

1 在看了一些文章后猜想可能可以用于判断企业上下游的方法：

1. 通过企业A能否成为企业B供应商，可确定AB企业的上下游关系：若A企业能成为B企业供应商，则A为上游企业，B为下游企业。同样若企业A与企业C存在竞争关系，则企业C与企业B可能也存在上下游关系
2. 标准行业分类 SIC代码
 - 标准行业分类 (SIC) 代码是四位数的数字代码，根据公司的业务活动对公司所属的行业进行分类。CRSP标准工业分类 (SIC) 代码可以按行业识别现有客户和潜在客户。因此使用SIC代码可以通过识别出企业间的客户，从而可确定企业间的上下游关系。
 - 通过SIC还可以识别同类公司，从而识别同行业的竞争对手，可能可以确定上下游关系。
 - 尽管如今标准工业分类代码大部分被六位数的北美工业分类系统 (NAICS) 所取代已被替换，但包括 SEC 在内的政府机构和一些公司今天仍在使用 SIC 代码。因此**猜想SIC代码是否可以作为上下游判断中的检验性代码，以提高判断的准确性。**
3. 如今标准工业分类代码大部分被六位数的北美工业分类系统 (NAICS) 所取代。使用分层的六位数编码系统，将所有经济活动分为 20 个不同的行业部门。其中五个部门主要是生产商品的部门，而其余 15 个部门提供某种类型的服务。每家公司都会收到一个主要的 NAICS 代码，表明其主要业务线。此主要代码由在过去一年中为指定地点的公司产生最大收入的代码定义确定。研究NAICS的代码，NAICS 代码的前两位数字表示公司经营的最大业务部门。第三位数字表示公司的子行业，第四位数字表示公司所属的行业组，代码的第五位数字反映了公司特定的经营行业。第六位也是最后一位数字表示公司的特定国家行业。与SIC代码相同，**使用NAICS代码同样可以按行业识别现有客户和潜在客户，并确定企业间上下游关系。**

2 找到Shahrur的论文，对通过基准投入产出矩阵来判断企业上下游关系的方法进行进一步的学习，但可惜全文阅读需要订阅，没能找到更多通过基准投入产出矩阵来判断企业上下游关系的论文了，这个方法的研究只能暂时停下。

3 重复阅读《Fee_Thomas_2004_Sources of gains in horizontal me》

文章中在第14页中讲述了如何将客户的缩写和完整的公司数据相连接的算法。文章中所用到的简单的模糊查询，以下代码已经能够实现，并且在昨天的基础上已经能无视大小写误差：

```
import re

file_list = [
    {
        "type": "dir",
        "size": "123",
        "name": "access.log",
    },
    {
        "type": "dir",
        "size": "123",
        "name": "access.log.gz",
    },
    {
        "type": "dir",
        "size": "123",
        "name": "error.log",
    },
    {
```

```

        "type": "dir",
        "size": "123",
        "name": "access-auth.log",
    },
    {
        "type": "dir",
        "size": "123",
        "name": "Access-Auth.log",
    },
]

def fuzzy_finder(key, data):
    """
    模糊查找器
    :param key: 关键字
    :param data: 数据
    :return: list
    """
    # 结果列表
    suggestions = []
    key = key.lower()
    pattern = '.*%s.*' % key
    # 编译正则表达式
    regex = re.compile(pattern)
    for item in data:
        # 检查当前项是否与regex匹配,并无视大小写,以小写来进行匹配
        name = item['name'].lower()
        match = regex.search(name)
        if match:
            # 如果匹配,就添加到列表中
            suggestions.append(item)

    return suggestions

# 搜索关键字
keys = "AcCeSs"
result = fuzzy_finder(keys, file_list)
print(result)

```

文章中关于辨别客户和供应商的方法是先生成数据库来识别客户，再通过反转数据库来识别供应商，这段的论述看的其实不太理解，但文章中也有语句表明：供应商可直接从Compustat的行业细分文件中确定，也许从这里可以找到新的突破口。