CS practice project report template

Project Name: Investigating the people’s impression of Kamen Rider Build and the possible reasons of some negative assessment.

Your Name: Lisa Xu

# Problem definition

## 1.1 What is the project about? (What is it? Why you want to do it? Possible applications)

It is about the people’s impression of a typical teleplay of Tokusatsu, which is the nineteenth of the works in series and facing the problem that carrying on history and starting the new future. It is interesting and worthwhile to find the appraisal from the audience, which can show us the wishes and results of the works at the end of the era, Heisei.

## 1.2 What are the success criteria? (What are your goals that will guide you through design and implementation and can be evaluated)

Gaining the assessments from comments and expressing by making a figure and a word cloud. Doing some critical thinking of the possible reasons for the negative evaluation.

# Design overview

How to solve the problem in part 1? You need to design your software system, design about the input, process, and output parts.

Please list the Identifier tables if needed.

INPUT:

requrl = 'https://movie.douban.com/subject/27073290/' + \

'?' + 'start=' + str(start) + '&limit=20' + '&sort=new\_score&status=P'

PROCESS

Describe your key algorithms in flowchart or pseudocode.

Key algorithm 1: **Get Comments**

file = open("hw.txt", "w", encoding='utf-8') # open a file to store comments

def getComments(pageNum):

eachCommentList = []

eachDateList = []

if pageNum > 0:

start = (pageNum-1) \* 20

else:

return False

requrl = 'https://movie.douban.com/subject/27073290/' + \

'?' + 'start=' + str(start) + '&limit=20' + '&sort=new\_score&status=P'

print(requrl)

resp = request.urlopen(requrl)

html\_data = resp.read().decode('utf-8') # translation

soup = bs(html\_data, 'html.parser') # using BeautifulSoup to analyse

# finding all parts of <div, class = 'comment'>

comment\_div\_lits = soup.find\_all('div', class\_='comment')

for item in comment\_div\_lits:

if item.find\_all('p'):

# finding where comment at <span>

eachCommentList.append(item.find\_all('span', class\_='short')[0].string)

tmpDate = item.find\_all('span')[-2].string

eachDateList.append(tmpDate)

return eachCommentList, eachDateList # combining comments

Key algorithm2: **Main (splitting words and saving comments into text)**

def main():

commentList = []

dateList = []

for i in range(50):

num = i + 1

[commentList\_temp, dateList\_temp] = getComments(num)

commentList.append(commentList\_temp)

dateList.append(dateList\_temp)

commentList = reduce(operator.add, commentList)

dateList = reduce(operator.add, dateList)

dataTmp = {'comments': commentList[:], 'date': dateList[:]}

df2 = pd.DataFrame(dataTmp)

pd.DataFrame(df2).to\_excel("text-movie.xls",

sheet\_name="sheet1", index=False, header=True)

comments = ''

final = ''

for k in range(len(commentList)):

comments = comments + (str(commentList[k])).strip()

final+=comments

file.write(final)

file.close()

pattern = re.compile(r'[\u4e00-\u9fa5]+')

filterdata = re.findall(pattern, comments) # eliminating punctuations

cleaned\_comments = ''.join(filterdata)

segment = jieba.lcut(cleaned\_comments)

words\_df = pd.DataFrame({'segment': segment}) # splitting

stopwords = pd.read\_csv("stopwords.txt", index\_col=False, quoting=3, sep="\t", names=['stopword'], encoding='utf-8')

#stop words, which are stored in stopwords.txt and could add new word in it

words\_df = words\_df[~words\_df.segment.isin(stopwords.stopword)]

words\_stat = words\_df.groupby(by=['segment'])[

'segment'].agg({"计数": numpy.size})

words\_stat = words\_stat.reset\_index().sort\_values(

by=["计数"], ascending=False)

print(words\_stat.head()) # counting

word\_frequence\_list = []

for key in word\_frequence:

temp = (key, word\_frequence[key])

word\_frequence\_list.append(temp)

main()

Key algorithm 3: **Producing Picture**

back\_color = imread('bg.png') # analyzing background picture

wc = WordCloud(background\_color='white', # background color

max\_words=16000, # maximum number of words

mask=back\_color, # using this parameter to make words cloud, width and height will be ignored if this parameter is not void

max\_font\_size=30, # maximum size of words

stopwords=STOPWORDS.add('tm'), # using built-in stop words

font\_path="simhei.ttf", # changing script

random\_state=42, # return PIL color for each color

# width=1000, # width

# height=860 #l ength

)

jieba.add\_word('假面骑士')

jieba.add\_word('build')

jieba.add\_word('exaid')

jieba.add\_word('ex-aid')

# open the file where stored comments

text = open('hw.txt', encoding = 'utf-8').read()

def stop\_words(texts):

words\_list = []

word\_generator = jieba.cut(texts, cut\_all=False) # return a iterator

with open('stopwords.txt', encoding = 'utf-8') as f:

str\_text = f.read()

str\_text.encode(encoding = 'utf-8')

unicode\_text = str\_text # make string turn to unicode

f.close() # one word in a row

for word in word\_generator:

if word.strip() not in unicode\_text:

words\_list.append(word)

return ' '.join(words\_list) # space

text = stop\_words(text)

……

OUTPUT:

wc.generate(text)

# color based on the color of background picture

image\_colors = ImageColorGenerator(back\_color)

# show

plt.imshow(wc)

# close x-axis

plt.axis('off')

# make word cloud

plt.figure()

plt.imshow(wc.recolor(color\_func=image\_colors), interpolation="bilinear")

plt.axis('off')

# save picture

wc.to\_file(‘build.png’)

# Implementation

List the key techniques used in your software, like arrays, searching, sorting, dictionaries, python modules (third-party, like matplotlib, numpy, scipy, sklearn etc.)

Tech #1: jieba

Why you use tech 1 in your project?

To split Chinese words.

How do you use it? Put your code screenshots here.

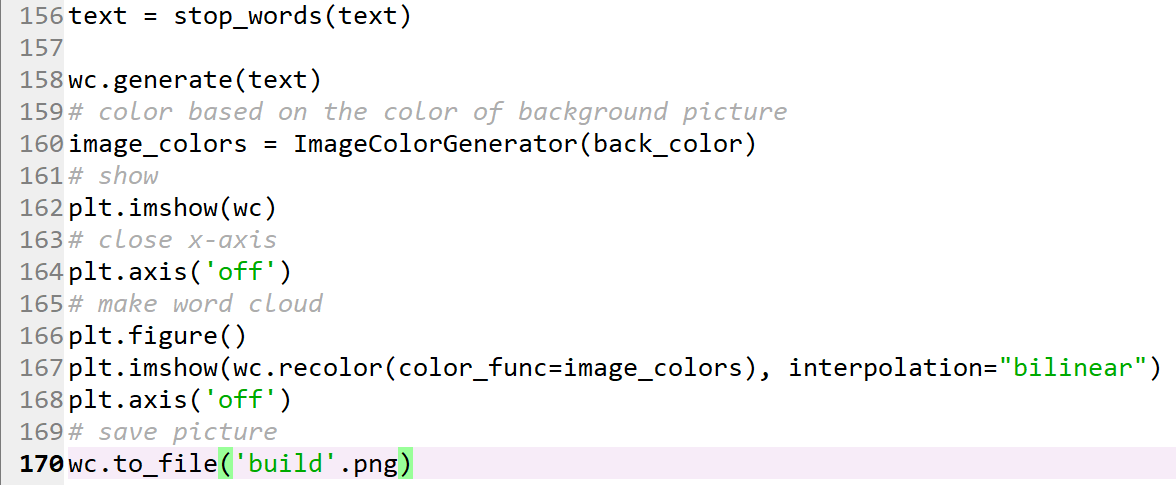


Tech #2: matplotlib

Why you use tech 2 in your project?

To produce picture.

How do you use it? Put your code screenshots here.

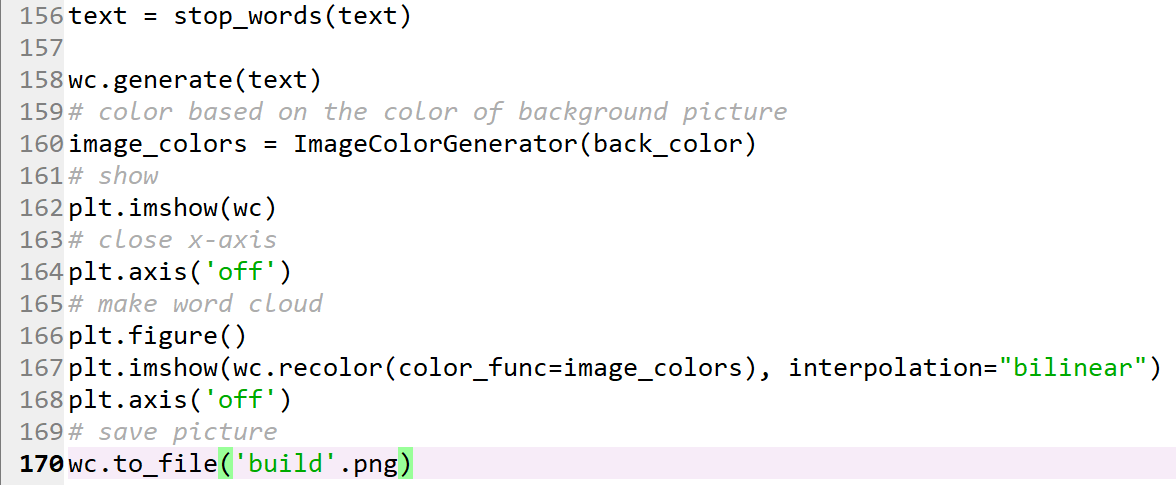


Tech #3: wordcloud

Why you use tech 3 in your project?

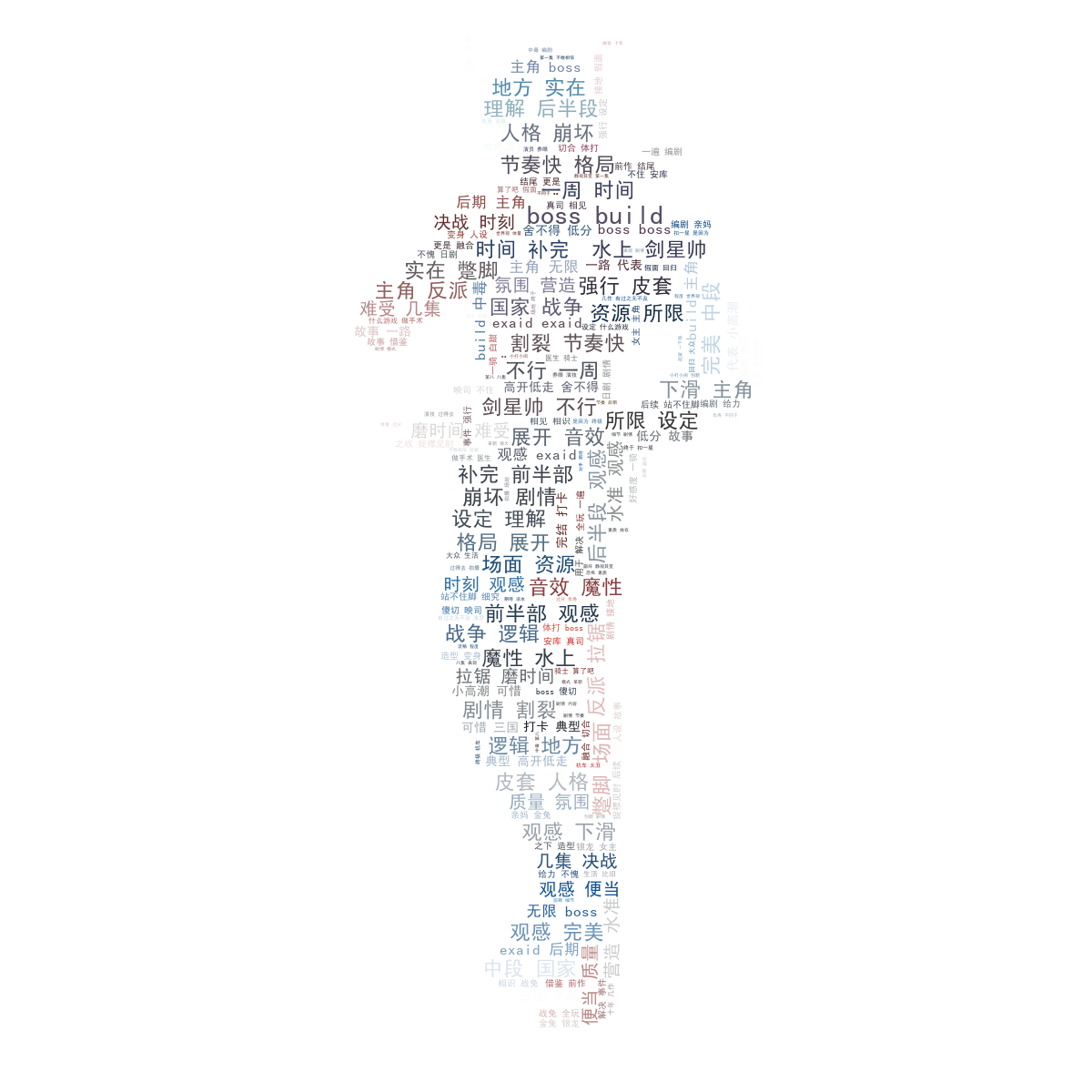
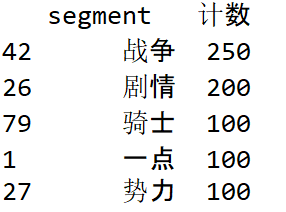
To derive word cloud.

How do you use it? Put your code screenshots here.



# Results

Show the outcomes/results of your projects

****

# Evaluation and conclusion

Evaluate the project based on your success criteria in Part 1.

Meet all of the criteria.

What is the conclusion of the project?

From the frequency of words, we can find that there is a heated debate about the content and main characters. From the word cloud, there are some praises for the work.

While, there are also some negative comments, such as, decreasing, doing things halfway and so forth. One of the most possible reasons is the postponement of the next era.

I did some critical thinking. There was a subtitle about Build in 2018, which is the Heisei Final. But, nowadays, this subtitle had disappeared. As a result, after I researched the name of the new era was delayed in 2018. Because of the concerns of creating confusion and giving rise to heated debate on the era name quickly shifting of public's attention from Emperor Akihito to his successor, even before his abdication on April 30, 2019, they decided to delay the next era. Therefore, Build was not the end of Heisei to the series anymore. Then, the company needs to take some investment money out of Build to prepare another work for this generation. Build is not as important as before.

# References

List all the materials you referred to.

<https://segmentfault.com/a/1190000010473819>

<https://ask.hellobi.com/blog/yuguiyang1990/9445>

<https://github.com/wudithu08/icc-AL-CS-2021/blob/master/Assessments/files/resources-textmining.zip>

<https://github.com/wudithu08/icc-AL-CS-2021/blob/master/Assessments/files/douban-movie.zip>

<https://www.cnblogs.com/delav/p/7845539.html>

<https://www.nippon.com/en/behind/l10797/japan-may-defer-announcement-of-new-era-name.html>