数据预处理：数据集成

覃雄派

数据预处理：数据集成

提纲

- 数据预处理：数据集成
  - 实体匹配
  - 编辑距离

数据预处理：数据集成

- 数据预处理：数据集成

数据预处理 | 探索式分析 | 分析与建模 | 结果呈现与解读

Cleaning, Transformation &Integration

数据预处理：数据集成

数据预处理：数据集成

- 数据集成（Data Integration）
  - 整合多源数据，形成统一的数据视图

Brand	Product	Region	Price
Apple	iPhone6S	Beijing	4000
Apple	iPhone6SP	Beijing	5000
Samsung	Galaxy S7	Beijing	3500

Name	Loc	Sales
6S 4.7’	Bei Jing	40K
6S 5.5’	Bei Jing	30K
S7	Bei Jing	35K

数据集成

Brand	Product	Loc	Price	Sales
Apple	iPhone6S	Beijing	4000	40K
Apple	iPhone6SP	Beijing	5000	30K
Samsung	Galaxy S7	Beijing	3500	35K

数据分析

销售地点和销售额之间是否有相关性?

数据预处理：数据集成

- 数据集成的基本任务

Data Source 1 (from CourSys)

First Name	Last Name	Mark
Michael	Jordan	50
Kobe	Bryant	48

Data Source 2 (from survey)

Name	Background
Mike Jordan	C++, CS, 4 years
Kobe Bryant	Business, 2 years

Integrated Data

Name	Mark	Background
Michael Jordan	50	C++, CS, 4 years
Kobe Bryant	48	Business, 2 years

思考：请指出数据集成有哪些难点问题？

数据预处理：数据集成

- 数据集成的基本任务
  - 模式映射（Schema Mapping）
    - 创建一个全局模式：将不同数据源的局部模式映射到全局模式
    - 例子： (First Name, Last Name) → Name
  - 实体匹配（Entity Resolution）
    - 此处进一步介绍
  - 数据融合（Data Fusion）
    - 解决不同数据源属性值的冲突与不一致

想要了解数据集成的更多细节？
AnHai Doan, Alon Halevy, and Zachary Ives. Principles of Data Integration.Morgan Kaufmann, 1st edition (2012)

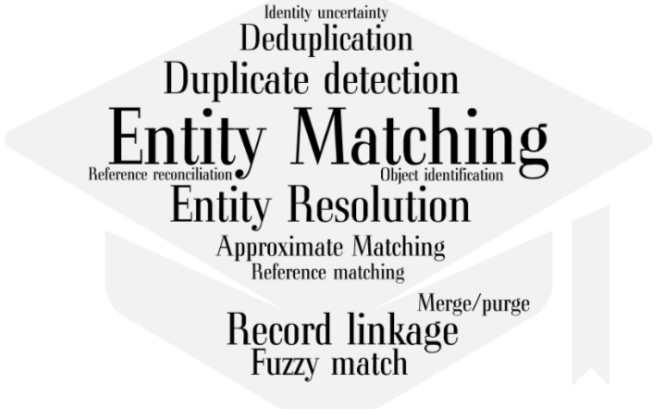
数据预处理：数据集成

- 实体匹配（Entity Matching）
  - 数据集成的核心问题
  - 将表征现实世界中同一实体的不同数据记录匹配起来

不同数据源的手机商品

Apple	iPhone6S	Beijing	4000
Apple	iPhone6SP	Beijing	5000
Samsung	Galaxy S7	Beijing	3500

Name	Loc	Sales
Apple 6S 4.7’	Bei Jing	40K
Apple 6S 5.5’	Bei Jing	30K
Samsung S7	Bei Jing	35K



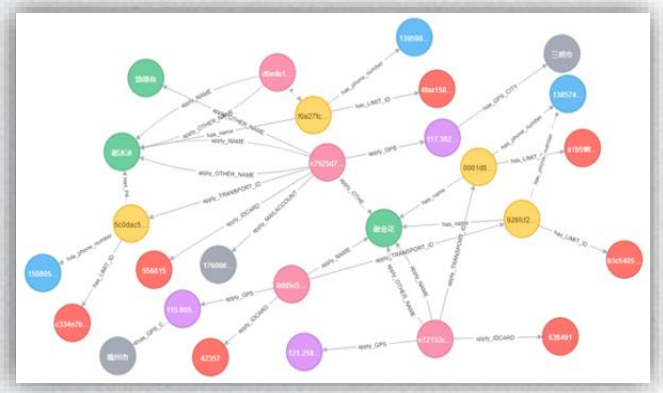
颇为讽刺的是，实体匹配本身就有很多不同的别名

数据预处理：数据集成

- 实体匹配举例：知识图谱构建

互联网借贷行为数据

P2P借贷记录	通讯录
通话详单	淘宝交易
银行账单	…



实体匹配问题

用户	淘宝收货地址
A	河北保定东韩村
B	河北省保定市定兴县东韩村，到定兴县给我打电话1583xxx1234
C	东韩村，河北省定兴县
D	东韩村 072650

机构	地址
证监会	金融街富凯大厦

同一实体？

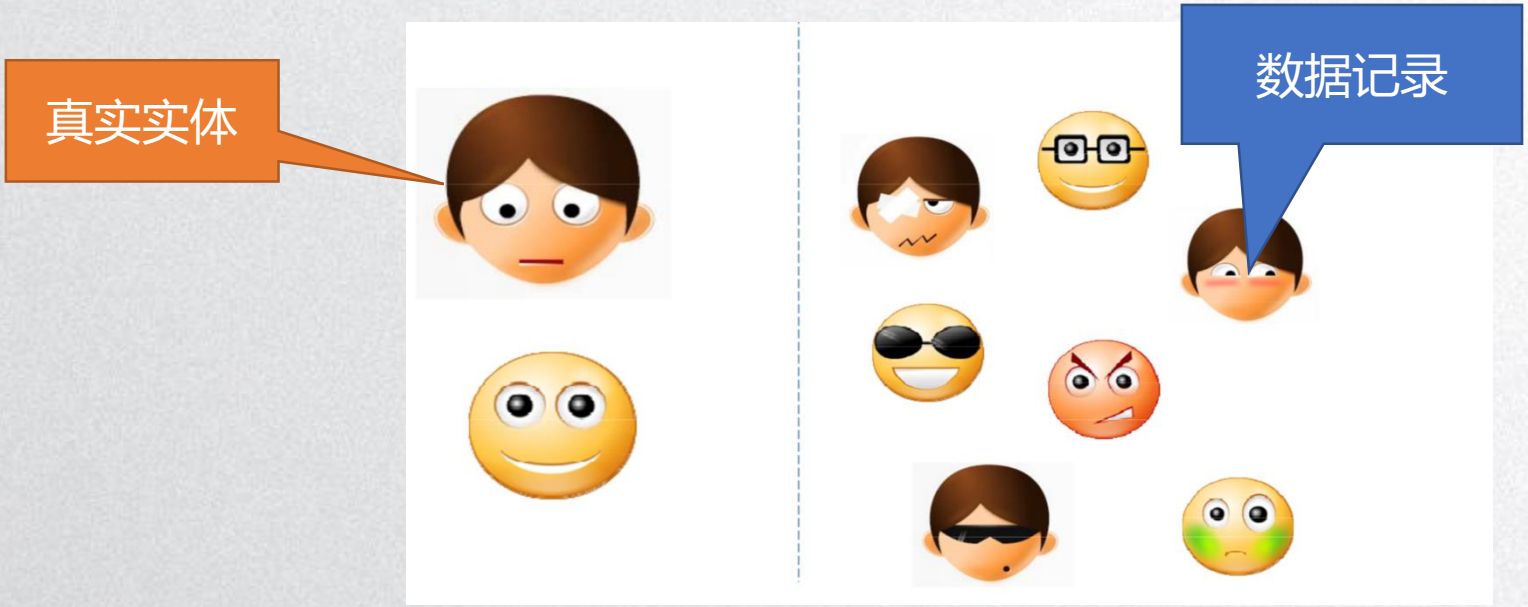
机构	地址
证券监督管理委员会	北京市金融大街19号富凯大厦

数据预处理：数据集成

- 实体匹配
  - 将表示同一实体的不同记录统一起来
- 实体消歧
  - 将表示不同实体的相同记录区分开来

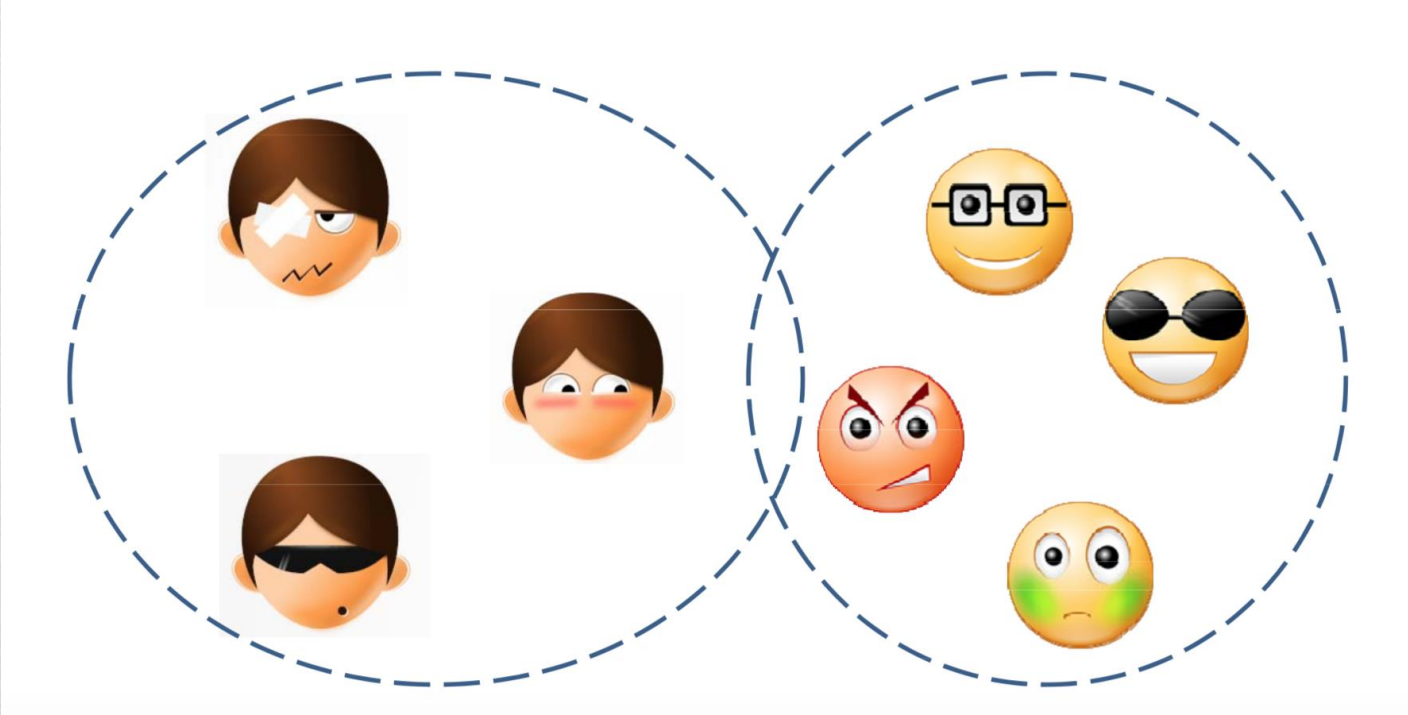
数据预处理：数据集成

- 实体匹配问题定义



数据预处理：数据集成

- 实体匹配：将表示同一实体的数据记录聚在一起



数据预处理：数据集成

- 实体匹配方法举例
  - 先看一个具体的场景：机构名称匹配

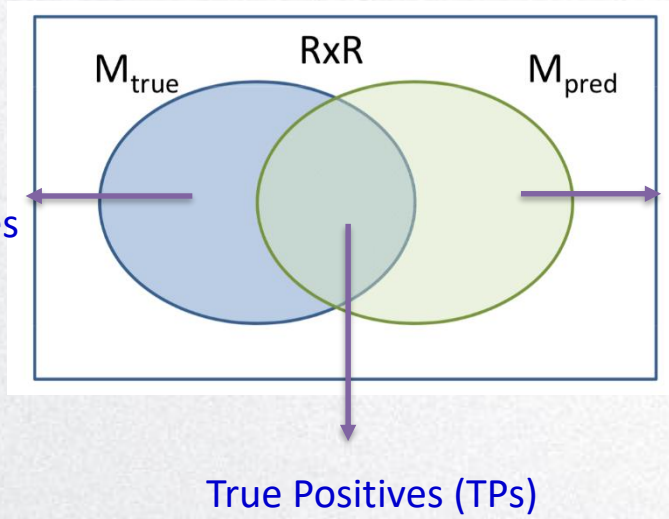
	id1	affil1
0	7927	, IBM Almaden Research Center, 650 Harry Road,...
1	7930	, IIT Bombay
2	7987	, University of California, San Diego, USA
3	5613	28msec Inc., Zurich, Switzerland
4	9530	28msec, Inc.
...	...	...
2255	8434	York University, Canada
2256	8949	York University, Toronto ON, Canada
2257	63	York University, Toronto, ON, Canada
2258	5677	Zhejiang University, China, China
2259	9735	Zhejiang University, Hangzhou, China

2260 rows × 2 columns

不完全相同的字符串

数据预处理：数据集成

- 实体匹配的评测
  - R表示一组数据记录的集合
  - M表示匹配的数据记录对
  - N表示不匹配的数据记录对
  - E表示记录集合R中包含的实体集合
  - 标准答案M_T、N_T、E_T：真实世界
  - 判别结果M_P、N_P、E_P：算法输出



- 准确率Precision=|TPs|/|M_P|
- 召回率Recall=|TPs|/|M_T|

数据预处理：数据集成

- 匹配方法 - 步骤1：提取匹配特征
  - 给定一对待匹配的记录，我们计算出他们的特征向量，其中每一维称为一个特征
    - 例如：比较两条论文记录是否是一篇论文，可以从不同角度计算相似度，组成特征向量
      - 第一作者姓名相似度
      - 标题相似度
      - 发表期刊/会议相似度
      - 发表时间相似度
      - 等等
    - 上述相似度可以是布尔值（匹配/不匹配），也可以是实数（基于某种相似度度量方法）

数据预处理：数据集成

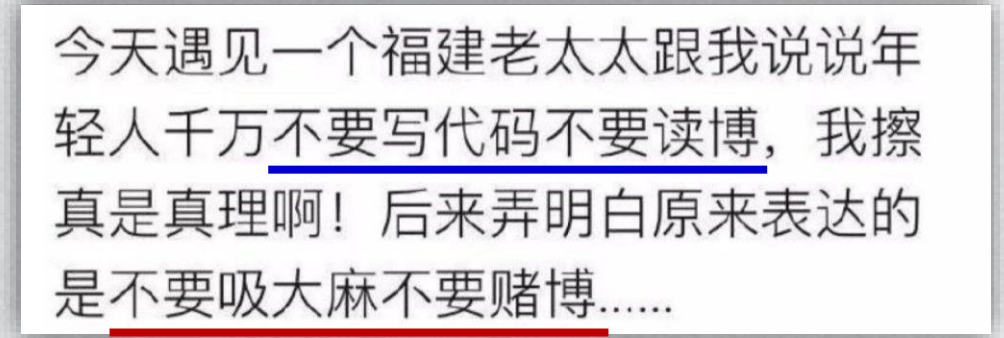
- 匹配方法 - 步骤2：计算匹配特征的相似度
  - 布尔属性：直接判断相等/不相等
  - 数值属性：比较数字之间的差值
  - 文本属性：引入相似度函数进行度量
    - 编辑距离
      - 也称Levenshtein距离
    - 基于集合的函数
      - 如Jaccard函数、Dice函数
    - 基于向量的函数
      - 向量的余弦距离
      - 权重方案，如TF-IDF

适合拼写类差异

适合较长的文本

数据预处理：数据集成

- 编辑距离Edit Distance
  - 有些数据源在录入的时候可能会存在错误
    - 例如数据源A错将作者William Cohen录入成了Willliam Cohon，导致作者无法匹配
  - 在中文录入中，有时会受口音影响



数据预处理：数据集成

- 编辑距离Edit Distance
  - 定义三种针对单一字母的操作
    - 插入：在任意位置插入一个字母，it → its
    - 删除：删除单词的任一字母，she → he
    - 替换：替换单词的任一字母，shot → shit
  - 一个单词可由一组操作变为另一单词
    - 例子：如下操作将kitten变为sitting
      - 替换：kitten → sitten
      - 替换：sitten → sittin
      - 插入：sittin → sitting

数据预处理：数据集成

- Edit Distance度量相似性
  - 定义相似性
    - 编辑距离edit：将查询词q变为任意单词w所需的最少操作次数
    - 相似性定义

$$\frac{\mathrm{edit}(q,w)}{\max\{|w|,|q|\}}$$

https://en.wikipedia.org/wiki/Edit_distance

数据预处理：数据集成

- 编辑距离Edit Distance计算 – 动态规划算法

定义D(i,j) 为从字符串*s1..si* 到 *t1..tj*最少的编辑操作次数

$$= \min \begin{cases} D(i-1,j-1) + d(si,tj) & //substitution/copy \quad 左上 \\ D(i-1,j)+1 & //insert \quad 上 \\ D(i,j-1)+1 & //delete \quad 左 \end{cases}$$

(其中 d(c,d)=0 如果 c=d, 否则等于1)

另外初始化 D(i,0)=i 以及 D(0,j)=j

数据预处理：数据集成

- 编辑距离Edit Distance计算 – 动态规划算法
  - 看一个实例
  - 假设有字符串s1为jary，和字符串s2为jerry，现在求s1和s2的编辑距离，也就是把s2转换为s1的最少编辑操作步
  - 首先，我们建立如下的矩阵，并且初始化该矩阵

源

目标

		j	a	r	y
	0	1	2	3	4
j	1				
e	2				
r	3				
r	4				
y	5				

  - 从源串的第一个字符(“j”)开始，从上至下与目标串进行对比

数据预处理：数据集成

- 编辑距离Edit Distance计算 – 动态规划算法
  - Min（左上角+0或者1，上+1，左+1 ）
  - 比如，第一次，源串第一个字符“j” 与目标串的“j”对比，左+1、上+1、左上+0或者1三个值中取出最小的值0，因为两字符相等，所以填上0
  - 接着，依次对比“j”→“e”、“j”→“r”、“j”→“r”、“j”→“y”等进行处理，直到扫描完目标串，得到的结果如下

		j	a	r	y
	0	1	2	3	4
j	1	0			
e	2	1			
r	3	2			
r	4	3			
y	5	4			

数据预处理：数据集成

- 编辑距离Edit Distance计算 – 动态规划算法
  - 按照上面的方法，遍历整个源串的各个字符，与目标串的各个字符对比，填写各个单元格，各个单元格的变化如下表所示

		j	a	r	y
	0	1	2	3	4
j	1	0	1		
e	2	1	1		
r	3	2	2		
r	4	3	3		
y	5	4	4		
		j	a	r	y
	0	1	2	3	4
j	1	0	1	2	
e	2	1	1	2	
r	3	2	2	1	
r	4	3	3	2	
y	5	4	4	3	
		j	a	r	y
	0	1	2	3	4
j	1	0	1	2	3
e	2	1	1	2	3
r	3	2	2	1	2
r	4	3	3	2	2
y	5	4	4	3	2

数据预处理：数据集成

- 处理完最后一列，则最后一列的最后一个值，为最短编辑距离
  - 即jary和jerry的编辑距离为2
  - 也就是， jary插入r得到jarry，把a改成e得到jerry

		j	a	r	y
	0	1	2	3	4
j	1	0	1		
e	2	1	1		
r	3	2	2		
r	4	3	3		
y	5	4	4		
		j	a	r	y
	0	1	2	3	4
j	1	0	1	2	
e	2	1	1	2	
r	3	2	2	1	
r	4	3	3	2	
y	5	4	4	3	
		j	a	r	y
	0	1	2	3	4
j	1	0	1	2	3
e	2	1	1	2	3
r	3	2	2	1	2
r	4	3	3	2	2
y	5	4	4	3	2

参考如下箭头

		j	a	r	y
	0	1	2	3	4
j	1	0	1	2	3
e	2	1	1	2	3
r	3	2	2	1	2
r	4	3	3	2	2
y	5	4	4	3	2

数据预处理：数据集成

- 集合相似性
  - 举例：Jaccard函数
    - 核心想法：交集大小除以并集大小
    - 应用于单词（汉字）集合
      - 人民大学 vs 中国人民大学 （Jaccard相似度0.67）
    - 也可以应用于单词的字母集合表示
      - 将单词表示成其包含字母的集合
      - 例：scholo表示为{s, c, h, o, l}
      - 例：scholar表示为{s, c, h, o, l, a, r}
    - 字母集合的Jaccard函数
      - scholo和scholar字母集合的交集大小为5，并集大小为7，所以计算出的Jaccard函数值为0.71

数据预处理：数据集成

- 匹配方法 - 步骤3：记录对判别
  - 给定记录 x 和 y 的特征向量（每一维是某个特征上的相似性），输出匹配/不匹配的结果
  - 直观解决方案：
    - 将特征向量各个维度加权求和，得到综合分数，如果分数大于某个阈值，则判为匹配

$$\text{E.g., } 0.5\times 1^{st}\text{-}author\text{-}match\text{-}score + 0.2\times venue\text{-}match\text{-}score + 0.3\times title\text{-}match\text{-}score \geq 0.8$$

    - 利用领域规则，对匹配进行判别

$$\text{E.g., } (1^{st}\text{-}author\text{-}match\text{-}score > 0.7 \text{ AND } venue\text{-}match\text{-}score > 0.8) \text{ OR } (title\text{-}match\text{-}score > 0.9 \text{ AND } venue\text{-}match\text{-}score > 0.9)$$

数据预处理：数据集成