《Python 语言程序设计》期末课程设计

一、课程设计简介

作为 Python 课程期末的课程设计，我所选择的题目是**数据可视化**。

设计的程序名为**中文词云 Maker**.

“词云”这个概念由美国西北大学新闻学副教授里奇·戈登提出：“词云”就是对网络文本中出现频率较高的“关键词”予以视觉上的突出，形成“关键词云层”或“关键词渲染”，从而过滤掉大量的文本信息，使浏览网页者只要一眼扫过文本就可以领略文本的主旨。

在 Python 编程中，我们可以使用词云的方法来提取文档中的词语，生成词云图片，实现数据可视化。

通常的词云设计中，按词语出现频率对词语的字体大小进行排序，本次设计中则选择将各词语（句子）的字体大小设为**随机大小**。

代码所实现的功能是：通过读取用户给定的中文文本文档的内容，对文档进行分词，再读取用户给定的白底图片蒙版，生成具有一定形状的词云图片。

此外，本 Python 程序设计使用 Tkinter 库实现可视化界面，使用 GitHub 上出名的 jieba 库实现中文的分词处理，编译器为 pycharm.

二、Python 代码实现

```
from tkinter import *
import tkinter.filedialog
from tkinter import ttk
from datetime import *
from wordcloud import WordCloud, STOPWORDS
import jieba
import numpy as np
from PIL import Image



w = Tk()
w.title('中文词云 Maker v0.02')
w.geometry('400x560')
```

```
#   读取 txt 类型的文件
def readtxt():
    filename = tkinter.filedialog.askopenfilename()
    if filename != '' and filename[-3:] == 'txt':
        rtxt_lb.config(text='已选择文件: '+filename)
        rtxt_btn.config(text='重新选择 txt 文件')
    else:
        rtxt_lb.config(text='请选择 txt 文件!')


#   读取图片类型的文件
def readimg():
    filename = tkinter.filedialog.askopenfilename()
    if filename != '' and filename[-3:] in ['jpg', 'png']:
        rimg_lb.config(text='已选择文件: '+filename)
        rimg_btn.config(text='重新选择图片文件')
    else:
        rimg_lb.config(text='请选择图片文件!')


def choosefont(*args):
    del font[1]
    if choice_list.get() == '微软雅黑':
        font.insert(1, '微软雅黑/msyh.ttc')
    elif choice_list.get() == '宋体':
        font.insert(1, 'simsun.ttc')
    elif choice_list.get() == '黑体':
        font.insert(1, 'simhei.ttf')
    else:
        font.insert(1, 'Arial/arial.ttf')
```

```
#   制作词云的主函数
def maker_do():
    #   读取 label 中存储的文件目录信息
    if rtxt_lb.cget('text')[:1] == '已' and rimg_lb.cget('text')[:1] == '已':
        txtfile = rtxt_lb.cget('text')[7:]
        imgfile = rimg_lb.cget('text')[7:]
        #   默认输出目录为图片所在的目录,并命名
        now_time = str(datetime.now().strftime('-%Y%m%d-%H%M%S'))
        list_imgfile = list(imgfile)
        list_imgfile.insert(-4, now_time)
        savefile = ''.join(list_imgfile)
        #   修改 label 指示
        course_lb.config(text='制作中...')
    else:
        course_lb.config(text='请选择正确类型的文件！')
        return
    #   制作词云
    #   1|读取词云选项
    sc = scale.get()
    if sc > 10 or sc <= 0:
        course_lb.config(text='清晰度范围为 1~10！')
        return
    mw = max_words.get()
    mfs = max_font_size.get()
    rs = random_state.get()
    if (mfs - rs) < 0:
        course_lb.config(text='字体大小变动范围应小于最大字体大小！')
        return
    if mw <= 0 or mfs <= 0 or rs <= 0:
        course_lb.config(text='词云设置选项均应大于 0！')
```

```
            return
        #   2|读取 txt 文件内容到 text 中
        text = open(txtfile, 'r').read()
        #   3|利用 jieba 进行中文分词并生成字符串 wl
        wl = ''.join(jieba.cut(text))
        #   4|图片设置蒙版
        coloring = np.array(Image.open(imgfile))
        #   5|生成图云
        wc = WordCloud(scale=sc, background_color='White', max_words=mw, mask=coloring,
                       max_font_size=mfs, random_state=rs, font_path=''.join(font), stopwords=stop_words)
        wc.generate(wl)
        #   6|保存到图片所在目录中
        wc.to_file(savefile)
        course_lb.config(text='已存储为文件: '+savefile)


#   在主界面置入读取 txt 文件的标签和按钮
rtxt_lb = Label(w, text='')
rtxt_lb.pack(ipady=3, pady=5)
rtxt_btn = Button(w, text='选择 txt 文件', command=readtxt)
rtxt_btn.pack(ipady=3, pady=5)
#   在主界面置入读取图片文件的标签和按钮
rimg_lb = Label(w, text='')
rimg_lb.pack(ipady=3, pady=5)
rimg_btn = Button(w, text='选择图片文件', command=readimg)
rimg_btn.pack(ipady=3, pady=5)
#   字体选项
Label(w, text='词云字体:').pack(pady=5)
choice = StringVar()
choice_list = ttk.Combobox(w, textvariable=choice)
```

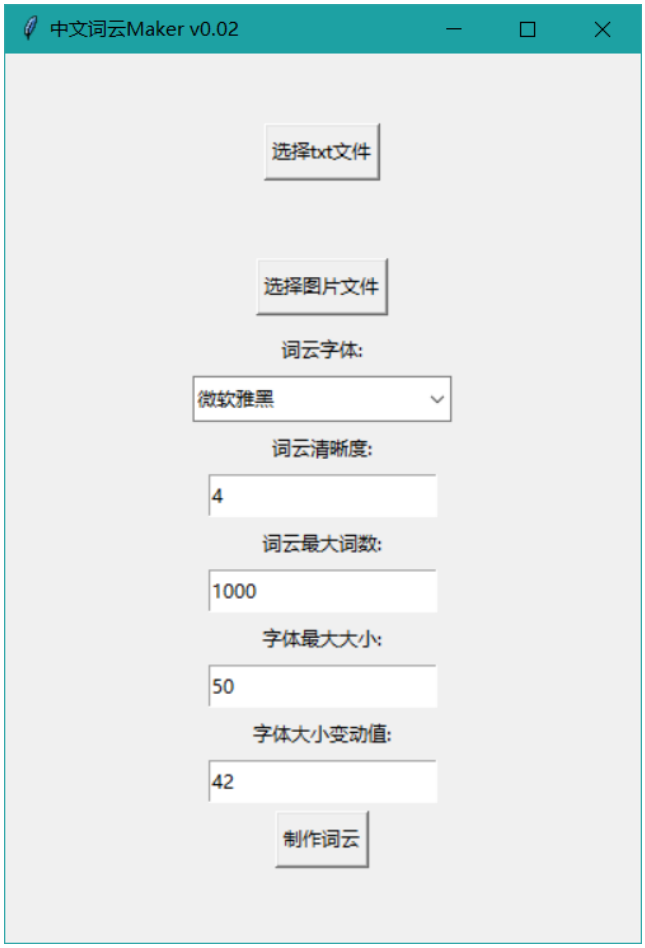
```
choice_list['values'] = ('微软雅黑', '宋体', '黑体', 'Arial')
choice_list.current(0)    # 默认选项为微软雅黑
font = ['C:/WINDOWS/Fonts/', '微软雅黑/msyh.ttc']    # 默认字体为微软雅黑(常规)
choice_list.bind("<<ComboboxSelected>>", choosefont)
choice_list.pack(ipady=3)
#   生成词云选项
Label(w, text='词云清晰度:').pack(pady=5)
scale = IntVar()    # 图片清晰度
input_scale = Entry(w, textvariable=scale)
input_scale.pack(ipady=3)
Label(w, text='词云最大词数:').pack(pady=5)
max_words = IntVar()    # 最大词数
input_max_words = Entry(w, textvariable=max_words)
input_max_words.pack(ipady=3)
Label(w, text='字体最大大小:').pack(pady=5)
max_font_size = IntVar()    # 字体最大大小
input_max_font_size = Entry(w, textvariable=max_font_size)
input_max_font_size.pack(ipady=3)
Label(w, text='字体大小变动值:').pack(pady=5)
random_state = IntVar()    # 字体大小变动范围((max_font_size - random_state) ~ max_font_size)
input_random_state = Entry(w, textvariable=random_state)
input_random_state.pack(ipady=3)
scale.set(4)    # 设置默认清晰度为 4
max_words.set(1000)    # 设置默认最大词数为 1000
max_font_size.set(50)    # 设置默认字体最大大小为 50
random_state.set(42)    # 设置默认字体大小变动范围为 42
#   添加词云暂停词
stop_words = set(STOPWORDS)
stoped_words = ['said', '你好', '撤回', '表情', '图片', 'QQ', '红包']
```

```
for key in stoped_words:
        stop_words.add(key)
#   开始按钮和进程提示
start_btn = Button(w, text='制作词云', command=maker_do)
start_btn.pack(ipady=3, pady=5)
course_lb = Label(w, text='')
course_lb.pack(ipady=3, pady=5)


w.mainloop()
```

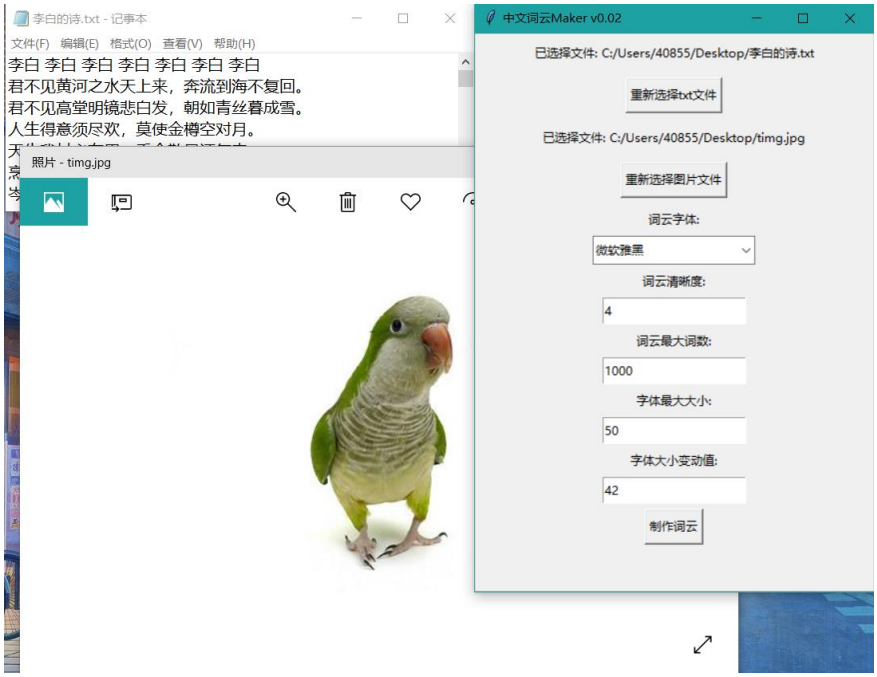
三、程序运行结果

程序的主界面如下，包含“选择 txt 文件”、“选择图片文件”、“制作词云”按钮，“词云字体”的下拉选择框，“词云清晰度”、“词云最大词数”、“字体最大大小”和“字体大小变动值”四个可输入选项。各选项默认值如图所示。



3.1 主界面

首先点击“选择 txt 文件”和“选择图片文件”，分别对应欲制作词云的文档和词云蒙版。然后可以修改字体（包括“微软雅黑”、“宋体”、“黑体”和“Arial”），词云清晰度（1~10，越高词云越清晰）等，其中字体大小变动值应小于字体最大大小。下图以鹦鹉为例进行操作。

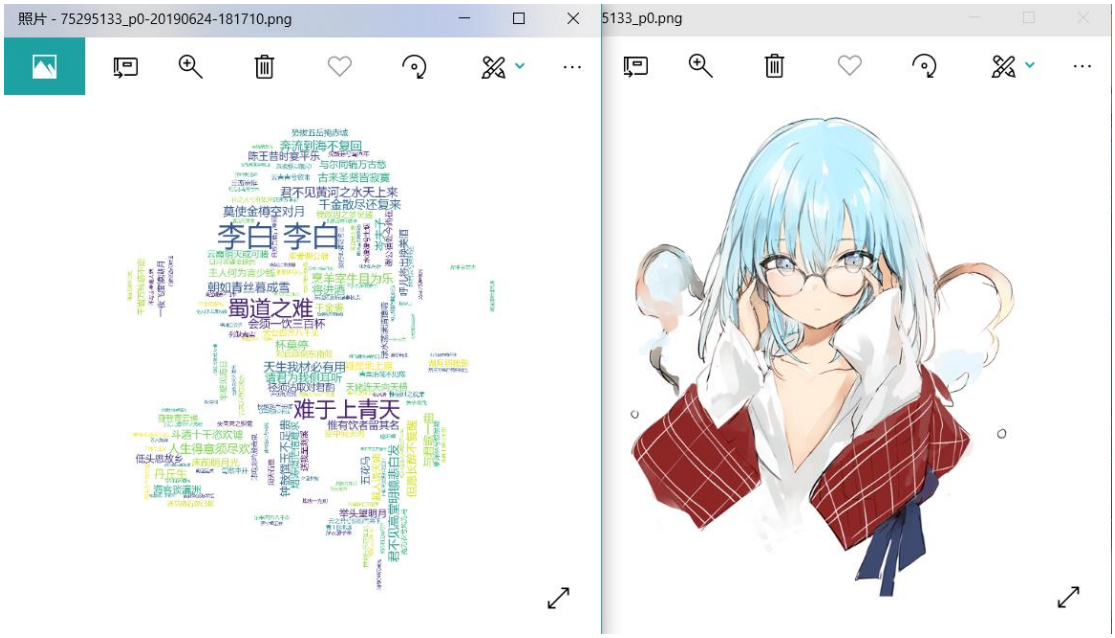


3-2 选择鹦鹉进行图云制作

结果如下图所示，在鹦鹉图片所在目录的位置生成了一个新的图片文件，就是软件制作的词云图片。对比词云图片和鹦鹉原图，具有较高的相似性，词云的数据可视化处理成功！

3-3 鹦鹉词云图片结果比较

同理，选择其它图片可以进行词云制作。由于使用图片蒙版的形式，因此图片白色的部分会作为背景色略去，对于部分图片并不能实现很好的词云制作。如下图所示。但总体而言可以实现具有较高相似性的词云制作。



3-4 插画词云图片结果比较

四、附录

4-1 鹦鹉原图

4-2 鹦鹉词云图

4-3 插画原图

4-4 插画词云图