实验二：词频统计与可视化

Design by W.H Huang | Direct by Prof Feng

1 实验目的

通过本次实验，你应该：

- 熟悉 *hadoop+ Spark*下编程环境
- 掌握基于*Spark*的基本*MAP REDUCE* 操作
- 掌握基本大数据可视化工具
- 独立完成本次青年群体择偶观分析实验

本次实验需小组内分工合作完成两个任务：

1. *WordCount* 词频统计

   你将会使用到 *jieba* 分词 & 基于 *pySpark* 的基本 *MAP REDUCE* 操作进行词频统计，在指定数据集上大数据分析青年群体择偶观倾向。

2. 大数据可视化

   你将使用 *echars & WordCloud* 两个可视化库来进行大数据可视化，小组独立完成核心代码编写、测试。

2 实验准备

2.1 上传文件

在开始实验前， 首先要将代码及相关资源上传到服务器。该小节将介绍如何使用FTP软件将本地（Windows）文件上传到服务器(Linux)。

1. 下载软件

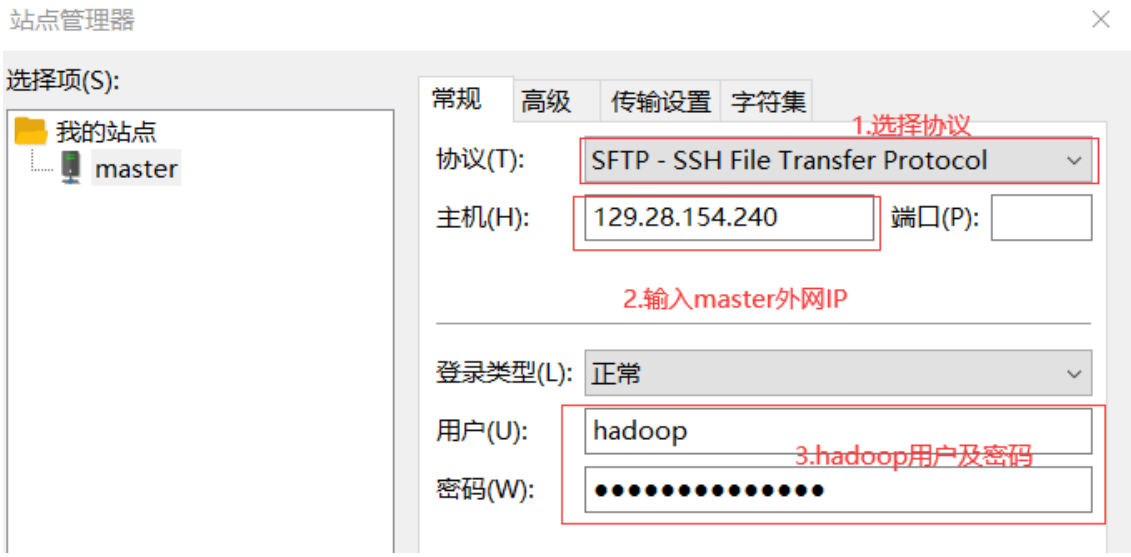
   FTP工具我们选择`Filezilla`，下载地址：Filezilla下载

   点击进行下载安装，安装过程较为简单不再赘述。

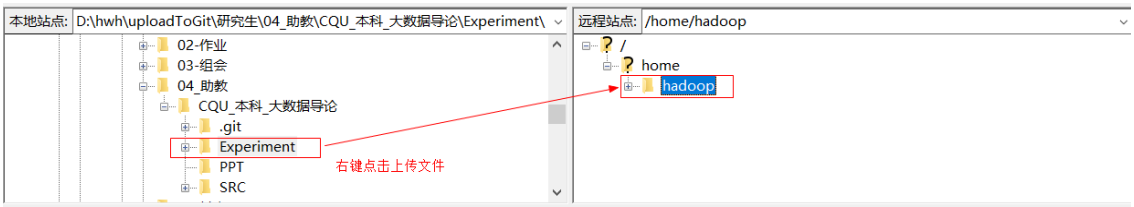
2. 连接服务器

   依次点击：文件 --> 站点管理器 --> 新站点



3. 上传文件

   如下图所示，左侧为本地文件，右侧为服务器文件目录（默认为 `/home/hadoop`）



   上传完毕后，可在服务器上查看文件：

```
[hadoop@master Experiment]$ cd /home/hadoop/Experiment/
[hadoop@master Experiment]$ ll
total 20
drwxrwxr-x 2 hadoop hadoop 4096 Jan 26 18:54 Ex1_SettingUpEnvironment
drwxrwxr-x 5 hadoop hadoop 4096 Jan 27 00:34 Ex2_WordCount
```

2.2 安装相关库

- 安装 `jieba`

```
sudo pip3 install jieba -i https://pypi.tuna.tsinghua.edu.cn/simple
```

- 安装 `wordcloud`

```
sudo pip3 install wordcloud -i https://pypi.tuna.tsinghua.edu.cn/simple
```

- 安装 `pyecharts`

```
sudo pip3 install pyecharts -i https://pypi.tuna.tsinghua.edu.cn/simple
sudo pip3 install snapshot-selenium
```

- 安装驱动

  `pyechars` 模块保存图片需要安装相应驱动：

```
sudo yum install https://dl.google.com/linux/direct/google-chrome-
stable_current_x86_64.rpm
sudo yum install chromedriver.x86_64
```

2.3 设置日志级别

> 由于`spark`在运行时会打印非常多的日志，为了便于调试观察，我们设置日志级别为`WARN`。

以下为全局设置日志级别方式，你也可在代码中临时设置`sc.setLogLevel("WARN")`（详见`ex3.pdf`）。

1. 切换到`conf`目录

```
cd /usr/local/spark/conf
```

2. 设置配置文件

```
cp log4j.properties.template log4j.properties
vim log4j.properties
```

修改`log4j.rootCategory=WARN,console`

```
# Set everything to be logged to the console
log4j.rootCategory=WARN, console
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.target=System.err
```

3 实验流程

在实验开始之前，我们强烈建议你按照以下流程完成实验：

1. **命令行**下完成代码**单元测试**
2. 单元测试无误，将代码填充在相应给出的`py`文件函数中
3. `spark-submit`方式提交代码

> ☺ 如何在命令行下完成单元测试？

1. 启动`pyspark`

```
cd /usr/local/spark
bin/pyspark
```

> ⚠ 后续实验都是在集群环境下（本次实验不需要），启动`pyspark`应该按以下方式：
>
> - 启动集群
>
> ```
> # 启动hadoop集群
> cd /usr/local/hadoop
> sbin/start-all.sh
> # 启动spark集群
> cd /usr/local/spark
> sbin/start-master.sh
> sbin/start-slaves.sh
> ```
>
> - 启动`pyspark`
>
> ```
> bin/pyspark --master spark://master:7077
> ```

2. 命令行下单元测试

例如，本次实验要求你完成`jiebaCut`函数编写：

> 完成下列指定位置编码，使得`str`为所有答案拼接而成的字符串
>
> ```
> def jiebaCut(answers_filePath):
>  """
>  结巴分词
>  :param answers_filePath: answers.txt路径
>  :return:
>  """
>  # 读取answers.txt
>  answersRdd = sc.textFile(answers_filePath) # answersRdd每一个元素对应answers.txt每一行
>
>  # 利用SpardRDD reduce()函数,合并所有回答
>  # 【现在你应该完成下面函数编码】
>  str = answersRdd.reduce(lambda )
>
>  # jieba分词
>  words_list = jieba.lcut(str)
>  return words_list
> ```
>
> ⚠ 命令行模式下，不用设置`SparkContext、SparkSession`实例：
>
> ```
> conf = SparkConf().setAppName("ex2").setMaster("local")
> sc = SparkContext(conf=conf)
> ```
>
> 会自动生成实例`sc`，可直接使用！

首先定义`answers_filePath`，查询此前代码指定按照如下方式进行拼接：

```
# 结巴分词
words_list = jiebaCut("file:///" + SRCPATH + "answers.txt")
```

```
>>> answers_filePath =
'file:///home/hadoop/Experiment/Ex2_WordCount/src/answers.txt'
```

按照流程读入文件：

```
>>> answersRdd = sc.textFile(answers_filePath)
>>> answersRdd.take(10)  # 展示前10行数据验证
```

现在你可以尝试在命令行下 **实时交互** 完成`str = answersRdd.reduce(lambda )`这行代码完整编写。

例如，你可以如此进行测试：

```
>>> answersRdd.reduce(lambda a,b: a+b)
'★★★更新于4个月以后★7月14日晚★★★写这个答案时，刚刚过完春节，在家被催婚心烦意乱，随手刷到，一时兴起...'
```

会实时显示交互结果，验证是否编码正确。

3. 提交代码

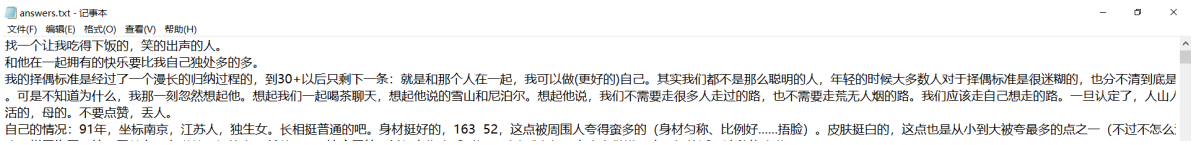
在命令行下单元测试后，便可以填写在相应`py`文件中。

最后通过`spark-submit`方式提交代码。相应如何提交，在实验后有详细介绍，这里不再赘述。

3.1 数据集介绍

> 本次实验数据集来源于2019级研究生@***W.H Huang***，数据集包含了知乎全站 **择偶观** 相关问题下所有 *答案&作者信息* 。原**完整**数据存储于`mysql`数据库，出于简化实验数据部署等目的，本次实验**仅使用其中部分数据**以`txt`文本形式展示。

数据集 `answers.txt` 每一行代表一个完整回答，一共有3W条回答，如下图：

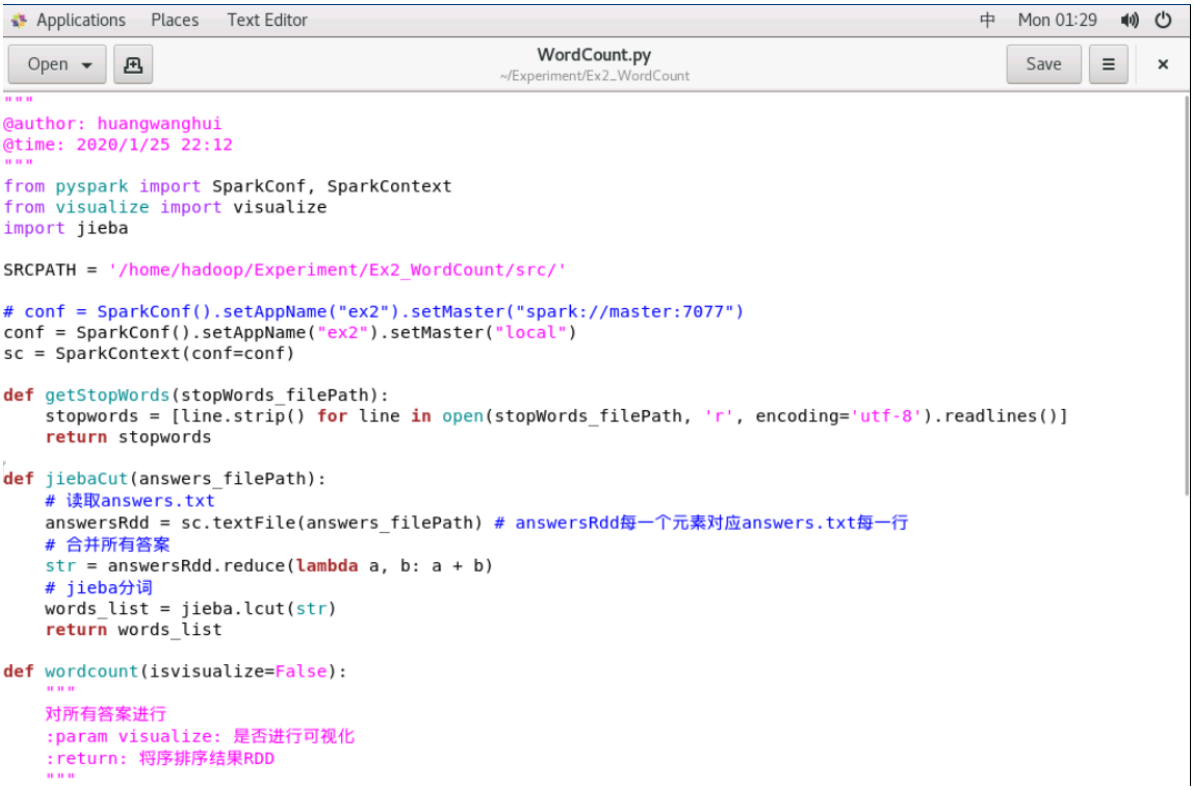


每一条回答均已进行简单清洗去除图片、视频URL、HTML标签等。

3.2 WordCount.py

3.2.1 完成编码

你现在应该独立完成`WordCount.py`编码，可在服务器上查看：



`WordCount.py` 有`3`个函数，它们的作用如下：

- `getStopWords`：读取`stop_words.txt`所有停用词，返回一个`python List`
- `jiebaCut`：将所有答案合并，并进行分词，返回一个`python List`
- `wordcout`：核心函数，利用`SparkRdd`完成词频统计

`jiebaCut`

完成下列指定位置编码，使得`str`为所有答案拼接而成的字符串

```
def jiebaCut(answers_filePath):
    """
    结巴分词
    :param answers_filePath: answers.txt路径
    :return:
```

```
    """
    # 读取answers.txt
    answersRdd = sc.textFile(answers_filePath) # answersRdd每一个元素对应
answers.txt每一行

    # 利用SpardRDD reduce()函数,合并所有回答
    # 【现在你应该完成下面函数编码】
     str = answersRdd.reduce(lambda  )

    # jieba分词
    words_list = jieba.lcut(str)
    return words_list
```

`wordcout`

完成下面指定位置编码，使得 `resRdd` 包含所有词频统计结果，且降序排列。

打印 `resRdd` 前十个元素应该为如下结果，`resRdd.take(10)`：

```
[('身高', 2627), ('家庭', 2018), ('父母', 2002), ('性格', 1882), ('男生', 1640),
 ('朋友', 1618), ('条件', 1568), ('学历', 1445), ('女生', 1380), ('感情', 1301)]
```

```
def wordcount(isvisualize=False):
    """
    对所有答案进行
    :param visualize: 是否进行可视化
    :return: 将序排序结果RDD
    """
    # 读取停用词表
    stopwords = getStopWords(SRCPATH + 'stop_words.txt')

    # 结巴分词
    words_list = jiebaCut("file://" + SRCPATH + "answers.txt")

    # 词频统计
    wordsRdd = sc.parallelize(words_list)

    # wordcount：去除停用词等同时对最后结果按词频进行排序
    # 完成SparkRDD操作进行词频统计
    # 提示：你应该依次使用
    #      1.filter函数进行停用词过滤&去除长度=1的词汇
    #      2.map进行映射，如['a','b','a'] --> [('a',1),('b',1),('a',1)]
    #      3.reduceByKey相同key进行合并 [('a',2),('b',1)]
    #      4.sortBy进行排序，注意应该是降序排序
    #【现在你应该完成下面函数编码】
    resRdd = wordsRdd.filter(lambda word: ) \
                     .filter(lambda word: ) \
                     .map(lambda word: ) \
                     .reduceByKey(lambda a, b: ) \
                     .sortBy(ascending=False, numPartitions=None,keyfunc=lambda
x: x[1])\

    # 可视化展示
    if isvisualize:
        v = visualize()
        # 饼状图可视化
        pieDic = v.rdd2dic(resRdd,10)
        v.drawPie(pieDic)
```

```
        # 词云可视化
        wwDic = v.rdd2dic(resRdd,50)
        v.drawWorcCloud(wwDic)
    return  resRdd
```

3.2.2 提交代码

此时主函数代码，设置可视化为`False`：

```
if __name__ == '__main__':

    # 进行词频统计，不进行可视化
    resRdd = wordcount(isvisualize=False)
    print(resRdd.take(10))  # 查看前10个
```

1. 切换到`spark`目录

   ```
   cd /usr/local/spark
   ```

2. 提交代码

   > ⚠ 如果启动了集群需要先关闭：
   >
   > ```
   > cd /usr/local/hadoop
   > sbin/stop-all.sh
   > ```
   >
   > ```
   > cd /usr/local/spark
   > sbin/stop-all.sh
   > ```
   >
   > 因为本次实验并非在集群环境下运行。

   ```
   bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
   ```

3. 查看结果

   你应该得到如下结果：

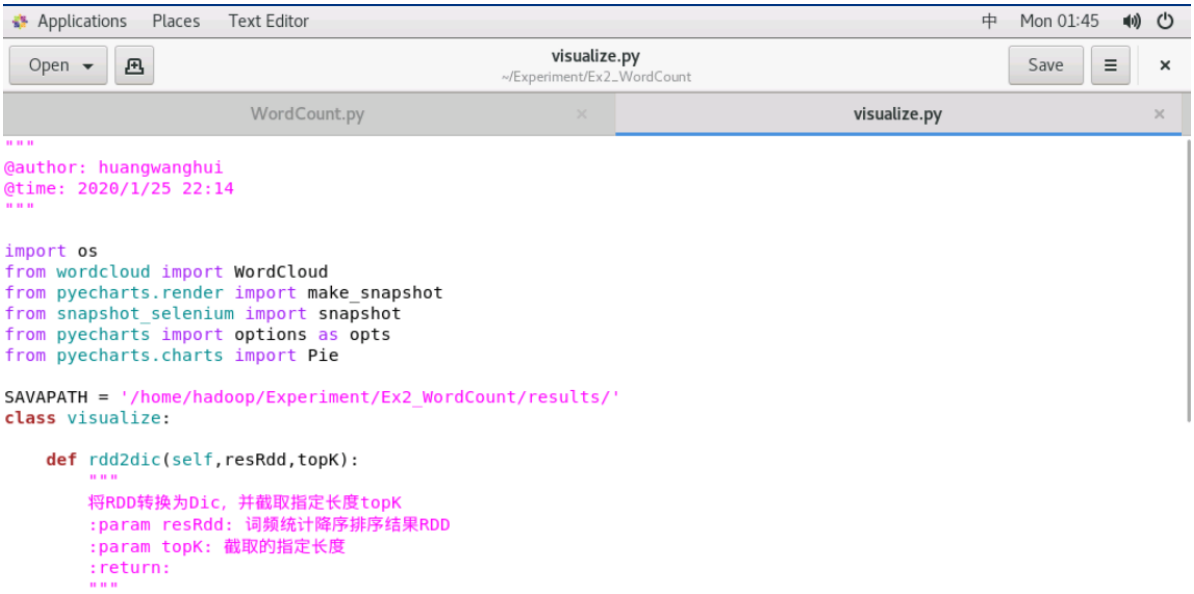
```
[hadoop@master spark]$ bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
20/01/27 01:34:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 1.790 seconds.
Prefix dict has been built successfully.
20/01/27 01:35:31 WARN TaskSetManager: Stage 1 contains a task of very large size (18076 KB). The maximum recomm
ended task size is 100 KB.
[('身高', 2627), ('家庭', 2018), ('父母', 2002), ('性格', 1882), ('男生', 1640), ('朋友', 1618), ('条件', 1568),
 ('学历', 1445), ('女生', 1380), ('感情', 1301)]
```

3.3 visualize.py

在本节你应该完成对`visualize.py`核心代码编写。

3.3.1 完成编码

可在服务器上查看`visualize.py`文件如下：



`visualize.py` 一共有 `3` 个函数，它们的作用如下：

- `rdd2dic`： 将 `resRdd` 转换为 `python Dic`，并截取指定长度 `topK`
- `drawWorcCloud`：进行词云可视化，同时保存结果
- `drawPie`： 进行饼图可视化，同时保存结果

`rdd2dic`

完成下面指定位置编码，将 `resRDD` 转换为 `python Dic`，并截取指定长度

```
def rdd2dic(self,resRdd,topK):
    """
    将RDD转换为Dic，并截取指定长度topK
    :param resRdd: 词频统计降序排序结果RDD
    :param topK: 截取的指定长度
    :return:
    """
    # 提示：SparkRdd有函数可直接转换
    #【现在你应该完成下面函数编码】

    resDic =
    # 截取字典前K个
    K = 0
    wordDicK = {}
    for key, value in resDic.items():
        # 完成循环截取字典


    return wordDicK
```

3.3.2 提交代码

此时主函数代码，设置可视化为 `True`：

```
if __name__ == '__main__':

    # 进行词频统计，并可视化
    resRdd = wordcount(isvisualize=True)
```

1. 切换到`spark`目录

```
cd /usr/local/spark
```

2. 提交代码

```
bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
```

> 如果出现可视化错误，`selenium. common. exception. webdriver exception: message: chrome not reachable`
>
> 通常是因为chromedriver程序占用了端口，控制台登陆重启服务器即可。相关讨论可见 issue#5 @lympassion

3. 查看结果

查看目录`/home/hadoop/Experiment/Ex2_WordCount/results`：

可以看出：`身高、家庭、性格、父母、学历`等是青年群体择偶最在意的几个特质。

4 实验小结

本次数据集仅使用了其中一部分：

- 对完整数据集感兴趣同学，可联系助教获取
- 对数据集`爬虫`感兴趣同学，参照博客记录 爬虫实战3：模拟登陆知乎并爬取任意帖子数据

接下来的实验，你将会进一步学习在分布式集群下进行大数据分析，如：在集群执行任务、`hdfs`的使用等。同时你也将开始了解基本机器学习算法在大数据分析的应用，这将包括`Kmeans`、`SVM`等经典机器学习算法。

我们将尽量设计有趣、生动的实例来帮助你理解。最后，恭喜你完成第一个在基于`Spark`平台的大数据分析小项目，希望你从中获得了不少乐趣 :) 。