdn.net/ztf312/article/details/88672225>[OL]
- [12] <https://www.jb51.net/article/179958.htm>[OL]
- [13] https://blog.csdn.net/weixin_44084325/article/details/89764990[OL]
- [14] <https://wenku.baidu.com/view/04cla20ecdbff121dd36a32d7375a417866fclc0.html> [OL]
- [15] <https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.Series.str.contains.html> [OL]
- [16] https://blog.csdn.net/weixin_43746433/article/details/95353814[OL]
- [17] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html [OL]
- [18] <https://pandas.pydata.org/pandas-docs/version/0.18.1/text.html> [OL]
- [19] <https://pastebin.com/nRt3rVPr> [OL]
- [20] <https://wenwen.sogou.com/z/q854553321.htm> [OL]
- [21] <https://blog.csdn.net/davidchang365/article/details/58594922>[OL]
- [22] <https://www.jb51.net/article/123567.htm>[OL]
- [23] <https://blog.csdn.net/qql32618817/article/details/80430841>[OL]
- [24] <https://blog.csdn.net/yoggiecda/article/details/105710863>[OL]
- [25] <https://blog.csdn.net/AlanGuoo/article/details/76522429?locationNum=7&fps=1> [OL]
- [26] <https://jingyan.baidu.com/article/bad08ele8d2e9809c851212b.html> [OL]
- [27] <https://blog.csdn.net/bqwl8744018044/article/details/79934285>[OL]
- [28] <http://www.xwlunwen.com/jsjlunwen/59653.html> [OL]
- [29] <https://blog.csdn.net/MachineLearner/article/details/104638647>[OL]
- [30] <https://www.yebaike.com/wenhua/125486.html> [OL]