

# 基于知识图谱的语言理解

---

# 本章大纲

---

- 语言理解概述
- 理解实体
- 理解概念
- 理解属性

# 概述

---

# 语言理解的重要性

- 语言理解是重要的思维活动之一
  - 语言是思考的工具
  - 正是语言表达和语言理解能力将我们与动物区分开来
- 
- 机器的语言理解
    - 其输入是自然语言，输出则是语言认知的各种结果，包括实体、概念、关系、场景、主题以及内涵等
    - 涵盖从浅层次到深层次等各种形式的理解任务



机器理解人类语言是实现智能信息处理和机器智脑的重要途径

# 语言理解的挑战——自然语言的复杂性

---

- 表达的**多样性**

- 例如：“妻子”“老婆”“夫人”“娘子”“太太”“内人”“拙荆”

- 表达的**歧义性**

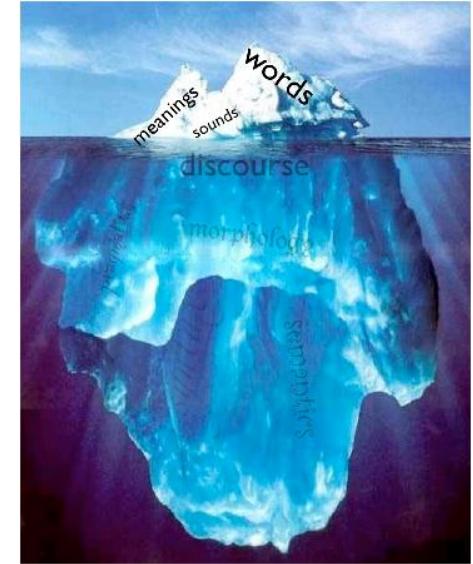
- “苹果”：①水果；②科技公司
  - “青藏高原”：①地理名词；②歌曲名

- **上下文关联**

- “倒了一杯水”：①往杯子里倒水；②把杯子里的水倒出去

# 语言理解的挑战——机器理解自然语言困难

- 人类的语言理解建立**人类认知**为基础
- 认知需要大量的**背景知识**
- 语言理解的冰山现象
  - 沟通主体之间的“**共识**”所构成的巨大背景知识库是 人类彼此理解的基础

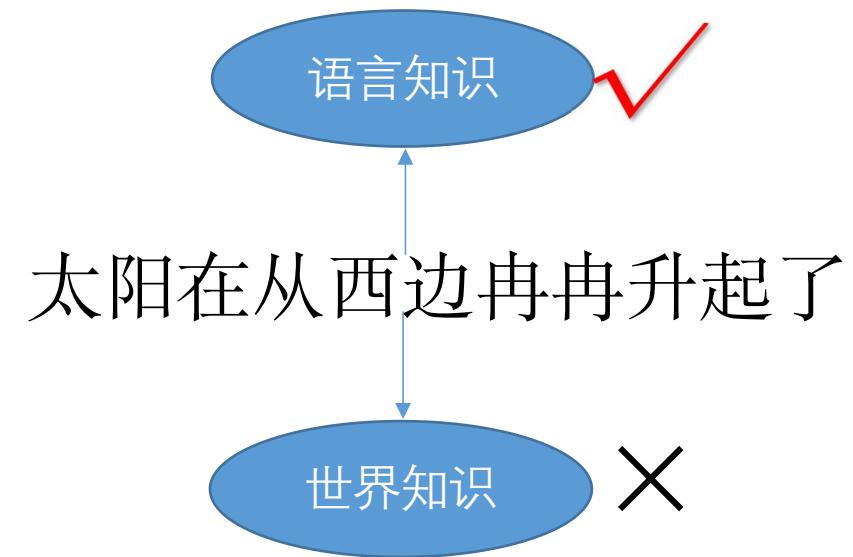


New *Frozen* Boutique to Open at Disney's Hollywood Studios

The diagram illustrates the interconnected nature of language and context. At the top, a news headline is shown: "New *Frozen* Boutique to Open at Disney's Hollywood Studios". Below it, there are three images: a poster for the movie "Frozen", the Disney logo, and the logo for "Disney's Hollywood Studios". Arrows point from each of these images down to their respective Wikipedia pages: "/wiki/Frozen\_(2013\_film)", "/wiki/The\_Walt\_Disney\_Company", and "/wiki/Disney's\_Hollywood\_Studios". This visualizes how a single statement can depend on a complex web of shared knowledge and context.

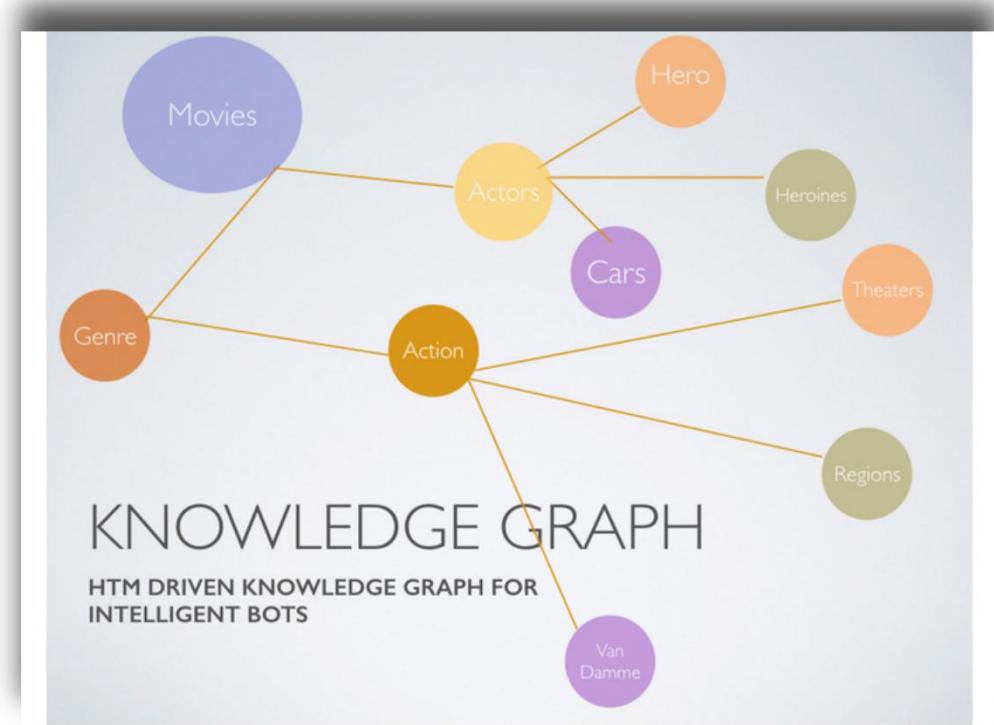
# 机器语言理解需要良好的背景知识

- 传统NLP使用知识的尝试
  - 人工定义的语法规则：难穷举
  - 领域专家定义的本体：规模有限
  - 从文本中自动挖掘语法或语义模式：非结构化抽取仍然困难
- 语言知识、语法知识不足以理解世界
- 机器的语言理解需要世界知识



# 知识图谱是一个好的选择

- 知识图谱的优势
  - 规模巨大
    - 动辄数十亿实体
  - 关系多样
    - DBpedia包含数千种语义关系
  - 结构友好
    - RDF三元组
  - 质量精良
    - 多源交叉验证；众包验证

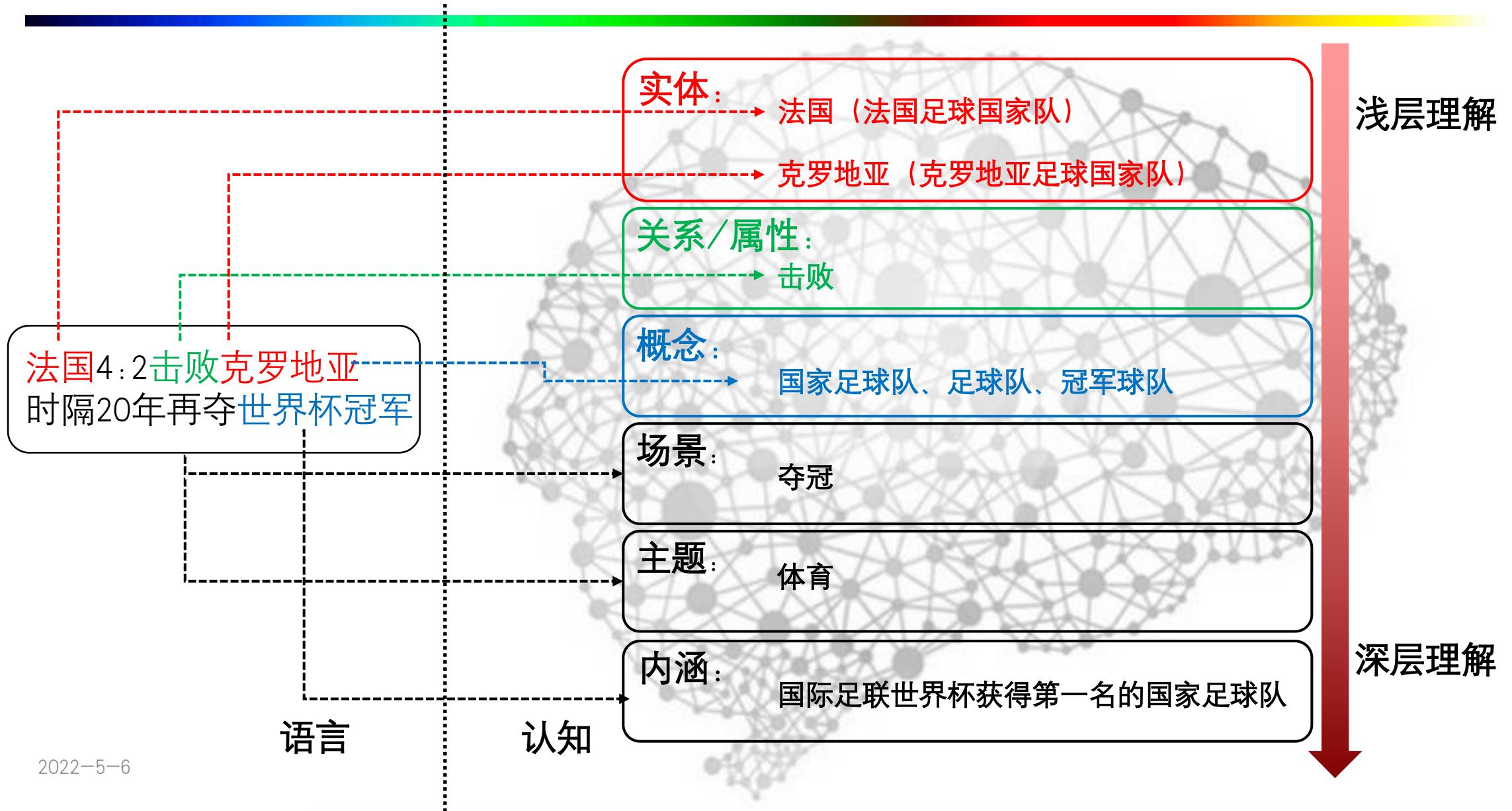


# 语言理解任务

---

- 机器语言理解是指，机器在接收自然语言输入后，形成相应的内在表示的过程。
- 根据内在表示的不同，语言理解主要分为以下三类任务
  - 语法解析（Syntactic Parsing）
  - 语义解析（Semantic Parsing）
  - 特定的知识表示或者其中的某个片段
    - 从自然语言形式的文本映射到知识图谱中的实体、概念、关系、路径以及子结构。
- 从第三类语言理解角度看，语言理解的本质是从文本到知识库的映射。

# 语言理解



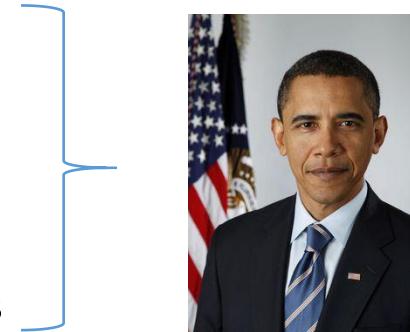
# 实体理解

---

# 实体理解

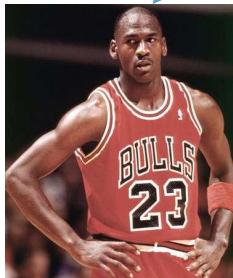
- 理解指代：同一个实体存在不同指代 (mention)

Barack Obama  
Barack H. Obama  
President Obama  
Senator Obama  
President of the United States



- 实体消歧/实体链接：同一个指代词可能指代多个不同实体

NBA Michael Jordan



Machine Learning Michael Jordan



# 理解实体指代

- 利用在线百科丰富的结构化信息构建实体—指代映射表



# 实体链接：问题描述

- 将文本中的实体指代链接到知识库中特定实体
  - 主要问题：指代词歧义



知识图谱

# 实体链接：基本流程



# 实体链接：基本模型

- 输入
  - 文本：以及上下文已识别指代集  $M = (m_1, m_2, \dots, m_n)$ ，知识库  $K$
- 输出：链接实体列表  $\Gamma = (e_1, e_2, \dots, e_n)$

- 模型：
$$\Gamma_{\text{best}} = \arg \max_{\Gamma} (\sum_{i=1}^n \varphi(m_i, e_i) + \psi(\Gamma))$$



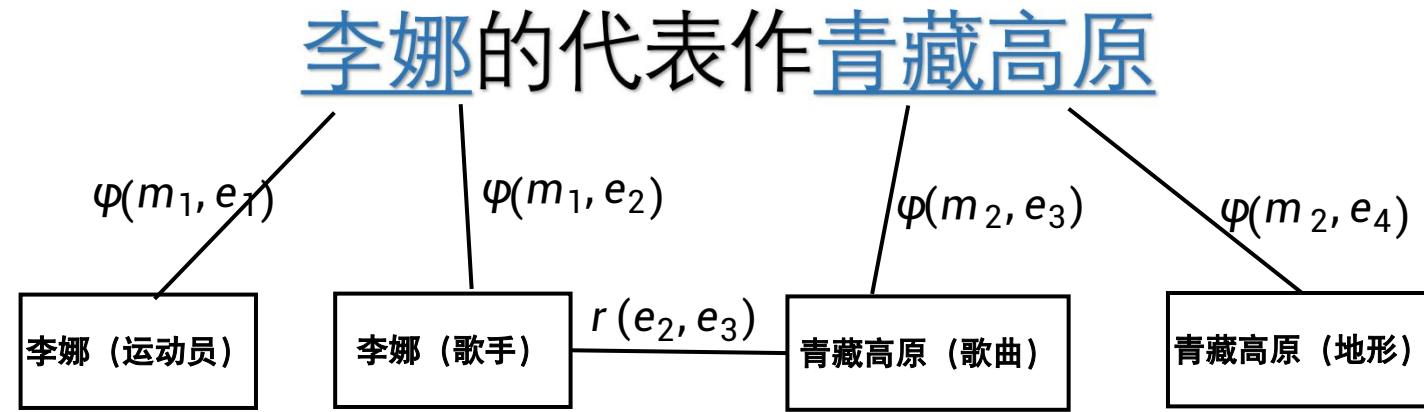
局部匹配函数：刻画指代与实体的匹配程度计算

$$\varphi(m_i, e_i) = \sum_k w_k f_k(e_i, m_i)$$

全局匹配函数：考虑实体对之间的语义关联强度

$$\psi(\Gamma) = \sum_{e_i \in \Gamma, e_j \in \Gamma} r(e_i, e_j)$$

# 实体链接：模型示例



# $\varphi$ 局部特征

- 上下文无关特征
  - 指代与实体名的字符相似度
    - 实体的指代与实体名通常是有一定相似度的
      - Eg: “周杰伦”与“杰伦”、“周董”
  - 实体流行度 (popularity)
    - $p(e = \text{北京}(\text{中华人民共和国首都})) >$
    - $p(e = \text{北京}(\text{北宋时期行政区划})) >$
    - $p(e = \text{北京}(\text{诗歌}))$
  - 百科锚文本先验概率
    - $p(e|A = \text{'李娜'})$
    - 从百科文本中统计出来，依赖大量锚文本

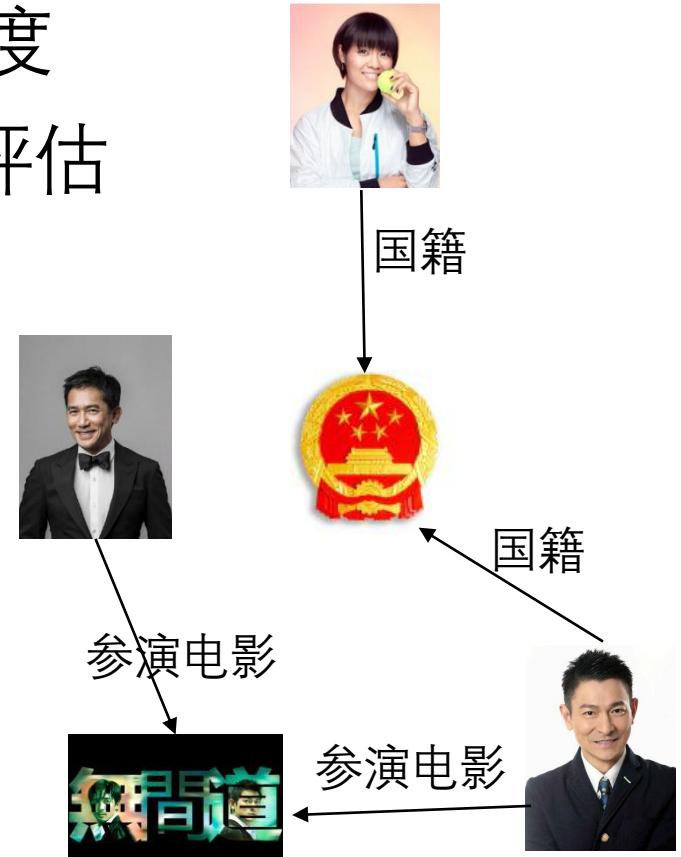
## • 上下文有关特征

- 文本相似度：候选实体相关文本与上下文文本的文本相似度
  - Eg, 利用候选实体及上下文的词袋向量或概念向量，计算相应的 cosine similarity

$$f(me) = \frac{\sum_{w \in \text{Coext} \cap (m \cup T_{\text{ext}}(e))} \text{TFIDF}_m(w) \times \text{TFIDF}_e(w)}{\sqrt{\sum_{w \in \text{Coext} \cap (m)} \text{TFIDF}_m(w)} \times \sqrt{\sum_{w \in T_{\text{ext}}(e)} \text{TFIDF}_e(w)}}$$

# $\psi$ 全局特征

- 候选实体( $e_1$ )与上下文实体( $e_2$ )之间的语义相关度
- 基于实体在知识图谱中的邻居集合 $U_1$ 和 $U_2$ 进行评估
- Jaccard相似度:  $JAC(u_1, u_2) = \frac{|U_1 \cap U_2|}{|U_1 \cup U_2|}$
- 互信息:  $PM(u_1, u_2) = \frac{|U_1 \cap U_2|/|W|}{|U_1|/|W| * |U_2|/|W|}$   
(联合概率除以各自独立的概率)
- 规范化谷歌距离:  $NG(u_1, u_2) = 1 - \frac{\log(\max(|U_1|, |U_2|)) - \log(|U_1 \cap U_2|)}{\log(|W|) - \log(\min(|U_1|, |U_2|))}$   
(对集合大小偏差做规范化)
- Adamic Adar 相似度:  $AA(u_1, u_2) = \sum_{n \in A \cap B} \log \left( \frac{1}{\text{degree}(n)} \right)$   
(对popular的公共邻居进行惩罚)

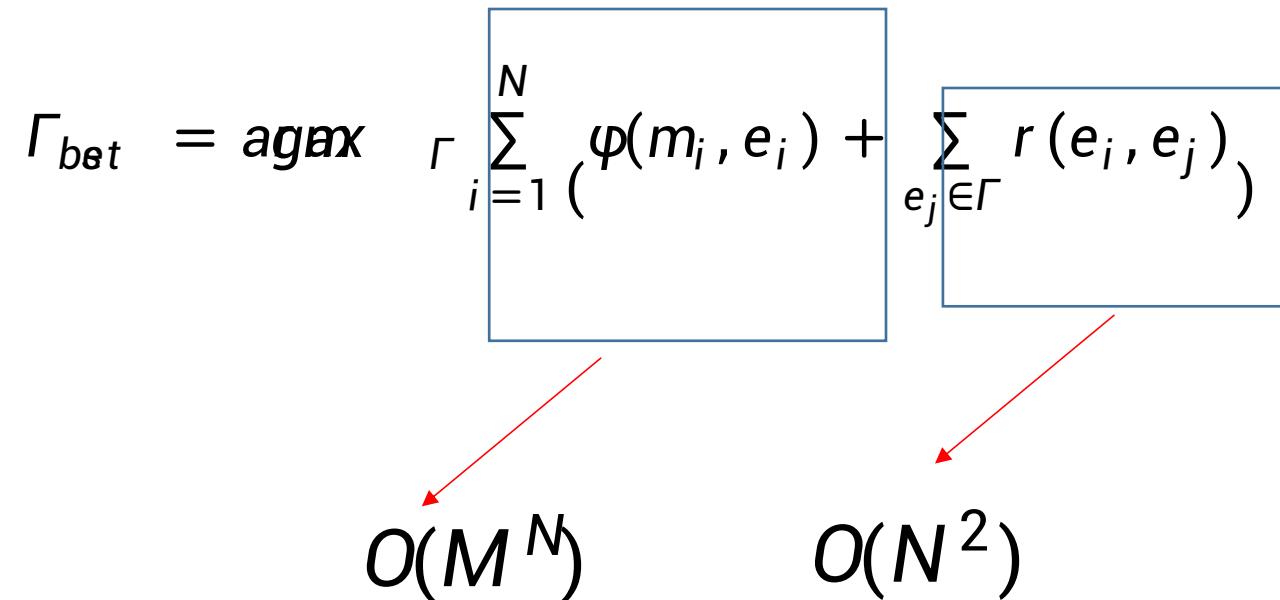


# 复杂度

- 最优实体链接问题是NP难问题
- 链接方案 $\Gamma$ 的搜索空间为 $O(N^2 \times M^N)$ 
  - $M$ 为每个指代的候选实体数， $N$ 为指代数量
- 近似算法
  - 局部化
  - 图算法

$$\Gamma_{best} = \arg\max_{\Gamma} \left( \sum_{i=1}^N \varphi(m_i, e_i) + \sum_{e_j \in \Gamma} r(e_i, e_j) \right)$$

$O(M^N)$        $O(N^2)$



# 局部近似计算

- 原目标函数

$$\Gamma_{best} = \operatorname{argmax}_{\Gamma} \sum_{i=1}^N (\varphi(m_i, e_i) + \sum_{e_j \in \Gamma} r(e_i, e_j))$$

- 先只利用局部函数 $\varphi$ 算出一个局部最优方案

$$\Gamma_{local} = \operatorname{argmax}_{\Gamma} \sum_{i=1}^N \varphi(m_i, e_i)$$

- 等价于独立地为每个 $m_i$  匹配最优的实体:  $\operatorname{argm}_{e_i} \varphi(m_i, e_i)$
- 时间复杂度:  $O(NM)$
- 完全忽略候选实体之间的相容程度, 存在较多的信息损失

# 改进1

- 根据局部链接方案  $\Gamma_{local}$ ，重新计算指代的最优实体

$$\Gamma_{best} \approx \operatorname{argmax}_{\Gamma} \sum_{i=1}^N (\varphi(m_i, e_i) + \sum_{e_j \in \Gamma_{local}} r(e_i, e_j))$$

$\Gamma_{local}$  中的实体  $e_j$  与候选实体  $e_i$  的相容性

- Milne 只考虑无歧义的指代，构造相应的局部链接方案  $\Gamma_{local}$
- 复杂度： $O(N^2M)$

李娜（歌手）	青藏高原（地形）	嫂子颂（歌曲）
李娜（歌手）	青藏高原（歌曲）	嫂子颂（歌曲）
李娜（运动员）	青藏高原（地形）	嫂子颂（歌曲）
李娜（运动员）	青藏高原（歌曲）	嫂子颂（歌曲）
李娜（教授）	青藏高原（地形）	嫂子颂（歌曲）
李娜（教授）	青藏高原（歌曲）	嫂子颂（歌曲）

全局实体消歧时间复杂度为  $O(M^N)$



李娜的代表作有青藏高原和嫂子颂

李娜（歌手）	嫂子颂（歌曲）
李娜（运动员）	嫂子颂（歌曲）
李娜（教授）	嫂子颂（歌曲）

当对“李娜”进行消歧时，不考虑歧义指代“青藏高原”，可节省枚举空间，其时间复杂度为  $O(MN)$

图 12-2 全局特征局部化示例

# 改进2

- 上述方法问题
  - 只利用了每个上下文指代的**局部最优候选实体**帮助其他指代的全局消歧
  - 一旦局部最优候选实体识别有误，就会影响全局最优链接的质量
- Tagme考虑了每个上下文指代的**所有候选实体**对链接方案的影响
$$r'(e_i, m_j) \propto \sum_{e_k \in \text{cand}(m_j)} (\varphi(m_j, e_k) \times r(e_i, e_k))$$
  - 其中 $\text{cand}(m_j)$ 为 $m_j$ 的所有可能候选实体集合
  - $r'(e_i, m_j)$ 可以视作对全局评分 $r(e_i, e_j)$ 的一种改进，最终求解公式为

$$\Gamma_{best} \approx \arg\max \left( \sum_{i=1}^N (\varphi(m_i, e_i) + \sum_{j=1}^N r'(e_i, m_j)) \right)$$

# 图算法

- Mention—Entity Graph

- 点：指代与实体
  - 边

- 指代—实体边权：局部分数  $\varphi(m, e_i)$

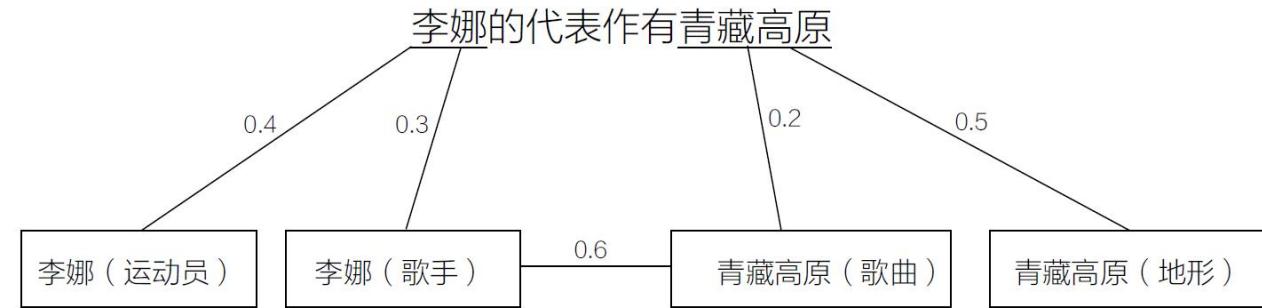
- 实体—实体边权：实体相关度  $r(e_i, e_j)$

- 问题模型

- Given a mention—entity graph, find a *subgraph* with maximal accumulative weight

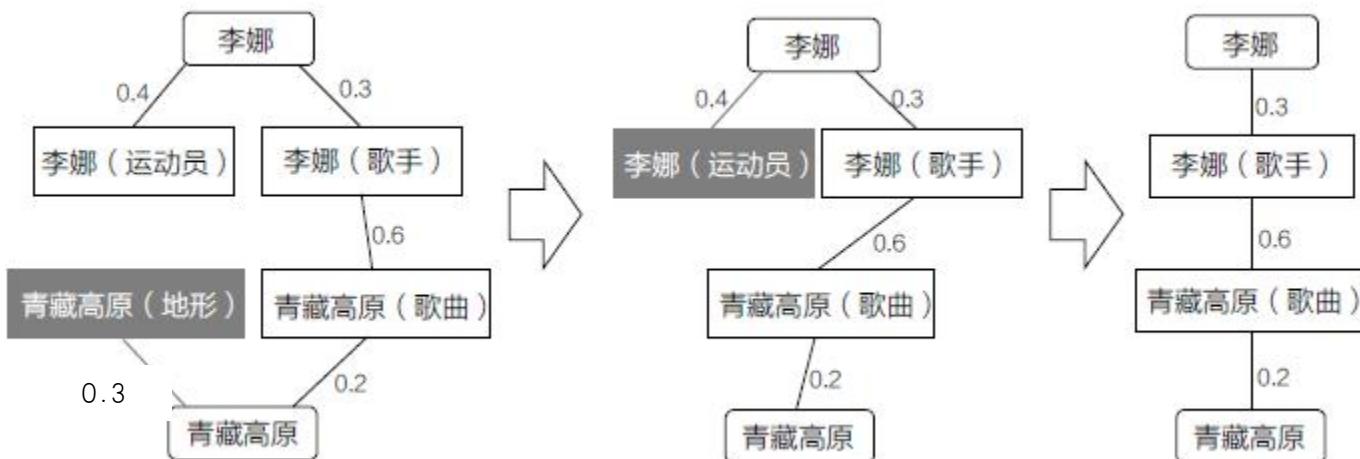
- Such that contain all mention nodes and exactly one mention—entity edge for each mention

- NP难(Steiner—tree problem )：贪心算法



# 图修剪算法

- 输入：实体与指代的加权图
- 输出：每个指代只有一条边的子图
- 算法
  - Step1、筛选与任何指代都不甚相关的实体
  - Step2、贪心删除“不合群”节点



## 算法流程

- 对于每个实体，计算它们距离所有指代节点的距离
- 保留最近的( $5 \times$ 指代数量)个实体，舍弃其他的
- 当图中存在非唯一实体时
  - 删除临接边权和最小的非唯一实体以及相邻边
  - 如果删掉的是加权度最小的点，则将当前剩下的图加入解的集合
- 在最后解的集合里暴力枚举搜索最优的方案

# 短文本实体链接一问题

---

- 应用场景
  - 搜索引擎上的查询短语
  - 广告关键词
  - 影视作品的字幕
  - ...
- 特点：上下文稀缺、噪音多、不规范
  - E.g. 冰与火之歌有多少卷，冰与火之歌有多少集
- 问题：难以关联上下文词语和实体
  - 卷->书籍；集->电视

# 短文本实体链接—方法

- 利用概念作为主题
  - 计算上下文与实体的主题凝聚度

$$\text{sim}_c(m, t) = \cos(\mathbf{v}_c(m), \mathbf{v}_c(t))$$

- 实体的主题向量  $\mathbf{v}_c(t)$  根据实体的概念直接获得
- 上下文的主题向量  $\mathbf{v}_c(m)$  可通过加权聚合每个词的主题向量获得

$$\mathbf{v}_c(m) = \sum_{w \in \text{context}} (m) \frac{\mathbf{v}_c(w)}{D(w, m)}$$

词语与指代的距离

- 每个词的概念向量可从实体的描述性文本计算

$$r(w, c) = \sum_{t \in E} \frac{n(w|t) \cdot r(t, c)}{\sum_{c'} r(t, c')}$$

实体t与概念c的关系权重

词语w出现在实体t的描述文本中的次数

冰与火之歌有多少卷

$$\begin{aligned}\mathbf{v}_c(w: \text{有}) &= \{\text{文学: 0.01, 影视: 0.02, 音乐: 0.01}\} \\ \mathbf{v}_c(w: \text{多少}) &= \{\text{文学: 0.02, 影视: 0.01, 音乐: 0.01}\} \\ \mathbf{v}_c(w: \text{卷}) &= \{\text{文学: 0.2, 影视: 0.05, 音乐: 0.01}\}\end{aligned}$$



$$\begin{aligned}\mathbf{v}_c(m: \text{冰与火之歌}) &= \{ \\ \text{文学: } 0.01/1 + 0.02/2 + 0.2/3 &= \mathbf{0.09}, \\ \text{影视: } 0.02/1 + 0.01/2 + 0.05/3 &= 0.04, \\ \text{音乐: } 0.01/1 + 0.01/2 + 0.01/3 &= 0.02\}\end{aligned}$$

# 跨语言实体链接—问题分类

- 任务：将源语言文本链接到目标语言知识库的实体上
- 可以帮助完善小语种知识库

• 方法分类 第一类方法 唱歌的李娜 ————— 机器翻译 ————— Li Na who sings ————— 英文实体链接方法 ————— Li Na(singer)

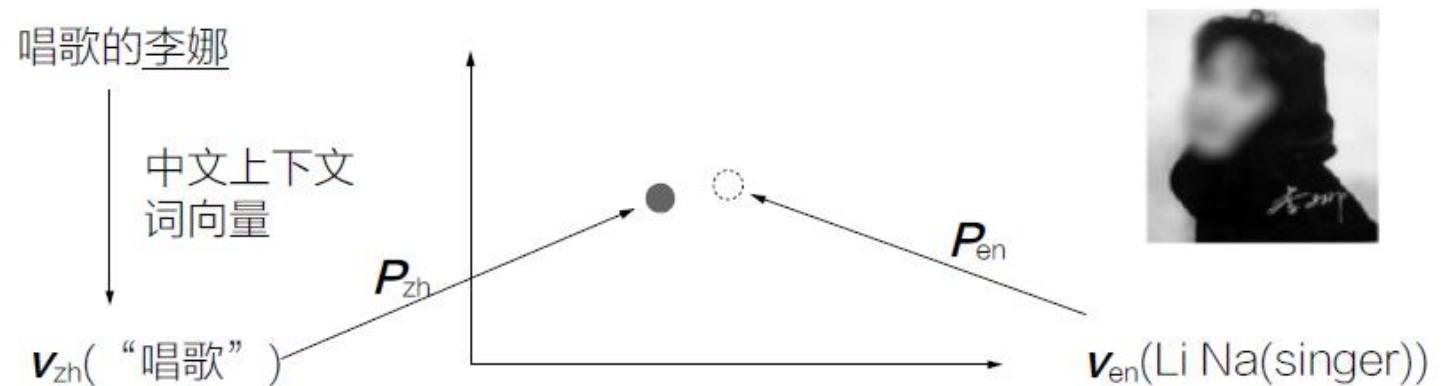
第二类方法 唱歌的李娜 ————— 直接链接 ————— Li Na(singer)

第三类方法 唱歌的李娜 ————— 中文实体链接方法 ————— 李娜（歌手） ————— 维基百科 ————— Li Na(singer)

- 第一类和第三类可以利用现有的模型解决，着重介绍第二类

# 跨语言实体链接—解决思路

- 关键：计算上下文中的中文词语和英文实体的相关度
  - 通过维基百科锚文本训练词向量与实体向量的统一表示
  - 利用典型相关分析（Canonical Correlation Analysis）
  - 计算不同语言转换矩阵 $P_{zh}$ 和 $P_{en}$
  - 将中文词语和英文实体映射到同一空间

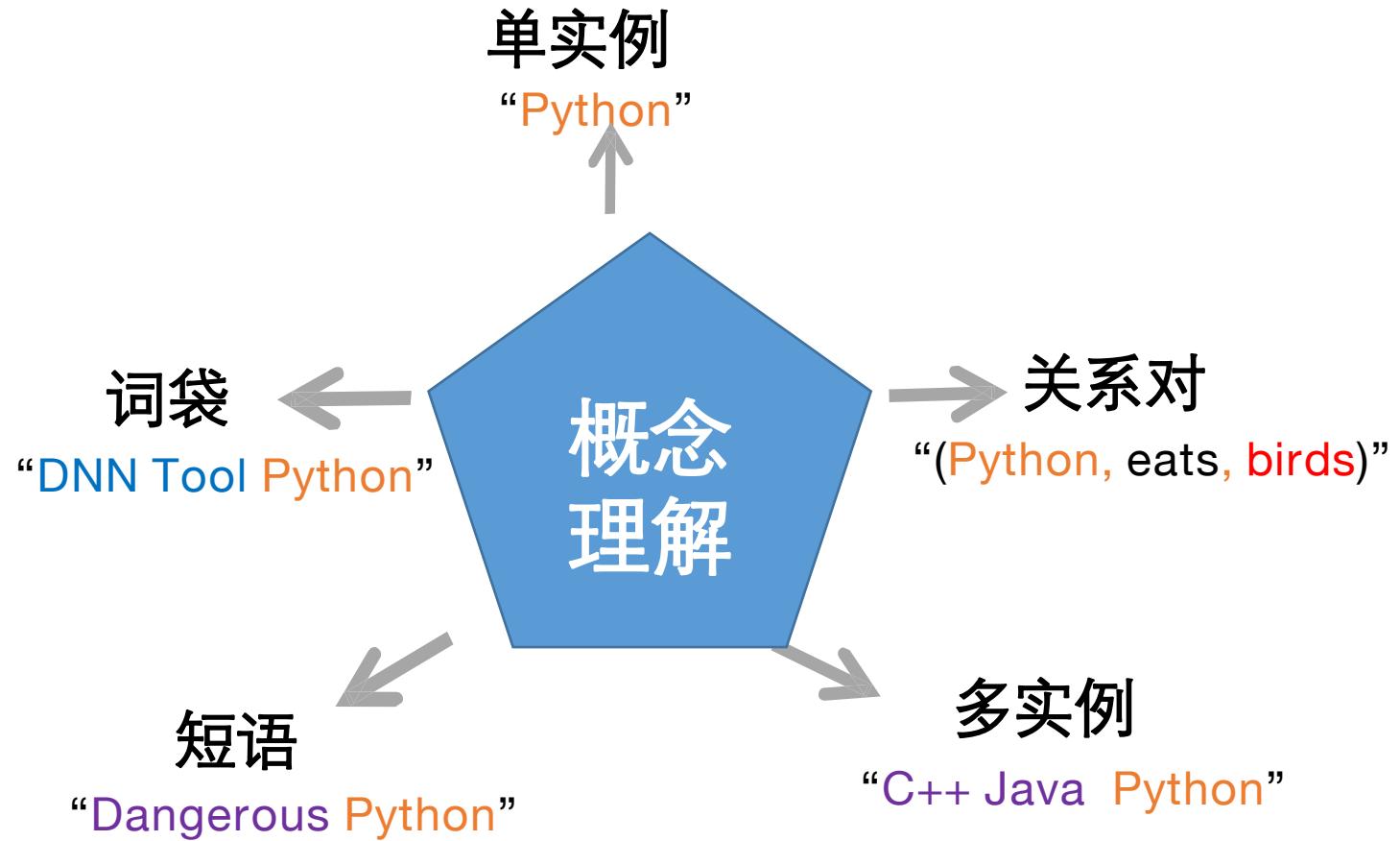


# 概念理解

---

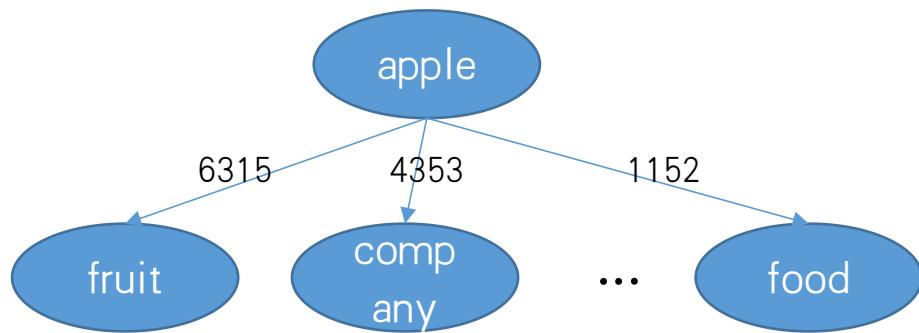
# 概念理解

- 概念理解
  - 对于特定形式的输入产生相应的概念
  - 输入可以是单实例、多实例(或者词袋)、短文本以及句子等
- isA 关系对概念理解起着至关重要的作用

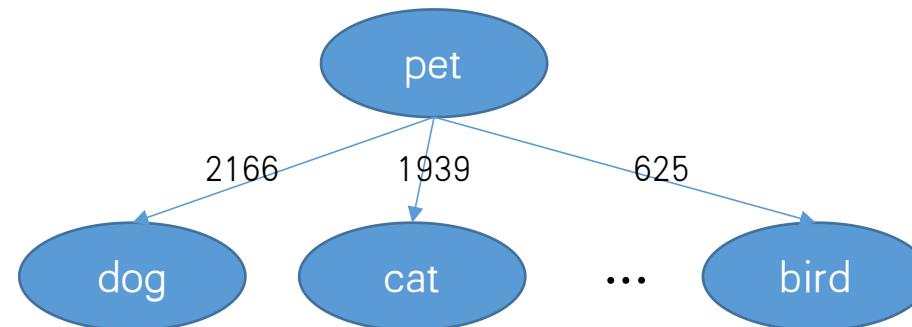


# 单实例概念理解

- 输入：一个实例
- 前提条件：每个isA关系都有在语料库中统计的频率信息
- 典型性
  - 给定一个实体（或概念）的情况下，人们有多大的可能想到某个概念（或实体）



- $p(c|e) = \frac{n(e,c)}{\sum_{c_i} n(e, c_i)}$
- $p(fruit | apple) = \frac{n(apple, fruit)}{n(apple, fruit) + \dots + n(apple, food)}$



- $p(e|c) = \frac{n(e,c)}{\sum_{e_i} n(e_i, c)}$
- $p(dog | pet) = \frac{n(dog, pet)}{n(dog, pet) + \dots + n(bird, pet)}$

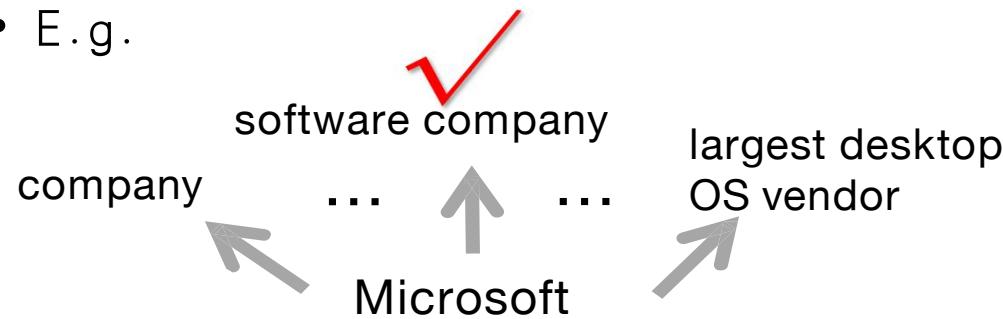
# 单实例概念理解：基本水平分类

- 基本水平分类

- 对一个实例合适层级的概念化
- 粒度既不会太粗，也不会太细

- 如何找到基本概念？

- 如果从实体能够很容易地联想到某个概念，同时从该概念也能够很容易地联想到给定的实体，那么这样的概念往往就是基本概念
- E.g.



- Formally,

- $R_{\text{Rep}}(e, c) = P(c|e) * P(e|c)$
- $bc(e) = \arg \max_c \text{Rep}(e, c)$

Category Level	Informative?	Distinctive?
Superordinate	No	Yes
Basic-level	Yes	Yes
Subordinate	Yes	No

Basic-level  
conceptualization

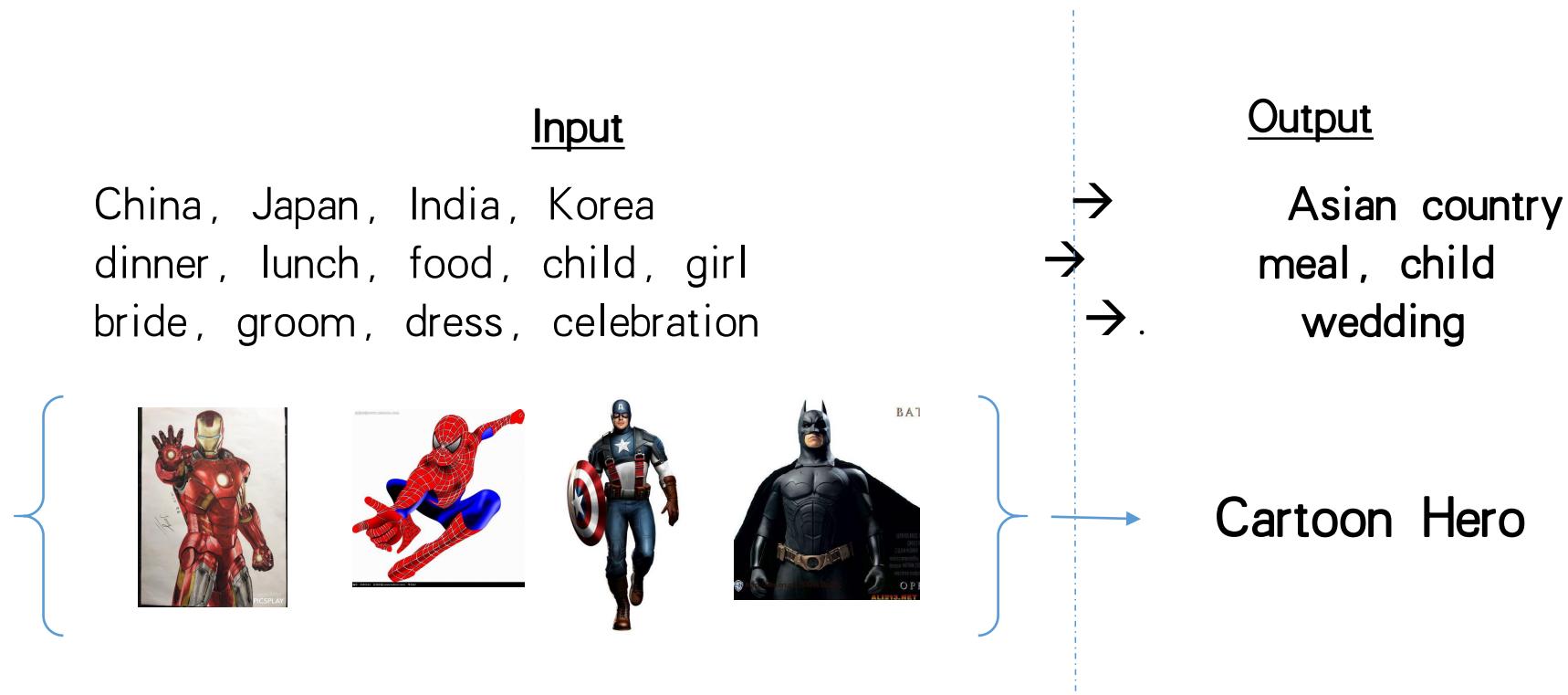
A process of finding **concept nodes** having  
**shortest expected distance** with  $e$

Given  $e$ , the  $c$  should be its  
typical concept (**shortest distance**)

Given  $c$ , the  $e$  should be its  
typical entity (**shortest distance**)

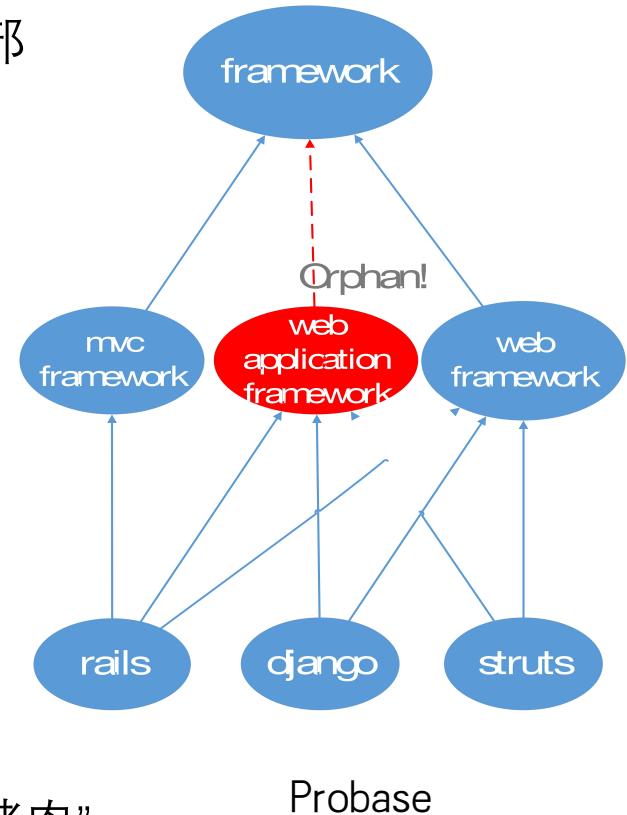
# 多实例概念理解：词袋概念理解

- 输入数据可能是多个实例，这些实例可以体现为实体，也可以是实体的词汇描述，差生相应的概念集合
- 词袋概念理解：输入一组标签或实体，使用较少的几个概念进行概括



# 多实例概念理解：词袋概念理解

- 挑战
  - 语义覆盖: 概念应该尽可能多地覆盖输入中的单词或短语，否则部分输入字词的信息将丢失
  - 最少概念: 用尽可能少的概念来概括语义
- 动机
  - 用概念图谱获取后选概念
  - 用最小描述长度 (minimal description length) 选择合适的概念
- 应用
  - 主题词标记
    - 主题词: 没有具体语义的词袋
    - 通过概念化将主题词标记为具有语义信息的概念
  - 语言理解
    - 动词角色标记，如动词“吃”的直接宾语包括“苹果”、“早餐”、“猪肉”等，可以将其概念化为一些概念，如“水果”、“餐食”、“肉”等。



# 多实例概念理解：基于MDL的概念化

- 问题建模: Given a bag of tags  $X$ , find

$$C^* = \operatorname{argmin} CL(X, C)$$

$$CL(X, C) = \underbrace{L(C)}_{\text{Minimality: 概念集合的编码长度}} + \underbrace{L(X|C)}_{\text{Coverage: 给定概念集合的条件下词袋的编码长度}}$$

Minimality: 概念集合的编码长度

Coverage: 给定概念集合的条件下词袋的编码长度

# 多实例概念理解：其他问题

- 噪声容忍

- Example: {apple, banana, breakfast, dinner, pork, beef, **bullet**}

$$L^*(x|C) = \min \begin{cases} L(x), & \text{encode directly} \\ \log |C| + L(x|c), & \text{encode using } c \in C \end{cases}$$

- 属性信息融合

- Example: {population, president, location}, which triggers the concept country

$$P(c|x) = 1 - (1 - P_e(c|x))(1 - P_a(c|x))$$

# 短语概念理解

- 输入：一段包含一个或多个实例的短语

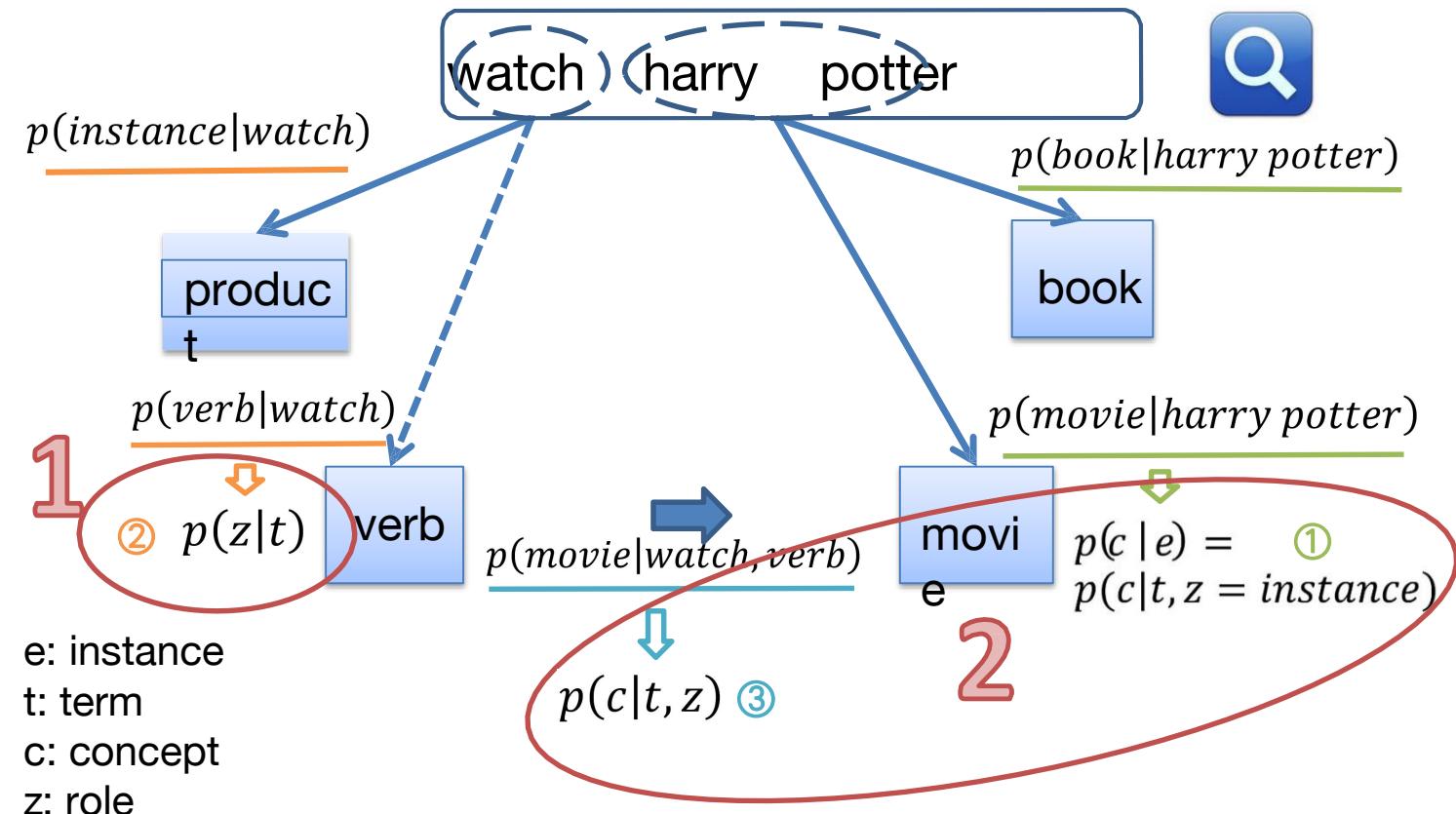
- E.g. *watch Harry Potter (movie)*

- 挑战：

- 复杂性：短语中往往伴随着形容词与动词等修饰词
- 语义稀疏：上下文稀缺

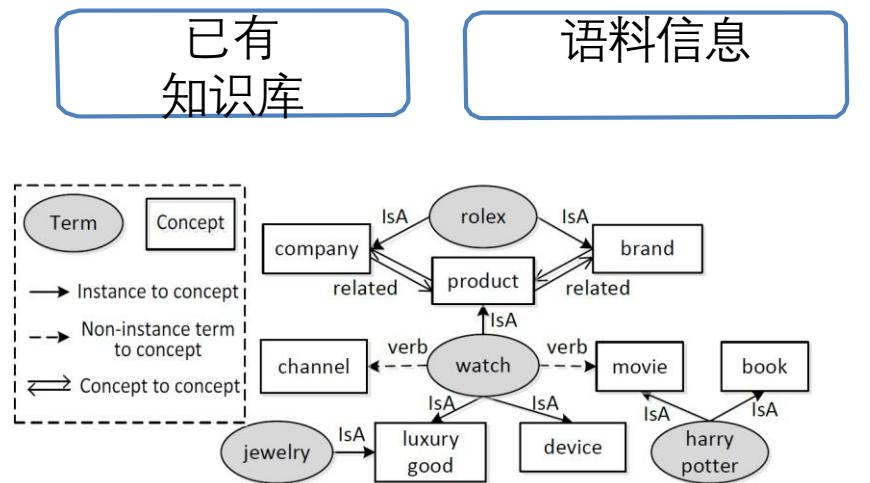
- 解决方案：语义网络

- 建立实例、动词和形容词等两者之间的关系

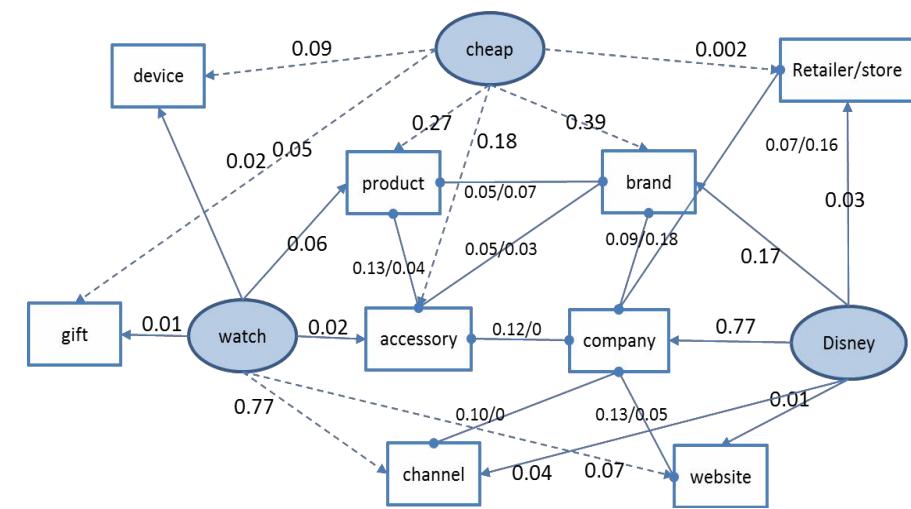


# 短语概念理解—语义网络

- 构造语义网络，将短语概念化转换为最大概率路径计算与选择问题



离线构建的语义网络



在线子图的随机游走 [Sun et al., 2005]

# 关系对概念理解

- 输入：一对有关系的实例，i.e.,  $(s, p, o)$ 
  - E.g. (Jordan, PlaysIn, Bull)
- 挑战
  1. Jordan 存在很多概念  $(c_s)$ , 如: NBA player, sport brand, etc.
  2. Bull 存在很多概念  $(c_o)$ , 如: animal, sports team, etc.
  3. “Jordan”和“Bull”的概念化任务之间是相互影响的
    - 如: When Bull isA basketball team, then Jordan isA NBA player.
- 解决方案
  - $P(c_o|o) \propto P_T(c_o|o)P_E(c_o|c_s)P_T(c_s|s)$ 
    - $P_T(c_*|*)$  是基于Probbase的实例\*的概念估计
    - $P(c_s|s)$  反之亦然.
  - $P_E(c_o|c_s)$  估计: 
$$P_E(c_o|c_s) = \frac{n(c_s, c_o)}{n(c_s)}$$

# 概念理解应用举例—电商搜索

- 场景：在淘宝平台搜索“popular smart cover iPhone X”
  - 返回 iPhones or smart covers?
  - 搜索意图：smart cover for iPhone, not iPhone
- 关键点：
  1. 理解 *cover* isA *accessory*, *iPhone* isA *device*.
  2. 当 *device* and an *accessory* 同时存在时，*accessory* 是核心词，*device* 是修饰词。
- 解决方案
  1. 从语料中挖掘(*coep t* [hed], *coep t* [mdfer], *soe* )
    - 其中，score表示两者共同出现在一个短文本中的可能性
    - 首先利用基于介词的文法规则识别核心词—修饰词之间的修饰关系，如 {*hed* [fo |of|wt h|i n|on|at] *mdfer* }。
  2. 得到 (aceoy [hed], deice [mdfer], 90%)

# 属性理解

---

# 理解概念／属性



单身汉是？



一个没结婚的男人

{类型=人，性别=男性，婚姻状况=未婚}

# 问题陈述

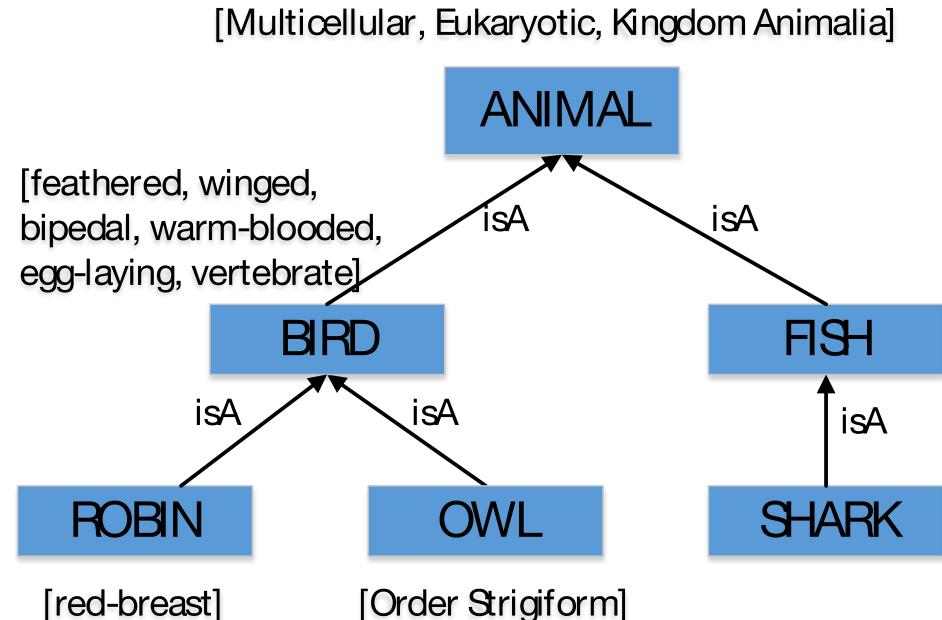
- 输入：一个类别
- 输出：一组定义性特征(Defining Feature)

- **类别** 和一组属性-值对

- 例如，Category “*Jay Chou albums*” is defined by type *album*, and Property-Value feature (*Singer*, *Jay Chou*).
    - 类别“films directed by Christopher Nolan”的定义性特征集合表示形式如下表

PV 特征	(director, Christopher Nolan)
类别特征	(type, Film)

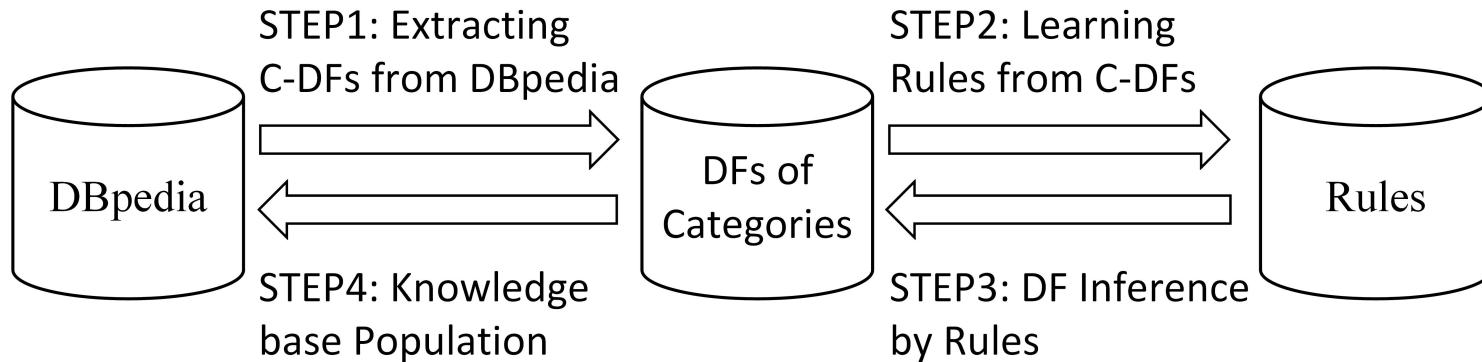
- 一个类别的**定义性特征**刻画了实体属于该类别的充分必要条件



Applications: build Semantic memory for machines

# Defining Feature 挖掘

- 框架



- Bootstrapping方法

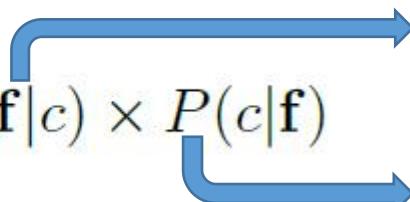
- Step 1: 使用打分函数找到某些类别的DF
- Step 2 & 3: 使用基于规则的方法获取更多类别的DF
- Step 4: 通过使用到目前为止发现的类别的DF来填充DBpedia

# Defining Feature挖掘—统计方法

- 目标函数

$$\hat{f} = \arg \max_{\vec{f} \subseteq F_c} \text{score}(c, f)$$

- 通过一个打分函数来估计一组特征集合 $f$ 是某个类别 $c$ 的DF的分值：

$$score(c, f) = P(f|c) \times P(c|f)$$

$$P(f|c) = \frac{\# \text{ of entities in } c \text{ that have } f}{\# \text{ of entities in } c}$$
$$P(c|f) = \frac{\# \text{ of entities in } c \text{ that have } f}{\# \text{ of entities that have } f}$$

- 挑战

- 枚举每一个特征子集具有**指数级复杂度**
- **频繁项集挖掘**进行事前剪枝

# Defining Feature挖掘—文本规则

## 统计+规则

- 利用统计方法解决样本丰富情况下的问题
- 再利用规则方法解决样本稀疏情况下问题

C	DFs
Films directed by Christopher Nolan	$\{(Type, Film), (director, Christopher Nolan)\}$
Films directed by James Cameron	$\{(Type, Film), (director, James Cameron)\}$
Films directed by Steven Spielberg	$\{(Type, Film), (director, Steven Spielberg)\}$
Films directed by David Fincher	$\{(Type, Film), (director, David Fincher)\}$
Films directed by Ben Affleck	$\{(Type, Film), (director, Ben Affleck)\}$

↓

C Pattern	DFs Pattern
Films directed by $(\cdot)$ *	$\{(Type, Film), (director, (\cdot))\}$

↓

Person

Left part of rule	Right part of rule
Films directed by $\langle Person \rangle$	$\{(Type, Film), (director, \langle Person \rangle)\}$

# References

---

- Ratinov, Lev—Arie et al. “Local and Global Algorithms for Disambiguation to Wikipedia.” ACL (2011).
- Hoffart, Johannes et al. “Robust Disambiguation of Named Entities in Text.” *EMNLP*(2011).
- Ganea, Octavian—Eugen and Thomas Hofmann. “Deep Joint Entity Disambiguation with Local Neural Attention.” *EMNLP* (2017).
- Bo Xu, Chenhao Xie, Yi Zhang, **Yanghua Xiao\***, Haixun Wang and Wei Wang, Learning Defining Features for Categories, (*IJCAI 2016*)
- Zhongyuan Wang, Haixun Wang, Jirong Wen and **Yanghua Xiao**, An Inference Approach to Basic Level of Categorization, (*CIKM 2015*)
- Xiangyan Sun, **Yanghua Xiao\***, Haixun Wang, Wei Wang, On Conceptual Labeling of a Bag of Words, (*IJCAI 2015*)
- Wanyun Cui, Xiyu Zhou, Hangyu Lin, **Yanghua Xiao\***, Seungwon Hwang,Haixun Wang and Wei Wang, Verb Pattern: A Probabilistic Semantic Representation on Verbs, (*AAAI 2016*)
- Zhongyuan Wang and Haixun Wang, Understanding Short Texts, in *the Association for Computational Linguistics (ACL) (Tutorial)*, August 2016.
- Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou, Understand Short Texts by Harvesting and Analyzing Semantic Knowledge, in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Volume: PP, Issue: 99, May 23, 2016.
- Zhiyi Luo, Yuchen Sha, Kenny Zhu, Seung—Won Hwang, and Zhongyuan Wang, Commonsense Causal Reasoning between Short Texts, in *the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, April 2016.
- Peipei Li, Haixun Wang, Kenny Q Zhu, Zhongyuan Wang, Xuegang Hu, and Xindong Wu, A Large Probabilistic Semantic Network Based Approach to Compute Term Similarity, in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Volume: 27, Issue: 10, October 1 2015.

# References

---

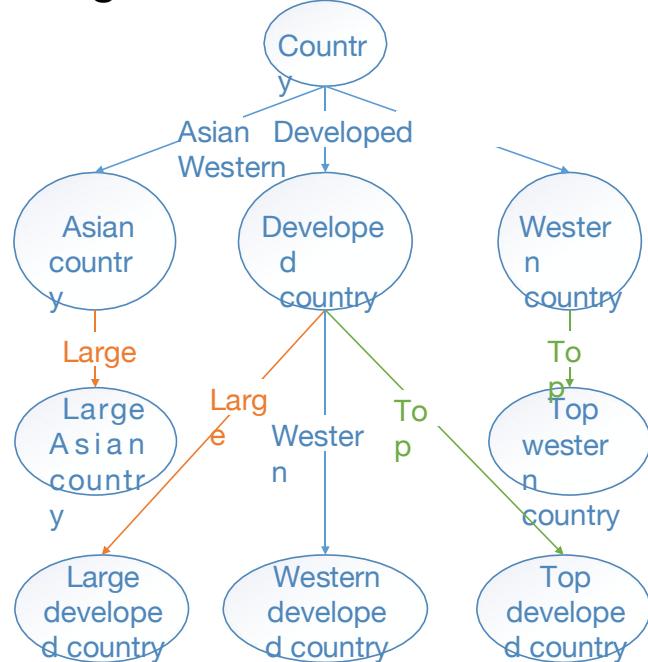
- Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao, [An Inference Approach to Basic Level of Categorization](#), in *ACM International Conference on Information and Knowledge Management (CIKM)*, ACM – Association for Computing Machinery, October 2015.
- Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan, and Zheng Chen, [Contextual Text Understanding in Distributional Semantic Space](#), in *ACM International Conference on Information and Knowledge Management (CIKM)*, ACM – Association for Computing Machinery, October 2015.
- Zhongyuan Wang, Fang Wang, Ji-Rong Wen, and Zhoujun Li, [Bring User Interest to Related Entity Recommendation](#), in *the 4th IJCAI International Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR 2015)*, July 2015.
- Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen, [Query Understanding through Knowledge-Based Conceptualization](#), in *IJCAI*, July 2015.
- Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou, [Short Text Understanding Through Lexical–Semantic Analysis](#), in *International Conference on Data Engineering (ICDE)*, April 2015. Best Paper Award
- Fang Wang, Zhongyuan Wang, Senzhang Wang, and Zhoujun Li, [Exploiting Description Knowledge for Keyphrase Extraction](#), in *PRICAI*, December 2014.
- Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji-Rong Wen, [Concept-based Short Text Classification and Ranking](#), in *ACM International Conference on Information and Knowledge Management (CIKM)*, ACM – Association for Computing Machinery, October 2014.
- Zhongyuan Wang, Haixun Wang, and Zhirui Hu, [Head, Modifier, and Constraint Detection in Short Texts](#), in *International Conference on Data Engineering (ICDE)*, 2014.
- Kai Zeng, Jiacheng Yang, Haixun Wang, Bin Shao, and Zhongyuan Wang, [A Distributed Graph Engine for Web Scale RDF Data](#), in *PVLDB*, August 2013.
- Taesung Lee, Zhongyuan Wang, Haixun Wang, and Seung-won Hwang, [Attribute Extraction and Scoring: A Probabilistic Approach](#), in *International Conference on Data Engineering (ICDE)*, , 2013.
- Peipei Li, Haixun Wang, Kenny Q. Zhu, Zhongyuan Wang, and Xindong Wu, [Computing Term Similarity by Large Probabilistic isA Knowledge](#), in *ACM International Conference on Information and Knowledge Management (CIKM)*, 2013.

# 多实例概念理解

问题：

短文本中的中心词、修饰词与约束检测

Model1: Non-Constraint Modifiers Mining: Construct Modifier Networks



“Large” and “Top” are pure modifiers

Example: Popular<sub>[modifier]</sub> smart cover<sub>[head]</sub> iphone 5s<sub>[constraint]</sub>

Model2: Head—Constraints Mining: Acquiring Concept Patterns

