**中国矿业大学计算机学院**

**2019 级本科生课程报告**

课程名称 《应用软件开发实践》

报告时间 2022年6月23日

学生姓名 潘逸冬

学 号 08192764

专 业 计算机科学与技术

任课教师 徐东红

评 分 表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 课程教学目标 | 考查方式与考查点 | 占比 | 得分 |
| 1 | **目标1：**能够采用结构化方法或面向对象方法分析系统需求。 | 通过学生答辩及软件验收情况，考察其知识熟练应用程度。考察撰写的报告和设计文稿与原有业务要求的贴近度，描述的清晰性、完整性、无歧义。 | 30% |  |
| 2 | **目标2：**综合考虑设计、测试、维护，对设计方案进行优化，开发满足系统需求和约束条件的软件系统、模块或算法流程。 | 通过学生答辩及软件验收和设计文档，考察学生是否开发完成了满足系统需求和约束条件的软件系统、模块或算法流程。 | 30% |  |
| 3 | **目标3**：熟悉软件开发过程，具有系统的工程研究与实践经历。 | 通过答辩，考察学生需求分析、方案设计、详细设计、编码、测试等各环节中对于软件开发和管理技术的综合应用情况。 | 15% |  |
| 4 | **目标4：**掌握软件需求分析、设计、编码、测试等环节的常用技术和工程开发工具。 | 通过答辩，考察学生在分析、设计、编码和测试过程中，对需求分析、软件设计、源代码版本管理、软件测试等计算机辅助软件工程工具的使用情况。 | 15% |  |
| 5 | 目**标5：**理解并遵守计算机职业道德和规范，具有良好的法律意识、社会公德和社会责任感。 | 通过应用软件开发综合实训环节的选题和设计文档，考察学生是否具有良好的法律意识、社会公德和社会责任感，是否理解并遵守计算机职业道德和规范。 | 10% |  |
| 总分 | | | |  |

**摘 要**

本文为软件杯项目A9：智能创作平台项目，通过文章直接生成标题与摘要过程的详细介绍。因为目前文章的标题、摘要都是由人工完成，由人工阅读完成后，通过自己的理解输入对应文本的摘要和标题，受主观性影响较大，会产生误判，因此得到的标题和摘要不够准确。现使用人工智能算法，通过训练集的训练，生成契合文章主题内容和有吸引力的优质标题和摘要，并开发出一套智能创作WEB系统，提升写作的效率和质量。本文从四个方面叙述：数据集分割、摘要生成、ROUGE评价函数、系统测试，包括原理解析，关键实现代码等，能够帮助快速理解设计人的思维过程。

关键词：自然语言处理；自然语言生成；ROUGE;Web开发

**ABSTRACT**

This article is the software Cup project A9: intelligent creation platform project. It introduces in detail the process of directly generating titles and abstracts through the article. At present, the title and abstract of the article are all completed manually. After the manual reading is completed, the abstract and title of the corresponding text are input through their own understanding, which is greatly affected by subjectivity and will lead to misjudgment. Therefore, the title and abstract obtained are not accurate enough. Now we use the artificial intelligence algorithm to generate high-quality titles and abstracts that are consistent with the theme and attractive content of the article through the training of the training set, and develop an intelligent creation web system to improve the efficiency and quality of writing. This paper describes from four aspects: data set segmentation, summary generation, rouge evaluation function, system testing, including principle analysis, key implementation codes, etc., which can help to quickly understand the thinking process of designers.

Key Words: naturallanguageprocessing; Natural language generation; ROUGE; Web development

**目 录**

[1 项目简介 5](#_Toc107049375)

[2 个人负责部分 5](#_Toc107049376)

[3 开发平台 5](#_Toc107049377)

[4 项目详述 5](#_Toc107049378)

[4.1数据集分割 5](#_Toc107049379)

[4.1.1算法介绍 5](#_Toc107049380)

[4.1.2代码实现 6](#_Toc107049381)

[4.2摘要生成 7](#_Toc107049382)

[4.2.1Textrank摘要算法介绍 7](#_Toc107049383)

[4.2.2代码实现 8](#_Toc107049384)

[4.3Rouge评价函数 8](#_Toc107049385)

[4.3.1ROUGE评价函数介绍 8](#_Toc107049386)

[4.3.2 代码实现 9](#_Toc107049387)

[4.4系统测试 11](#_Toc107049388)

[4.4.1模型评价 11](#_Toc107049389)

[4.4.2前端接口 11](#_Toc107049390)

[5 后期优化 13](#_Toc107049391)

[6 心得体会 13](#_Toc107049392)

[参考文献： 13](#_Toc107049393)

# 1 项目简介

A9-智能创作平台：目前文章的标题、摘要都是由人工完成，由人工阅读完成后，通过自己的理解输入对应文本的摘要和标题，受主观性影响较大，会产生误判，因此得到的标题和摘要不够准确。现需要参赛者使用人工智能算法，通过训练集的训练，生成契合文章主题内容和有吸引力的优质标题和摘要，并开发出一套智能创作WEB系统，提升写作的效率和质量。

# 2 个人负责部分

1. 数据集分割
2. 摘要生成
3. Rouge评价函数
4. 系统测试

# 3 开发平台

编程IDE选择Pycharm，开发语言选择Python3.7，深度学习框架选择pytorch1.8.1。

# 4 项目详述

## 4.1数据集分割

### 4.1.1算法介绍

因为官方提供的数据集和开发集是json数据格式，不符合训练模型中标题与正文分开的要求，所以需要对数据集进行分割。

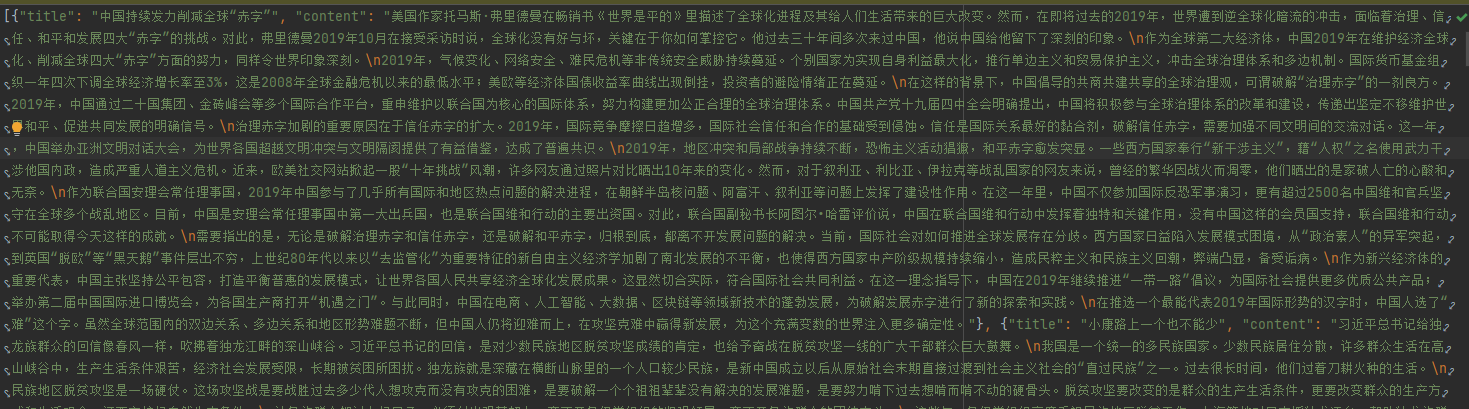


图1 官方提供的json格式的数据集

在Python的库中，有用于读取json数据格式的接口json.load()函数可以读取json数据并返回为Python中dict字典对象。两个标签分别为’titile’和’content’，用data存储json中的数据，并把’title’和’content’输出到文件中供后期训练。

### 4.1.2代码实现

关键代码：

1. f = open('../data/dev.json', encoding='utf-8')
2. data = json.load(f)
3. f.close()
4. **for** i **in** range(0, 1):
5. item = data[i]
6. with open('temp.src', 'w', encoding='utf-8') as src:
7. src.write(item['content'])
8. with open('test\_title.tgt', 'a', encoding='utf-8') as tgt:
9. tgt.write(item['title'] + '\n')

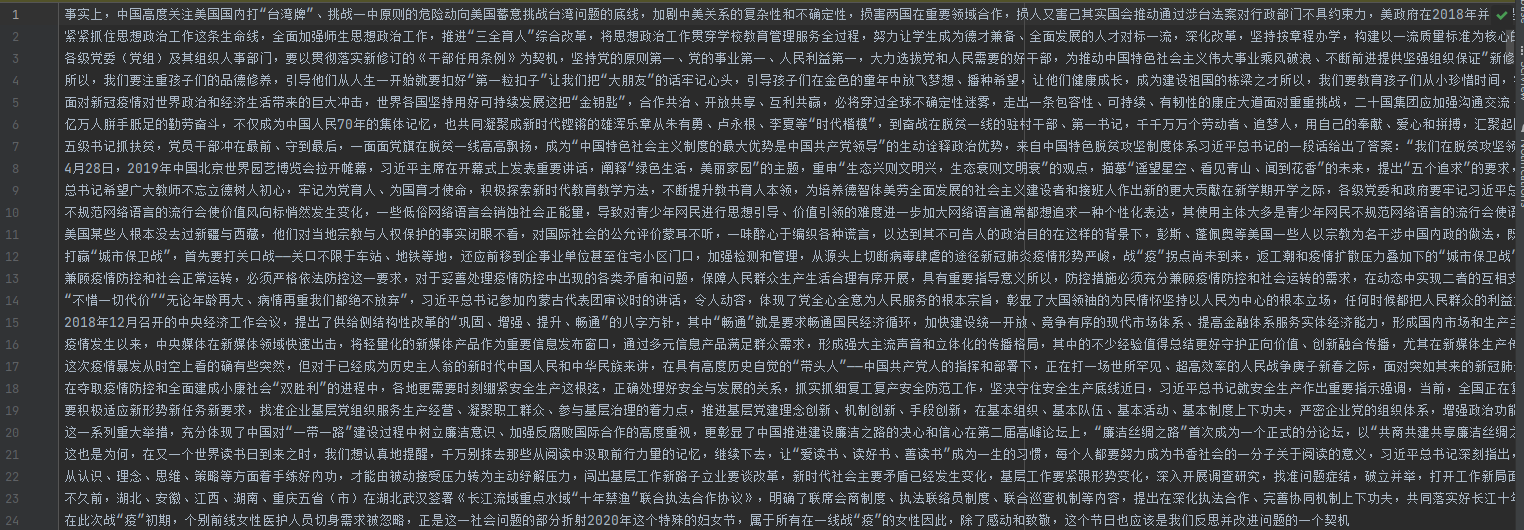


图2 拆分后的文章内容

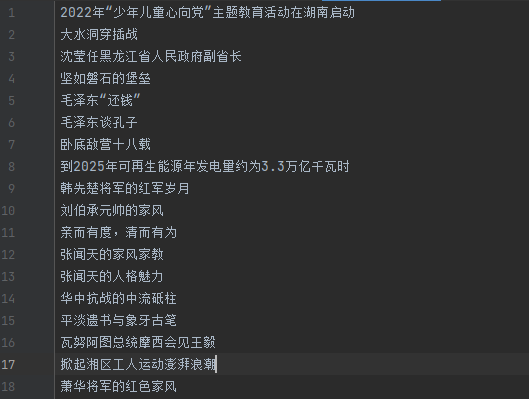


图3 拆分后的标题

## 4.2摘要生成

### 4.2.1Textrank摘要算法介绍

摘要生成使用Textrank算法，TextRank算法是由网页重要性排序算法PageRank算法迁移而来：PageRank算法根据万维网上页面之间的链接关系计算每个页面的重要性；TextRank算法将词视为“万维网上的节点”，根据词之间的共现关系计算每个词的重要性，并将PageRank中的有向边变为无向边。

其工作原理是将原文本拆分为句子，在每个句子中过滤掉停用词，并只保留指定词性的单词。由此可以得到句子的集合和单词的集合。

每个单词作为pagerank中的一个节点。设定窗口大小为k，假设一个句子依次由下面的单词组成：w1, w2, w3, w4, w5, ..., wn

w1, w2, ..., wk、w2, w3, ...,wk+1、w3, w4, ...,wk+2等都是一个窗口。在一个窗口中的任两个单词对应的节点之间存在一个无向无权的边。

基于上面构成图，可以计算出每个单词节点的重要性。最重要的若干单词可以作为关键词。若原文本中存在若干个关键词相邻的情况，那么这些关键词可以构成一个关键词组。将每个句子看成图中的一个节点，若两个句子之间有相似性，认为对应的两个节点之间有一个无向有权边，权值是相似度。

### 4.2.2代码实现

代码中参数设置窗口大小为2，权重为3。随后读取上步提取的content.src中的文本内容，使用Textrank算法生成摘要并输出到abstract.src中。

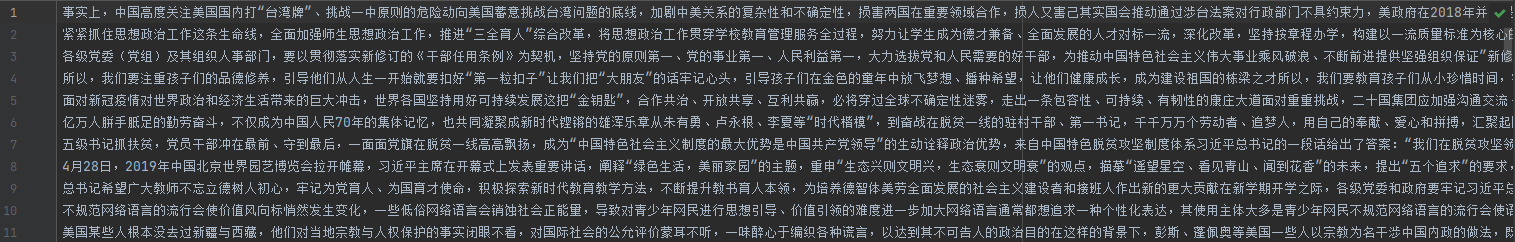


图4 生成的摘要

关键代码：

1. **def** getSummary(str):
2. string\_abstract = []  # 返回的字符串
3. text = str  # 传输入的字符串
4. # 创建分词类的实例
5. tr4w = TextRank4Keyword()
6. # 对文本进行分析，设定窗口大小为2，并将英文单词小写
7. tr4w.analyze(text=text, lower=True, window=2)
8. # 创建分句类的实例
9. tr4s = TextRank4Sentence()
10. # 英文单词小写，进行词性过滤并剔除停用词
11. tr4s.analyze(text=text, lower=True, source='all\_filters')
12. **for** item **in** tr4s.get\_key\_sentences(num=3):
13. # 打印句子的索引、权重和内容
14. # print(item.index, item.weight, item.sentence)
15. string\_abstract += item.sentence
16. **return** string\_abstract

## 4.3Rouge评价函数

### 4.3.1ROUGE评价函数介绍

ROUGE 指标主要是基于召回率 (recall) 的。ROUGE是一种常用的机器翻译和文章摘要评价指标，由 Chin-Yew Lin 提出，其在论文中提出了 4 种 ROUGE 方法：

ROUGE-N: 在 N-gram 上计算召回率

ROUGE-L: 考虑了机器译文和参考译文之间的最长公共子序列

ROUGE-W: 改进了ROUGE-L，用加权的方法计算最长公共子序列

在短摘要/标题式摘要任务中，ROUGE-1, ROUGE-2, ROUGE-L的表现很好,所以我们采用的是ROUGE-N与ROUGE-L公式如下。

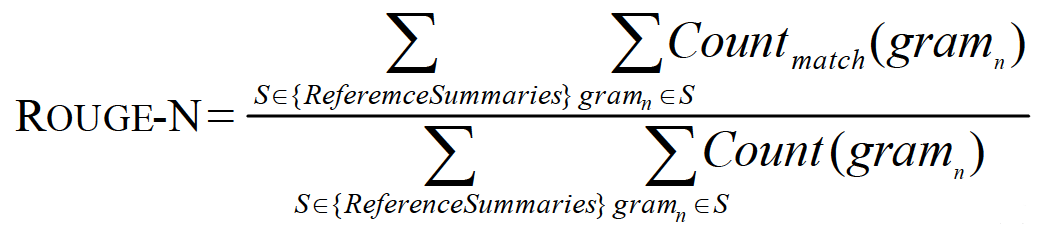


图5 ROUGE-N公式

分母是参考标题（人工标题/标准答案）中n-gram的总数量，分子是模型生成的摘要和参考摘要共有的n-gram的个数，即召回率。

Rouge-N: 是对应的Rouge-1 : 1-gram, Rouge-2: 2-gram,Rouge-3: 3-gram。

Rouge-L:L即是LCS(longest common subsequence，最长公共子序列)的首字母，因为Rouge-L使用了最长公共子序列。Rouge-L计算方式如下图：

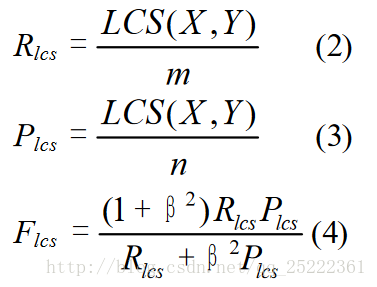


图6 ROUE-L公式

其中LCS( X , Y )LCS(X,Y)LCS(X,Y)是X和Y的最长公共子序列的长度，m,n分别表示参考标题和自动标题的长度，Rlcs, Plcs分别表示召回率和准确率。最后的Flcs即是我们所说的Rouge-L。

### 4.3.2 代码实现

关键代码：

1. # 读取摘要测试数据并放入数组
2. test\_data = []
3. with open('../dataprocess/abstract.txt', 'r', encoding='utf-8') as abstract:
4. lines = abstract.readlines()
5. **for** line **in** lines:
6. line = line.strip()
7. test\_data.append(line)
8. # 读取摘要相对应的标题并放入数组
9. title = []
10. with open('../dataprocess/title.txt', 'r', encoding='utf-8') as name:
11. lines = name.readlines()
12. **for** line **in** lines:
13. line = line.strip()
14. line = ' '.join(line)
15. title.append(line)
16. **print**('原标题：', title)
17. # 生成预测标题并放入数组
18. forecast = []
19. **for** text **in** test\_data:
20. with torch.no\_grad():
21. temp = bert\_model.generate(text, beam\_size=3)
22. forecast.append(temp)
23. **print**('预测标题', forecast)
24. rouge = Rouge()
25. **for** i **in** range(0, len(forecast)):
26. rouge\_score = rouge.get\_scores(forecast[i], title[i])
27. **print**(rouge\_score[0]["rouge-1"])
28. **print**(rouge\_score[0]["rouge-2"])
29. **print**(rouge\_score[0]["rouge-l"])

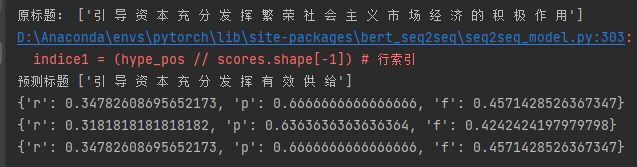


图7 部分测试数据的ROUGE评价

上图从上至下分别是ROUGE-1,ROUGE-2,ROUGE-L，范围都在[0,1]之间，数值越高说明与原标题越匹配，即预测的标题越准确。

但是从大量的测试集来看，ROUGE的数值还是有很大的波动，需要对训练生成的模型进行进一步的优化，更细的处理数据集。

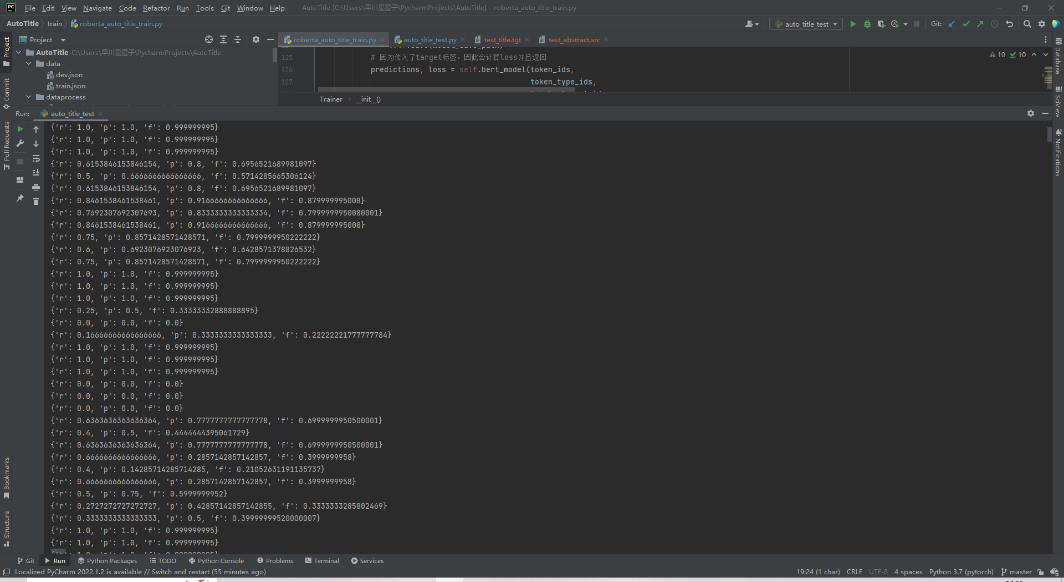


图8 大量测试集下的ROUGE评价

## 4.4系统测试

### 4.4.1模型评价

模型训练完后，用模型进行后期测试。首先读取生成的bin文件，并加载模型字典。随后将测试的文章生成摘要，并预测标题，与官方提供的标题使用ROUGE函数进行评估，查看生成标题的准确程度，并做适当调整。同时还需要关注生成标题所需要的时间，不能让生成标题的时间过长。

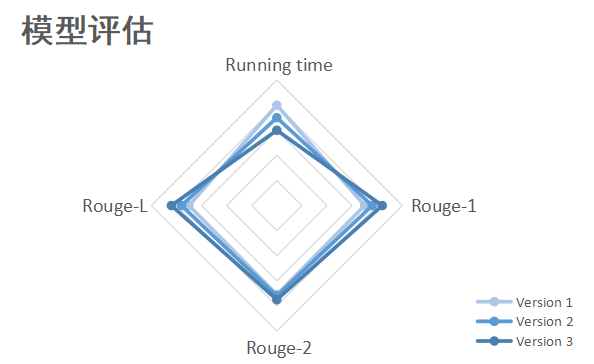


图9 生成标题评价图

### 4.4.2前端接口

因为需要在前端生成摘要和标题，所以需要将生产成标题与摘要的函数传送给前端。新建两个Python文件Web\_Abstract.py与Web\_Ttitle.py。其中定义了两个函数：Abstract()和Title()。

前端首先把用户输入的文本信息写入一个名为put\_article的文本文件中，随后调用Abstract()函数读取文章并用textrank算法生成摘要写入out\_summary的文本文件中，显示在前端界面中。接着调用Title()函数，读取out\_summary中的内容生成标题，放入out\_title的文本文件中在前端展示。

关键代码：

1. **def** Abstract():
2. string\_abstract = []
3. with open('put-article.txt', 'r', encoding='utf-8') as article:
4. text = article.read().replace('\n', '')
5. # 创建分词类的实例
6. tr4w = TextRank4Keyword()
7. # 对文本进行分析，设定窗口大小为2，并将英文单词小写
8. tr4w.analyze(text=text, lower=True, window=2)
9. # 创建分句类的实例
10. tr4s = TextRank4Sentence()
11. # 英文单词小写，进行词性过滤并剔除停用词
12. tr4s.analyze(text=text, lower=True, source='all\_filters')
13. **for** item **in** tr4s.get\_key\_sentences(num=3):
14. string\_abstract += item.sentence
15. summarize\_text = string\_abstract
16. with open('out-summary.txt', 'w', encoding='utf-8') as abstract:
17. **for** n **in** summarize\_text:
18. abstract.write(n)
20. **def** Title():
21. vocab\_path = "train/state\_dict/roberta\_wwm\_vocab.txt"  # roberta模型字典的位置
22. model\_name = "roberta"  # 选择模型名字
23. # 加载字典
24. word2idx = load\_chinese\_base\_vocab(vocab\_path)
25. # 定义模型
26. bert\_model = load\_bert(word2idx, model\_name=model\_name)
27. bert\_model.set\_device(device)
28. bert\_model.eval()
29. # 加载训练的模型参数
30. bert\_model.load\_all\_params(model\_path=auto\_title\_model, device=device)
31. test\_data = []
32. with open("out-summary.txt", 'r', encoding='utf-8') as abstract:
33. lines = abstract.readlines()
34. **for** line **in** lines:
35. line = line.strip()
36. test\_data.append(line)
37. forecast = []
38. **for** text **in** test\_data:
39. with torch.no\_grad():
40. temp = bert\_model.generate(text, beam\_size=3)
41. forecast.append(temp)
42. with open('out-title.txt', 'w', encoding='utf-8') as title:
43. s = "".join(forecast)
44. title.write(s)
45. **print**("预测标题：", forecast)

# 5 后期优化

因为目前的数据集还不是很多，并且前期在训练模型上面耗费了太多时间，在batch\_size与epoch的取值上没有花很多时间研究，目前训练的loss还是没有趋于稳定，在两个数上还需要花时间调整，可用图标的形式绘制每次loss的值，查看规律。并且需要收集更多的相关新闻加入数据集进行训练。

目前的测试数据还没有很多，光看rouge评价函数还不能看出模型的好坏，后面需要收集更多的数据进行测试。有时还会出现标题与摘要毫不相关的情况，需要进一步排查代码漏洞，寻找原因。同时生成标题的时间还需要进一步优化，越快越好。

# 6 心得体会

这是我第一次使用深度学习框架，以前的项目大都使用机器学习的方法，对我来说还是有不小的挑战。在这一个月的学习时间中，我慢慢理解了深度学习是怎样一种概念，是如何运作的，期间也遇到过很多问题，但经过我和我团队的讨论一一克服了。从读懂Roberta模型再到生成训练模型再到测试最后与前端进行交互，都对我编程能力有很大的提升。以前只接触过Java的Web开发，这次使用了Python的flask框架，更拓展我的知识面。通过这门课程参加了软件杯，在讨论群中学习别人的思维，对我来说都非常有意义。

# 参考文献：

Python TextRank4ZH 应用TextRank算法提取中文文本关键词、关键词组、关键句，<https://www.jianshu.com/p/b44445e53d19>

TextRank4ZH，<https://github.com/letiantian/TextRank4ZH>

文本生成：自动摘要评价指标 Rouge，https://blog.csdn.net/GJ\_0418/article/details/120944646