# 实验内容与要求

# 实验三 相似项发现

第一部分 初话python

1. 实验目的

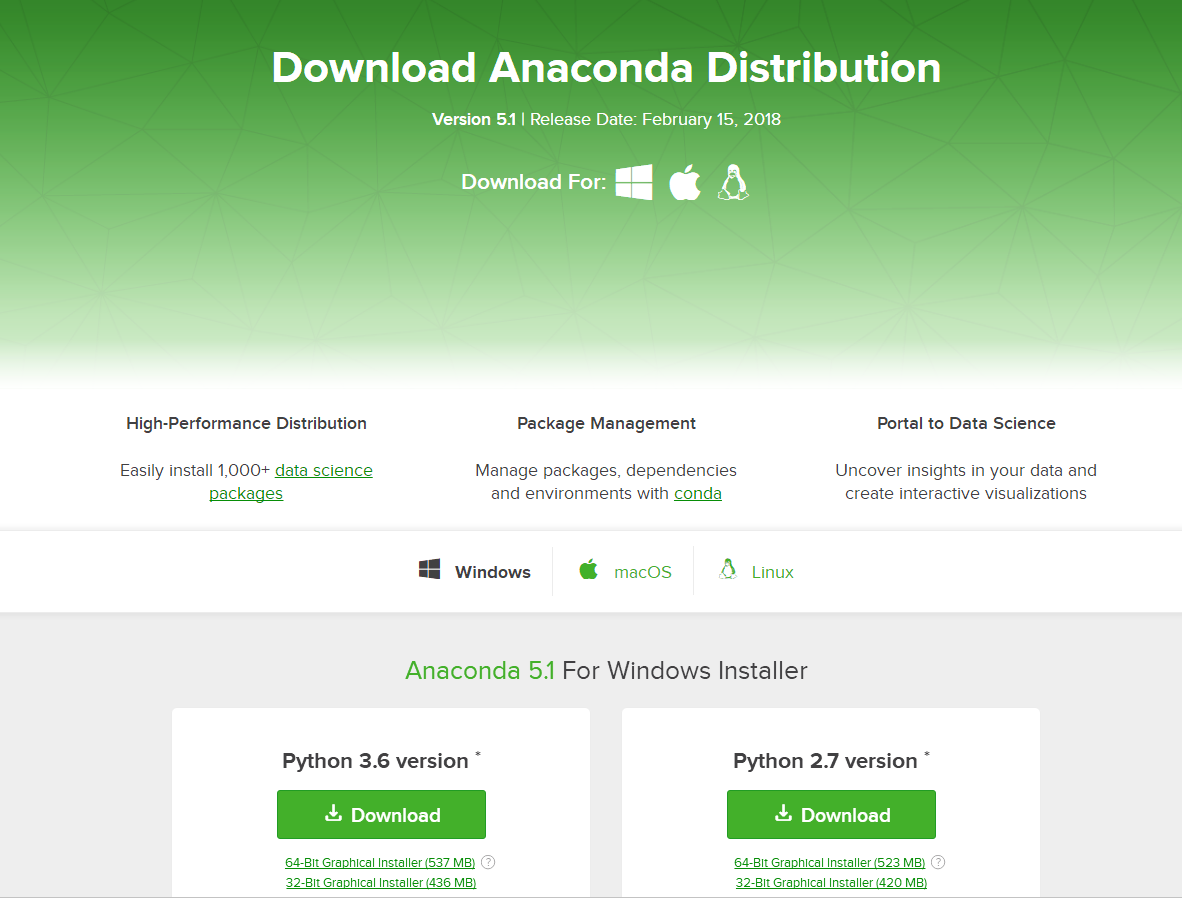
* 熟悉Python 环境搭建
* 掌握Python扩展包安装
* 掌握Python文件读写模式

1. 实验原理、方法和手段

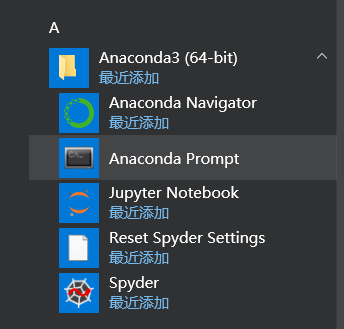
* Python 环境搭建

以win10系统为例：

1. 安装anaconda，官网地址：<https://www.anaconda.com/download/>



1. 根据需要选择python2.7或者python3.6下载，这里选择python3.6。
2. 安装好后，如下图所示，打开Anaconda Prompt，输入python，查看python版本号。



1. 输入以下命令即可联网升级 pip 工具

python -m pip install -U pip

* Python扩展包安装

由于Anaconda自带很多扩展包，省去很多安装的麻烦。但在实际中，需要更多的扩展包支持，通常有三种方法。

1. 在线直接安装：pip install +所需的扩展包名

eg. pip install numpy

1. 离线安装：在https://www.lfd.uci.edu/~gohlke/pythonlibs/下载所需的安装包，然后pip install +安装包名

eg. pip install scikit\_learn-0.19.1-cp36-cp36m-win\_amd64.whl

1. 源码安装：解压之后：

Python setup.py build

Python setup.py install

* Python文件读写模式

不同模式打开文件的完全列表：

|  |  |
| --- | --- |
| **模式** | **描述** |
| r | 以只读方式打开文件。文件的指针将会放在文件的开头。这是默认模式。 |
| rb | 以二进制格式打开一个文件用于只读。文件指针将会放在文件的开头。这是默认模式。一般用于非文本文件如图片等。 |
| r+ | 打开一个文件用于读写。文件指针将会放在文件的开头。 |
| rb+ | 以二进制格式打开一个文件用于读写。文件指针将放在文件的开头。一般用于非文本文件如图片等。 |
| w | 打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| wb | 以二进制格式打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。一般用于非文本文件如图片等。 |
| w+ | 打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| wb+ | 以二进制格式打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。一般用于非文本文件如图片等。 |
| a | 打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |
| ab | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |
| a+ | 打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。 |
| ab+ | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件用于读写。 |

示例

test.txt的原始文件如下：

早上好

您好

how are you?

如果我们在open文件后，没有进行任何读写，则从末尾加入

>>> with open('test.txt','r+') as f:

... f.writelines("长沙")

这时文件变成:

长沙好

您好

how are you?

>>> with open('test.txt','w+') as f:

... f.writelines("长沙")

这时文件变成:

长沙

第二部分 文本相似度计算

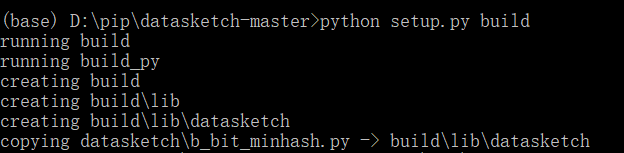
1. 实验目的
2. 能够编程实现最小哈希Jaccard估计值，并和Jaccard实际值对比
3. 熟悉局部敏感哈希原理
4. 实验内容
5. 掌握Jaccard相似度计算方法。
6. 掌握最小哈希计算，并和Jaccard相似度实际值对比。
7. 掌握局部敏感哈希原理，并计算所给数据集每一项ID的相似项。
8. 实验原理、方法和手段

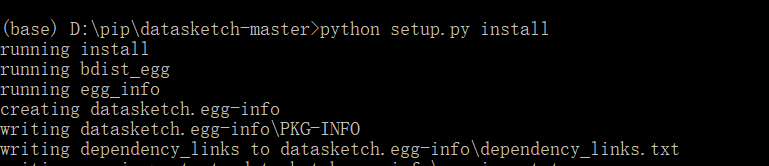
1、安装datasketch.

解压缩datasketch-master.zip，命令行安装：

Python setup.py build

Python setup.py install

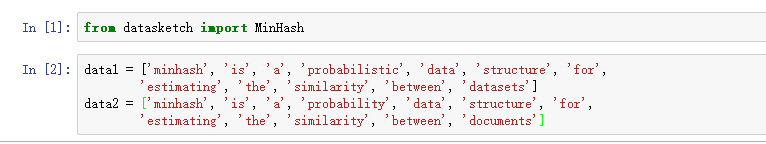




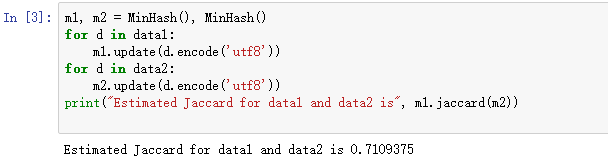
2、最小哈希Jaccard计算

* 导入安装好的datasketch，输入两组待测数据。

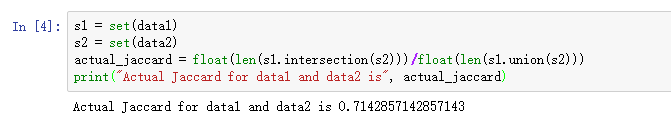
具体minhash实现可参考datasketch-master\datasketch\minhash.py，这里直接调用该模块。



* 计算Jaccard估值

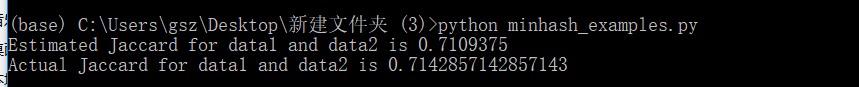


* Jaccard实际值



我们发现，估值近似等于实际值。

3、我们对源码进行抽取，进行编译，见<源码1文件夹>。



我们对源码进一步修改。通过自定义MinHash中使用的置换函数的数量来调整精度。

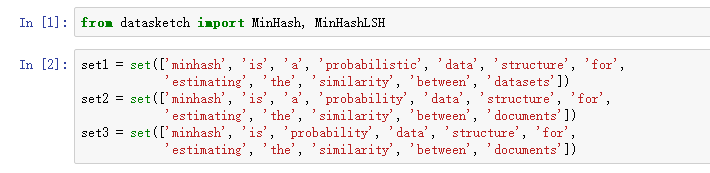
m = MinHash(num\_perm=256)

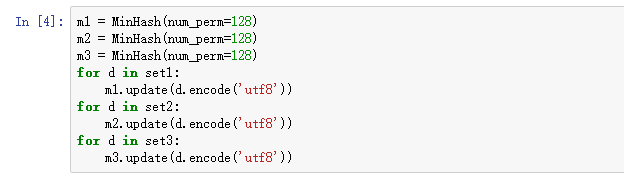
然而，更多的置换函数意味着：更多的哈希值需要被存储。因此需要选择合适数量的置换函数，确保精度的基础上性能更优化。

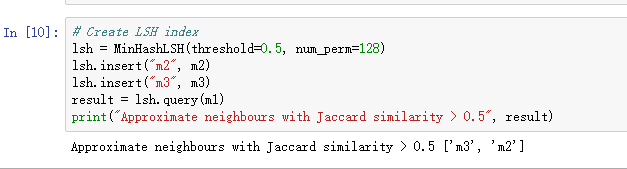
4、局部敏感哈希

当数据集很大时，通常采用构建签名矩阵的方法来进行相似度判定。但是当涉及到的文档很多时，单纯使用两两签名矩阵相互比较的方式需要比较的次数太多。因此，有人提出只比较相似度高的文档，相似度低的部分被略去，这就是局部敏感哈希的思想。

本次实验涉及的数据集不大，可以跳过签名矩阵的步骤，在计算哈希函数之后直接使用局部敏感哈希的方法筛选即可，示例代码如下：







1. 实验条件

运行带python编译环境的计算机1台。并根据实验需要安装必要的python库。

1. 实验步骤
2. 在源码1的基础上，测定出不同的哈希精度的条件下，最小哈希Jaccard估计值与实际Jaccard值之间的差别。
3. 读取文件夹下的products.txt，调用datasketch计算每一个ID的相似项，实验过程中注意选择合适的精度和阈值。
4. 思考题

**思考领域：命名实体识别/重名消歧**

在文件夹中有一个数据集CNKI-636603734021580000.txt，是同一姓名的作者近年来所发表论文的信息，但不是同一个作者所发表论文的信息，只是姓名相同。

该数据集包括了论文题名、合作者名、作者单位、文献来源、关键词。思考字段如何组合构建出最能识别出同一姓名作者之间的差异，以便于区分不同出相同姓名的不同作者。以流程图的形式展现出你的思考结果（使用相似项发现的方法）。