知识融合

大纲

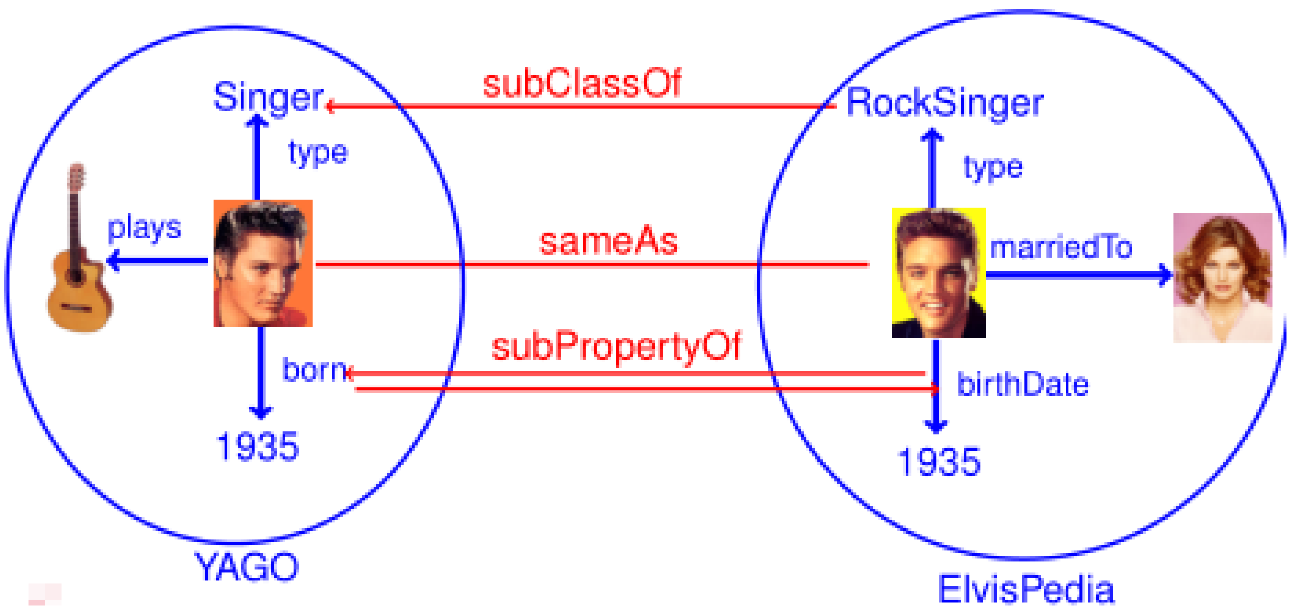
- 什么是知识融合
- 知识融合的基本技术流程
- 典型知识融合工具简介
- 典型案例简介
  - Zhishi.me
  - OpenKG.Link
- LIMES实战演练

什么是知识融合？

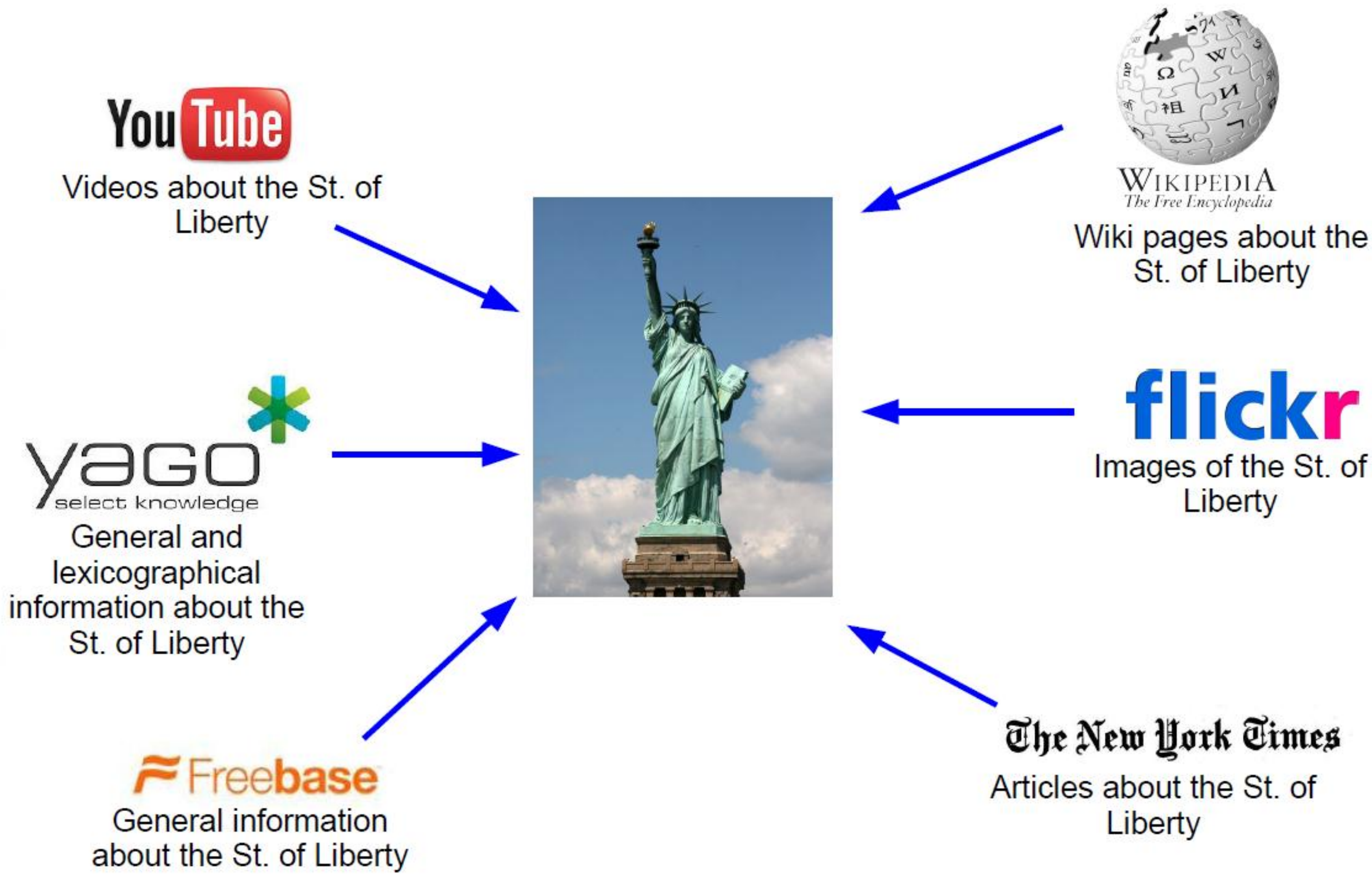
目标：融合各层面的知识

合并两个知识图谱(本体)，需要确认的是：

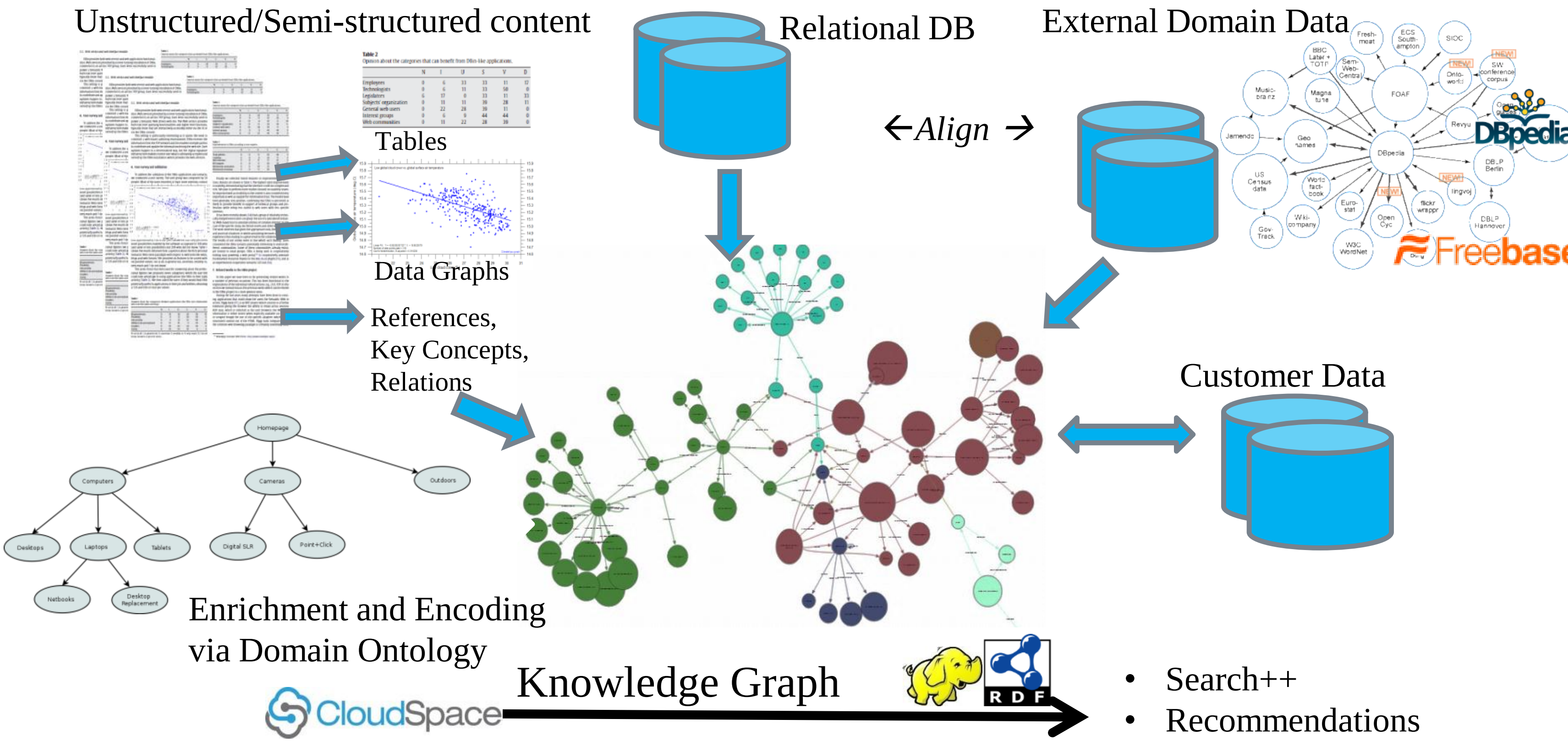
1. 等价实例
2. 等价类/子类
3. 等价属性/子属性



来源于不同知识库的同一实体

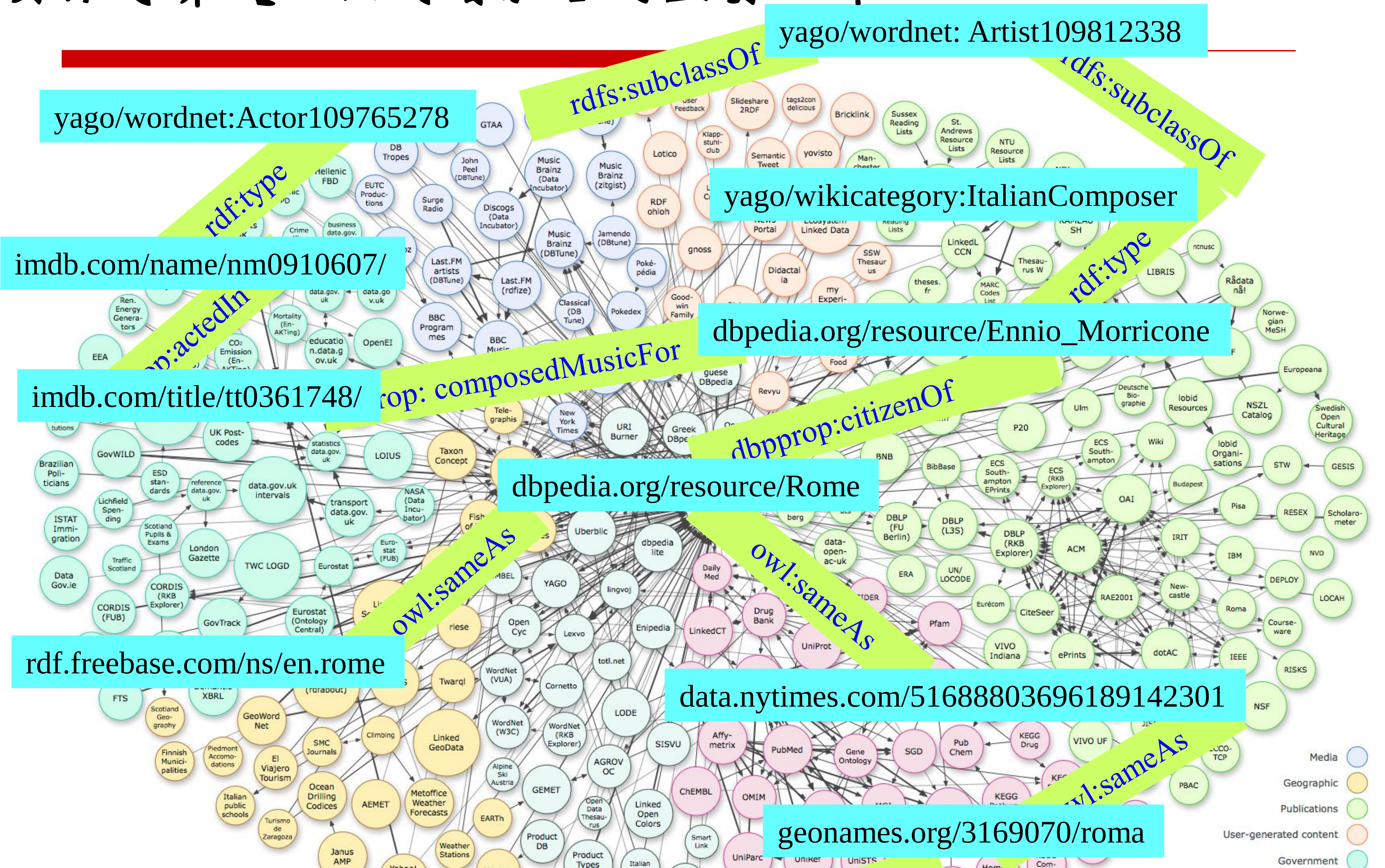


知识图谱的构建经常需要融合多种不同来源的数据

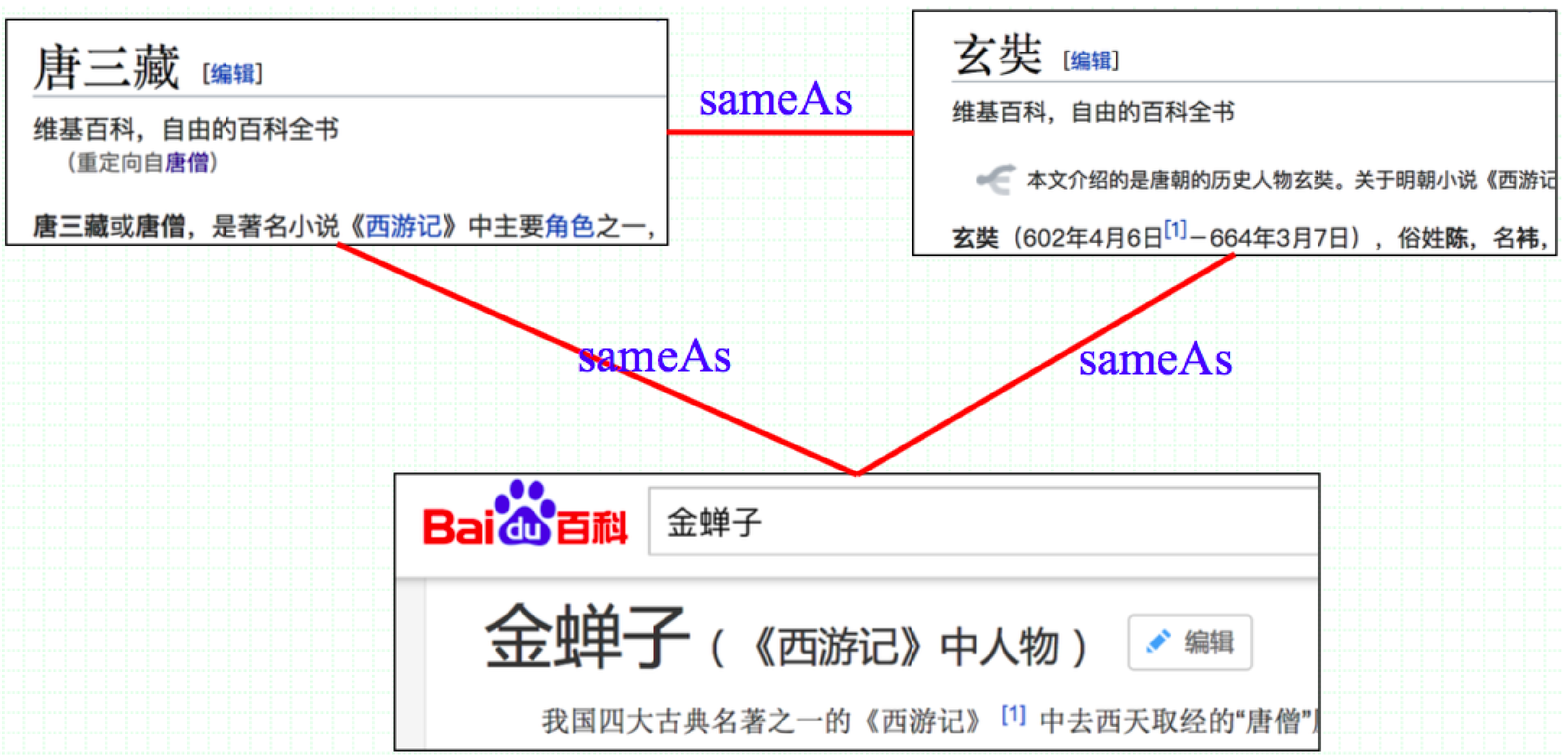


- Search++
- Recommendations
- Vertical applications
- Explorative interfaces

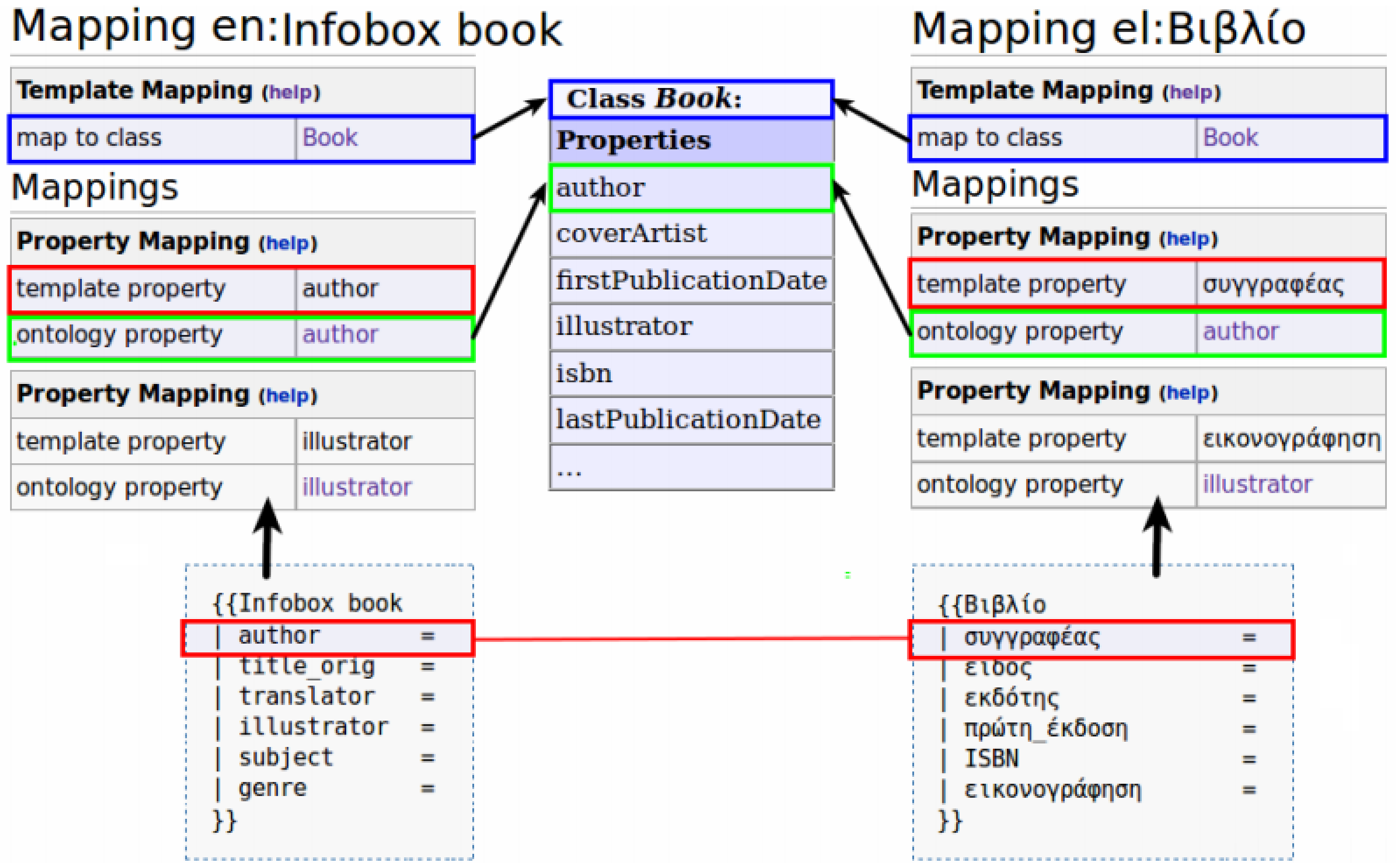
实体对齐是知识图谱融合的主要工作



中文百科中的等价实例


唐三藏 [编辑]
维基百科，自由的百科全书
（重定向自唐僧）
唐三藏或唐僧，是著名小说《西游记》中主要角色之一，
sameAs
玄奘 [编辑]
维基百科，自由的百科全书
本文介绍的是唐朝的历史人物玄奘。关于明朝小说《西游记
玄奘（602年4月6日[1]–664年3月7日），俗姓陈，名祎，
sameAs
sameAs
Baidu百科
金蝉子
金蝉子（《西游记》中人物）
编辑
我国四大古典名著之一的《西游记》[1] 中去西天取经的“唐僧”

概念层知识融合



http://mappings.dbpedia.org/

跨语言知识融合

Unified Medical Language System ®

- Abdominal distension
- Abdominal mass
- Abdominal pain
- Abnormal dermatoglyphics
- Abnormal heart sounds
- Abnormal reflexes
- Abnormal splitting of heart sounds
- Abnormal sputum
- Absence of uvula
- Acneiform lesions
- Acute confusional state
- Adrenal cortex hyperplasia
- Akathisia
- Alopecia
- Amaurosis fugax

咳嗽 症状 [症状起因] [诊断详述] [检查鉴别] [问医生] [找医院]

咳嗽是呼吸系统疾病最常见的症状之一，它是一种保护性神经反射，通过咳嗽产生呼气性冲击动作，能将呼吸道内的异物或分泌排出体外。[详细]

相关疾病 夏季感冒 | 病毒性支气管炎 | 病毒性喉炎 | 肾咳 | 痰饮

阴茎短 症状 [症状起因] [诊断详述] [检查鉴别] [问医生] [找医院]

小阴茎(micropenis)是指阴茎外观正常长度与直径比值正常，但阴茎体的长度小于正常阴茎长度平均值2.5个标准差以上。阴茎的长度是指用手提阴茎头尽量拉直即相当于阴茎充分勃起时从阴茎顶到耻骨联合的距...[详细]

相关疾病 男性生殖器畸形 | 小阴茎 | 类固醇5α-还原酶2缺乏综合征 | 先天性睾丸发育不全

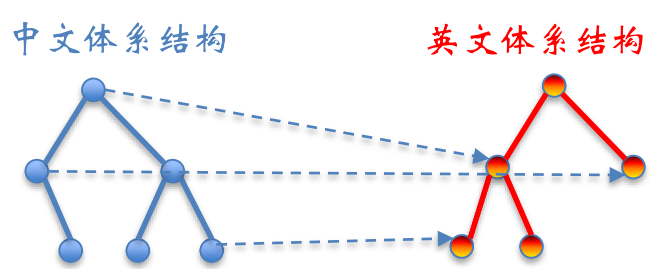
阴道出血 症状 [症状起因] [诊断详述] [检查鉴别] [问医生] [找医院]

阴道出血是女性生殖器官疾病常见的症状。出血可来自外阴、阴道、子宫颈和子宫内膜，但以来自子宫者为最多。阴道出血量固然可以危及生命，但如良性疾病所致者，预后良好；而出血量少的，也可能是恶性肿...[详细]

相关疾病 重度宫颈糜烂 | 胎热 | 血热崩漏 | 妊娠热病 | 原发性纤维蛋白溶解症

关节疼痛 症状 （别名：关节痛） [症状起因] [诊断详述] [检查鉴别] [问医生] [找医院]

关节炎或关节病牵涉范围非常广泛，种类繁多，病因各异，然而，普遍的临床症状均可表现关节疼痛，因此关节疼痛的鉴别诊断至关重要，但是有的病因已知，有的病因尚未明了，总之，明确诊断才是合理治疗疾...[详细]



R30 Pain associated with micturition
Excludes: psychogenic pain (F45.3)
小 便 疼 痛
不 包 含 ： 心 因 性 小 便 疼 痛 (F45.3)

R30.0 Dysuria
Strangury
小 便 困 难
尿 绞 痛

ICD编码

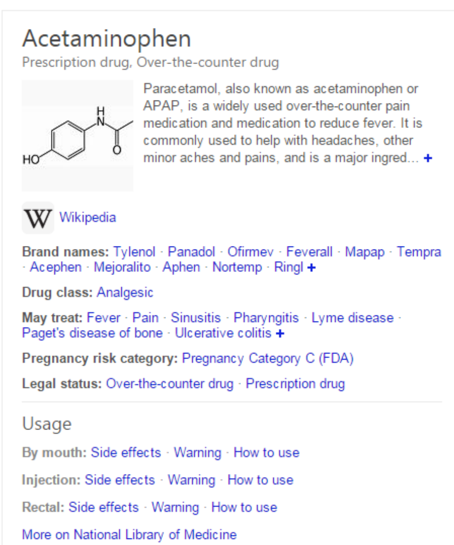
R30.1 Vesical tenesmus
膀 胱 里 急 後 重

R30.9 Painful micturition, unspecified

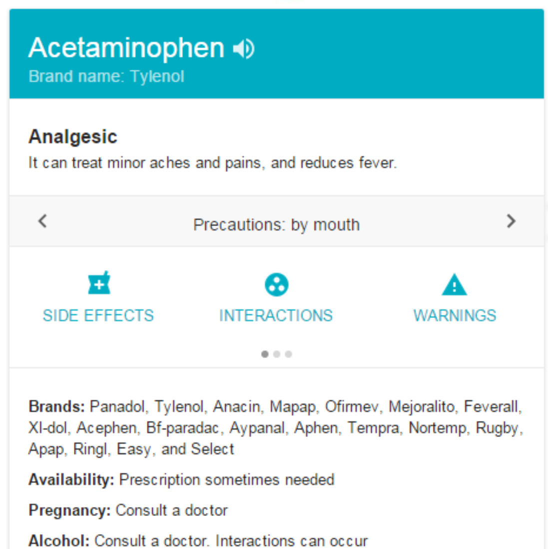
知识在线融合

实体：扑热息痛(Acetaminophen)

bing

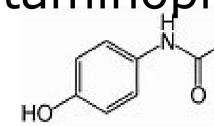


Google



合并后知识卡片

Acetaminophen



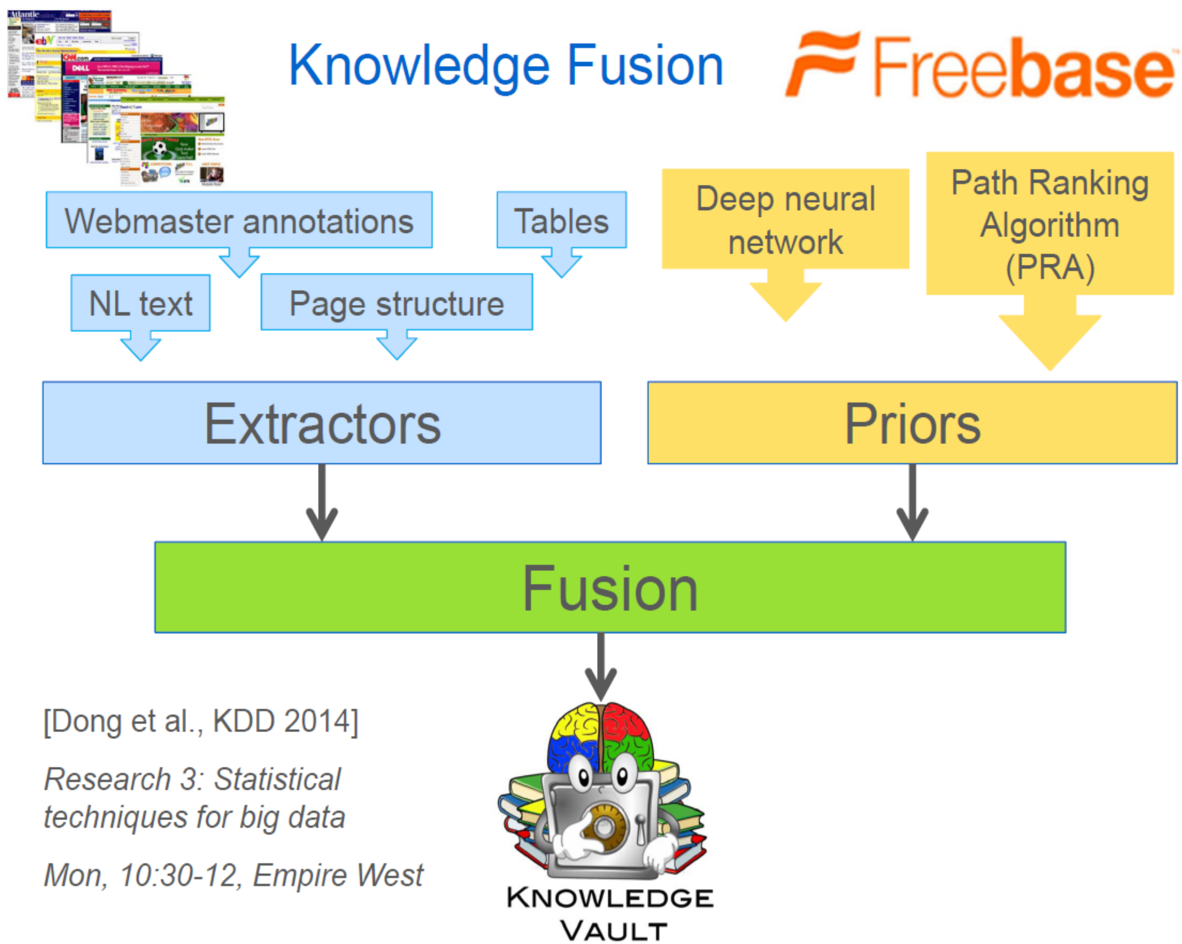
Brands: Tylenol , Panadol, Ofirmev, Feverall, Mapap, Tempra, Acephen, Mejoralito, Aphen, Nortemp

Drug class: Analgesic

Availability: Prescription sometimes needed

Pregnancy: Pregnancy Category C (FDA)

Google Knowledge Vault



[Dong et al., KDD 2014]

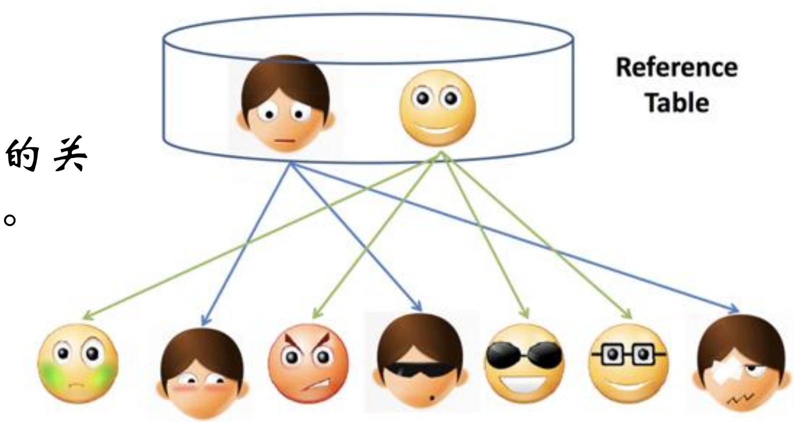
Research 3: Statistical techniques for big data

Mon, 10:30-12, Empire West

各种相关的名词术语

- 知识融合 (Knowledge Fusion)
- 本体对齐 (Ontology Alignment)
- 本体匹配 (Ontology Matching)
- Record Linkage (传统数据库领域)
- Entity Resolution (传统数据库领域)
- 实体对齐 (Entity Alignment)

基本的问题都是研究怎样将来自多个来源的关于同一个实体或概念的描述信息融合起来。



知识融合的主要技术挑战

- **数据质量的挑战**

  命名模糊，数据输入错误，数据丢失，数据格式不一致，缩写等

- **数据规模的挑战：**

  数据量大(并行计算)，数据种类多样性，不再仅仅通过名字匹配，多种关系，更多链接等

知识融合竞赛—OAEI

- OAEI

  OAEI (Ontology Alignment Evaluation Initiative)本体对齐竞赛，用来评估各种本体对齐算法，以达到评估，比较，交流以及促进本体对齐工作的目的。OAEI每年举办一次，结果公布在官网上。

- 主页

  http://oaei.ontologymatching.org/

- 2017年

  http://oaei.ontologymatching.org/2017/

OAEI-2017 Schedule

6.1 上传数据集
7.15 确定数据集
6.30 注册

7.15 可以开始提交，有固定格式
7.31 提交初步包装版本，zip文件，强制性

8.31 **SEALS client**提交最终版本，zip文件
9.15 **HOBBIT platform**提交最终版本注册

9.30 评估及结果分析

10.10 提交初步paper
10.21 会议
11.15 提交最终paper

OAEI-Tracks

序号	关注问题	关注点
1	Anatomy	解剖
2	conference	会议
3	Multifarm	不同语言会议数据
4	Interactive matching evaluation	交互式的匹配评估(用户交互下是否能改善匹配结果)
5	Large Biomedical Ontologies	生物
6	Disease and Phenotype	疾病及症状匹配
7	Process Model Matching	在更多具体的领域上，本体匹配系统性能测试
8	Instance Matching	实例匹配
9	HOBBIT Link Discovery	新增的，地理

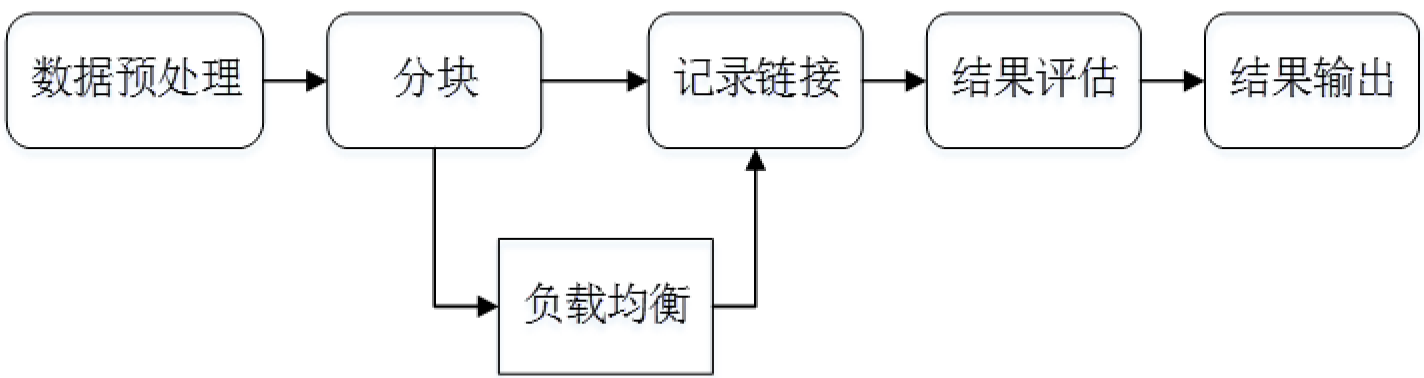
OAEI

- **SEALS:**
  - SEALS项目致力于评估语义网络技术。 它创建了便于进行评估的基础架构，组织评估活动，围绕此评估活动构建工具提供者和工具用户社区。参与者可以通过OAEI_SealsClient 提交。
  - tutorial.pdf
- **Hobbit：**
  - 该平台旨在对群集上的关联数据系统进行基准测试。
  - 官网： http://project-hobbit.eu/
  - Github： https://github.com/hobbit-project
  - Tutorial： https://github.com/hobbit-project/platform/wiki

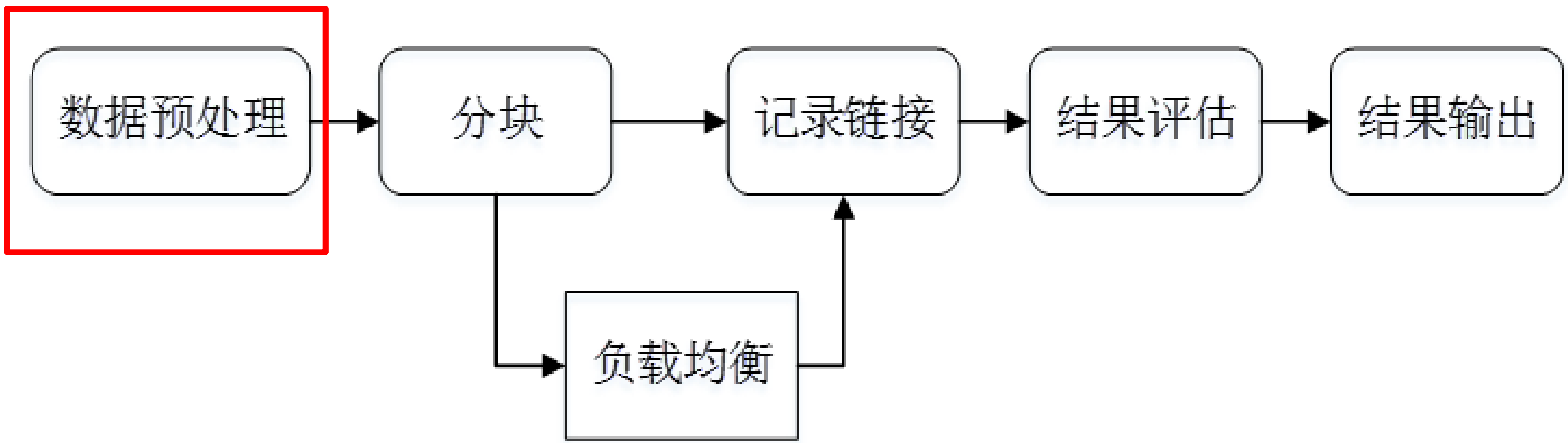
知识融合的基本技术流程

基本技术流程

知识融合一般分为两步，本体对齐和实体匹配两种的基本流程相似，如下：



数据预处理



数据预处理阶段，原始数据的质量会直接影响到最终链接的结果，不同的数据集对同一实体的描述方式往往是不相同的，对这些数据进行归一化处理是提高后续链接精确度的重要步骤。

数据预处理

- 语法正规化

  语法匹配：联系电话的表示方法

  综合属性： 家庭地址的表达方式

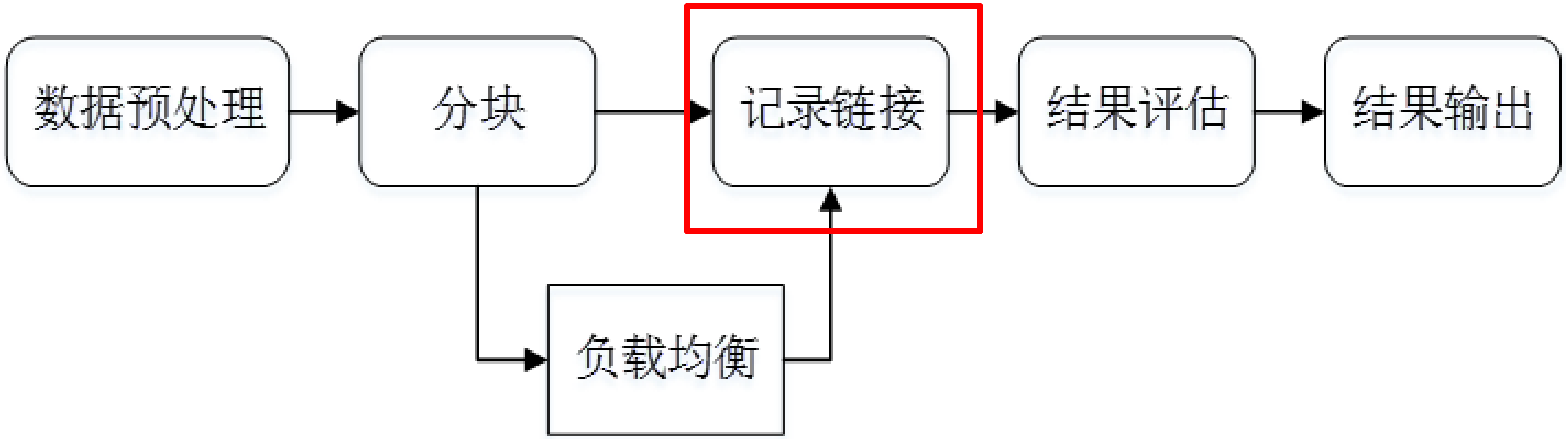
- 数据正规化

  移除空格，《》，“”，-，等符号

  输入错误类的拓扑错误

  用正式名字替换昵称和缩写等

记录链接



假设两个实体的记录x和y，x和y在第i个属性上的值是x_i,y_i，那么通过如下两步进行记录链接：

1. 属性相似度：
   综合单个属性相似度得到属性相似度向量
   $[sim(x_1, y_1), sim(x_2, y_2), \ldots, sim(x_N, y_N)]$
2. 实体相似度：
   根据属性相似度向量得到一个实体的相似度

怎样计算属性相似度

- 编辑距离

  Levenstein，Wagner and Fisher，Edit Distance with Affine Gaps

- 集合相似度计算

  Jaccard系数，Dice

- 基于向量的相似度计算

  Cosine相似度，TFIDF相似度

- ……

属性相似度——编辑距离

- Levenshtein distance (最小编辑距离)，目的是用最少的编辑操作将一个字符串转成另一个，如下：

$$'Lvensshtain' \xrightarrow{\text{插入 } 'e'} 'Levensshtain'$$

$$'Levensshtain' \xrightarrow{\text{删除 } 's'} 'Levenshtain'$$

$$'Levenshtain' \xrightarrow{\text{替换 } 'a' \to 'e'} 'Levenshtein'$$

上述将‘Lvensshtain’转换成‘Levenshtein’，总共的操作3次，编辑距离也就是3。

属性相似度——编辑距离

- Levenshtein distance是典型的动态规划问题，可以通过动态规划算法计算，如下：

$$\begin{cases} D(0,0) = 0 \\ D(i,0) = D(i-1,0) + 1 & 1 < i \leqq N \\ D(0,j) = D(0,j-1) + 1 & 1 < j \leqq M \end{cases}$$

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + 1 \end{cases}$$

其中，+1表示的是插入，删除和替换的代价。

属性相似度——编辑距离

- Wagner and Fisher distance是Levenshtein 的一个扩展，将这个模型中编辑操作的代价赋予了不同的权重，如下：

$$\begin{cases} D(0,0) = 0 \\ D(i,0) = D(i-1,0) + del[x(i)] & 1 < i \leqq N \\ D(0,j) = D(0,j-1) + del[y(j)] & 1 < j \leqq M \end{cases}$$

$$D(i,j) = \min \begin{cases} D(i-1,j) + del[x(i)] \\ D(i,j-1) + ins[y[j]] \\ D(i-1,j-1) + sub[x(i),y(j)] \end{cases}$$

其中，del 和 ins 以及 sub 分别是删除和插入以及替换的代价

属性相似度—编辑距离

- Edit Distance with affine gaps，在上面两种算法的基础上，引入了gap的概念，将上述的插入，删除和替换操作用用gap opening和gap extension代替，编辑操作的代价也就表示为：

$$Cost(g) = s + e * l$$

其中，s是open gap的代价，e是extend gap的代价，l是gap的长度。

属性相似度——编辑距离

- Edit Distance with affine gaps

结合前面‘Lvensshtain’转换的例子：

$$L\ \boxed{\varepsilon}\ v\ e\ n\ s\ s\ h\ t\ a\ \boxed{\varepsilon}\ i\ n$$

$$L\ e\ v\ e\ n\ s\ \boxed{\varepsilon}\ h\ t\ \boxed{\varepsilon}\ e\ i\ n$$

其中，$\boxed{\varepsilon}$代表一个gap，结合上述代价公式，若设置$s = 2$，$e = 1$上述编辑操作代价为$(2 + 1 * 1) * 4 = 12$。

属性相似度—集合相似度

- Dice系数用于度量两个集合的相似性，因为可以把字符串理解为一种集合，因此Dice距离也会用于度量字符串的相似性，Dice系数定义如下：

$$sim_{\text{Dice}}(s,t) = \frac{2|S \cap T|}{|S| + |T|}$$

以‘Lvensshtain’和‘Levenshtein’为例，两者相似度为 2*9/ (11+11)= 0.82

属性相似度—集合相似度

- Jaccard 系数适合处理短文本的相似度，定义如下：

$$sim_{Jaccard}(s,t)=\frac{|S\cap T|}{|S\cup T|}$$

  可以看出与Dice系数的定义比较相似。

- 两种方法，将文本转换为集合，除了可以用符号分格单词外，还可以考虑用n-gram分割单词，用n-gram分割句子等来构建集合，计算相似度。

属性相似度—基于向量的相似度

- TF-IDF主要用来评估某个字或者某个词对一个文档的重要程度。

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad idf_i = \log \frac{|D|}{1 + |\{j : t_i \in d_j\}|}$$

$$sim_{TF-IDF} = tf_{i,j} \times idf_i$$

比如某个语料库中有5万篇文章，含有“健康”的有2万篇，现有一篇文章，共1000个词，‘健康’出现30次，则$sim_{TF\text{-}IDF}$ = 30/1000 * log(50000/(20000+1)) = 0.012

怎样计算实体相似度

1. 聚合

   加权平均，手动制定规则，分类器

2. 聚类

   层次聚类，相关性聚类

3. 表示学习

怎样计算实体相似度

相似度得分向量：$[sim(x_1, y_1), \ldots, sim(x_N, y_N)]$

1. 聚合
   - 加权平均：$w_1 * sim(x_1, y_1) + \ldots + w_N * sim(x_N, y_N)$
   - 手动制定规则：$sim(x_1, y_1) > T_1$ and (or) …. $sim(x_i, y_i) > T_i$
   - 分类器：逻辑回归，决策树，SVM和条件随机场等

怎样计算实体相似度：机器学习方法

1. 聚合

- 问题：

  —训练集的生成

  —分类不均衡(更多不匹配的记录对)

  —误分类

  最关键的问题是需要生成训练集合。

- 方案

  —无监督/半监督(EM，生成模型等)

  —主动学习(众包等)

怎样计算实体相似度：机器学习方法

2. 聚类

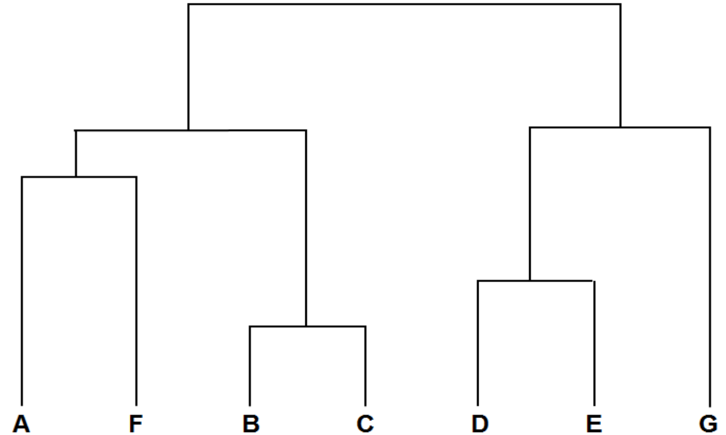
- 层次聚类
- 相关性聚类
- Canopy + K-means

怎样计算实体相似度：机器学习方法

2. 聚类

层次聚类

层次聚类 (Hierarchical Clustering) 通过计算不同类别数据点之间的相似度对在不同的层次的数据进行划分，最终形成树状的聚类结构。



怎样计算实体相似度：机器学习方法

2. 聚类

- 层次聚类

底层的原始数据可以通过相似度函数计算，类之间的相似度有如下三种算法：

a. SL(Single Linkage)算法

SL算法又称为最邻近算法 (nearest-neighbor)，是用两个类数据点中距离最近的两个数据点间的相似度作为这两个类的距离。

b. *CL* (Complete Linkage)算法

与SL不同的是取两个类中距离最远的两个点的相似度作为两个类的相似度。

c. AL (Average Linkage)算法

用两个类中所有点之间相似度的均值作为类间相似度。

怎样计算实体相似度：机器学习方法

2. 聚类：

- 层次聚类

举例：有左图的数据，用欧氏距离和SL进行层次聚类。

A	16.9
B	38.5
C	39.5
D	80.8
E	82
F	34.6
G	116.1

→

	A	B	C	D	E	F	G
A	0	21.60	22.60	63.90	65.10	17.70	99.20
B	21.60	0	1.00	42.30	43.50	3.90	77.60
C	22.60	1.00	0	41.30	42.50	4.90	76.60
D	63.90	42.30	41.30	0	1.20	46.20	35.30
E	65.10	43.50	42.50	1.20	0	47.40	34.10
F	17.70	3.90	4.90	46.20	47.40	0	81.50
G	99.20	77.60	76.60	35.30	34.10	81.50	0

B,C之间的欧式距离是：$\sqrt{(B-C)^2} = \sqrt{(38.5-39.5)^2}$

怎样计算实体相似度：机器学习方法

2. 聚类

层次聚类

上页B,C之间距离最小为1.00，将B，C组合，再次计算距离

	A	(B,C)	D	E	F	G
A	0	22.10	63.90	65.10	17.70	99.20
(B,C)	22.10	0	41.80	43.00	4.40	77.10
D	63.90	41.80	0	1.20	46.20	35.30
E	65.10	43.00	1.20	0	47.40	34.10
F	17.70	4.40	46.20	47.40	0	81.50
G	99.20	77.10	35.30	34.10	81.50	0

其中，单个数据与类之间的距离，如下：

$$D = \frac{\sqrt{(B-A)^2} + \sqrt{(C-A)^2}}{2} = \frac{21.6 + 22.6}{2}$$

怎样计算实体相似度：机器学习方法

2. 聚类

- 层次聚类

之后计算类之间的距离，如下是计算(A,F)和(B,C) 间的距离：

$$D = \frac{\sqrt{(A-B)^2} + \sqrt{(A-C)^2} + \sqrt{(F-B)^2} + \sqrt{(F-C)^2}}{4}$$

最终得到如下两个类：

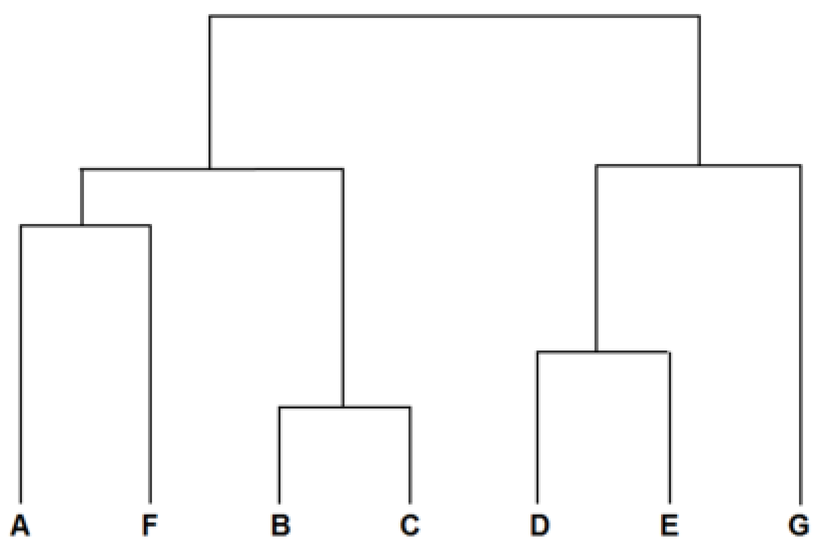
	(A,F,B,C)	(D,E,G)
(A,F,B,C)	0	60.59
(D,E,G)	60.59	0

怎样计算实体相似度：机器学习方法

2. 聚类

- 层次聚类

前面的每一步的计算结果以树状图的形式展现出来就是层次聚类树。



怎样计算实体相似度：机器学习方法

2. 聚类

- 层次聚类
- 相关性聚类

r_{xy}表示x,y被分配在同一类中， p_{xy}代表x,y是同一类的概率(x,y之间的相似度)， w^{+}_{xy} $(= p_{xy})$和 w^{-}_{xy} $(= 1\text{-}p_{xy})$分别是切断x,y之间的边的代价和保留边的代价。相关性聚类的目标就是使用最小的代价找到一个聚类方案。

$$minimize \sum r_{xy}\, w^{-}_{xy} + (1 - r_{xy})\, w^{+}_{xy}$$

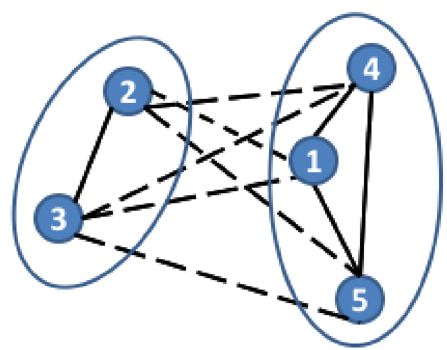
是个NP-Hard问题，可用贪婪算法近似求解。

怎样计算实体相似度：机器学习方法

2. 聚类

相关性聚类

举例，下图中，实线表示两数据点有关系，将其归为一类，会给最终结果贡献W $^{-}_{xy}$；虚线表示两数据点没有关系，将其归为一类，给最终结果贡献W $^{+}_{xy}$。



怎样计算实体相似度：机器学习方法

2. 聚类

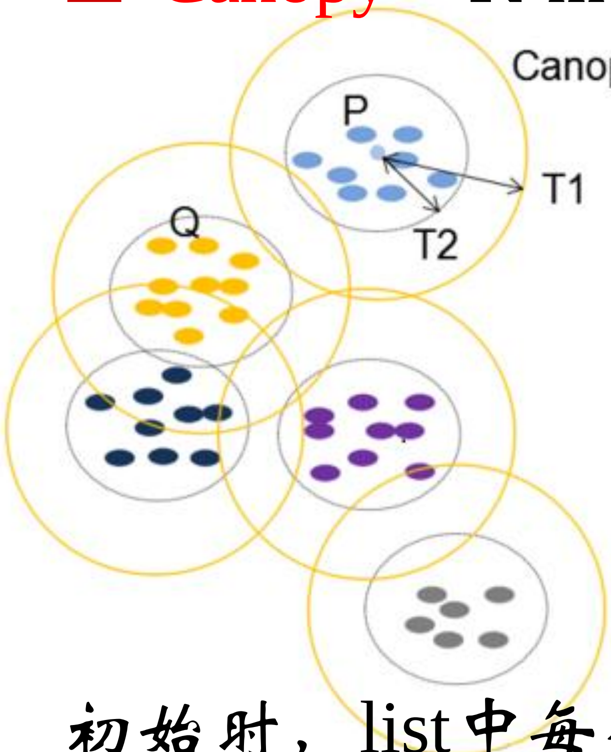
- 层级聚类
- 相关性聚类
- Canopy + K-means

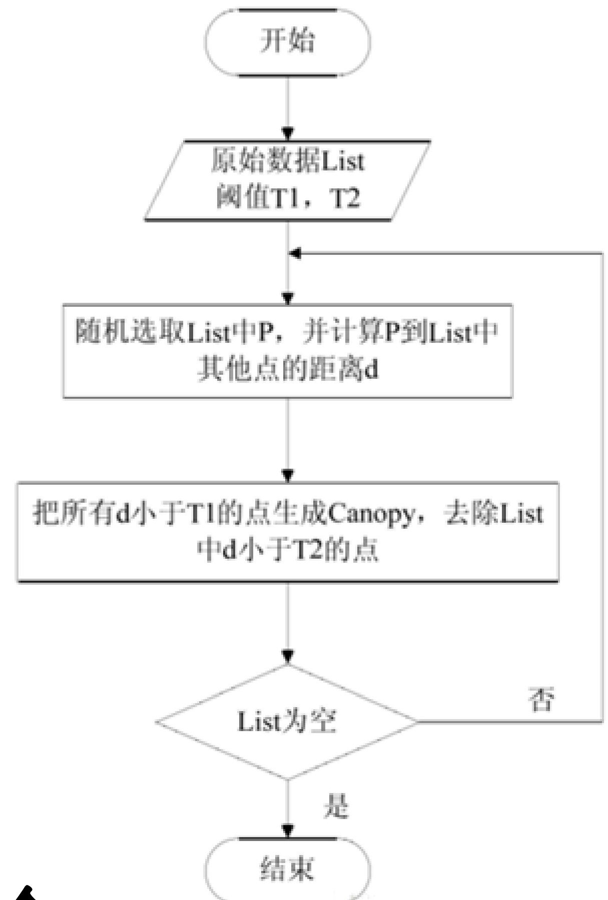
与K-means不同，Canopy聚类最大的特点是不需要事先指定k值(即clustering的个数)，因此具有很大的实际应用价值，经常将Canopy和K-means配合使用。

怎样计算实体相似度：机器学习方法

2. 聚类

- Canopy + K-means





初始时，list中每个点都是一个canopy。

怎样计算实体相似度：知识表示学习

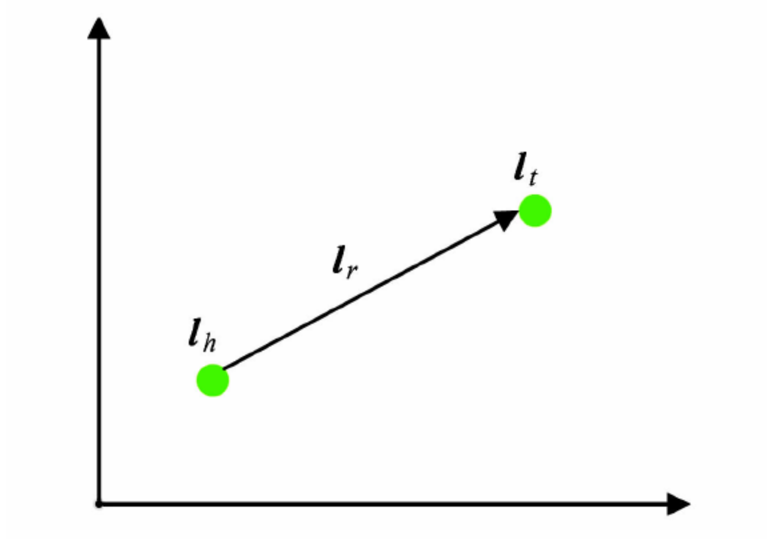
3. 知识表示学习 — 知识嵌入

将知识图谱中的实体和关系都映射低维空间向量，直接用数学表达式来计算各个实体之间相似度。这类方法不依赖任何的文本信息，获取到的都是数据的深度特征。

怎样计算实体相似度：知识表示学习

知识嵌入——TransE模型

三元组(h, r, t): 关系向量l_r表示为头向量l_h和尾向量l_t之间的位移

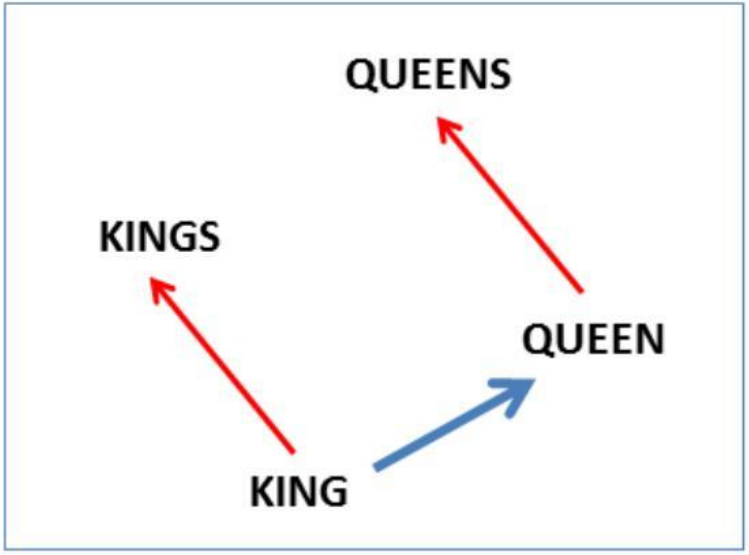


TransE希望

$$l_t = l_h + l_r$$

怎样计算实体相似度：知识表示学习

实体与向量之间的关系



KINGS – QUEENS = KING – QUEEN

则考虑两对实体之间存在某个相同的关系

怎样计算实体相似度：知识表示学习

实体与向量之间的关系

义勇军进行曲

中国国歌

$\xrightarrow{\text{作曲}}$ 聂耳

KG1中存在头实体：义勇军进行曲
KG2中存在头实体：中国国歌

义勇军进行曲 + 作曲 = 聂耳

中国国歌 + 作曲 = 聂耳

考虑义勇军进行曲和中国国歌是等价实体

怎样计算实体相似度：知识表示学习

- 如何将两个知识图谱嵌入到同一个空间
  - 桥梁：预链接实体对(训练数据)
  - 方法：
    - 联合知识嵌入：将两个KG的三元组糅合在一起共同训练，并将预链接实体对视为具有SameAS关系的三元组，从而对两个KG的空间进行约束

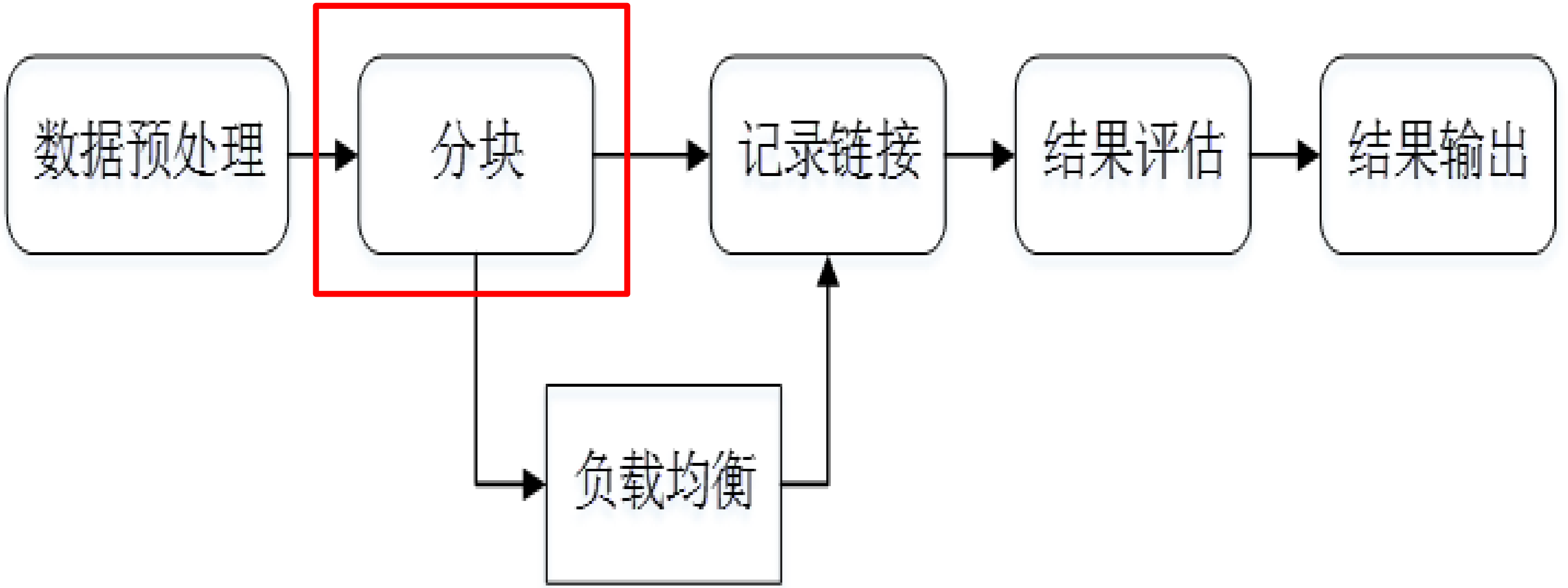
      具体实现： Hao Zhu et al. Iterative Entity Alignment via Knowledge Embeddings, IJCAI 2017
    - 双向监督训练：两个KG单独进行训练，使用预链接数据交替进行监督。

怎样计算实体相似度：知识表示学习

- 如何链接实体

KG向量训练达到稳定状态之后，对于KG1每一个没有找到链接的实体，在KG2中找到与之距离最近的实体向量进行链接，距离计算方法可采用任何向量之间的距离计算，例如欧式距离或Cosine距离。

分块：关注数据规模问题



分块 (Blocking)是从给定的知识库中的所有实体对中，选出潜在匹配的记录对作为候选项，并将候选项的大小尽可能的缩小。

分块

- 动机

  对记录直接进行一一链接的时间复杂度是O(|M| * |N|)。如下是直接对两个10^6条记录的数据集进行链接分块 (1000块)与不分块时间的对比：

Falcon-AO

10^6条记录的链接	无分块	分块（分为 m 块）
时间复杂度	10^{12}	10^9
用时（1ms 一次链接）	11.6 天	16 分钟

常用的分块方法

1. 基于Hash函数的分块

   对于记录x，有hash(x)=h_i,则x映射到与关键字h_i绑定的块C_i上。

   常见的hash函数：

   —字符串的前n个字

   —n-grams

   —结合多个简单的hash函数等

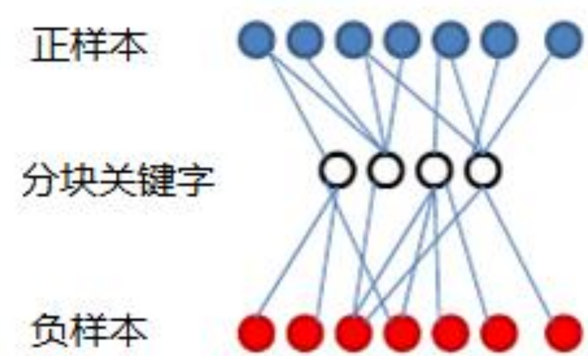
常用的分块方法

2. 邻近分块

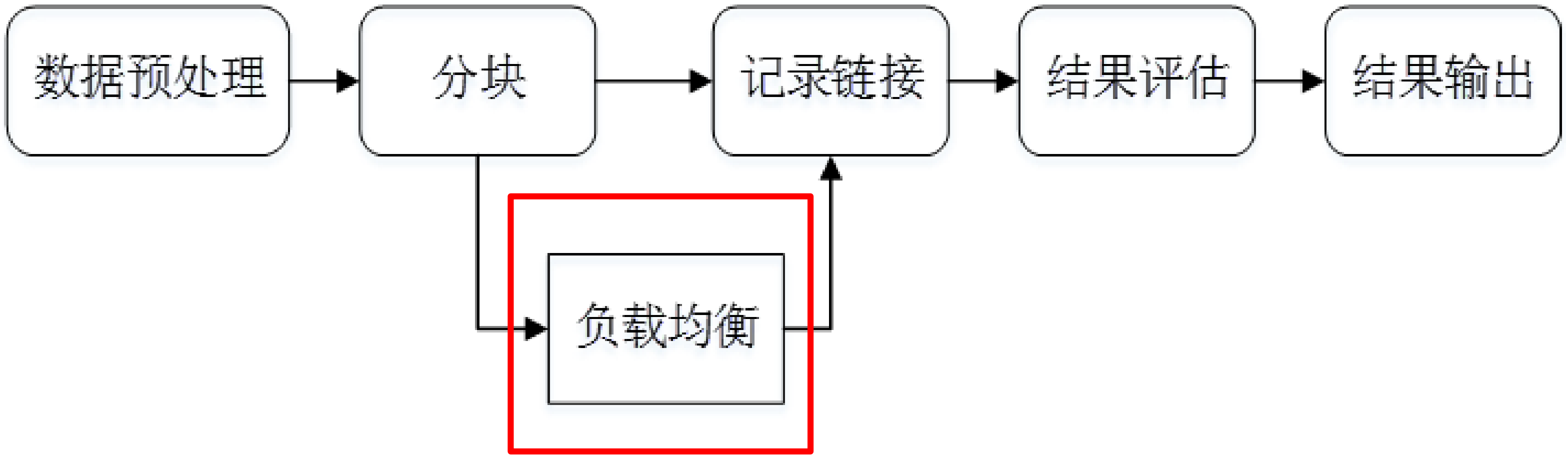
- □ Canopy聚类
- □ 排序邻居算法

生成关键字 → 关键字排序 → 移动窗口

- □ Red - Blue Set Cover



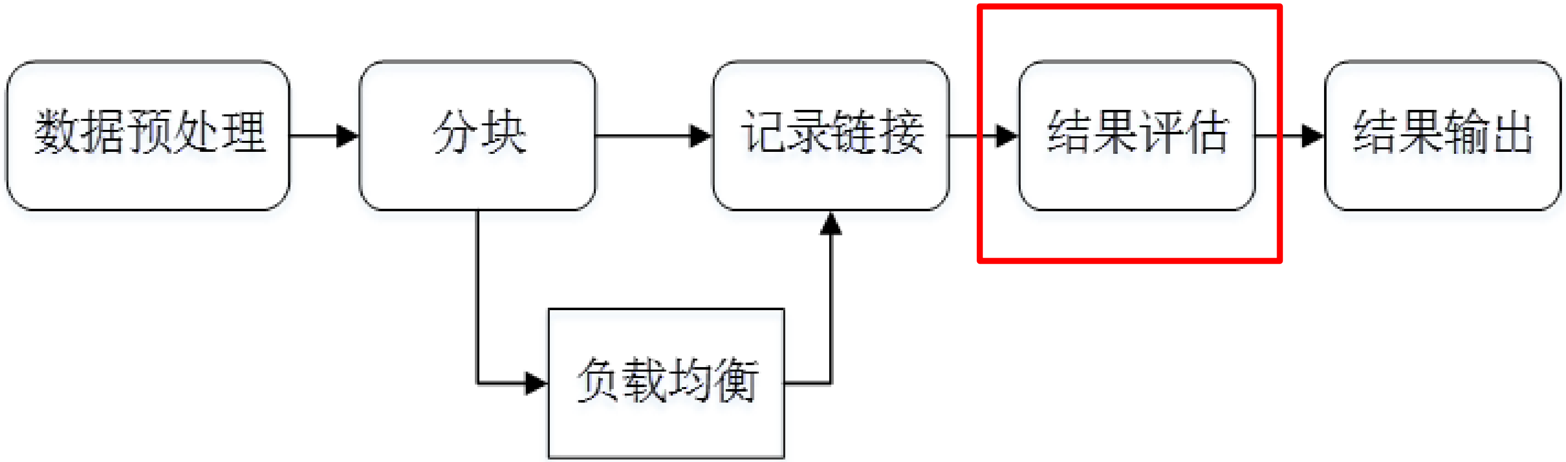
负载均衡



负载均衡 (Load Balance)来保证所有块中的实体数目相当，从而保证分块对性能的提升程度。

最简单的方法是多次Map-Reduce操作。

结果评估



- 准确率，召回率，F值
- 整个算法的运行时间

典型知识融合工具简介

本体对齐—Falcon-AO

- **简介**

  Falcon-AO是一个自动的本体匹配系统，已经成为RDF(S)和OWL所表达的Web本体相匹配的一种实用和流行的选择。编程语言为Java。

- **联系人**

  胡伟

- **主页**

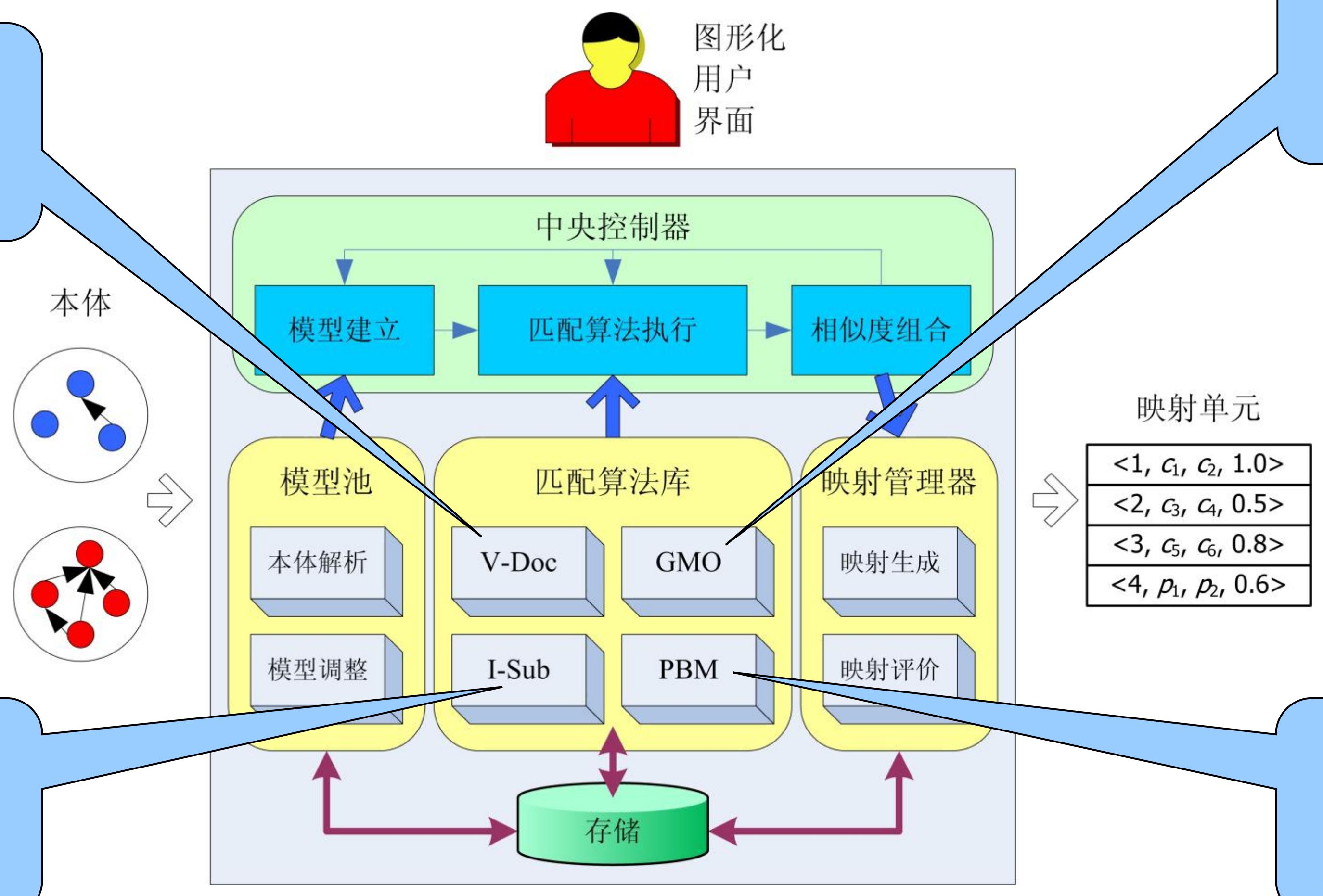
  http://ws.nju.edu.cn/falcon-ao/

本体匹配—Falcon-AO

□ 系统架构

基于虚拟文档的语言学匹配

基于本体RDF图结构的匹配

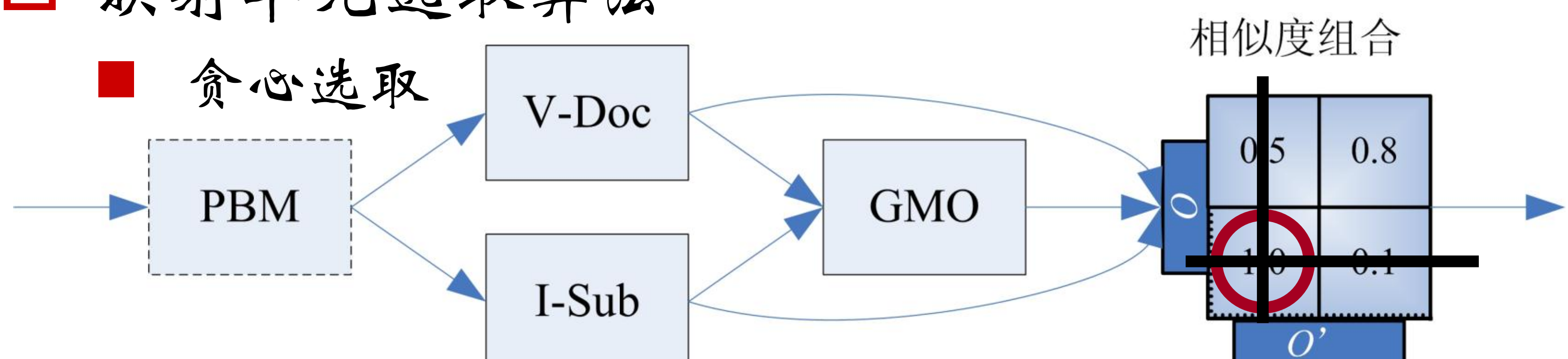


基于编辑距离的字符串匹配

基于分而治之的大本体匹配

相似度组合策略

- 语言学可比性
  - 语言学算法找到的映射单元数目对比本体概念数目
- 结构可比性
  - 本体间使用的原语的数目对比
- 映射单元集成
  - 语言学可比性和结构可比性分别分为高、中、低档
- 映射单元选取算法
  - 贪心选取



本体对齐—Falcon-AO

- Model Pool

  Parser用Jena将输入的本体导入到模型中，coordinator内置了一套协调规则用来调整模型。

- Matcher Library

  管理4种匹配方法，包括V-Doc , I-Sub, GMO 和PBM 。其中， V-Doc 和I-Sub是基于语言的匹配器，GMO是基于图结构的匹配器， PBM用分而治之的策略为大规模的本体进行块映射。

- Alignment Set

  以XML/RDF格式生成匹配文件，并用传统的准确率/召回率进行评测。

- Central Controller

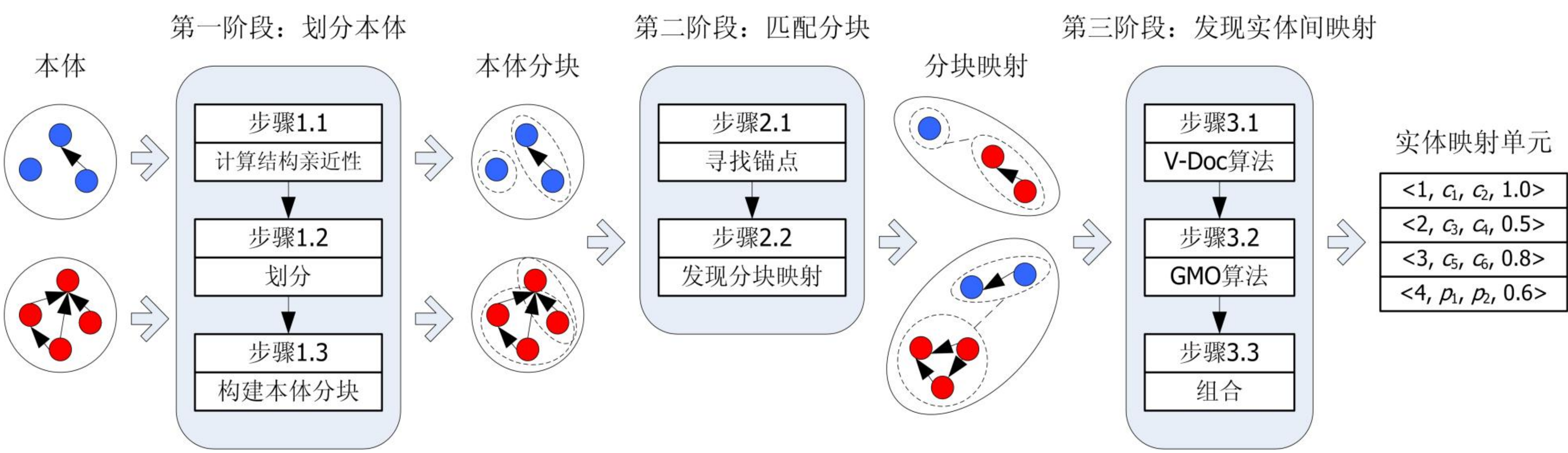
  手动调节参数，可以选择用Matcher Library 中的哪种方法进行匹配。

- Repository

  用来存储中间数据。

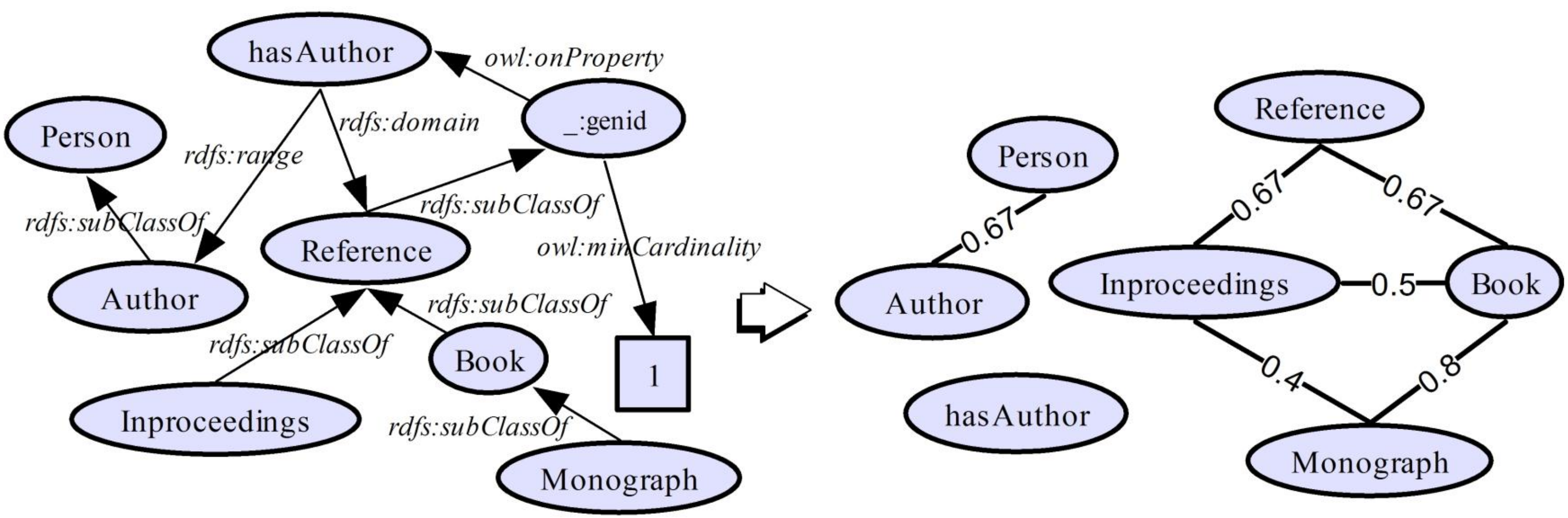
Falcon-AO —分块

- 一种基于分而治之策略的大型本体匹配方法



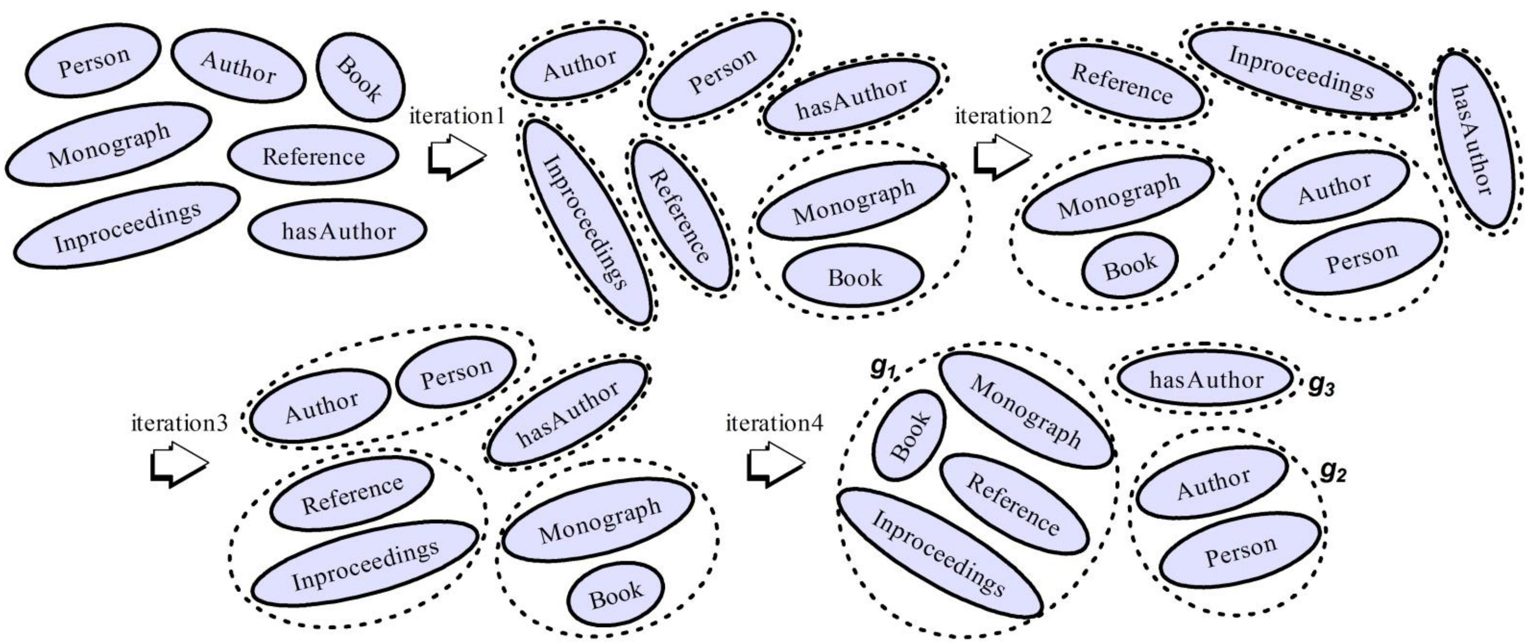
本体划分：概念间的结构亲近性计算

- 类：层次关系(公共父类)
- 属性：层次关系、定义域等



本体划分：本体划分算法

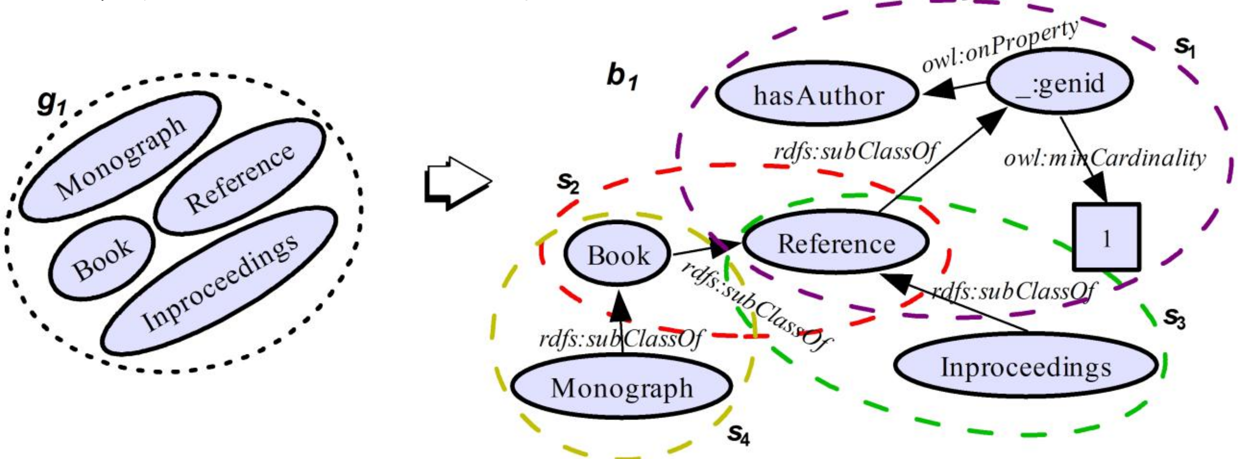
- 一个自底向上的划分算法
  - 每个聚类内结点间的内聚度较高
  - 不同聚类内结点间的耦合度较低



本体划分：本体分块的构建

- 定义：设O是一个本体，E是O中所包含概念的集合。针对E的一个划分G，把E分成一个聚类集合$G = \{g_1, g_2, \ldots, g_n\}$，满足(1) $\forall g_i, g_j$, $i, j = 1, 2, \ldots, n$且$i \neq j$，$g_i \cap g_j = \phi$；同时(2) $g_1 \cup \ldots \cup g_n = E$。设$b_i$是对应于$g_i$的唯一本体分块 $(i = 1, 2, \ldots, n)$。b_i是一个RDF句子的并集 $(b_i = sent_1 \cup \ldots \cup sent_m)$，其中每个$sent_k$ $(k = 1, 2, \ldots, m)$满足$sent_k$的主语属于g_i。$b_i, b_j, \forall i, j = 1, 2, \ldots, n$且$i \neq j$，$b_i \cap b_j = \phi$。

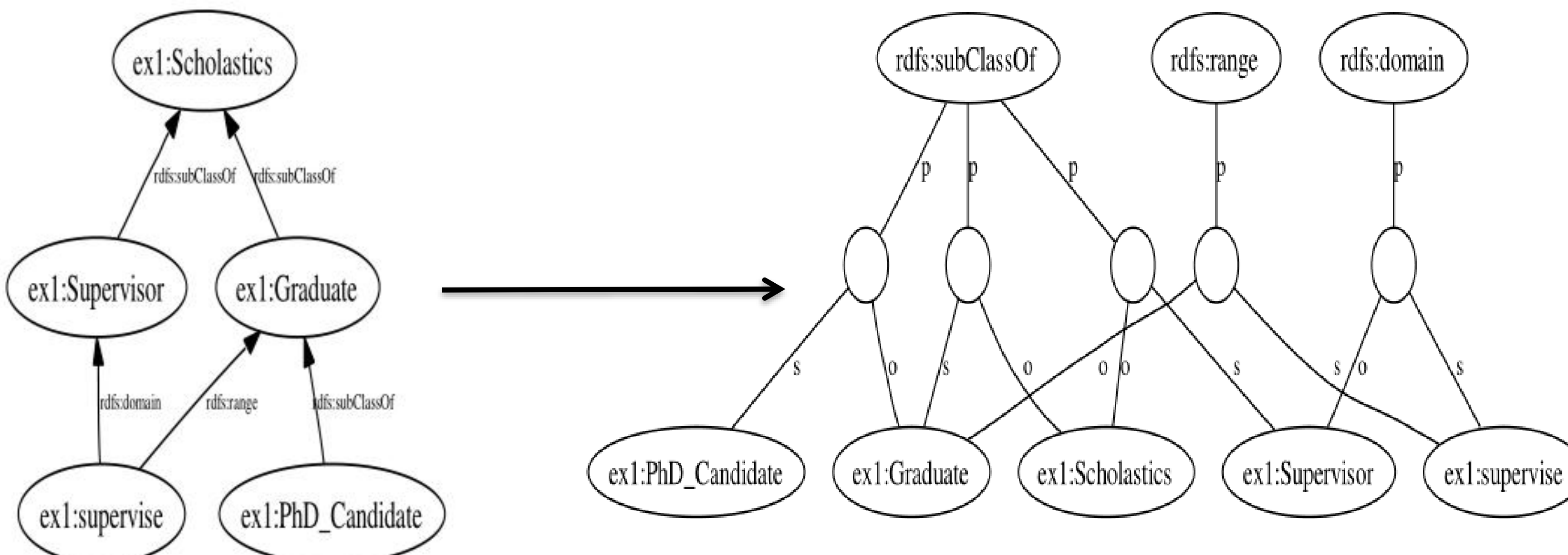
  - **RDF句子**：RDF三元组的集合，保证匿名结点完整性



本体对齐—Falcon-AO

- **GMO** (**G**raph **M**atch for **O**ntology)

  GMO是基于图结构的本体匹配方案，它使用RDF二部图来表示本体，并通过在二部图中递归传播相似性来计算实体和三元组之间的结构相似性。一般，用V-Doc和I-Sub的输出作为它的输入。



RDF图 RDF二部图

实体匹配—Dedupe

- 概要

  Dedupe是一个用于模糊匹配，记录去重 和实体链接的python库。

- github主页

  http://www.openkg.cn/tool/dedupe

- 作者

  Forest Gregg

  Derek Eder

实体匹配—Dedupe

1. 指定谓词集合 & 相似度函数

Dedupe的输入需要指定属性的类型，在内部为给每个属性类型指定一个谓词集合以及相似度计算方法。

右图是对‘比尔盖茨’的’name’属性的简单描述，将每个属性都映射上去，会形成一个大的谓词集合。

subject	predicate	object
比尔盖茨	name	比尔盖茨

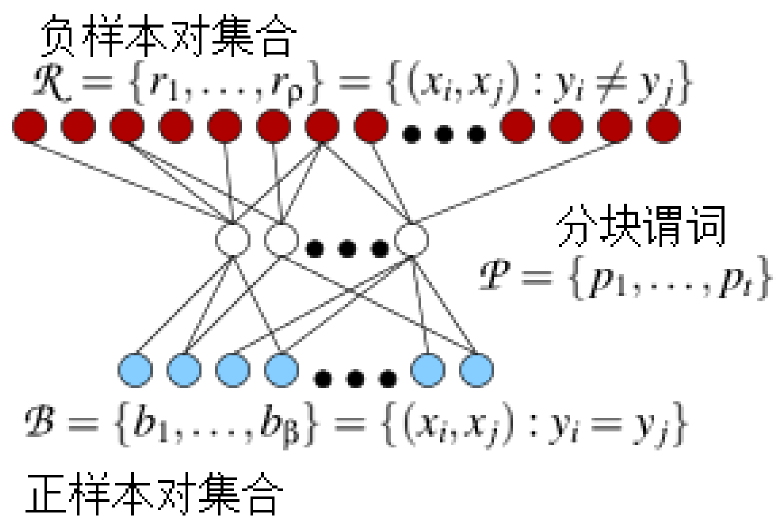
predicate	predicate type
name	string

predicate Type	predicate set	similarity function
string	{Contain Common Token⇔（比，尔，盖，茨）， Exact Match ⇔（比尔盖茨）， Same 1st Three Chars⇔（比尔盖） …}	affine gap edit distance

实体匹配—Dedupe

2. 训练Blocking： 通过 Red-Blue set cover 找到最优谓词集合来分块

最优谓词集合至少能覆盖95% (可以指定)的正样本对，负样本对被误分到同一个block中越少越好。



实体匹配—Dedupe

3. 训练逻辑回归(LR)模型

用用户标记的正负样本对训练LR模型，来进行分类。LR不能确定的会返回给用户进行标注(Active Learning)。

实体匹配—Limes

- 概要

  Limes是一个基于度量空间的实体匹配发现框架，适合于大规模数据链接，编程语言是Java。

- github主页

  http://openkg.cn/tool/limes

- Author

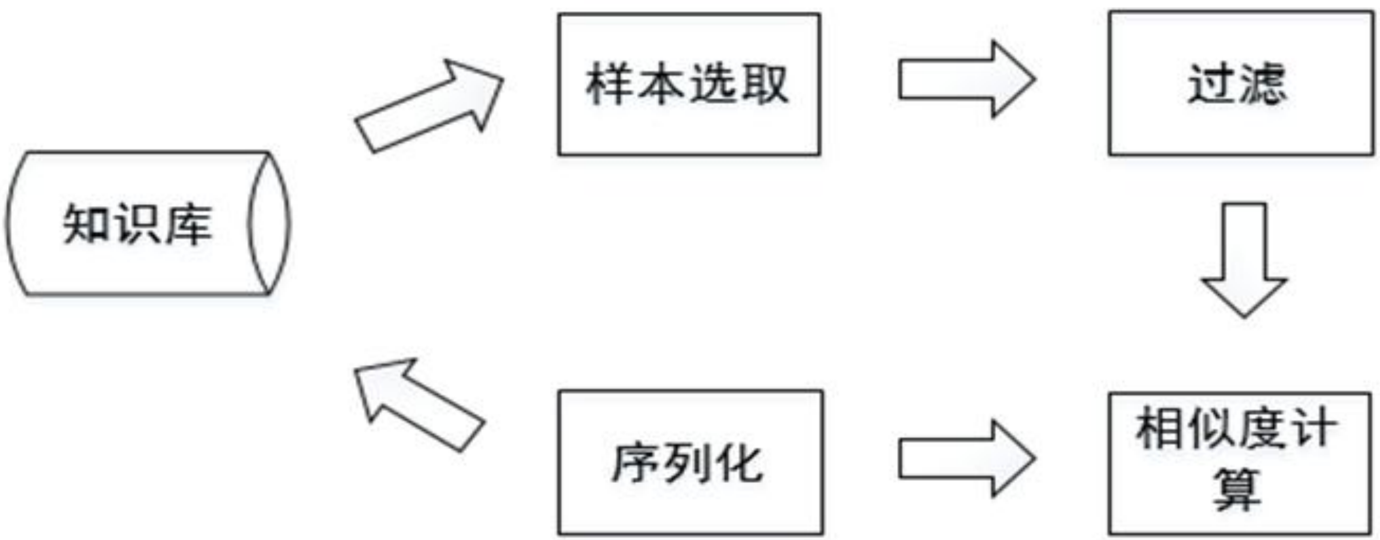
  Data Science Group at UPB

- 参考论文

  LIMES: a time-efficient approach for large-scale link discovery on the web of data. *IJCAI 2011*

实体匹配——Limes

□ 整体框架



实体匹配—Limes

整体流程

给定源数据集S，目标数据集T，阈值θ

1. 样本选取：从T中选取样本点E来代表T中数据。
2. 过滤：计算s ∈S与e ∈E之间的距离m(s, e)，利用三角不等式进行过滤。
3. 相似度计算：同上
4. 序列化：存储为用户指定格式

实体匹配—Limes

- 技术细节

1. 样本点选取

所谓样本点，也就是能代表距离空间的点。应该在距离空间上均匀分布，各个样本之间**距离尽可能大**。

实体匹配—Limes

- 技术细节

2. 三角不等式过滤

给定(A，m)，m是度量标准，相当于相似性函数，A中的点x,y和z相当于三条记录，根据三角不等式有：

$$m(x,y) \le m(x,z) + m(z,y)$$

上式通过推理可以得到：

$$m(x,y) - m(y,z) > \theta \Rightarrow m(x,z) > \theta$$

实体匹配—Limes

- 技术细节

2. 三角不等式过滤

上式中y相当于样本点。因为样本点E的数量是远小于目标数据集T的数量，所以过滤这一步会急剧减少后续相似性比较的次数，因而对大规模的web数据，这是非常高效的算法。

实体匹配—Silk

- 概要

  Silk是一个的集成异构数据源的开源框架。编程语言为python。

- Github主页

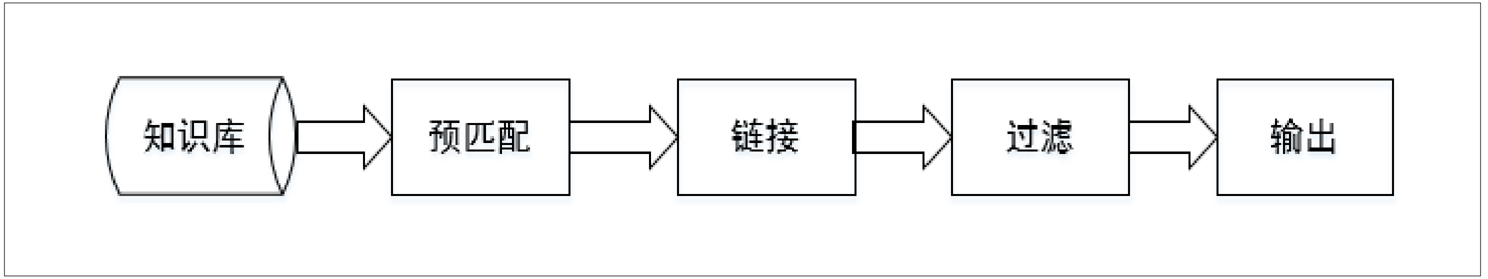
  https://github.com/silk-framework/silk

- Author

  UNIVERSITY of MANNHEIM

实体匹配——Silk

整体框架



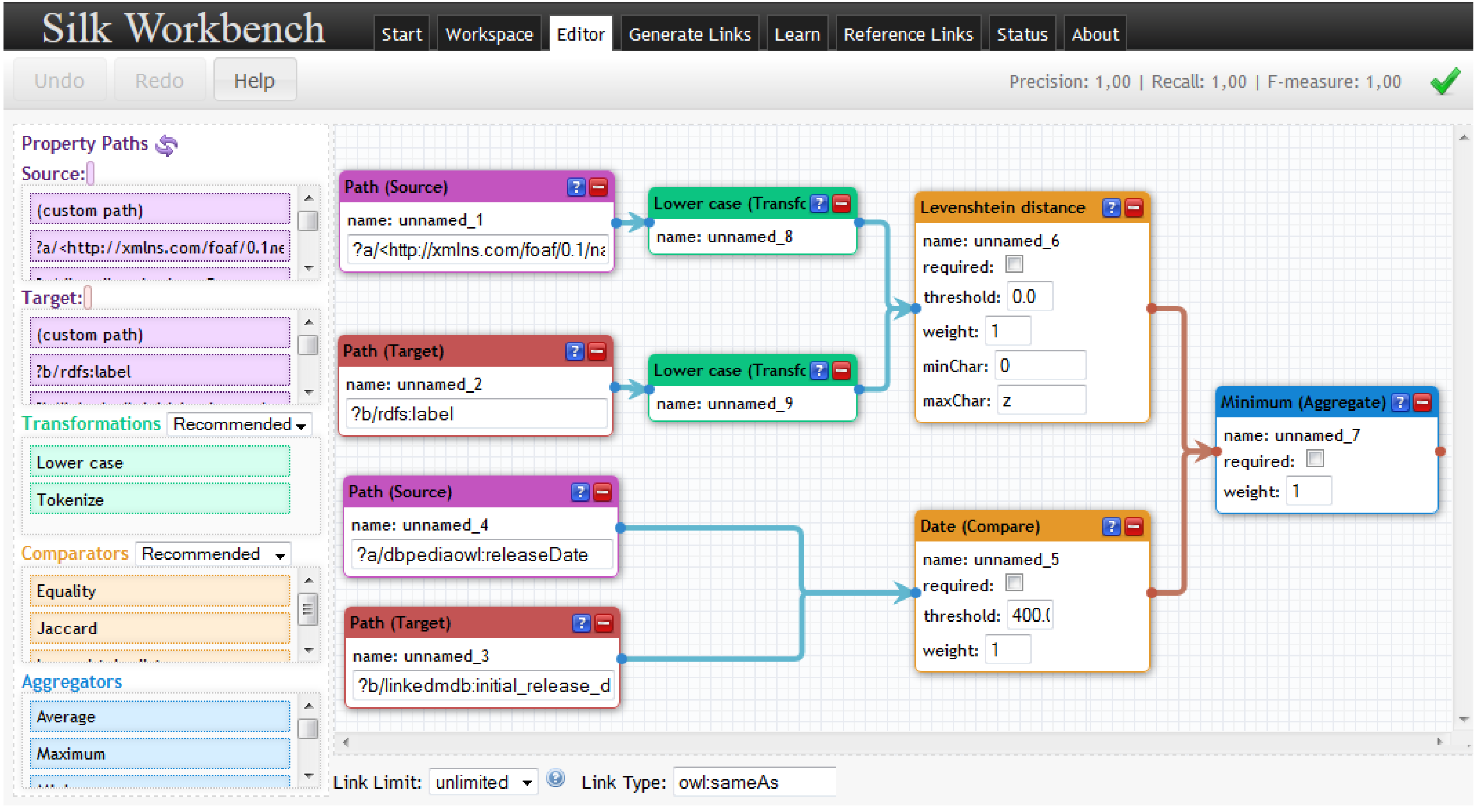
- 预处理：会将索引的结果排名前N的记录下来进行作为候选对，进行下一步更精准的匹配(损失精度)。
- 相似度计算：里面包含了很多相似度计算的方法。
- 过滤：过滤掉相似度小于给定阈值的记录对。

实体匹配—Silk

特点

1. 提供了专门的Silk-LSL语言来进行具体处理
2. 提供图形化用户界面—Silk Workbench，用户可以很方便的进行记录链接。

实体匹配—Silk



Silk Workbench
Start
Workspace
Editor
Generate Links
Learn
Reference Links
Status
About
Undo
Redo
Help
Precision: 1,00 | Recall: 1,00 | F-measure: 1,00
Property Paths
Source:
(custom path)
?a/<http://xmlns.com/foaf/0.1/na
Target:
(custom path)
?b/rdfs:label
Transformations
Recommended
Lower case
Tokenize
Comparators
Recommended
Equality
Jaccard
Aggregators
Average
Maximum
Path (Source)
name: unnamed_1
?a/<http://xmlns.com/foaf/0.1/na
Path (Target)
name: unnamed_2
?b/rdfs:label
Path (Source)
name: unnamed_4
?a/dbpediaowl:releaseDate
Path (Target)
name: unnamed_3
?b/linkedmdb:initial_release_d
Lower case (Transfo
name: unnamed_8
Lower case (Transfo
name: unnamed_9
Levenshtein distance
name: unnamed_6
required:
threshold: 0.0
weight: 1
minChar: 0
maxChar: z
Date (Compare)
name: unnamed_5
required:
threshold: 400.(
weight: 1
Minimum (Aggregate)
name: unnamed_7
required:
weight: 1
Link Limit: unlimited
Link Type: owl:sameAs

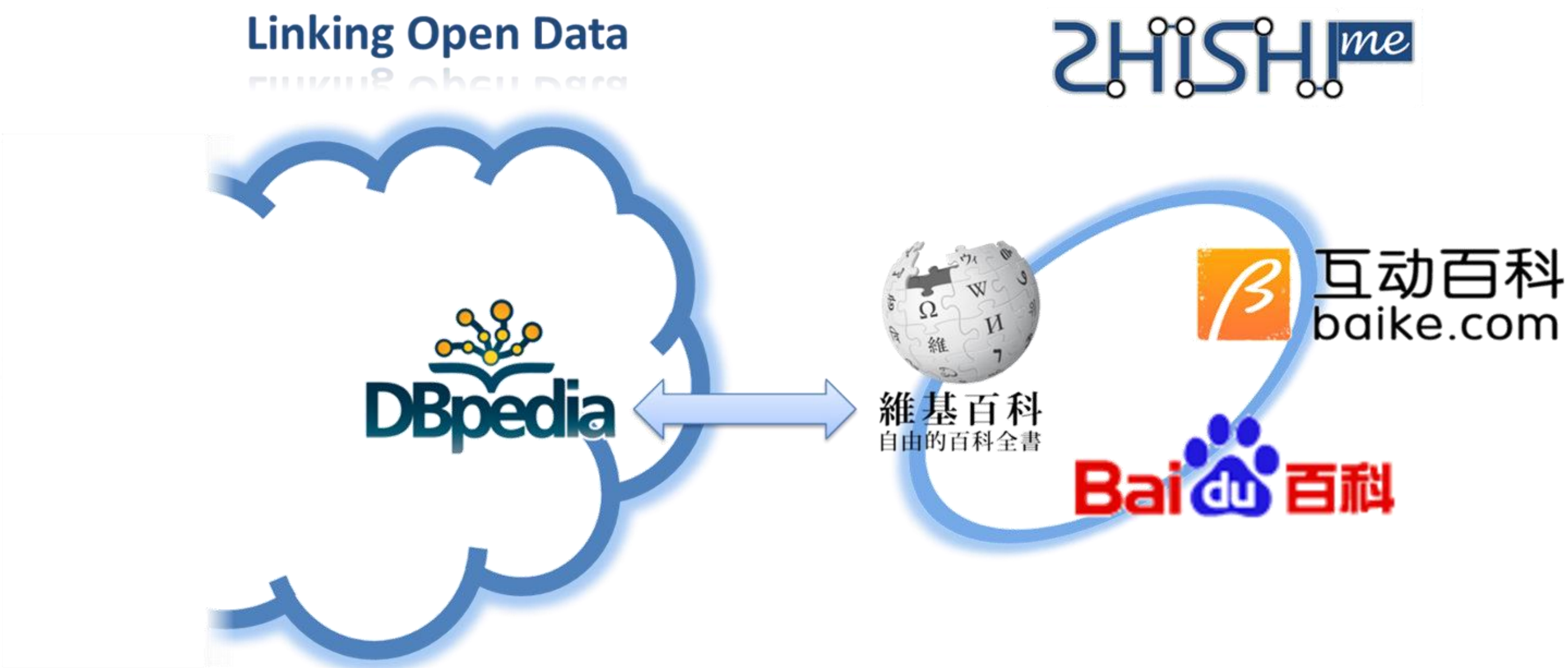
典型案例简介

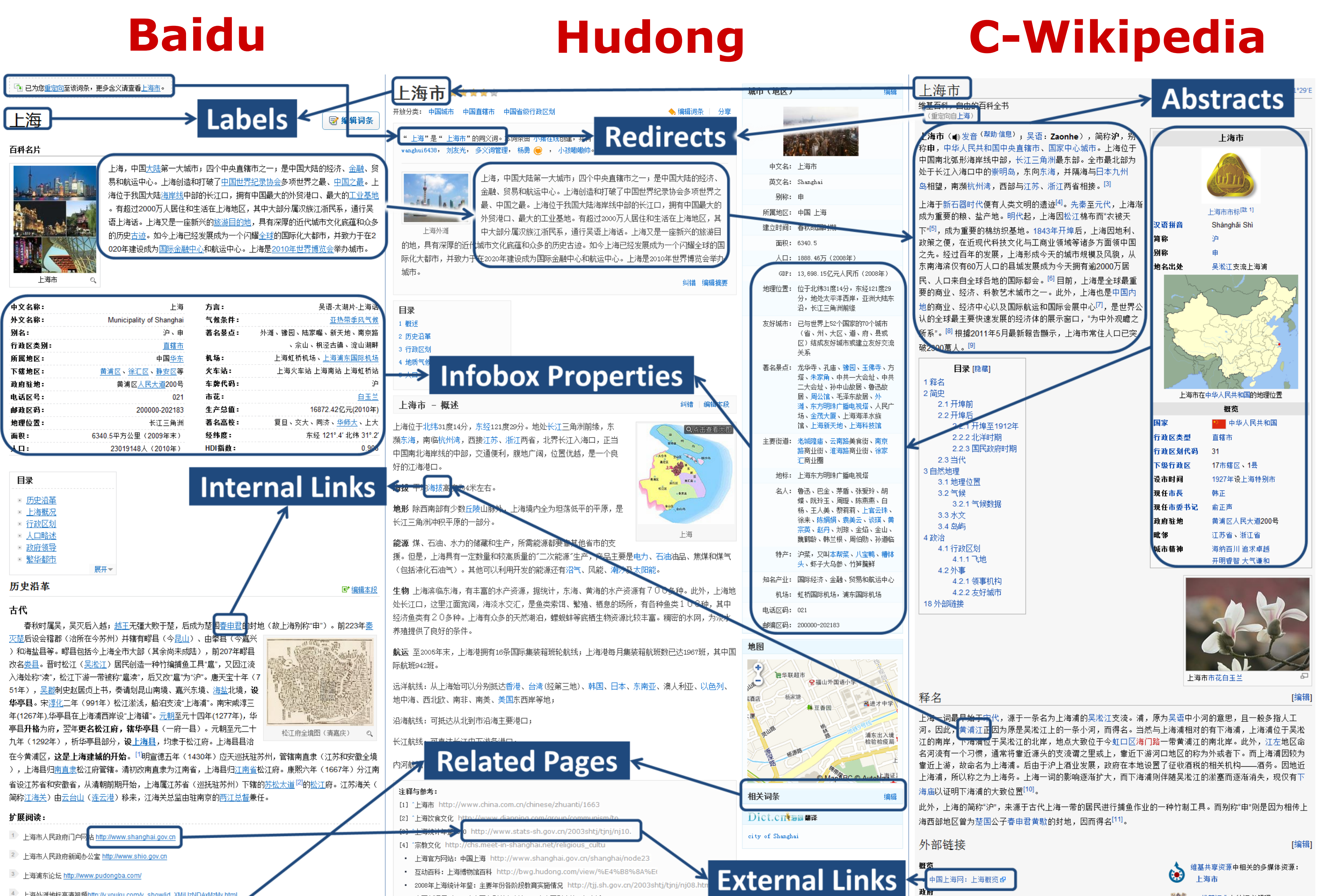
典型案例

- 案例一：Zhishi.me中的实体融合
- 案例二：OpenKG的链接百科

案例一：Zhishi.me中的实体融合

- Zhishi.me (http://zhishi.me) 是国内首次发布的大规模中文语义数据集，并致力于将已经发布数据集进行链接形成中文链接开放数据 (CLOD).





Baidu
Hudong
C-Wikipedia
Labels
Redirects
Abstracts
Infobox Properties
Internal Links
Related Pages
External Links

等价实体

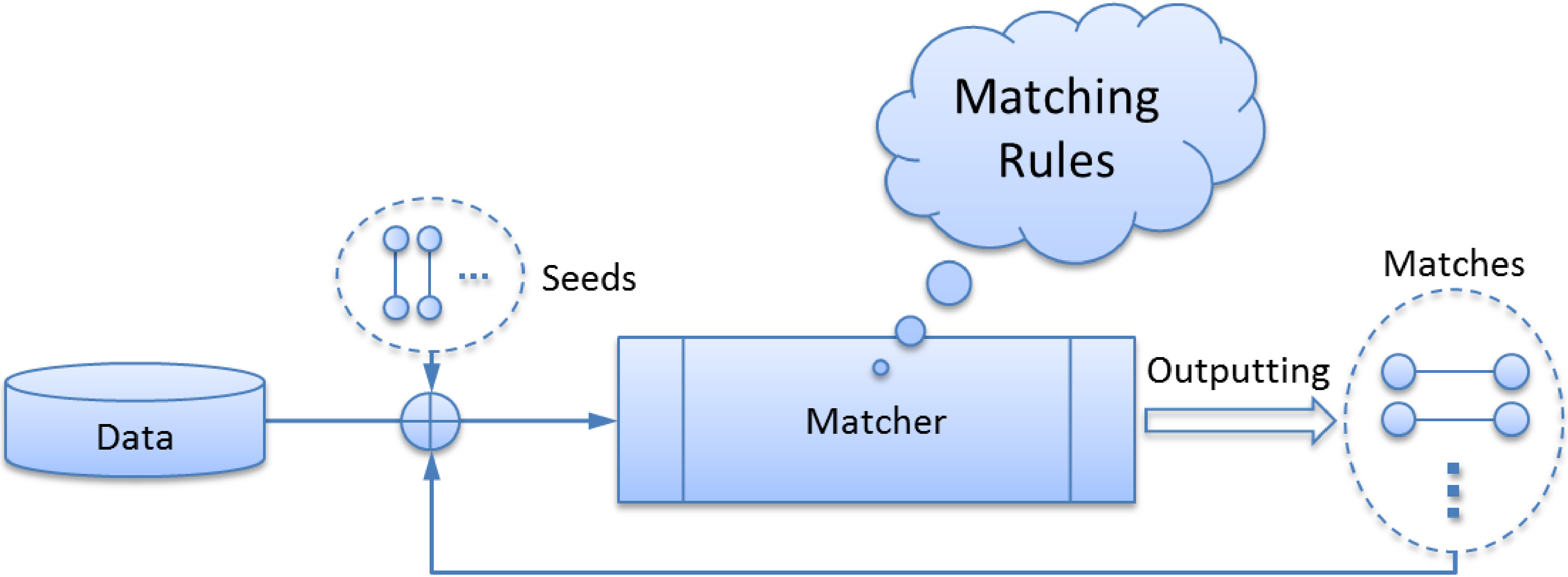
- Baidu Baike

baidu:大熊猫	
baidu:标签	"大熊猫"
baidu:拉丁学名	"Ailuropoda melanoleuca"
baidu:纲	baidu:哺乳纲
…	

- Hudong Baike

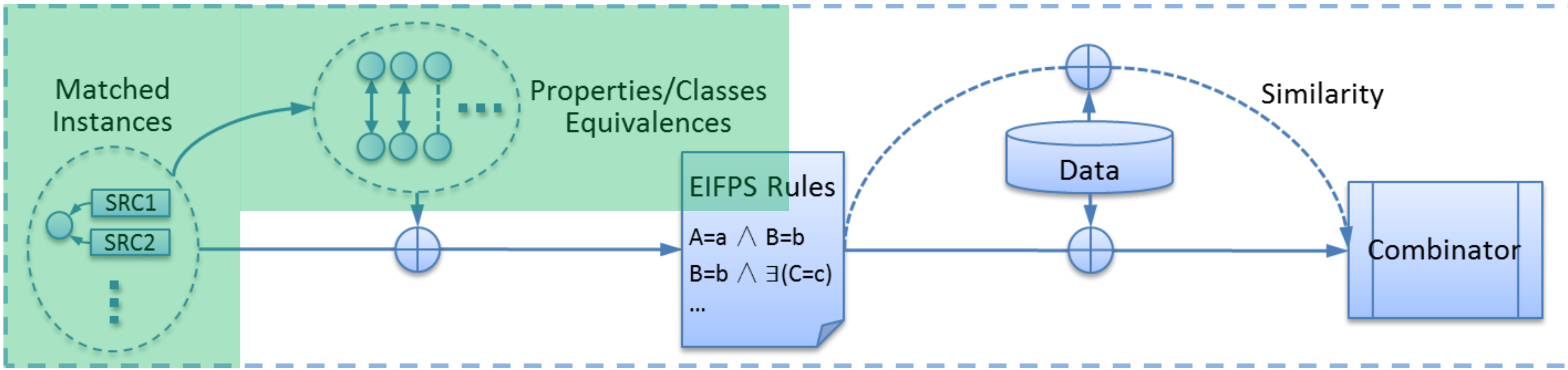
hudong:大熊猫	
hudong:中文学名	"大熊猫"
hudong:二名法	"Ailuropoda melanoleuca"
hudong:纲	hudong:哺乳纲
…	

解决方案



- 通过迭代，自动发现并修改特定数据集的匹配规则
  - 通过找出训练数据对中最具有判别力的数据来推导匹配规则

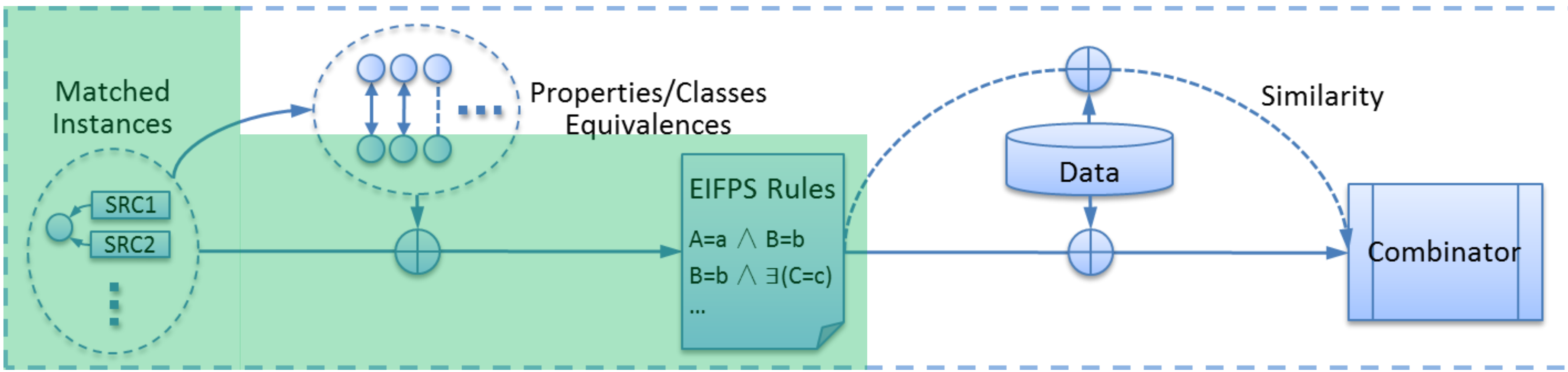
Workflow – 挖掘等价属性



- 合并现有已匹配的实例对的属性值

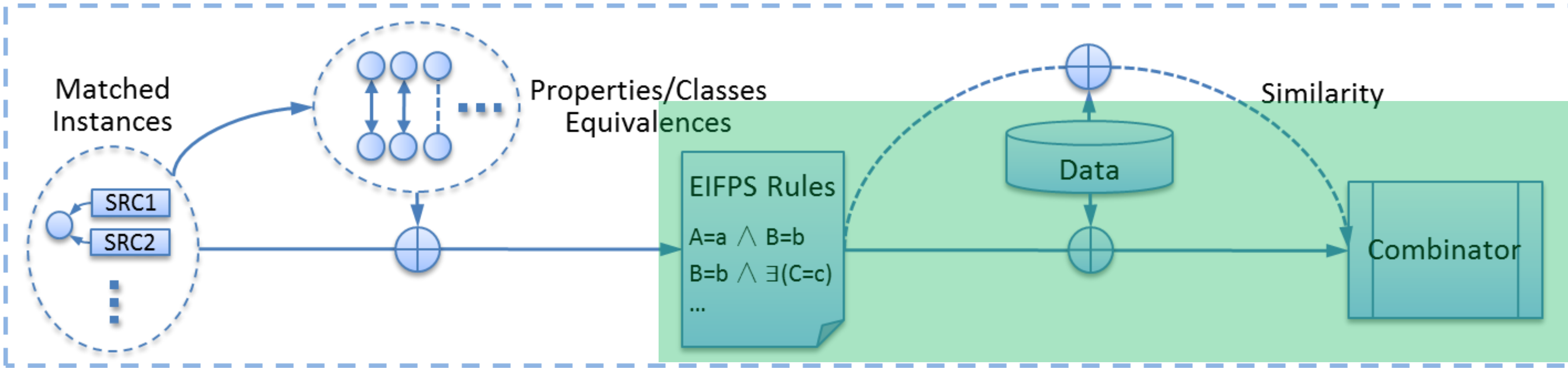
Values	Property_1	Property_2
"大熊猫"	baidu:标签	hudong:中文学名
"Ailuropoda melanoleuca"	baidu:拉丁学名	hudong:二名法
"白鳍豚"	baidu:标签	hudong:中文学名
"桂花"	baidu:标签	hudong:中文学名
…	…	…

Workflow – 挖掘匹配规则



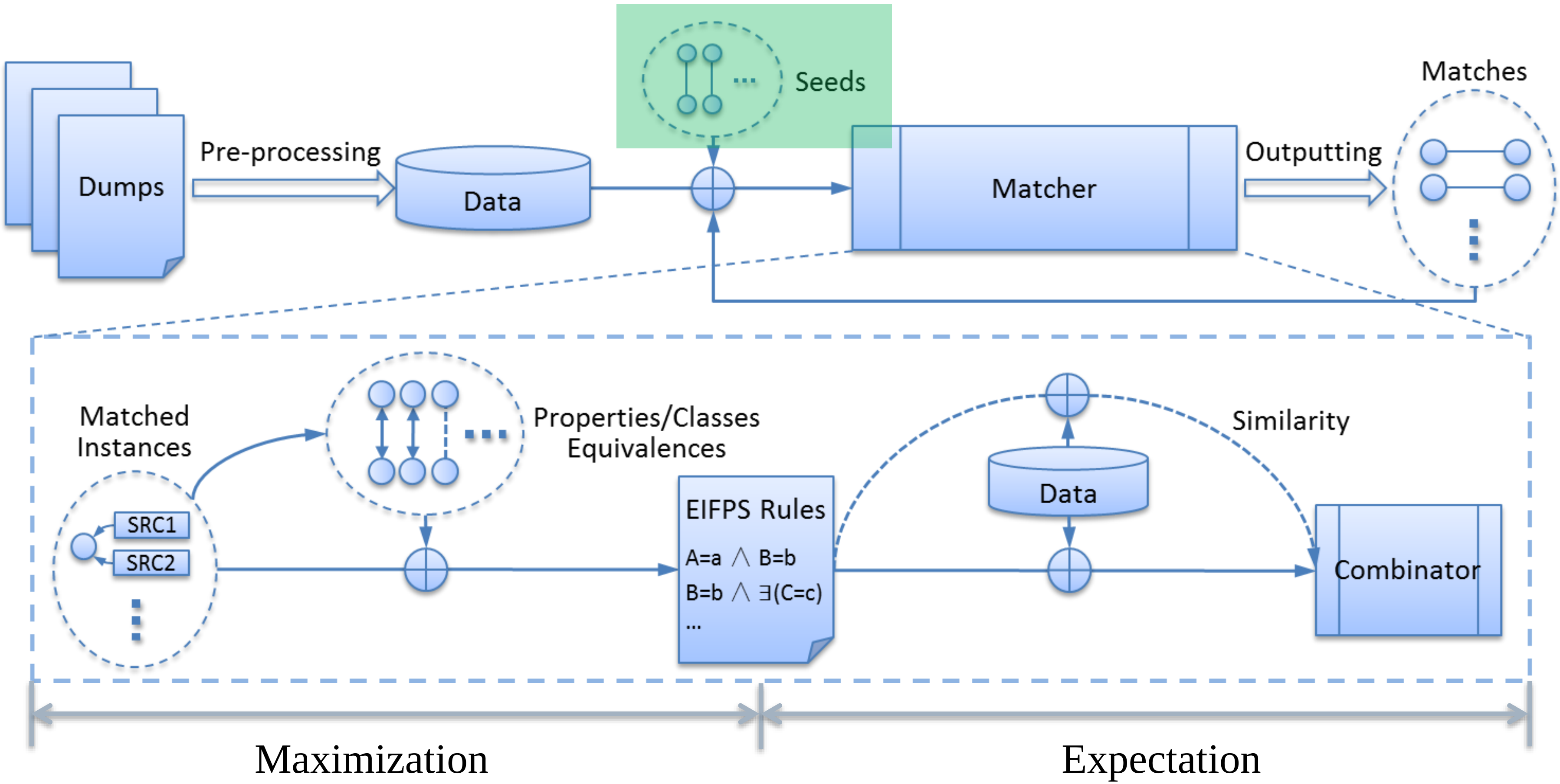
- 匹配规则 (frequent set mining):
  - baidu:x and hudong:x are matched, iff.
  - valueOf(baidu:标签) = valueOf(hudong:中文学名)
  - and
  - valueOf(baidu:拉丁学名) = valueOf(hudong:二名法)
  - and
  - valueOf(baidu:纲) = valueOf(hudong:纲)

Workflow – 生成匹配对



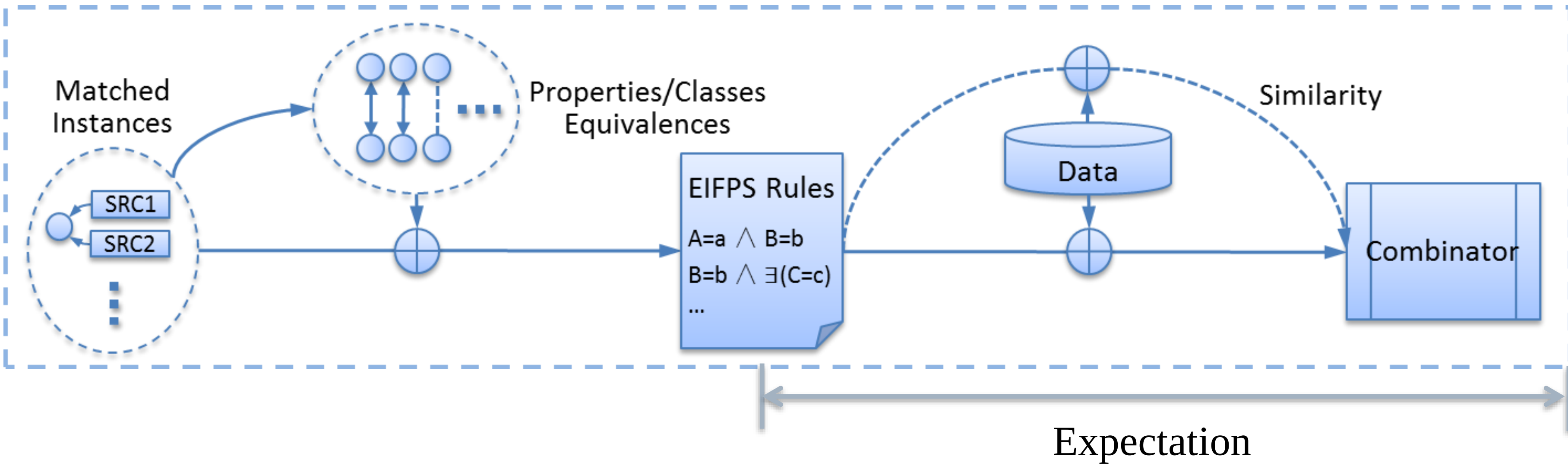
- 用得到的匹配规则处理未标记的数据，生成候选匹配对。
- Combiner用来计算候选匹配对的置信度。

Workflow – the Wrapper 算法



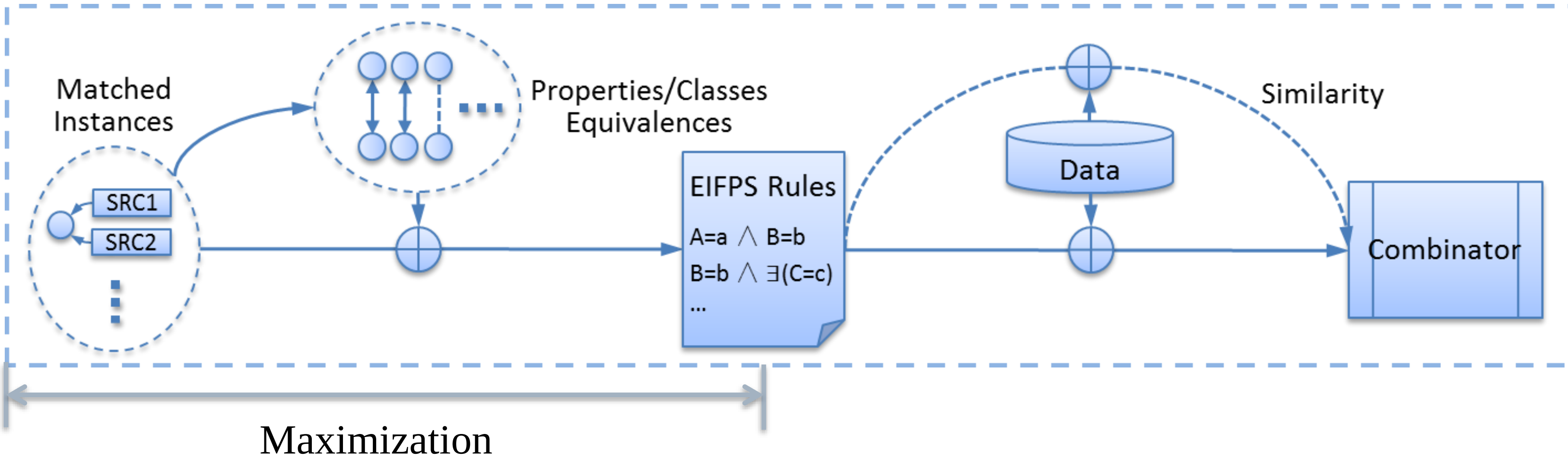
- Wrapper 是对EM迭代算法的封装

Workflow – E-step



- E-step用观测到的数据（已匹配的实例对)和目前估计的参数(匹配规则)来评估缺失数据(未匹配的实例对)。

Workflow – M-step



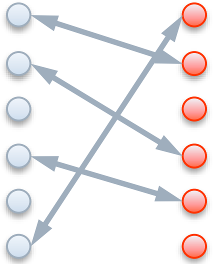
- 正如在E-step估计未匹配实例对一样，在M-step通过最大似然函数来估计参数
  - M: 匹配的实例对
  - Θ: 参数

$$L(\boldsymbol{\theta}; \boldsymbol{M}) = \Pr(\boldsymbol{M}|\boldsymbol{\theta}).$$

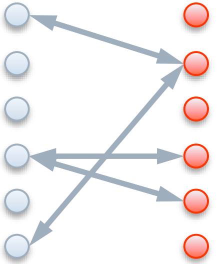
似然函数

$$L(\theta; M) \approx \frac{|\mathrm{ConnectedComponent}(M)|}{|\mathrm{Edge}(M)|}$$

- 假设每个数据集中不存在重复的实例，我们可以推断出，对于一个数据集中某个实例a1，在另一个数据集中能找到最多一个与此等价的实例a2。
- M中错误的matches会导致某个实例a1匹配到多个等价实例 (a2,a3…)，这与上面的假设相违背。

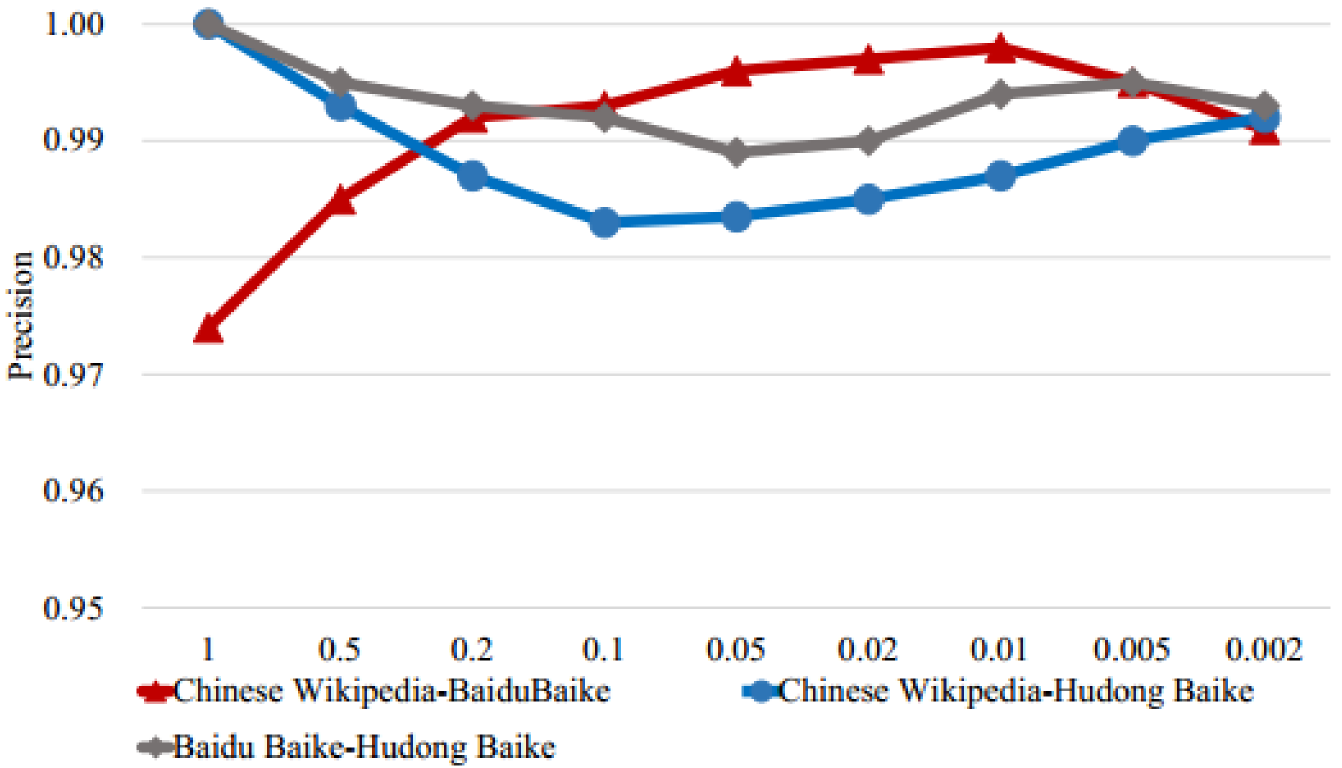


P=4/4=1



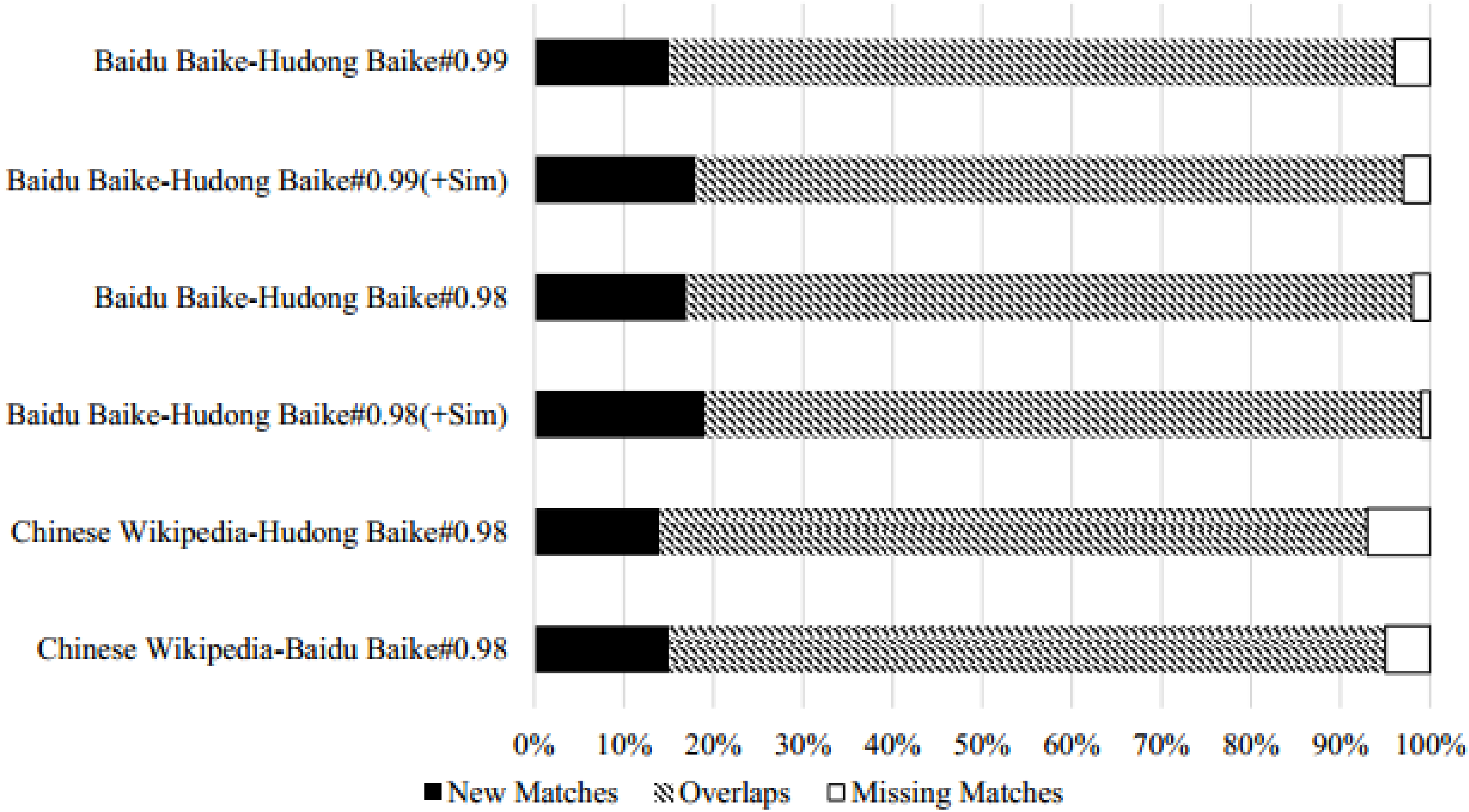
P=2/4=0.5

Evaluation – Precisions



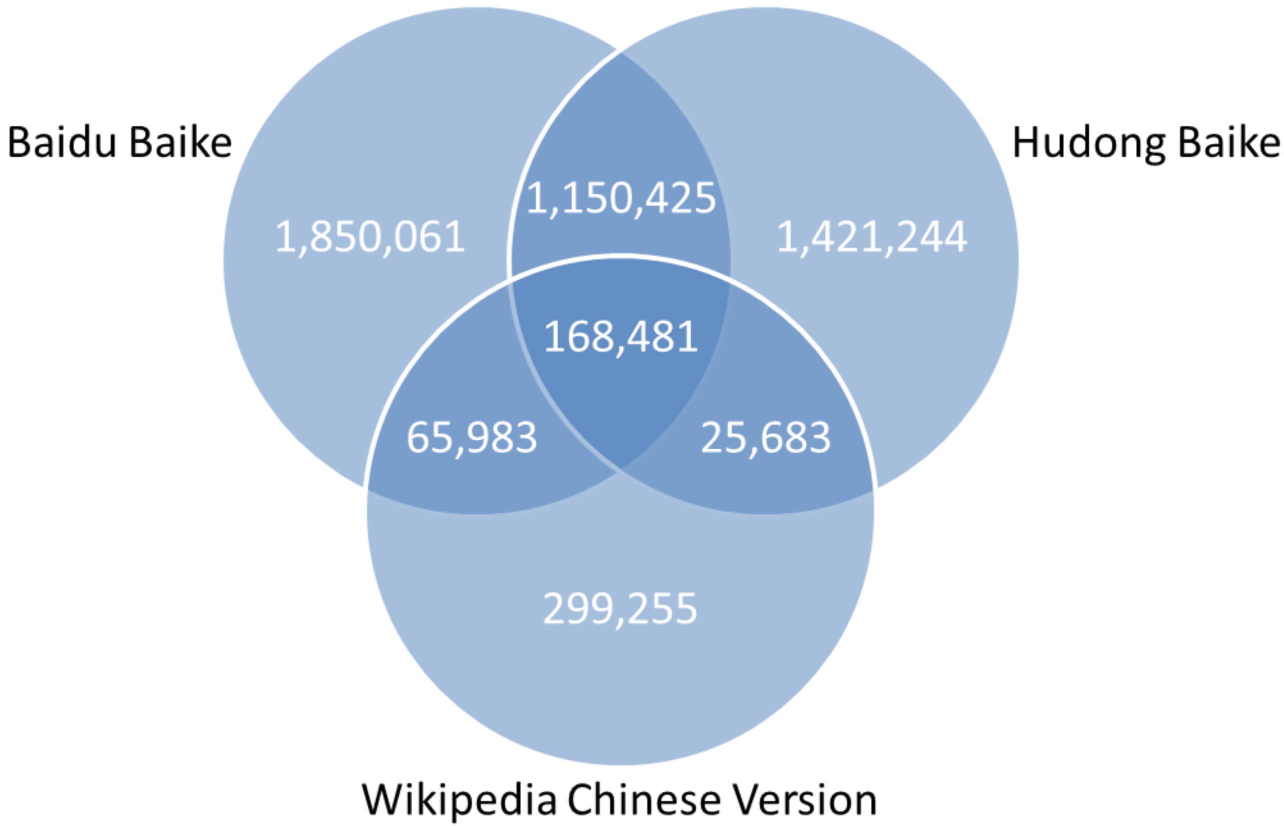
- 在输出matches中采样。
- X轴表示在现有的参考matches中选择作为初始seeds的比例。

Evaluation – 新发现的matches



- 匹配空间是由参考的matches和新产生的matches构成。

Results



- 准确度: ~97% (手动标记)

网页访问等价实例

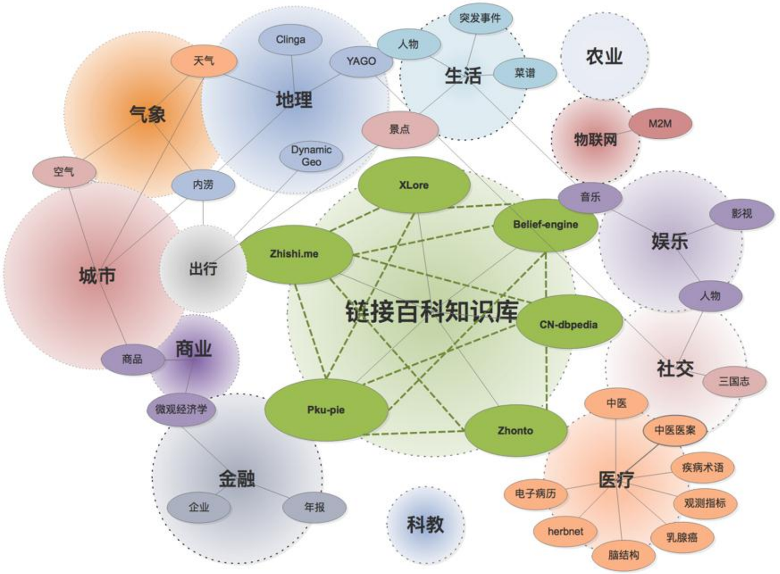
owl:sameAs
hudong:北京 (this)
baidu:北京
zhwiki:北京
MERGE PAGE

zhwiki:北京市
hudong:北京
baidu:北京
resources
Index
zhishi:abstract
infobox
dcterms:subject
zhishi:thumbnail
zhishi:relatedPage
<owl:sameAs> check box
owl:sameAs
zhwiki:北京市 (this)
baidu:北京 (this)
zhwiki:北京
hudong:北京 (this)
dbpedia:Beijing
hudong:北京市
baidu:北京市
MERGE PAGE
zhishi:abstract
北京有着三千余年的建城史和八百五十余年的建都史，最初见于记载的名字为"蓟"。民国时期，称北平。新中国成立后，是中华人民共和国的首都，简称"京"，现为中国四个中央直辖市之一，全国第二大城市及政治、交通和文化中心。北京位于华北平原北端，东南局部地区与天津相连，其余为河北省所环绕。它荟萃了元、明、清以来的中华文化，拥有众多名胜古迹和人文景观，是世界上拥有世界文化遗产最多的城市，每年有超过1亿4700万的旅客。
infobox
市花 月季、菊花
a merged statement
政府驻地 东城区正义路2号
下辖地区 海淀区 西城区 顺义区 大兴区 东城、西城、海淀、丰台、朝阳等
时区 UTC+8（东八区）
行政区类别 直辖市
友好城市 纽约、东京、巴黎
dcterms:subject
中华人民共和国直辖市 京津冀城市群 中国城市 首都 燕京 中国
zhishi:thumbnail
all other statements
zhishi:relatedPage
圆明园 SHOW MORE (182)

典型案例二：OpenKG的链接百科

□ 动机

目前OpenKG中已收录了76个中文数据集，目前已经融合几个百科类数据集Zhishi.me, CN-DBpedia, PKU-PIE和Belief Engine。



□ 数据集

http://openkg.cn/dataset

□ 链接结果

http://openkg.cn/dataset/links-encyclopedia

链接结果

名称	实体数
CN-DBpedia	16,546,273
Zhishi.me	10,337,505 (zhwiki, hudongbaike, baidubaike)
PKUBase	11,554,258
Belief Engine	1,423,828 (currently, hudongbaike)
XLore	10,856,042 (label only)

	CN-DBpedia	Zhishi.me	PKUBase	Belief Engine
CN-DBpedia	-	5,393,835	6,347,721	1,538,865
Zhishi.me	5,393,835	-	5,687,428	697,123
PKUBase	6,347,721	5,687,428	-	552,066
Belief Engine	1,538,865	697,123	552,066	-

详细结果

CN-DBpedia被链接实体	链接总数	Zhishi.me被链接实体
528,449	576,989	(zhwiki) 164,043
1,767,294	1,912,165	(hudongbaike) 1,065,991
2,754,271	2,904,681	(baidubaike) 1,960,806

PKUBase被链接实体	链接总数	CN-DBpedia被链接实体
6,320,826 (54.7%)	6,347,721	3,887,078 (23.5%)

PKUBase被链接实体	链接总数	Zhishi.me被链接实体
3,887,078 (33.6%)	5,687,428	5,668,632 (54.8%)

Zhishi.me被链接实体	链接总数	Belief Engine被链接实体
(hudongbaike) 697,123	697,123	697,123 (49.0%)

CN-DBpedia被链接实体	链接总数	Belief Engine被链接实体
1,455,305 (8.8%)	697,122	640,070 (45.0%)

PKUBase被链接实体	链接总数	Belief Engine被链接实体
546,670 (4.7%)	697,122	547,748 (38.5%)

链接方法

- 通过繁简转换后的实体名严格匹配
- 利用wikidata扩展别名

- CN-DBpedia与Zhishi.me的链接，使用了CN-DBpedia中具有别名含义的属性扩充实体别名
- 由于PKUBase里没有可用的具有别名含义的属性，因此它没有上述链接过程
- Belief Engine的链接过程与其他完全不同，使用互动百科的唯一“article title”与Zhishi.me中的互动百科实体进行链接

LIMES 实战演练

基于limes的中文知识图谱融合实战演练教程

Limes使用教程和数据集

数据与资源

Metrics实例测试数据

Tutorial中使用的Source和Target数据

浏览

MLALgorithm使用实例-领域数据集之间的匹配

浏览

MLALgorithm使用实例-领域与百科数据集之间的匹配

浏览

http://openkg.cn/dataset/limes-tutorial

谢谢大家！