使用的环境：python3.9.7
使用的库：

- xlwt(1.3.0)高版本会报错
- xlrd(1.2.0)高版本会报错
- cleanco(2.2)
- xlsxwriter(3.0.1)
- jellyfish(0.9.0)

> 1.执行 `1 estabish updownstream ( full name and abbreviation).py`文件 生成 `1 Allcompany.xlsx` 文件，里面保存用全称与缩写建立的上下游关系数据
> 2.执行`2 preprocess full name.py`文件生成`2 All Company name.xlsx`文件，保存对公司全称清理后的名称
> 3. 执行`3 preprocess abbreviation.py`文件，生成`3 Abbreviation Data.xlsx`文件, 保存对客户缩写清理后的名称
> 4. 执行`4 name matching.py`文件，生成`4 Match Company name.xlsx`文件，将第二步和第三步清理后名称匹配，记录下与缩写匹配相似度前三的名称
> 5. 执行`5 determine the unique name.py`文件，生成`5 Vaild Matched data`文件，在第四步的基础上除去匹配无效的数据，确定一定有效的数据，无法确定是否一定有效或无效的数据保留，后期比对后进一步确定。

1. 获取要处理的基本数据

执行程序名为`1 estabish updownstream ( full name and abbreviation).py`程序，

```
import xlrd
import xlwt

# 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')
```

上面最后一行在读取 *global firm names.xlsx* 文件，使用时请改成**对应文件保存的位置**

```
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')
```

上面最后一行在读取 *us_ names.xlsx* 文件，使用时请**改成对应文件保存的位置**

```
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))
# 将customer表中的Customer Name列和Global Company Key 提取出
customer_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'+
                                     '\Database Table\customer.xlsx')
```

上面最后一行在读取*customer.xlsx*文件位置，使用时请**改成对应文件保存的位置**

```
customer_worksh = customer_workbk.sheet_by_name('vozkv0ioajsw5wov')
customer_downstream = customer_worksh.col_values(colx = 2, start_rowx = 1)
customer_upkey = customer_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
customer_up_down = list(zip(customer_upkey,customer_downstream))


# global firm names 和 us names 的Global Company Key 和 Company Name列表合并后去重
Allnamekey_
lst = list(set( global_namekey+us_namekey))
#对customer_up_down列表也去重
customer_up_down = list(set(customer_up_down))
#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(upstream)','Downstream']
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])

cust_num = [[i[1],0] for i in Allnamekey_lst]
#比较 upstream key(customer_up_down[i][0])和global company key(Allnamekey_lst[j][1])
for upkey_down in customer_up_down:
    for name_key in Allnamekey_lst:
        if upkey_down[0] == name_key[1]:
            Allcompany_sheet.write(Allnamekey_lst.index(name_key)+1
                                   ,2+cust_num[Allnamekey_lst.index(name_key)][1]
                                   ,upkey_down[1])
            cust_num[Allnamekey_lst.index(name_key)][1] += 1


# 保存文件
Allcompany.save('1 Allcompany.xlsx')
```

最后一行是保存*Allcompany.xlsx*文件的位置，可以不修改，默认保存在同py文件的文件夹下。

> Allcomany.xlsx文件里保存的有
> 第一列：Global Company Key 每个公司特有的一串数字
> 第二列：Company Name(upstream) 上游公司的全称
> 第三列以后：Downstream 与上有公司对应的，下游公司的缩写

> 对数据进行了哪些处理：
>
> 1. 里面上游公司的名称没有重复，都只出现一次
> 2. 数据都是按照 Global Company Key 公司特有的关键字 从小到大排列

在我的电脑上代码执行大约需要25min
执行后结果如下：

	Global Company Key	Company Name(upstream)	Downstream		
2	001004	AAR CORP	Other Government and Defense	Commercial	
3	001013	ADC TELECOMMUNICATIONS INC			
4	001019	AFA PROTECTIVE SYSTEMS INC	Not Reported		
5	001045	AMERICAN AIRLINES GROUP INC			
6	001050	CECO ENVIRONMENTAL CORP	Foreign		
7	001062	ASA GOLD AND PRECIOUS METALS			
8	001072	AVX CORP	Electronic Distributors	Not Reported	
9	001075	PINNACLE WEST CAPITAL CORP	Retail residential electric service	Other sources	
10	001076	PROG HOLDINGS INC	Furniture and Bedding	Medical	
11	001078	ABBOTT LABORATORIES	Other Emerging Markets	International	
12	001082	SERVIDYNE INC			
13	001084	WORLDS INC			
14	001094	ACETO CORP	Europe	United States	
15	001096	MORGUARD CORP			
16	001097	ACMAT CORP -CL A			
17	001104	ACME UNITED CORP	E-commerce	International	
18	001117	BK TECHNOLOGIES CORP	U.S. Government	Industrial	
19	001119	ADAMS DIVERSIFIED EQUITY FD			
20	001121	ADAMS RESOURCES & ENERGY INC	Not Reported		
21	001161	ADVANCED MICRO DEVICES	Sony Interactive Entertainment LLC	Not Reported	
22	001166	ASM INTERNATIONAL NV	10 Customers	South Korea	
23	001173	AEROSONIC CORP			
24	001177	AETNA INC	Medicaid	U.S. Federal Government	
25	001186	AGNICO EAGLE MINES LTD	Not Reported	4 Customers	
26	001209	AIR PRODUCTS & CHEMICALS INC	Merchant	On-site	
27	001210	AIR T INC	International	DELTA AIR LINES INC	

2. 对获取到的全称进行预处理:

执行`2 preprocess full name.py`文件
从第一步中得到的Allcompany.xlsx中获取所有公司的Allcpkey(每个公司特有的一串数字）和Allcpname（公司全名）

注 :== 第一行路径对应上面**第一部分代码中生成的Allcompany文件位置**

```
'''
从Allcompany中读取数据
 Allcpkey 用于保存公司缩写
 Allcpname用于保存公司全称
 '''
Allcompany = xlrd.open_workbook(r'C:\Users\jc\Documents\大学'
                                +'\\0大三其他\\_金融数据挖掘科研课题'
                                +'\Allcompany.xlsx')
Allcompany_sheet = Allcompany.sheet_by_name('sheet1')
Allcpkey = Allcompany_sheet.col_values(colx = 0, start_rowx = 1)
Allcpname = Allcompany_sheet.col_values(colx = 1, start_rowx = 1)
```

创建表Allcpdata,全称All Company Data表格，用于存储全称公司的：

> Global Company key, Full name, company type , country ， Cleande Full Name

其中 Cleande Full Name 表示去除了公司后缀的全称,用于后续与缩写的匹配
country 用于存储公司可能所在的国家，有些无法识别出就空着

1. 创建All company match表格的表头

```
# 1.创建All Company Data表格，写好表头
Allcpdata_wb = xlwt.Workbook()
Allcpdata = Allcpdata_wb.add_sheet('sheet1')
name_list = ['Global Company key', 'Full name',  'company type' , 'country'
,'Cleaned Full Name']

for i in range(len(name_list)):
    Allcpdata.write(0, i , name_list[i])
```

2. 将Global Company key写进表中 第i+1行,第0列

```
for i in range(len(Allcpkey)):
    Allcpdata.write(i+1, 0 , Allcpkey[i])
```

3. 将Full name写进表中 第i+1行，第1列

```
for i in range(len(Allcpname)):
    Allcpdata.write(i+1, 1, Allcpname[i])
```

4. 将company type写入表中 第i+1行，第2列, 用 cleanco中typesources()和match()判断

```
for i in range(len(Allcpname)):
    Allcpdata.write(i+1, 2, matches(Allcpname[i], typesources()))
```

5. 将country 写入表中 第i+1行，第3列, 用cleanco中match()和countrysources()判断

```
for i in range(len(Allcpname)):
    Allcpdata.write(i+1, 3,matches(Allcpname[i],countrysources()))
```

6. 清理不必要的后缀与符号，并转换为小写字符 写入表中，第i+1行，第4列，用clean_name()判断。

```
for i in range(len(Allcpname)):
    Allcpdata.write(i+1, 4, (clean_name(Allcpname[i]).lower()))
```

> **关于 clean_name 函数解释如下**:
>
> 1. 用cleanco中的basename删除后缀

```
(A.S./ N.V./  SA/SG S.A.与这一步共同进行单独处理)
(-CL A,-CL B,-CL C ,-CL I)单独处理
```

> 2. 删除未识别出的后缀

```
ETF , A/S  , -ADR , INC-OLD,group,
```

> 3. 再用cleanno中的basename删除一次，确保删除干净了

以下是clean_name方法，**基本能将company name中所有后缀与国家名都删除，并不丢失此外的其他字符。**

```
def clean_name(string):
    '''
    1. 用cleanco中的basename删除后缀
    (A.S./ N.V./  SA/SG S.A.与这一步共同进行单独处理)
    (-CL A,-CL B,-CL C ,-CL I)单独处理
    2.删除未识别出的后缀
    ETF , A/S  , -ADR , INC-OLD,group,
    3. 再用cleanno中的basename删除一次，确保删除干净了
    ---
    return 公司清理后缀后的名称
    '''
    #1.
    # 截取最后5个判断是否有-CL_?类后缀，如果有则删去
    str_suffix0 = string[:len(string)-6:-1][::-1].replace(' ','')
    if '-CL' in str_suffix0:
        string = string[:string.find('-')]
    string = re.sub(r'[\W]',' ',string) # 首先删去字符影响，替换为空格
    string = cleanco.basename(string) # 用cleanco里自带的basename做第一次后缀清除
    clean_suffix0 = ['A/S','N.V.','SA/AG','S.A.']
    for cs in clean_suffix0:
        if cs in string:
            string = string[:string.rfind(cs[0])]
    #2.
    # 进行第二次后缀清理
```

```
    # 截取最后 5个字符,并删除其中的空格 为什么是5个（因为根据观察，自动清除的后缀里，最长的是GROUP，所以取最后5个判断不会有漏）
    str_suffix = string[:len(string)-6:-1][::-1].replace(' ','')
    clean_suffix = ['CO','ETF','AS','ADR','SA','AG','OLD','GROUP'] # 判断是否包含这些后缀
    for cs in clean_suffix:
        if cs in str_suffix:
            string = string[:string.rfind(cs[0])]
    #3.
    #用basename进行第三次后缀处理
    string = cleanco.basename(string)
    return string
```

最后进行文件的保存：

```
Allcpdata_wb.save('2 All Company Data.xlsx')
```

数据处理结果如下：

	Global Company key	Full name	company type	country	Cleaned Full Name
2	001004	AAR CORP	Corporation	PhilippinesUnited States of America	aar
3	001013	ADC TELECOMMUNICATIONS INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	adc telecommunications
4	001019	AFA PROTECTIVE SYSTEMS INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	afa protective systems
5	001045	AMERICAN AIRLINES GROUP INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	american airlines
6	001050	CECO ENVIRONMENTAL CORP	Corporation	PhilippinesUnited States of America	ceco environmental
7	001062	ASA GOLD AND PRECIOUS METALS	Limited Liability Company	Norway	asa gold and precious metals
8	001072	AVX CORP	Corporation	PhilippinesUnited States of America	avx
9	001075	PINNACLE WEST CAPITAL CORP	Corporation	PhilippinesUnited States of America	pinnacle west capital
10	001076	PROG HOLDINGS INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	prog holdings
11	001078	ABBOTT LABORATORIES			abbott laboratories
12	001082	SERVIDYNE INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	servidyne
13	001084	WORLDS INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	worlds
14	001094	ACETO CORP	Corporation	PhilippinesUnited States of America	aceto
15	001096	MORGUARD CORP	Corporation	PhilippinesUnited States of America	morguard
16	001097	ACMAT CORP -CL A	Corporation	PhilippinesUnited States of America	acmat
17	001104	ACME UNITED CORP	Corporation	PhilippinesUnited States of America	acme united
18	001117	BK TECHNOLOGIES CORP	Corporation	PhilippinesUnited States of America	bk technologies
19	001119	ADAMS DIVERSIFIED EQUITY FD			adams diversified equity fd
20	001121	ADAMS RESOURCES & ENERGY INC	Corporation	PhilippinesUnited States of AmericaUnited States of America	adams resources energy
21	001161	ADVANCED MICRO DEVICES			advanced micro devices
22	001166	ASM INTERNATIONAL NV	Limited Liability Company		asm international
23	001173	AEROSONIC CORP	Corporation	PhilippinesUnited States of America	aerosonic

3.对获取到的缩写进行预处理

```
执行`3 preprocess abbreviation.py`文件，基本步骤与第二步差不多
```

第一步， 从 `customer.xlsx` 中读取数据， company_abbr 用于保存公司缩写, company_upkey用于保存用于识别缩写对应的上游公司的特殊数字串

第一行中custome_wb的路径对应保存customer.xlsx的位置，使用时注意修改

```
customer_wb = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'+
                                 '\Database Table\customer.xlsx')
customer = customer_wb.sheet_by_name('vozkv0ioajsw5wov')
```

```
company_abbr = customer.col_values(colx = 2, start_rowx = 1)
company_upkey = customer.col_values(colx = 0, start_rowx = 1)
# 组成列表
abbr_upkey_zip = list(set((zip(company_abbr,company_upkey))))
# 第i组： abbr_upkey_zip[i][0]公司缩写 abbr_upkey_zip[i][1]上游公司key
```

第二步，创建表Abbrdata,全称Abbreviation Data表格，用于存储缩写公司的：
Global Company key(upstream), Abbreviation, company type , country ， Cleaned Abbreviation
其中 Cleaned Abbreviation 表示去除了公司后缀的缩写,用于与公司全称的匹配
country 用于存储公司可能所在的国家，有些无法识别出就空着。
1.创建Abbreviation Data表格，写好表头

```
Abbrdata_wb = xlwt.Workbook()
Abbrdata = Abbrdata_wb.add_sheet('sheet1')
name_list = ['Global Company key(upstream)', 'Abbreviation',
              'company type' , 'country' ,'Cleaned Abbreviation']

for i in range(len(name_list)):
    Abbrdata.write(0, i , name_list[i])
```

2.将Global Company key(upstream)写进表中 第i+1行,第0列

```
for i in range(len(abbr_upkey_zip)):
    Abbrdata.write(i+1, 0 , abbr_upkey_zip[i][1])
```

3.将Abbreviation写进表中 第i+1行，第1列

```
for i in range(len(abbr_upkey_zip)):
    Abbrdata.write(i+1, 1, abbr_upkey_zip[i][0])
```

4.将company type写入表中 第i+1行，第2列，用 cleanco中typesources()和match()判断

```
for i in range(len(abbr_upkey_zip)):
    Abbrdata.write(i+1, 2, matches(abbr_upkey_zip[i][0], typesources()))
```

5.将country 写入表中 第i+1行，第3列, 用cleanco中match()和countrysources()判断

```
for i in range(len(abbr_upkey_zip)):
    Abbrdata.write(i+1, 3,matches(abbr_upkey_zip[i][0],countrysources()))
```

6.清理不必要的后缀与符号，并转换为小写字符 写入表中，第i+1行，第4列，用cleanco.clean_name()判断

```
for i in range(len(abbr_upkey_zip)):
    Abbrdata.write(i+1, 4, (clean_name(abbr_upkey_zip[i][0]).lower()))

Abbrdata_wb.save('3 Abbreviation Data.xlsx')
```

最后保存数据在`3 Abbreviation.xlsx`
数据展示如下：

	Global Company key(upstream)	Abbreviation	company type	country	Cleaned Abbreviation
2	186360	Russia			russia
3	008099	POLR			polr
4	024797	Clarion Water			clarion water
5	030098	Not Reported			not reported
6	138122	Ildong Pharmaceutical Co Ltd	LimitedCorporation	Hong KongIsraelNew ZealandPakistanUnited KingdomUnited States of AmericaUr	ildong pharmaceutical
7	016456	U.S. Department of Veterans Affairs			u s department of veterans affairs
8	006494	Latin America			latin america
9	154758	Private pay and other	Private Company		private pay and other
10	007923	Metals, Construction			metals construction
11	140044	Japan			japan
12	011169	U.S. Government			u s government
13	018699	China			china
14	025893	BrightView Holdings Inc	Corporation	PhilippinesUnited States of AmericaUnited States of America	brightview holdings
15	135990	Texas			texas
16	160570	Medium- and Heavy-duty Truck OEMs			medium and heavy duty truck oems
17	004412	France			france
18	026721	United Kingdom			united kingdom
19	062823	International			international
20	018888	International			international
21	135990	Business Solutions Customers			business solutions customers
22	119574	Jenne Corporation	Corporation	United States of America	jenne
23	022632	McKesson Corp	Corporation	PhilippinesUnited States of America	mckesson
24	187690	Community Health Network			community health network
25	025747	Americas			americas
26	001704	Micron Technology Inc.	Corporation	PhilippinesUnited States of AmericaUnited States of AmericaUnited States of Am	micron technology
27	032303	Not Reported			not reported
28	064072	Trading customers			trading customers
29	027780	Best Buy Co. Inc.	CorporationCorporation	PhilippinesUnited States of AmericaUnited States of AmericaUnited States of Am	best buy

4.进行第一次数据匹配

执行`4 name matching.py`文件
将第二步中清理后的**公司全名**的数据与第三步中清理后的**缩写**数据进行匹配。

```
import xlrd
import xlwt
import xlsxwriter
import re
import jellyfish

'''
从All Company data.xlsx 中读取公司名清理后的名字
Cleaned Full Name(E列:清理后的名字)
'''
```

1. 读取All company Data.xlsx中数据：
   cleaned_name:清理后的公司全称数据，用于匹配
   full_name:公司原本的全称数据，用于匹配时写入最终表格
   key:公司KEY,可以识别公司全称的数字串
   下面第一行代码中的路径对应**第二步中生成的2 All company Data.xlsx**的位置，使用时请注意修改

```
Allcompanydata_wb = xlrd.open_workbook(r'C:\Users\jc\Documents\大学'
                                       +'\\0大三其他\\_金融数据挖掘科研课题'
                                       +'\\PyProgram'
                                       +'\2 All Company Data.xlsx')
Allcompanydata = Allcompanydata_wb.sheet_by_name('sheet1')

cleaned_name = Allcompanydata.col_values(colx = ord('E')-ord('A'), start_rowx = 1)
full_name = Allcompanydata.col_values(colx = 1, start_rowx = 1 )
key = Allcompanydata.col_values(colx = 0, start_rowx = 1 )
```

下面第一行对应第3步中生成的3 Abbreviation Data.xlsx文件，使用时请注意修改

2. 从Abbreviation.xlsx中读取数据：
   clabbr: 表示清理后的客户公司缩写数据 cleaned abbrevaiton
   abbr: 表示清理前原本的客户缩写 abbreciation
   upkey: 表示缩写客户对应的上游公司KEY

```
'''
从Abbreviation.xlsx中读取 客户缩写 与 上游公司KEY
'''
Abbreviation_wb = xlrd.open_workbook(r'C:\\Users\\jc'
                                     +'\\Documents\\大学\\0大三其他'
                                     +'\\_金融数据挖掘科研课题\\PyProgram'
                                     +'\3 Abbreviation Data.xlsx')
Abbreviation = Abbreviation_wb.sheet_by_name('sheet1')
clabbr = Abbreviation.col_values(colx = ord('E')-ord('A'), start_rowx = 1)
abbr = Abbreviation.col_values(colx = 1, start_rowx = 1)
upkey = Abbreviation.col_values(colx = 0, start_rowx = 1)
# 上游公司KEY 和 客户缩写 客户清理以后的缩写 组成元组（不可修改，并去掉重复的）
upkey_abbr = list(set(zip(upkey,abbr,clabbr)))
```

3. 创建保存匹配结果的表格Match companyname.xlsx,并创建表头

```
Matchcpname_wb = xlwt.Workbook()
Matchcpname = Matchcpname_wb.add_sheet('sheet1')
name_list = ['Upstream key','customer abbreviation ','Full name','similarity',
              'Full name','similarity','Full name','similarity'
              ,'Cleaned Abbreviation']# 0 1

for i in range(len(name_list)):
    Matchcpname.write(0, i , name_list[i])

'''
第一列：Upstrean key ---- upkey_abbr[i][0]
第二列：customer abbreviation ---- upkey_abbr[i][1]
第三列及以后： 匹配出的公司全称 / JW算法算出的相似度
第九列：Cleaned Abbreviation 清理后的缩写数据，用于下一步的匹配
'''
```

4. 进行数据的匹配和写入，详细步骤在注释中都已讲明

```
# 1. 写入第一列Upstream key,第二列customer abbreviation,第九列，清理后的数据
for i in range(len(upkey_abbr)):
    Matchcpname.write(i+1, 0 , upkey_abbr[i][0])
    Matchcpname.write(i+1, 1, upkey_abbr[i][1])
    Matchcpname.write(i+1, 8, upkey_abbr[i][2])

# 2.写入第三列及以后的数据
# (1) 遍历一遍 upkey_abbr 中的 upkey---ukabbr[0], clabbr---ukabbr[2]
for uk_abbr in upkey_abbr:
    indexnow = upkey_abbr.index(uk_abbr)  # 表示读取到第几个缩写了

    matchcp = []    # 用于临时储存匹配相似度大于0.3时：[公司名，相似度]
    # (2) 遍历一遍 cleaned_name 中的 clname
    for clname in cleaned_name:
            # 用JW算法进行匹配，并用JW算法计算出相似度
        similarity = jellyfish.jaro_winkler_similarity(uk_abbr[2], clname)
        # 当相似度大于0.5时，写入临时数组中
        if(similarity > 0.5):
            matchcp.append([clname,similarity])
```

```
    #当名称匹配后没有大于0.5的相似度的数据时，跳过本次写入，相当于没有匹配到合适的公司
    if len(matchcp) == 0 :
        continue

       # 按匹配度顺序从大到小排序，并取出前三的[公司名，相似度]
    matchcp = sorted(matchcp, key=lambda matchcp:matchcp[1],reverse =True)

    #如果没有匹配到三个的话（匹配出的个数小于3），就全写进去
    if len(matchcp) < 3:
        col = 0
        for i,j in matchcp:
            Matchcpname.write(indexnow+1, col+2,
                              full_name[cleaned_name.index(i)])
            Matchcpname.write(indexnow+1, col+3, j )
            col += 2

    else:
    # 如果大于3个的话，就只写前三个
        for i,j in zip(range(0,5,2),range(3)):
            Matchcpname.write(indexnow+1, i+2,
                              full_name[cleaned_name.index(matchcp[j][0])])
            Matchcpname.write(indexnow+1, i+3,
                              matchcp[j][1])
```

最后保存数据在 4 Match Company name.xlsx文件中

```
Matchcpname_wb.save('4 Match Company name.xlsx')
```

数据展示如下：

Upstream key	customer abbrevation	Full name	similarity	Full name	similarity	Full name	similarity	Cleaned Abbrevation
184702	Cisco Systems Inc	CISCO SYSTEMS INC	1	OSI SYSTEMS INC	0.918414918	CASA SYSTEMS INC	0.880769231	cisco systems
175342	Yunnan Taoping IoT Co., LTD	YUNNAN TIN CO LTD	0.911111111	YUNNAN BAIYAO GROUP CO LTD	0.874352548	YUNNAN LUOPING ZINC & ELEC	0.872222222	yunnan taoping iot
142540	Europe	EUPE CORP BHD	0.911111111	EUROTEL SA	0.90952381	EUROESPES SA	0.9	europe
019526	South Carolina	SOUTH CAROLINA ELEC & GAS CO	0.92173913	SOUTHERN CALIFORNIA GAS CO	0.89047619	SOUTHERN CALIF BANCORP	0.884415584	south carolina
024878	Not Reported	NORBORD INC	0.828571429	NOTORIOUS PICTURES SPA	0.810740741	NORTH COPPER CO LTD	0.807407407	not reported
030397	Iskenderun Demir Ve Celik AS	ISKENDERUN DEMIR VE CELIK AS	1	ISEWAN TERMINAL SERVICE CO	0.763658615	EREGLI DEMIR VE CELIK FABRIK	0.760225564	iskenderun demir ve celik
026839	Kohl's Corp	KOHL'S CORP	1	KOHNAN SHOJI CO LTD	0.825	KOHSOKU CORP	0.822222222	kohl s
018494	Kentucky	KENTUCKY POWER	0.914285714	KENTUCKY UTILITIES CO	0.888888889	KENTUCKY BANCSHARES INC	0.884210526	kentucky
023714	Managed Care & Other Third Party Payors	MANAC INC	0.827027027	MANGOLD AB	0.791891892	MARR	0.762162162	managed care other third party payors
100644	Norway	NOVORAY CORP	0.879365079	NORDA ASA	0.875555556	NORMA GROUP SE	0.875555556	norway
022674	Small Business Administration (SBA)	SASINI PLC	0.754545455	SALORA INTERNATIONAL LTD	0.75030303	SABAA INTERNATIONAL	0.74784689	small business administration sba
026368	Outside of the United States	LOTUS EYE HOSPITAL&INSTITUTE	0.765010352	OUE LTD	0.761904762	UNITED STATES DIESEL-HEAT LP	0.736349206	outside of the united states
022674	Utah	UAGH INC	0.85	UTAH MEDICAL PRODUCTS INC	0.838095238	TA CORPORATION LTD	0.833333333	utah
179817	Merchant sales	MERCHANTS TRUST PLC	0.885627706	MERCHANTS FINANCIAL GROUP	0.881077694	MERCATOR MINERALS LTD	0.879271709	merchant sales
032345	Mobile device OEMs	MOBILICOM LTD	0.9	MOBILE WORLD INVESTMENT CORP	0.879408213	MOBILE LADS CORP	0.863636364	mobile device oems
005242	Gas and convenience	GASCO INVERSIONES SA	0.806518748	GAN LTD	0.775438596	GUANGDONG TECSUN SCIENCE	0.7750387	gas and convenience
164132	Pier 1 Imports Inc	PIER 1 IMPORTS INC/DE	0.933333333	PIERER MOBILITY AG	0.856190476	PIERIS PHARMACEUTICALS INC	0.820779221	pier 1 imports
031914	Alabama	ALABAMA POWER CO	0.907692308	ALABAMA GRAPHITE CORP	0.8875	ALTABA INC	0.879365079	alabama
064163	Europe	EUPE CORP BHD	0.911111111	EUROTEL SA	0.90952381	EUROESPES SA	0.9	europe
004213	International	INTERNATIONAL COAL GROUP INC	0.944444444	INTERNATIONAL CARE CO	0.944444444	ZK INTERNATIONAL GROUP	0.9375	international
001718	Circle K Denmark A/S	CIRCLE SPA	0.875	CIRCLE ENTERTAINMENT INC	0.853333333	CIRCLE STAR ENERGY CORP	0.85	circle k denmark
066393	International	INTERNATIONAL COAL GROUP INC	0.944444444	INTERNATIONAL CARE CO	0.944444444	ZK INTERNATIONAL GROUP	0.9375	international
008902	True Value Co	TRUE VALUE CO	1	TRUEBLUE INC	0.915	TRUECALLER AB	0.88	true value
004842	Dick's Sporting Goods, Inc.	DICKS SPORTING GOODS INC	0.94047619	BIG 5 SPORTING GOODS CORP	0.866900093	DIC CORPORATION	0.8	dick s sporting goods
007085	Lowe's Companies Inc.	LOWE'S COS INC	0.925	ROWAN COMPANIES LTD	0.85	DONG SUH COMPANIES INC	0.844907407	lowe s companies
010598	Florida	FLORA CORP LTD	0.942857143	FLORIDA GAMING CORP	0.9	FLORIDIENNE SA/NV	0.880519481	florida
034941	TC Energy Keystone Pipeline LP	TC ENERGY CORP	0.866666667	TECH-TOP ENGINEERING CO LTD	0.77	QTC ENERGY PCL	0.767195767	tc energy keystone pipeline

5. 进行匹配数据的筛查：保留有效数，删去无效数据，无法确定是否一定有效或一定无效的数据暂时保留，后期进行比对

执行`5 determine the unique name.py`文件

```
1. 首先读取Match Company name-copy.xlsx中的数据
'''
import xlrd
import xlwt



def write_vaild_data(i):
    '''
    记录一定有效的similarity=1的数据
    '''
    Vaild.write(i+1, 0,uk_abbr[0])# 第一列是global key(upstream)
    Vaild.write(i+1, 1,uk_abbr[1])# 第二列是 Abbreviation
    Vaild.write(i+1, 2, cpname[i][0])# 第三列是 similarity==1时的名称
    Vaild.write(i+1, 3, similarity[i][0])# 第四列是 similarity

def write_spare_data(i):
    '''
    记录不一定有效，也不一定无效的备用数据
    '''
    Vaild.write(i+1, 0,uk_abbr[0])# 第一列是global key(upstream)
    Vaild.write(i+1, 1,uk_abbr[1])# 第二列是 Abbreviation
    for n in range(0,3):
        Vaild.write(i+1, 2*(n+1), cpname[i][n])# cpname写入2，4，6
    for n in range(0,3):
        Vaild.write(i+1, 2*(n+1)+1, similarity[i][n])# similarity写入3，5，7
```

下面第一行中路径为第四步中生成的 4 Match Company name.xlsx 文件路径，使用时请注意修改

1. 首先，读取`4 Match Company name.xlsx`文件，读取其中的全部数据：

```
Matchcpname_wb = xlrd.open_workbook(r'C:\\Users\\jc\\Documents\大学\\0大三其他'
                                    +'\\_金融数据挖掘科研课题\\PyProgram'
                                    +'\\4 Match Company name.xlsx')
Matchcpname = Matchcpname_wb.sheet_by_name('sheet1')
```

```
# 第一列为upstream key ,第二列为 abbreviation，第9列为Cleaned abbreviation
upstream_key  = Matchcpname.col_values(colx = 0 , start_rowx = 1)
abbr = Matchcpname.col_values(colx = 1 ,start_rowx = 1)
cl_abbr = Matchcpname.col_values(colx = 8, start_rowx = 1)
```

将 upstream key,abbreviation,cleaned abbreviation打包为`upkey_abbr`的元组列表

```
upkey_abbr = list(zip(upstream_key,abbr,cl_abbr))
```

打包后继续匹配数据：

```
# 第三列和第四列为全名1与相似度1
cpname1 = Matchcpname.col_values(colx = 2,start_rowx = 1)
similarity1 = Matchcpname.col_values(colx = 3,start_rowx = 1)
# 第五列和第六列为全名2与相似度2
cpname2 = Matchcpname.col_values(colx = 4,start_rowx = 1)
similarity2 = Matchcpname.col_values(colx = 5,start_rowx = 1)
# 第七列和第八列为全名3与相似度3
cpname3 = Matchcpname.col_values(colx = 6,start_rowx = 1)
similarity3 = Matchcpname.col_values(colx = 7,start_rowx = 1)
```

将获取的数据继续打包,
`cpname1`,`cpname2`,`cpname3`分别对应与缩写匹配的第1，2，3个公司名
`similarity1`， `similarity2`,`similarity3`分别对应匹配的公司名与缩写的相似度

```
cpname = list(zip(cpname1,cpname2,cpname3))
similarity = list(zip(similarity1,similarity2,similarity3))
```

2. 按照以下步骤，对数据进行保留和删除
   保留有效数据，跳过无效数据
   (0) len(cpname1) == 0 时，直接跳过，一定属于无效数据
   (1) similarity == 1：一定属于有效数据
   (2) 当abbr属于国家名时：一定属于无效数据
   (3) 当abbr中是`数字+customer`的形式：匹配出的属于无效数据
   (4) 当cl*abbr只在一个`cpname[i]*`里面完全出现时，属于有效数据，只保留匹配的唯一cpname[i]_`
   (5) 当similarity1-similarity2 > 0.1 ,第一个数据为有效数据
   (5) 当cl_abbr 在cpname1,cpname2,cpname里都有时，匹配出的数据属于无效数据
   (6) 当不属于无效数据也不是一定有效的数据，保留cpname123的数据，进行人工匹配

先创建 Vaild Matched data.xlsx，用于保存最终数据
第一列为 global key(upstream) 对应上游公司的key
第二列为 Abbreviation 对应公司缩写
第三列及以后为 Vaild matched data 有效数据和 Similarity 相似程度

```
Vaild_wb = xlwt.Workbook()
Vaild = Vaild_wb.add_sheet('sheet1')
name_list = ['Global key(upstream)','Abbreviation ','Vaild matched name'
             ,'similarity','Spare name','similarity','Spare name','similarity'
             ,'Reason']# 0 1

for i in range(len(name_list)):
    Vaild.write(0, i , name_list[i])
```

按照上面缩写原则，判断数据是否有效或无效：
若有效，则写入后continue，进入下一个数据的有效或无效的判断
若无效，直接写明无效原因后continue，进行下一个数据有效或无效的判断

```
nullabbr = ['Americas','Florida','Canada','Japan','United States','Europe'
           ,'China','Italy','New York', 'Customers','Reason','Australia']
for i in range(0,10):
    nullabbr.append(str(i)+' Customers')

# 0.遍历upkey和abbr 当数据有效时id_data = 1,无效时id_data = 0
for uk_abbr in upkey_abbr:
    id_data = 1
    i = upkey_abbr.index(uk_abbr) # 此时下标
    # upkey和abbr对应的cpname_和similarity_分别是cpname_[i]和similarity_[i]
    # 0. abbr的cpname长度是否为0
    if len(cpname1[i]) == 0:
        id_data = 0

    if id_data == 0:
        continue
    else:
        # 1. 如果相似度=='1',一定是有效数据
        if similarity[i][0] == 1:
            # 都是i+1行
            write_vaild_data(i)# 记录数据后跳过本次循环，进入下一次循环
```

```
            Vaild.write(i+1, 8, 'similarity=1 数据一定有效 ')# 写入原因
            continue

        # 2. abbr是否属于国家地区名或’数字+customer'
        for nabbr in nullabbr:
            if nabbr in uk_abbr[1]:# 如果属于国家地区名或数字，数据无效
                id_data = 0
                # 写入原因
                Vaild.write(i+1, 8, '包括无法识别是否正确的字符串，数据一定无效')
                break

        if id_data == 0:
            continue
        else:
            # 3.当uk_abbr[2]只在cpname[i][_]其中一个出现时，出现的那个数据有效
            appear_n = [0,0,0]
            for n in range(0,3):
                if uk_abbr[2] in cpname[i][n].lower():
                    appear_n[n] = 1
            if appear_n.count(1) == 3: # 4. 数据中全包含cl_abbr,数据无效
                id_data = 0
                # 写入原因
                Vaild.write(i+1, 8, '匹配数据中都包含缩写字符串'+
                                    '，无法判断，匹配一定无效')

            # 5. 数据中只有一个全包含cl_abbr,数据有效且可写入
            if appear_n.count(1) == 1:
                for n in appear_n:
                    if n == 1:
                        # 第一列是global key(upstream)
                        Vaild.write(i+1, 0,uk_abbr[0])
                        # 第二列是 Abbreviation
                        Vaild.write(i+1, 1,uk_abbr[1])
                        # 第三列是 appear_n[n]==1时的名称cpname[i][n]
                        Vaild.write(i+1, 2, cpname[i][n])
                        # 第四列是 similarity[i][n]
                        Vaild.write(i+1, 3, similarity[i][n])
                        # 写入原因
                        Vaild.write(i+1, 8,
                                    '数据中只有一个完全包含，缩写数据一定有效')
```

```
            else:
                if id_data == 0:
                    continue
                else:
                    if(similarity[i][0]-similarity[i][1]>0.1):
                        # 6. 当第1个和第2个的相似度大于0.1时，第一个为有效数据
                        # 写入原因
                        Vaild.write(i+1, 8, '前后两字符串相似度之差大于0.1，'
                                            +'数据一定有效')
                        write_vaild_data(i) # 写入一定有效的数据
                        continue

                    # 7. 不满足以上任意一种情况，不一定有效，也不一定无效
                    # 则三种情况都写入，进行人工比对
                    Vaild.write(i+1, 8, '应人工识别，备用数据')# 写入原因
                    write_spare_data(i)
```

最终保存数据在`5 Vaild Maatched data.xlsx`文件中

```
Vaild_wb.save('5 Vaild Matched data.xlsx')
```

数据最终结果如下：

	Global key(upstream)	Abbreviation	Vaild matched name	similarity	Spare name	similarity	Spare name	similarity	Reason
2	184702	Cisco Systems Inc	CISCO SYSTEMS INC	1					similarity=1 数据一定有效
3	175342	Yunnan Taoping IoT Co., LTD	YUNNAN TIN CO LTD	0.911111111	YUNNAN BAIYAO GROUP CO LTD	0.874352548	YUNNAN LUOPING ZINC & ELEC	0.872222222	应人工识别，备用数据
4									包括无法识别是否正确的字符串，数据一定无效
5	019526	South Carolina	SOUTHERN CALIFORNIA GAS CO	0.89047619					数据中只有一个完全包含，缩写数据一定有效
6	024878	Not Reported	NORBORD INC	0.828571429	NOTORIOUS PICTURES SPA	0.810740741	NORTH COPPER CO LTD	0.807407407	应人工识别，备用数据
7	030397	Iskenderun Demir Ve Celik AS	ISKENDERUN DEMIR VE CELIK AS	1					similarity=1 数据一定有效
8	026839	Kohl's Corp	KOHL'S CORP	1					similarity=1 数据一定有效
9									匹配数据中都包含缩写字符串，无法判断，匹配一定无效
10	023714	Managed Care & Other Third Party Payors	MANAC INC	0.827027027	MANGOLD AB	0.791891892	MARR	0.762162162	应人工识别，备用数据
11	100644	Norway	NOVORAY CORP	0.879365079	NORDA ASA	0.875555556	NORMA GROUP SE	0.875555556	应人工识别，备用数据
12	022674	Small Business Administration (SBA)	SASINI PLC	0.754545455	SALORA INTERNATIONAL LTD	0.75030303	SABAA INTERNATIONAL	0.74784689	应人工识别，备用数据
13									包括无法识别是否正确的字符串，数据一定无效
14	022674	Utah	UTAH MEDICAL PRODUCTS INC	0.838095238					数据中只有一个完全包含，缩写数据一定有效
15	179817	Merchant sales	MERCHANTS TRUST PLC	0.885627706	MERCHANTS FINANCIAL GROUP	0.881077694	MERCATOR MINERALS LTD	0.879271709	应人工识别，备用数据
16	032345	Mobile device OEMs	MOBILICOM LTD	0.9	MOBILE WORLD INVESTMENT CORP	0.879408213	MOBILE LADS CORP	0.863636364	应人工识别，备用数据
17	005242	Gas and convenience	GASCO INVERSIONES SA	0.806518748	GAN LTD	0.775438596	GUANGDONG TECSUN SCIENCE	0.7750387	应人工识别，备用数据
18	164132	Pier 1 Imports Inc	PIERER MOBILITY AG	0.856190476					数据中只有一个完全包含，缩写数据一定有效
19	031914	Alabama	ALABAMA POWER CO	0.907692308	ALABAMA GRAPHITE CORP	0.8875	ALTABA INC	0.879365079	应人工识别，备用数据
20									包括无法识别是否正确的字符串，数据一定无效
21									匹配数据中都包含缩写字符串，无法判断，匹配一定无效
22	001718	Circle K Denmark A/S	CIRCLE SPA	0.875	CIRCLE ENTERTAINMENT INC	0.853333333	CIRCLE STAR ENERGY CORP	0.85	应人工识别，备用数据
23									匹配数据中都包含缩写字符串，无法判断，匹配一定无效
24	008902	True Value Co	TRUE VALUE CO	1					similarity=1 数据一定有效
25	004842	Dick's Sporting Goods, Inc.	DICKS SPORTING GOODS INC	0.94047619	BIG 5 SPORTING GOODS CORP	0.866900093	DIC CORPORATION	0.8	应人工识别，备用数据
26	007085	Lowe's Companies Inc.	LOWE'S COS INC	0.925	ROWAN COMPANIES LTD	0.85	DONG SUH COMPANIES INC	0.844907407	应人工识别，备用数据
27									包括无法识别是否正确的字符串，数据一定无效
28	034041	TC Energy Keystone Pipeline LP	TC ENERGY CORP	0.966666667	TECH-TOP ENGINEERING CO LTD	0.77	OTC ENERGY PCL	0.767406767	应人工识别，备用数据