数据预处理

将global firm name 、us names 表中的数据从excel中提出来后去重，仅剩58008条公司名称数据，也就是说共58008所公司需要和143229个customer name（缩写）进行匹配（还没匹配出来）

```
import xlwt
import difflib


# global firm names 和 us names 的 Global Company Key 和 Company Name 提取
global_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'+
                                   '\Database Table\global firm names.xlsx')
global_worksh = global_workbk.sheet_by_name('0x77igavdumz8vul')
global_cpnames = global_worksh.col_values(colx = 7, start_rowx = 1)
global_cpkey = global_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
global_namekey = list(zip(global_cpnames, global_cpkey))


us_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'
                               +'\Database Table\\us names.xlsx')
us_worksh = us_workbk .sheet_by_name('76aqys7wh9axjpme')
us_cpnames = us_worksh.col_values(colx = 9, start_rowx = 1)
us_cpkey = us_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
us_namekey = list(zip(us_cpnames,us_cpkey))


# 两个文件的Global Company Key 和 Company Name列表合并后去重
Allnamekey_lst = list(set( global_namekey+us_namekey))
#按照Global Company Key进行排序
Allnamekey_lst.sort(key=lambda x:x[1])

# 将Global Company Key 和 Company Name 写入Allcompany表中
Allcompany = xlwt.Workbook()
Allcompany_sheet = Allcompany.add_sheet('sheet1')
name_list = ['Global Company Key','Company Name']
for i in name_list:
    Allcompany_sheet.write(0, name_list.index(i), i)

for namekey in Allnamekey_lst:
    Allcompany_sheet.write(Allnamekey_lst.index(namekey)+1,0,namekey[1])
    Allcompany_sheet.write(Allnamekey_lst.index(namekey)+1,1,namekey[0])

# 将customer表中的Customer Name列提取出，并与Allcompany 表中的 company name对比
customer_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata\Database
Table\customer.xlsx')
customer_worksh = customer_workbk.sheet_by_name('vozkvOioajsw5wov')
customer_names = customer_worksh.col_values(colx = 2, start_rowx = 1)
# customer_names 是客户缩写，Allcompany_names是公司全名
Allcompany_names = Allcompany_sheet.col_values(colx = 1,start_rowx = 1)
```

```
# 保存文件
Allcompany.save('Allcompany.xlsx')
```

论文阅读

查找有没有其他提供Compustat的公司全名和客户缩写的方法时对以下论文进行了阅读，大部分文章都是一句带过，不会详细叙述名称匹配和数据获取的过程，应该从字符串查找算法方面的论文入手学习才比较好：

1. Financial benefits and risks of dependency in triadic supply chain relationships.
2. Supply chain collaboration : impact on collaborative advantage and firm performance
3. Concentrate supply chain membership and financial performance: chain- and firm-level perspectives
4. 中国进口与全球经济增长:公司投资的国际证据
5. The U.S. syndicated loan market: Matching data
6. The Effects of Corporate and Operations Resources Similarity on the Acquisition Performance of the Acquiring Firms: The Role of Prior Acquisition Experience and Size
7. Financial reporting fraud and CEO pay-performance incentives
8. 基于供应链关系的股票收益预测研究
9. 一种改进的字符串匹配模型研究

字符串匹配算法

BM算法

1. **从右到左**字符比较法
2. 坏字符原则：当文本串中的某个字符跟模式串的某个字符不匹配时，我们称文本串中的这个失配字符为坏字符，此时模式串需要向右移动，移动的位数 = 坏字符在模式串中的位置 - 坏字符在模式串中最右出现的位置。此外，如果"坏字符"不包含在模式串之中，则最右出现位置为-1。
3. 好后缀原则：当字符失配时，后移位数 = 好后缀在模式串中的位置 - 好后缀在模式串上一次出现的位置，且如果好后缀在模式串中没有再次出现，则为-1。
   坏字符和好后缀原则综合使用，哪个移动距离少用哪个

   尝试用于customer name 和 company name的匹配

The U.S. syndicated loan market: Matching data 中的匹配理论

该论文的匹配理论是用于匹配跨数据集的数据匹配时，没有可以用的公共字段时。
记录 i 为不同数据集中的**公共字段**，定义一个**可变指示变量**$\gamma_{i,j}$ 给每一个**记录对** j，

当 记录 i 和这两个记录（记录对j) 都匹配的时候,$\gamma_{i,j}$记为 1 ,否则记为0.

每个y_j对应一个对记录j和公共字符串$i = 1, 2, 3, 4 \ldots N$匹配的结果。

本文理论应该可以用于上下游的建网