# 知识图谱基础

赵 军（jzhao@nlpr.ia.ac.cn）

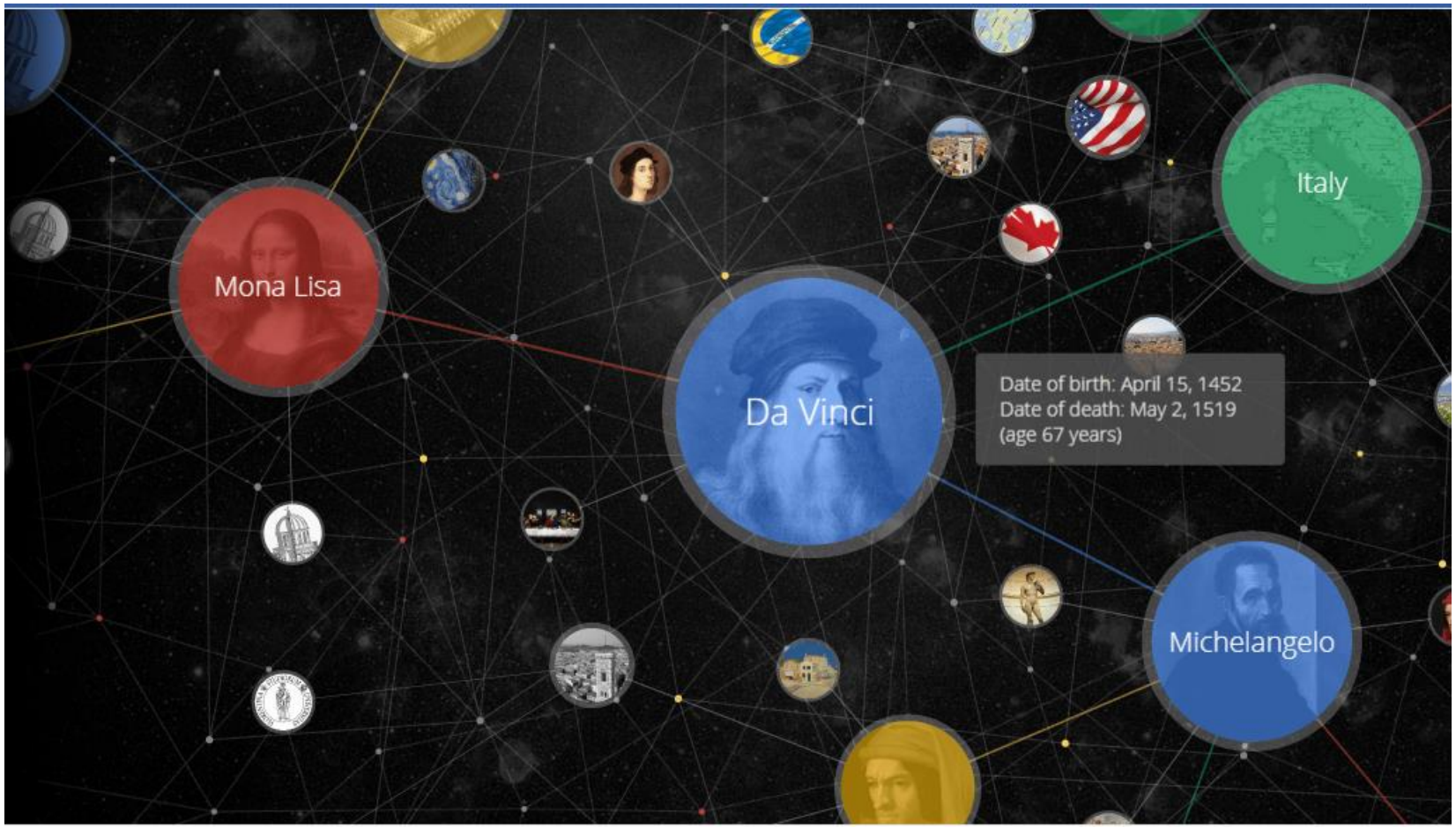
中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

# 提纲

- 引言
- 知识图谱发展历史
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用
- 课程介绍

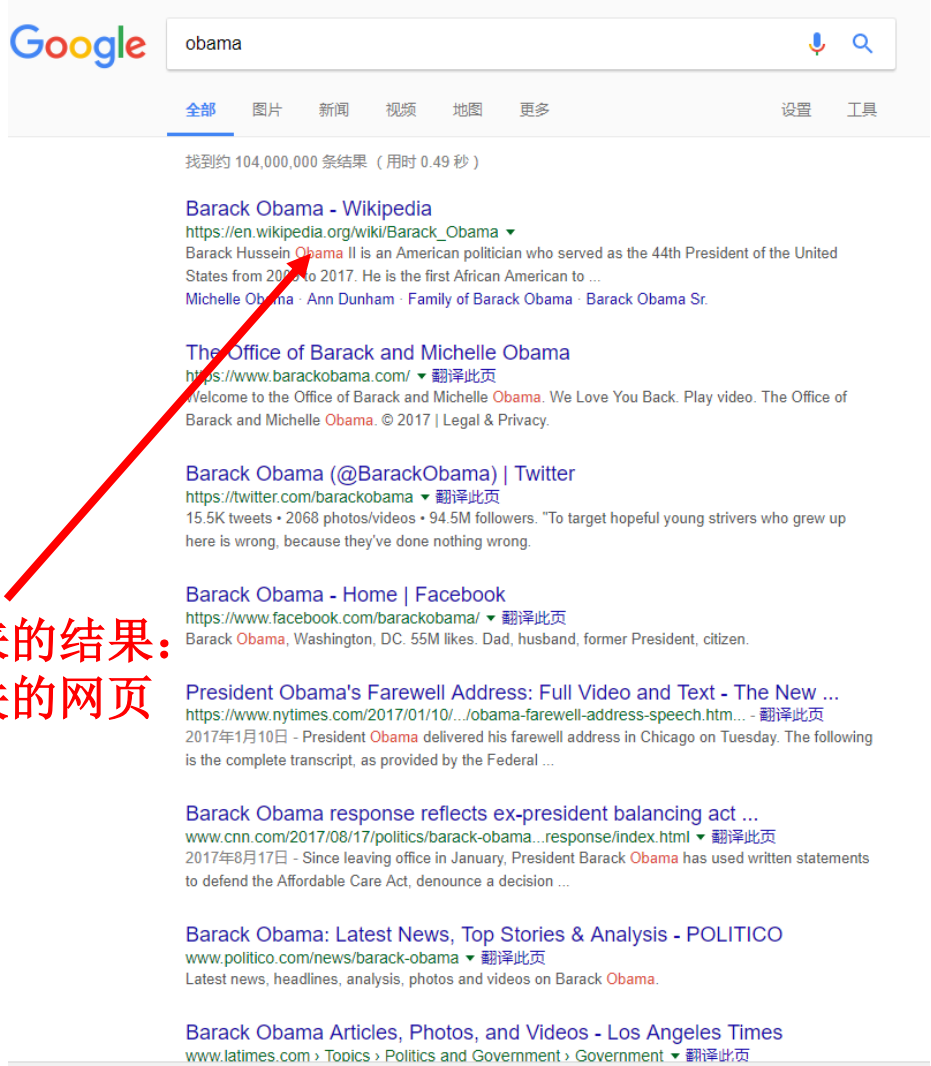
# Google Knowledge Graph




Video：Introducing the Knowledge Graph

Video：Explore lists and collections with Google search

# KG对搜索的改变




原来的结果：
相关的网页

现在的结果：
精准的知识

# 分类浏览

行业百科 hangye.baike.com 三百六十行 行行有百科

请输入搜索内容 搜索

## 百科分类

**生活**
美食 | 宠物 | 家庭 | 购物 | 服饰 | 家居 | 情感
出行 | 美容 | 日用品 | 手工 | 鲜花礼品

**旅游**
世界名城 | 中国城市 | 中国古迹 | 中国旅游景点
名山名河 | 户外 | 自助游 | 主题游

**教育**
高校 | 幼儿教育 | 中小学教育 | 技能培训
校园 | 网络教育 | 资格认证 | 高等教育 | 考试
职场学习 | 留学

**体育**
球类运动 | 休闲运动 | 比赛 | 俱乐部 | 彩票
武术 | 养生运动 | 体育器材 | 智力体育

**健康**
疾病 | 两性 | 整形 | 心理 | 保健 | 饮食健康
母婴 | 养生 | 药品 | 中老年 | 中医 | 美容
医疗器械

**文化**
历史 | 文学 | 宗教 | 民族 | 民俗 | 戏曲 | 工艺
语言 | 文物古迹 | 国学 | 文化遗产 | 绘画
书法 | 雕塑 | 设计 | 视觉 | 建筑 | 综艺
艺术鉴定 | 拍卖收藏 | 科普

**娱乐**
电影 | 电视剧 | 音乐 | 舞蹈 | 综艺节目 | 演出
明星

**社会**
政治 | 军事 | 法律 | 社会问题 | 战争
灾难事故 | 民生 | 时事评论 | 公益慈善
社会现象 | 服务 | 企业 | 福利保障

**游戏/动漫**
电子游戏 | 电脑游戏 | 掌机游戏 | 手机游戏
网络游戏 | 桌面游戏 | 卡牌游戏 | 网页游戏
运动游戏 | 语言游戏 | 益智游戏 | 漫画 | 动画

**财经**
股票 | 房产 | 投资 | 理财 | 汽车 | 基金 | 贷款
外汇 | 财税 | 保险 | 经管 | 银行 | 债券

**科技**
通信 | IT | 技术 | 数码 | 家电 | 手机 | 互联网
软件 | 硬件 | 能源 | 电子商务 | 科学探索

**工业技术**
原材料 | 电子电工 | 通信技术 | 建筑建材
五金 | 机械 | 石油化工 | 仪器仪表 | 安全防护

**农业种植**
蔬菜 | 水果 | 园林花卉 | 农药肥料 | 水产
畜禽 | 饲料兽药

**商务服务**
翻译 | 物流 | 中介 | 酒店餐饮 | 印刷包装
广告公关 | 展览 | 维修

## 电影

**SeediqBale百科**
《赛德克·巴莱太阳旗》和《赛德克·巴莱彩虹桥》两部分。魏德圣导演重新剪辑的版本。影片讲述的是1930年，台

赛德克·巴莱业内人士好评为何如潮？
骄傲才是生命，我们就要民族骄傲性！！
赛德克·巴莱为何又叫座又叫卖？

**韩国影视百科**
韩剧影视资讯介绍，逐步完善中。

杜珂
《加油金先生》
当下韩剧也玩穿越 细数韩古装剧三大特点

**香港电影金像奖百科**
香港电影金像奖标志香港电影金像奖于1982年由《电影双周刊》创办。这是从《电影双周刊》每年邀请影评人评选

是变态还是复仇？新闻与【微电影】引人深思
为什么说金像奖没有争议有些遗憾？
金像奖影后叶德娴还有哪些影视作品？

**黄金大劫案百科**
《黄金大劫案》是由李健，王文，黄渤主演的喜剧动作片。讲述了一个出身贫寒的小混混无意中卷进了一个地下党策

《黄金大劫案》里东北方言会成亮点吗？
大劫案，不想当英雄的混混不是情圣吗？
说说《黄金大劫案》的观后感

**新少林寺百科**
《新少林寺》由香港动作片导演陈木胜执导，汇聚了刘德华、谢霆锋、成龙、范冰冰等两岸三地实力战将，影片总投

你了解少林武功吗？
成龙为何在《新少林寺》中改变戏路？
谢霆锋在《新少林寺》中是大反派吗？

**匹夫百科**
《匹夫》是由杨树鹏执导，黄晓明、张译、张歆艺等主演的热血匪帮传奇电影，于2012年4月24日上映。匹夫》都

《匹夫》黄晓明为什么耍大牌？
《匹夫》黄晓明为什么拍激情？
《匹夫》张歆艺腿青手肿怎么不叫苦？

**新少林寺**
《新少林寺》是由中国大陆和香港英皇电影公司联合制片出品，在获得少林寺正式授权与大力支持下，由陈木胜执导

可以到少林寺体验禅修吗？
范冰冰是《新少林寺》的出品人吗？
新建的少林寺真的被剧组烧掉了吗？

**电影百科**
汇集包括十大经典电影在内的电影影评和电影观后感，解析科幻电影悬疑电影的前台幕后，罗列电影明星生活趣闻和

3•15电影质检报告
《遗落战境》之丢失的世界！
霍比特人3----中土世界的告别

# 推荐

京东

**机器学习实践应用**

人工智能触手可及 让数据起舞 用算法扩展业务边界 阿里机器学习专家力作 实战经验分享

李博 著

京 东 价：**￥57.80** [8.4折] [定价：~~￥69.00~~] (降价通知)

累计评价 900+

促销信息：以下促销可在购物车任选其一

满减 每满100元，可减30元现金 详情 >>

加价购 满10元另加16.90元，或满20元另加9.90元，或满30元另加6.90元，即可在购物车换购热销商品 详情 >>

优 惠 券：满105减6 满200减50 满200减16 更多>>

增值业务：礼品包装

配 送 至：北京海淀区三环到四环之间 有货，支持 99元免基础运费 | 货到付款

服 务：由 京东 发货，并提供售后服务。23:00前下单,预计明天(09月08日)送达

人气单品

机器学习实战

￥48.70

Python机器学习算法

￥57.80

统计学习方法

￥27.20

机器学习【首届京东文学奖-年度新锐入围作

￥77.00

终极算法：机器学习和人工智能如何重塑世界

￥44.20

TensorFlow技术解析与实战

￥66.20

# 推理

Baidu百度
刘德华老婆的女儿的爸爸 刘德华老婆的女儿的爸爸
百度一下
网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»
百度为您找到相关结果约2,420,000个
搜索工具
刘向惠父亲：
刘德华
刘德华（Andy Lau），1961年9月27日出生于中国香港，演员、歌手、作词人、制片人。1981年出演电影处女作《彩云曲》。1983年主演的武侠剧《神雕侠侣》在香港获得62点的收... 详情>>
来自百度知识图谱 | 报错
刘德华老婆的女儿的爸爸_百度图片
image.baidu.com - 查看全部7,142张图片
刘德华老婆有几个儿子 刘德华婚姻路大盘点_搜狐新闻_搜狐网
2015年10月29日 - 刘天王刘德华到底有几个孩子和老婆是大家非常关注的焦点,接下来小编将为您揭秘这个惊天大秘密。据台湾媒体报道,52岁的刘德华(华仔)有了女儿刘向蕙后...
mt.sohu.com/20151029/n... - 百度快照

相关人物
展开
朱丽倩
罗美薇
刘向蕙
刘德盛
张柏芝
关之琳
成龙
李菩蜜
相关演员
展开
郭富城
林凤娇
狄波拉
吴孟达

# 背后的知识图谱

- Freebase：4000多万实体，上万个属性关系，24多亿个事实三元组
- 百度百科：实体1000万
- 互动百科：实体800万，5万个分类，68亿文字

# 相关的知识图谱

- 语言知识图谱
  - WordNet：155, 327个单词，同义词集117,597个，同义词集之间由22种关系连接
- 事实性知识图谱
  - OpenCyc： 23.9万个实体，1.5万个关系属性，209.3万个事实三元组
  - DBpedia：400多万实体，48,293种属性关系，10亿个事实三元组
  - YAGO2 ：980万实体，超过100个属性关系， 1亿多个事实三元组

# 相关的知识图谱

- 领域知识图谱
  - Kinships：描述人物之间的亲属关系，104个实体，26种关系，10,800个三元组
  - UMLS：医学领域，描述医学概念之间的联系，135个实体，49种关系，6,800个三元组。
  - Cora：2,497个实体，7种关系，39,255个三元组
- 机器自动构建的知识图谱
  - NELL： 519万实体，306种关系， 5亿候选三元组
  - Knowledge Vault: 4500万实体，4469种关系，2.7亿三元组

# 产业界：以阿里为例

- 以商品、产品、品牌、条码、分类为核心
- 整合关联了百科、国家行业标准等9大类一级本体，包含了百亿级别的三元组，形成了巨大的知识图谱

- 应用于搜索、前端导购、平台治理、智能问答、品牌商运营等核心、创新业务

# 产业界-其它应用

- Google
- Bing
- 百度

- 微软小冰
- 公子小白

聊天机器人

- IBM Watson Health

决策支持

- Siri
- Google Now
- 微软小娜
- 百度度秘

私人助理

- Apple Watch
- Ticwatch

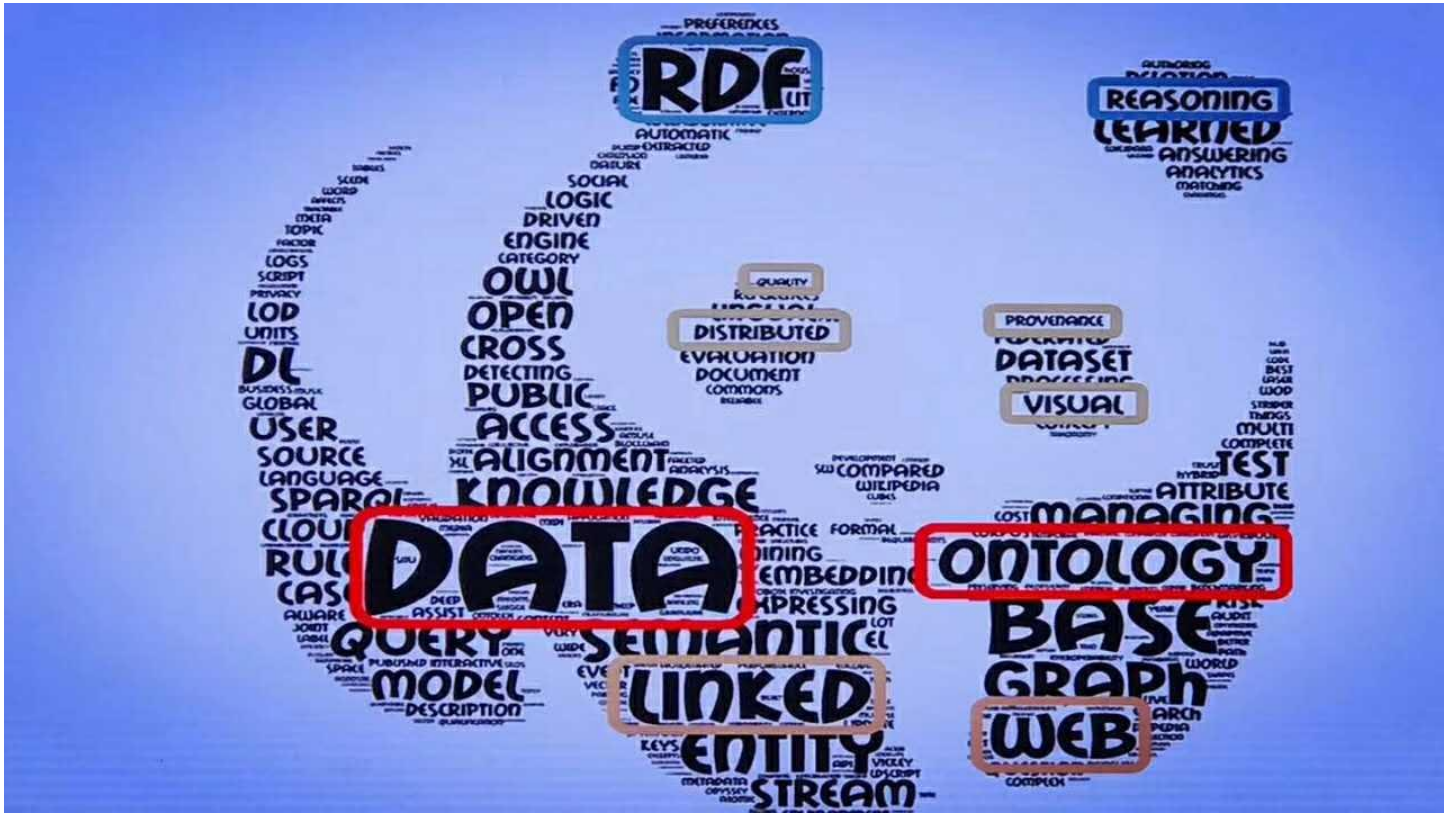
智能硬件

- 智能家居
- 智能厨房

智能硬件

# 学术界-国际

- 国际上顶级会议相关的文章和专题：云图



# 学术界-国内

## ■全国知识图谱与语义计算大会

| 2017年8月26日（知识图谱学术讲习班，西华大学） | |
|---|---|
| 上午8:45-12:20 | T1：知识图谱导论 |
| 下午2:00-5:20 | T2：知识图谱构建 |
| 晚上19:30-21:00 | T5：知识图谱虚拟化 |
| **2017年8月27日（知识图谱学术讲习班，西华大学）** | |
| 上午9:00-12:20 | T3：知识获取方法 |
| 下午2:00-5:20 | T4：知识图谱实践 |
| 晚上 | 语言与知识计算专委会工作 |

| 2017年8月28日（CCKS 2017主会，成都金牛宾馆） | |
|---|---|
| 上午8：30-12：20 | 特邀报告：Open Knowledge,Corporate Knowledge and the Analytics of Dynamic Unstructured Data（Prof. Hans Uszkoreit，德国人工智能研究中心） |
| | 特邀报告:人工智能的下一个前沿：知识计算（林钦佑 研究员，微软亚洲研究院） |
| | 知识图谱顶级会议回顾及进展报告（程龚 王泉） |
| | 知识图谱顶级会议回顾及进展报告（肖仰华 王志春） |
| 下午1：30-6：00 | 评测竞赛论文、论文Session及Poster、Demo环节 |
| 晚上 | 晚宴 |

| 2017年8月29日（CCKS 2017主会，成都金牛宾馆） | |
|---|---|
| 上午9：00-12：20 | 特邀报告：Knowledge will Propel Machine Understanding of Big Data（Amit Sheth教授，Wright State University） |
| | 特邀报告：显式知识在深度学习对话模型中的应用与挑战（赵军 研究员，中科院自动化所） |
| | 知识图谱工业界论坛Panel：知识图谱与认知智能 |
| 下午1：30-6：00 | 知识图谱工业界论坛 |
| | 闭幕式 |

| 会议 | 地点 | 时间 |
|---|---|---|
| 全国知识图谱与语义计算大会(CCKS 2017) | 成都 | 2017年8月26-29日 |
| 全国知识图谱与语义计算大会(CCKS 2016) | 北京 | 2016年9月19-22日 |
| 第三届全国中文知识图谱研讨会(KG 2015) | 宜昌 | 2015年11月9日-10日 |
| 第二届全国中文知识图谱研讨会(KG 2014) | 苏州 | 2014年10月17日 |
| 第一届全国中文知识图谱研讨会(KG 2013) | 武汉 | 2013年10月12日 |

# 课程目的

- 以知识图谱为代表的知识工程的基本问题和基本方法
- 基于知识图谱的语义计算的关键技术
- 使得学生初步具备建立通用或领域知识图谱，以及基于知识图谱进行数据分析、文本分析应用的能力

- 脉络梳理
- 兼顾学术和应用
- 教学相长

# 提纲

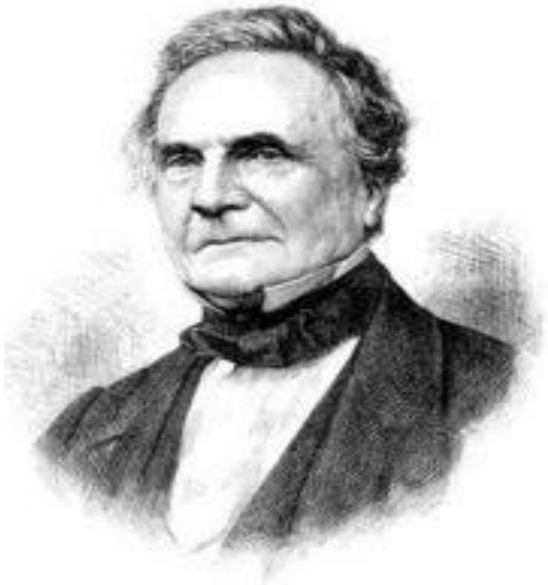
- 引言
- 知识图谱发展历史
  - 知识图谱和人工智能
  - 知识图谱和万维网
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用
- 课程介绍

# 人工智能

- 人工智能（Artificial Intelligence），也称机器智能（Machine intelligence），是计算机科学的一个分支，试图了解智能的本质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。
- 1956年，以麦卡赛、明斯基、罗切斯特和申农等为首的一批有远见卓识的年轻科学家在一起聚会，共同研究和探讨用机器模拟智能的一系列有关问题，并首次提出了“人工智能”这一术语，它标志着“人工智能”这门新兴学科的正式诞生。
- E. A. Feigenbaum《人工智能手册》（1981年）：人工智能是计算机科学的一个分支，涉及智能计算机系统的设计，该系统显示人类行为中与智能相关的特征。

# 孕育（1956年之前）

- 1821年，“现代计算之父和人工智能鼻祖”，英国数学家Babbage，发明了两台最先进的差分机和分析机，其中分析机的设计思想与现代电子计算机十分相似。他认为一台通用计算机系统应该具有5个主要组成部分：输入、存储器、运算器、控制器和输出，为电子计算机的发展奠定基础。

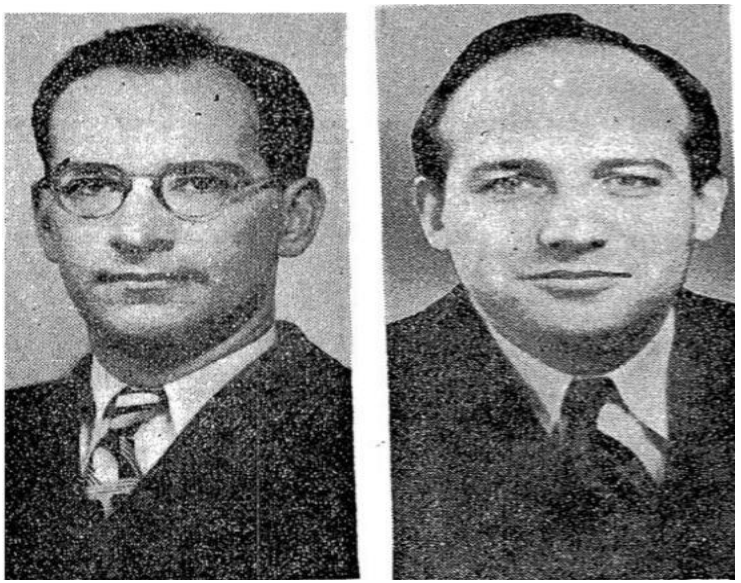


# 孕育（1956年之前）

- 1945年，普林斯顿大学教授，“计算机之父”和“博弈论之父”冯·诺依曼（John von Neumann），提出“数字计算机的数制采用二进制；计算机应该按照程序顺序执行”，——冯诺依曼理论，也称为冯诺依曼体系结构。冯诺依曼理论对计算机的设计具有决定性的影响，至今仍为电子计算机设计者所遵循。

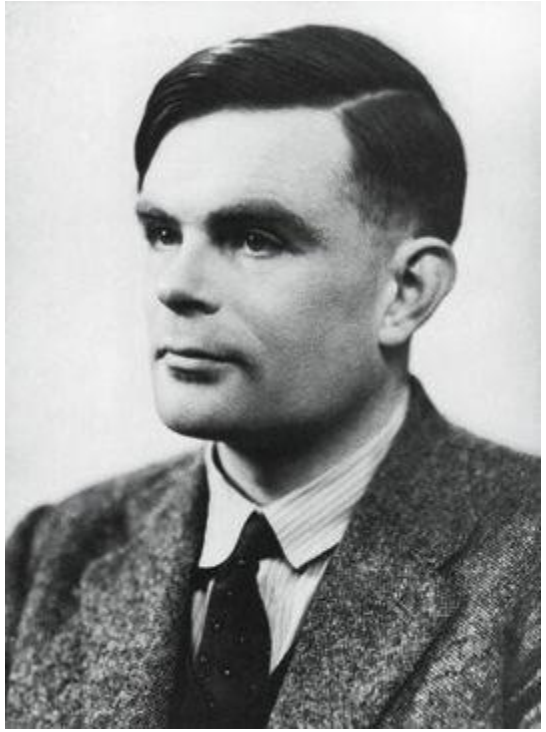
# 孕育（1956年之前）

- 1946年，宾夕法尼亚大学两位青年科学家，38 岁的物理学家、气象学家约翰·威廉·莫希利（John William Mauchly）博士和他的助理、26 岁的项目总工程师小普雷斯伯·埃克特（J. Presper Eckert Jr.）研制成功世界上第一台通用电子计算机ENIAC（Electronic Numerical Integrator And Computer）。



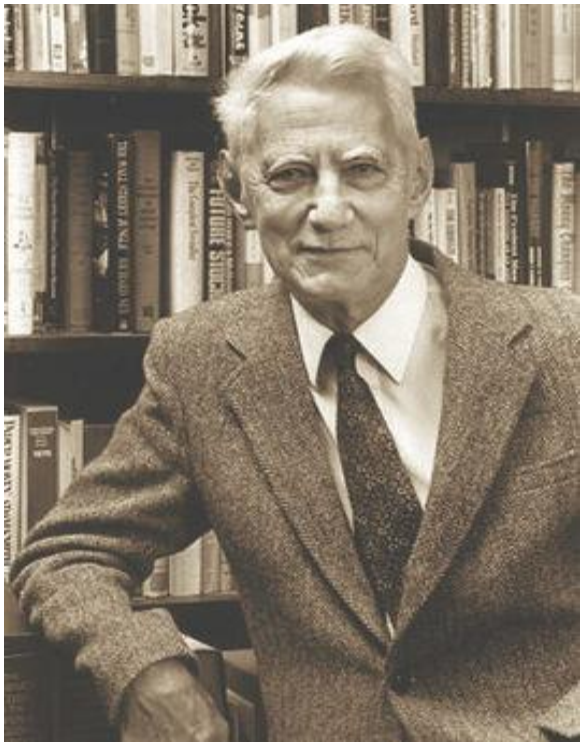
# 孕育（1956年之前）

- 1947年，英国天才数学家A. Turing发表了一篇人工智能的预言性文章“计算机和智能”，提出了计算机能否思考的问题，他肯定机器是可以思维的，并提出了著名的图灵测试，成为一种测试机器是不是具备人类智能的方法。
  - 一个人在不接触对方的情况下，通过一种特殊的方式，和对方进行一系列的问答，如果在相当长时间内，他无法根据这些问题判断对方是人还是计算机，那么，就可以认为这个计算机具有同人相当的智力，即这台计算机是能思维的。



# 孕育（1956年之前）

- 1950年美国数学家、信息论的创始人香农（Claude Elwood Shannon）在《科学美国人》发表有关计算机下棋的文章。
- 在1950年第一个估计出国际象棋的穷举复杂度大概在10^120种变化左右，具体数字被后人称为“香农数”。
  - 围棋的穷举复杂度又远远超出国际象棋，达到了惊人的10^360。
  - 作为比较，目前可观测宇宙中的原子总数据估计“只有”10^75个。



# 缘起和形成（1956-1970）

- Dartmouth Artificial Intelligence Conferences
- 1956年在达特茅斯学院召开的夏季讨论班，为期一个月
- 会议的议题：“智能的任何其他特性的每一个方面都应能被精确地加以描述，使得机器可以对其进行模拟。”
- 组织者
  - 达特茅斯学院的数学系助理教授麦卡锡（John McCarthy）
  - 哈佛大学数学与神经学初级研究员明斯基（Marvin Minsky）
  - IBM信息研究经理罗切斯特（Nathaniel Rochester）
  - 贝尔电话实验室数学家，信息论的创始人克劳德•香农（Claude Shannon）
- 标志人工智能成为一门真正的学科。达特茅斯会议之后的几年，人工智能得到蓬勃发展。

2006年，五十年后，当事人重聚达特茅斯。（左起：摩尔，麦卡锡，明斯基，赛弗里奇，所罗门诺夫）

# 缘起和形成（1956-1970）

- 解决疑难问题是人工智能早期的主要成果，它导致了问题求解技术的发展，其中启发式搜索技术取得很大的进展。
  - 1956年，纽厄尔和西蒙合作研制成功了“逻辑理论机LT”（the Logic Theory Machine），是第一个处理符号而不是数字的计算机城西，是用机器证明数学定理的最早尝试。
  - 1956年，塞缪尔研制成功“跳棋程序”，该程序在1962年击败了它的设计者，1962年击败了美国某个州的跳棋冠军。
  - 1960年，纽厄尔和西蒙又研制成功“通用问题求解程序GPS(General Problem Solving)系统”，解决不定积分、三角函数、代数方程等十几种性质不同的问题，机器定理证明成为人工智能研究的基本课题之一。
  - 1960年，麦卡锡提出并研制成功“表处理语言LISP”，不仅能处理数值，还可以处理符号。适用于符号微积分计算、数学定理证明、数理逻辑中的命题演算、博弈、图像识别等人工智能研究的多个领域，成为一种重要的人工智能程序设计语言。

# 发展和实用化（1971-1980）

- 科学家们认识到：人工智能系统要解决复杂问题，不能只有一般的问题求解方法，还需要领域**专门知识**。1970年开始，科学家们提出各种知识表示技术，搜索技术也日趋成熟。
- 有代表性的专家系统：
  - 1965年，斯坦福大学教授，人工智能鼻祖费根鲍姆和化学家C.Djerassi等研制出第一个专家系统DENDRAL，可以根据分子式及其质谱数据推断分子结构。
  - 1971年，MIT研发的数学专家系统MACSYMA投入使用，可以求解600多种数学问题。
  - 1974年，绍特里夫研制成功医疗专家系统MYCIN，可以为细菌感染疾病提供抗菌剂建议，该系统解决了一系列人工智能应用技术问题，包括知识表示、搜索策略、人机联系和知识获取等。
  - 1976年，斯坦福大学开始研制地质勘探专家系统PROSPECTOR，1982年，该系统准确地预测了华盛顿州一个勘探地段的钼矿位置。

# 发展和实用化（1971-1980）

- 知识表示方法的进展：
  - 1968年，Quillian提出了语义网络知识表示法，试图解决记忆的心理学模型。后来，Simmon在利用语义网络表示自然语言方面取得了很大的进展。
  - 1974年，Minsky提出了知识表示的另一种方法——框架理论，能够较好地描述范围较广的一类问题，可供智能系统理解视觉、自然语言和其他复杂行为，使得知识表示理论更趋于完善。
- 非精确推理方面的进展：主要研究如何处理专家知识的不确定性和推理证据的不确定性。
  - 1976年，R. O. Duda提出在PROSPECTOR系统中采用主观贝叶斯理论，该方法后来被很多专家系统采用。

# 发展和实用化（1971-1980）

- 归结原理：
  - 1965年Robinson提出归结原理
  - 接着柯瓦斯基提出以逻辑为基础的程序设计语言PROLOG（PROgramming in LOGic）的思想。
  - PROLOG是一阶谓词逻辑的Horn子集为语法，以Robisson的归结原理为工具，加上深度优先控制策略，形成的人工智能通用程序设计语言。
  - PROLOG的特点是能像人脑一样自动进行**逻辑推理**。是比LISP更高级的人工智能语言，被广泛应用于专家系统、自然语言处理、问题求解和数理逻辑等人工智能的多个领域。
  - 1972年，科麦瑞尔在法国马赛大学实现了第一个Prolog系统。

# 知识工程和专家系统（1980-1990）

- 1977年，在第五届国际人工智能会议上，美国斯坦福大学计算机科学家费根鲍姆教授(E．A．Feigenbaum) 发表特约文章“人工智能的艺术：知识工程课题及实例研究”，系统地阐述了“专家系统”的思想，并提出了“知识工程”的概念，确定了知识在人工智能中的重要地位。
- 1984年，“*ARTIFICIAL INTELLIGENCE: THE SECOND COMPUTER AGE*”

  ***Knowledge is power, and the computer is an amplifier of that power. We are now at the dawn of a new computer revolution… Knowledge itself is to become the new wealth of nations.***

- 知识工程很快渗透到人工智能的各个领域，促进了人工智能从实验室研究走向实际应用。

# 国际象棋（1990-）：搜索

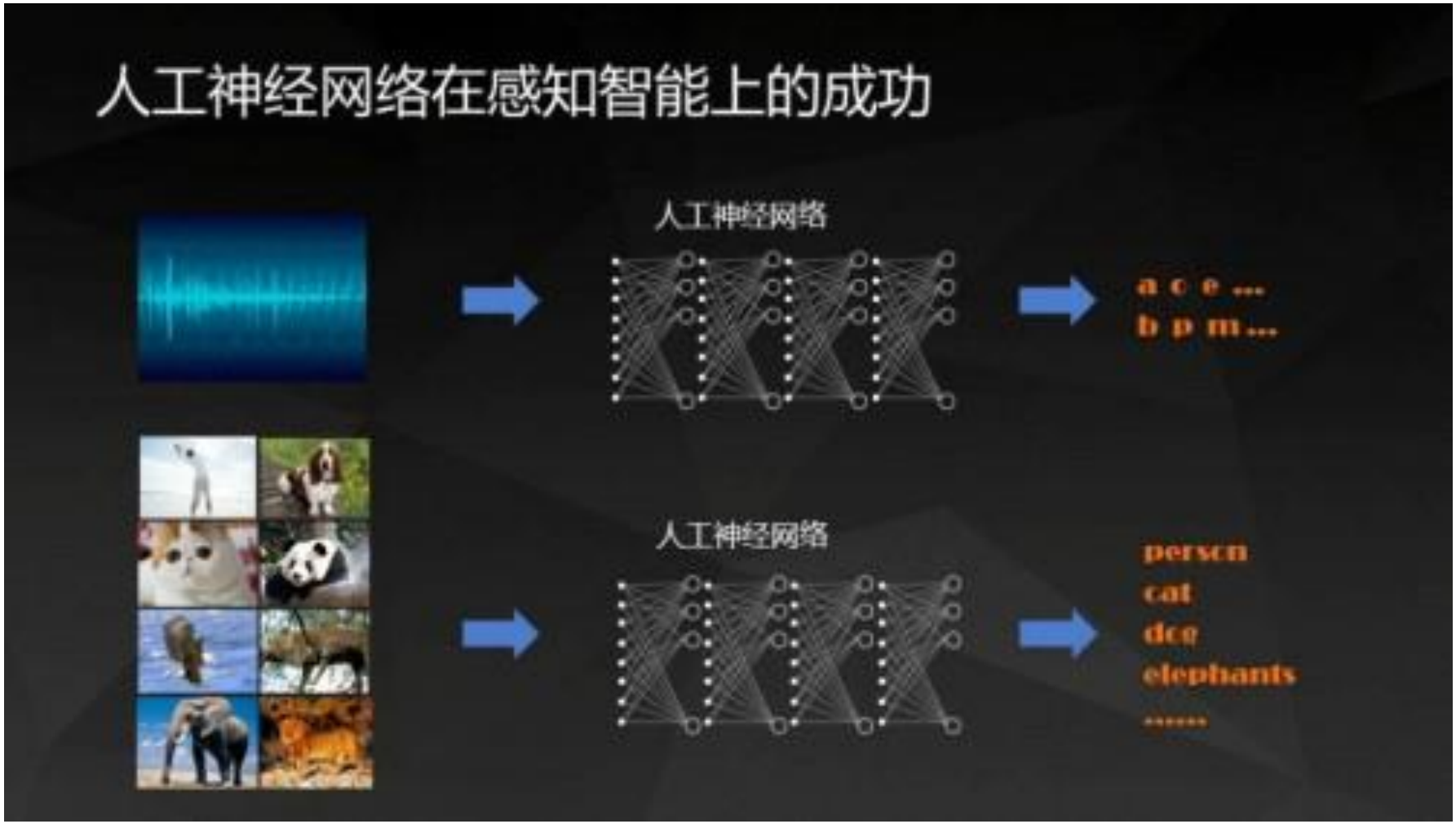
- 深蓝 (Deep Blue)
  - 美国IBM公司生产的一台超级国际象棋电脑，重1270公斤，有32个大脑（微处理器），每秒钟可以计算2亿步。
  - 输入了一百多年来优秀棋手的对局两百多万局。
  - 深蓝在国际象棋合理的步骤范围内，把棋局的10的46次方的可能性优化到一个可计算的范围，并给出所有的可能性。
  - 1997年的深蓝可搜寻及估计随后的12步棋，而一名人类象棋好手大约可估计随后的10步棋。
- 人机大战
  - 1996年2月10日，超级电脑深蓝首次挑战世界冠军卡斯帕罗夫，但以2-4落败。
  - 1997 年 5 月 11 日，深蓝首次击败了卡斯帕罗夫（2胜1负3平）。

# 深度问答（1990-）：DeepQA

- 沃森 (Watson)
  - 美国IBM公司生产的一台超级问答系统
  - 90台IBM服务器、360个CPU组成，每个CPU主频可达4.1GHz
  - 15TB内存、每秒可进行80万亿次运算
  - 存储了大量图书、新闻和电影剧本资料、辞海、文选和《世界图书百科全书》(World Book Encyclopedia)等海量的资料
  - 核心技术是DeepQA

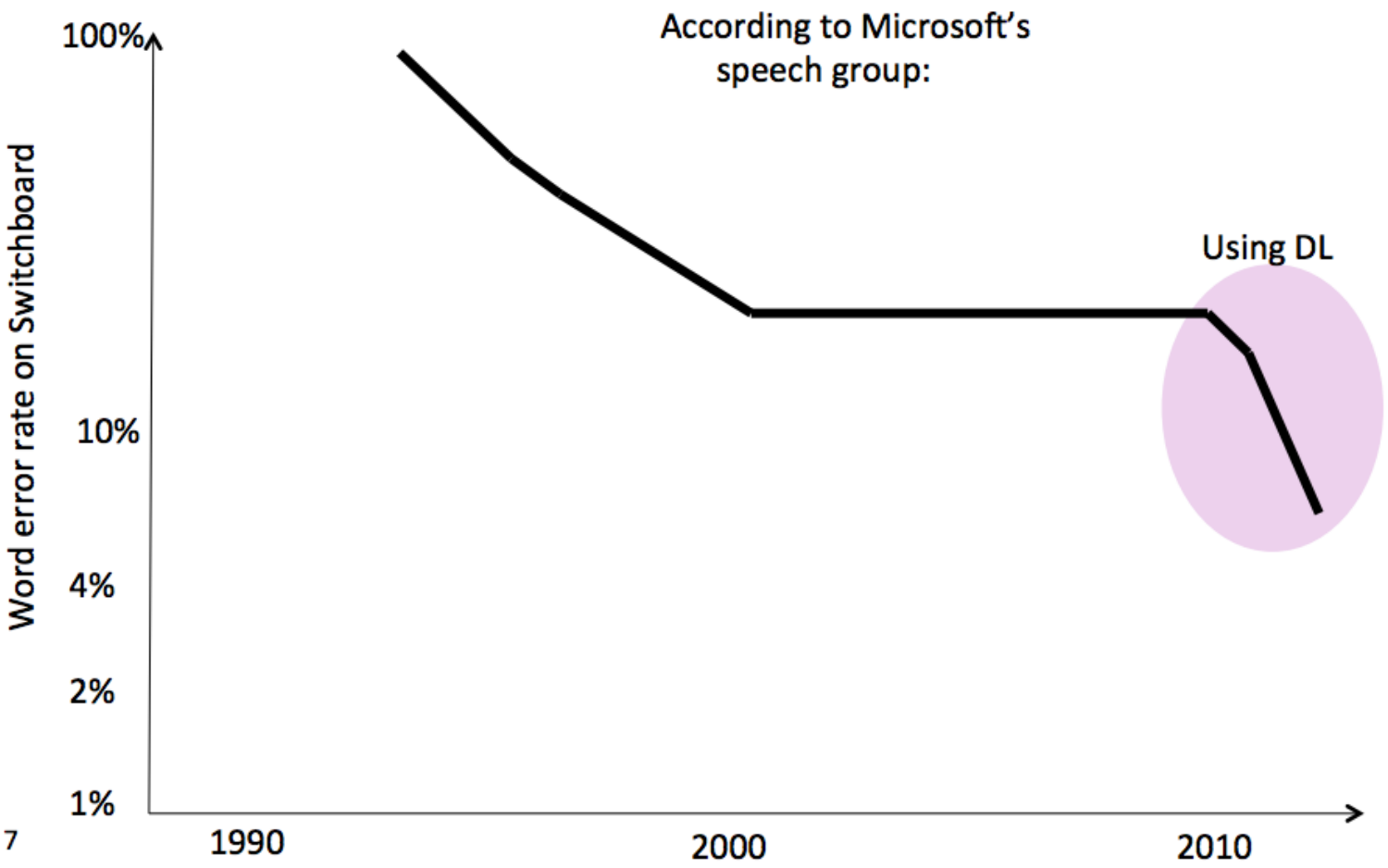
- 人机大战
  - 2011年，沃森在美国智力竞赛节目《危险边缘Jeopardy!》中上演“人机问答大战”，战胜人类选手

# 深度学习的突破：语音、图像和视频




http://www.csdn.net/article/1970-01-01/2821893

# 深度学习的突破：语音识别




http://humphreysheil.blogspot.com/2014/10/named-entity-recognition-short-tutorial.html

# 深度学习的突破：自然语言处理

## 关系分类

Convolution Neural Network

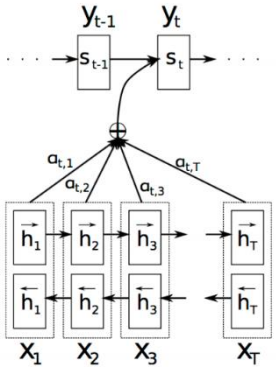
| Classifier | Feature Sets | F1 |
|---|---|---|
| SVM | POS, stemming, syntactic patterns | 60.1 |
| SVM | word pair, words in between | 72.5 |
| SVM | POS, stemming, syntactic patterns, WordNet | 74.8 |
| MaxEnt | POS, morphological, noun compound, thesauri, Google n-grams, WordNet | 77.6 |
| SVM | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | - | 74.8 |
| | POS, NER, WordNet, syntactic tree | 77.6 |
| MVRNN | - | 79.1 |
| | POS, NER, WordNet, syntactic tree | 82.4 |
| **Proposed** | word pair, WordNet | **82.7** |

CNN: Zeng et al. Coling 2014

| Classifier | Feature set | $F_1$ |
|---|---|---|
| SVM | POS, WordNet, prefixes and other morphological features, depdency parse, Levin classes, PropBank, FanmeNet, NomLex-Plus, Google $n$-gram, paraphrases, TextRunner | 82.2 |
| RNN | Word embeddings | 74.8 |
| | Word embeddings, POS, NER, WordNet | 77.6 |
| MVRNN | Word embeddings | 79.1 |
| | Word embeddings, POS, NER, WordNet | 82.4 |
| CNN | Word embeddings | 69.7 |
| | Word embeddings, word position embeddings, WordNet | 82.7 |
| Chain CNN | Word embeddings, POS, NER, WordNet | 82.7 |
| FCM | Word embeddings | 80.6 |
| | Word embeddings, depedency parsing, NER | 83.0 |
| CR-CNN | Word embeddings | 82.8† |
| | Word embeddings, position embeddings | 82.7 |
| | Word embeddings, position embeddings | **84.1**† |
| SDP-LSTM | Word embeddings | 82.4 |
| | Word embeddings, POS embeddings, WordNet embeddings, grammar relation embeddings | **83.7** |

LSTM: Xu et al. EMNLP 2015

## 机器翻译



| System | Training | MT06 | MT02 | MT03 | MT04 | MT05 | MT08 |
|---|---|---|---|---|---|---|---|
| MOSES | MERT | 32.74 | 32.49 | 32.40 | 33.38 | 30.20 | 25.28 |
| RNNSEARCH | MLE | 30.70 | 35.13 | 33.73 | 34.58 | 31.76 | 23.57 |
| | MRT | 37.34 | 40.36 | 40.93 | 41.37 | 38.81 | 29.23 |

[Bahdana et al. ICLR 2015]

# 深度学习的突破：围棋

- 阿尔法围棋（AlphaGo）
  - 由位于英国伦敦的Google旗下DeepMind公司一款围棋人工智能程序。
  - 主要工作原理是利用深度神经网络学习大量围棋棋谱。
  - 在传统蒙特卡洛搜索树的基础上，结合了一个价值网络和一个策略网络，使得系统能够有效降低搜索空间。
- 人机大战
  - 2015年10月，阿尔法围棋以5:0完胜欧洲围棋冠军、职业二段选手樊麾；
  - 2016年3月对战世界围棋冠军、职业九段选手李世石，并以4:1的总比分获胜。

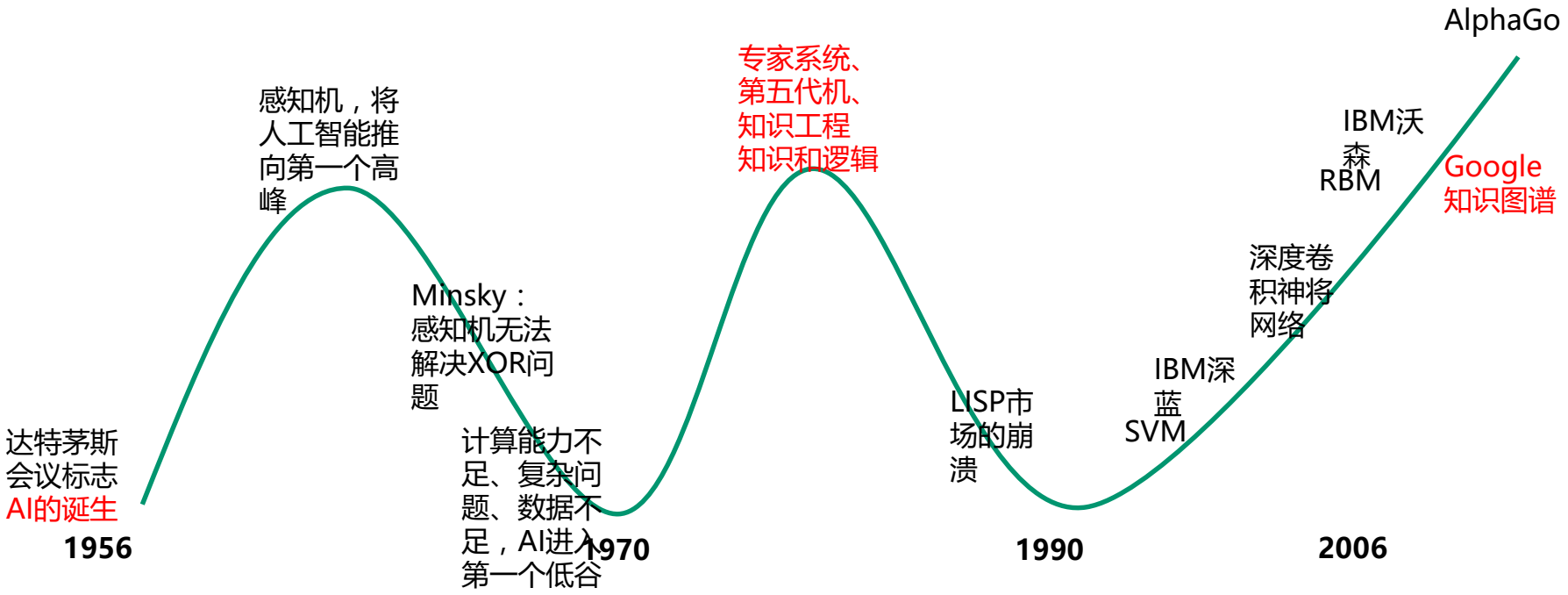
# 深度学习, 为什么成功?

- 模型
  - 深度神经网络：更丰富的连接网络可以建模更多的复杂任务
  - 152 layers [ResNet 16], 1000+layers [DN with Stochastic Depth 16]
  - 每2.4年，神经网络中隐含层节点翻番
- 资源
  - 每个类别5,000个标注样本可以达到理想的性能
  - 超过亿级标注样本就可以超过人类能力
- 计算
  - 更快的CPU、GPU和TPU，更快的连接网络
  - 更好的分布式计算平台

Larger networks are able to achieve higher accuracy on more complex tasks with larger datasets.

*(Ian Goodfellow, Yoshua Bengio et, al. Deep Learning, 2016)*

# 人工智能的发展历史


达特茅斯会议标志
AI的诞生
1956
感知机，将人工智能推向第一个高峰
Minsky：感知机无法解决XOR问题
计算能力不足、复杂问题、数据不足，AI进入第一个低谷
1970
专家系统、第五代机、知识工程
知识和逻辑
LISP市场的崩溃
1990
IBM深蓝
SVM
深度卷积神将网络
IBM沃森
RBM
2006
Google
知识图谱
AlphaGo

# 智能、人工智能和智能系统

- 智能

  **智能**是人类运用知识解决问题的能力

  演绎能力和归纳能力

- 人工智能

  研究怎样使计算机来模仿人脑所从事的推理、学习、思考、规划等思维活动，解决需人类专家才能处理的复杂问题。将人的思考过程、智能活动一部分机械化

- 智能系统

  **智能系统**利用人工智能技术解决实际问题，是知识工程研究的目标

# 人工智能的研究领域

- 专家系统
  - 一个智能的计算机程序，它运用知识和推理步骤来解决只有专家才能解决的复杂问题。
- 自然语言处理
  - 研究使用计算机理解和生成自然语言的基础理论和基本技术
- 机器学习
  - 研究如何使用计算机模拟或实现人类学习活动的科学
- 定理证明
  - 利用计算机证明非数值型的结果，即判断它们的真假。很多非数值领域的任务，如医疗诊断等都可以转化成一个定理证明问题。
- 博弈和游戏

# 人工智能的研究领域

- 机器人
  - 一种可再编程序的多功能的操作装置，是在电子学、控制理论、机械工程、仿生学和人工智能等各学科发展基础上出现的一门综合技术。
- 模式识别
  - 对表征事物或现象的各种形式的（数值的、符号的)信息进行处理和分析，以对事物或现象进行描述、辨认、分类和解释的过程。
- 计算机视觉
  - 研究如何使机器“看”的科学，更进一步的说，就是是指用摄影机和电脑代替人眼对目标进行识别、跟踪和测量等机器视觉，并进一步做图形处理，使电脑处理成为更适合人眼观察或传送给仪器检测的图像。
- 人工神经网络
  - 从信息处理角度对人脑神经元网络进行抽象， 建立某种简单模型，按不同的连接方式组成不同的网络。

# 提纲

- 课程介绍
- 知识图谱发展历史
  - 知识图谱和人工智能
  - 知识图谱和万维网
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用

# 互联网和英特网

- 互联网
  - internet
  - 泛指由多个计算机网络相互连接而成的一个大型网络
- 英特网
  - Internet
  - 特指由美国的ARPA网发展演变而来的、目前全球最大的一个互联网，它通过网络通讯协议TCP/IP实现互联
  - 因特网提供的主要应用有：万维网（WWW）、文件传输（FTP）、电子邮件（E-mail）、远程登录（Telnet）等

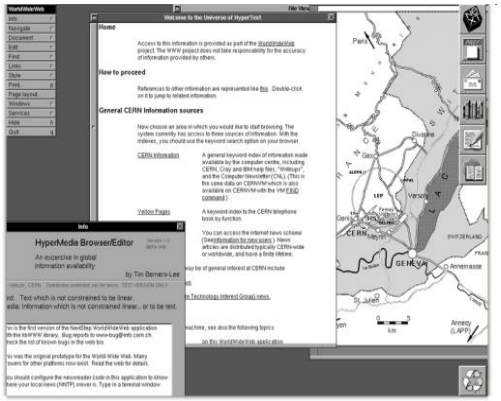
# 万维网

- World Wide Web(WWW，W3，Web)
- 英国人Tim Berners-Lee提出的一种因特网应用
- 思想：将互联网上的资源在一个网页中直观地表示出来，而在资源之间在网页上可以链来链去。
- 实现：
  - ✓ Web服务器：一个由许多互相链接的网页组成的系统。
    - 网页是WWW的基本信息单位，由文字、图片、动画、声音等多种媒体信息以及链接组成，用HTML编写，通过链接实现与其他网页的关联和跳转。
    - 每个网页都由一个“统一资源标识符”（URI）标识，这些资源通过超文本传输协议（HTTP）传送给用户，用户通过点击链接来获得资源。
  - ✓ Web客户端：常用浏览器，用户通过Web客户端访问浏览Web服务器上的网页。

http://info.cern.ch

# 世界上第一个万维网网站，1989

- 万维网之父：Tim Berners-Lee
- 技术
  - Internet
  - hypertext (HTML)
  - Web Browser
  - HTTP
- CERN
  - 欧洲核子研究组织，是世界上最大型的粒子物理学实验室，也是万维网的发祥地。它的内部深藏着一个升降机，整个机构位于瑞士日内瓦西部接壤法国的边境。



Engineer changed the World

# WWW的发展史

- World Wide Web (Web 1.0)
- Social Web (Web 2.0)
- Semantic Web (Web 3.0)
- Web Science

# Web 1.0

## 描述：

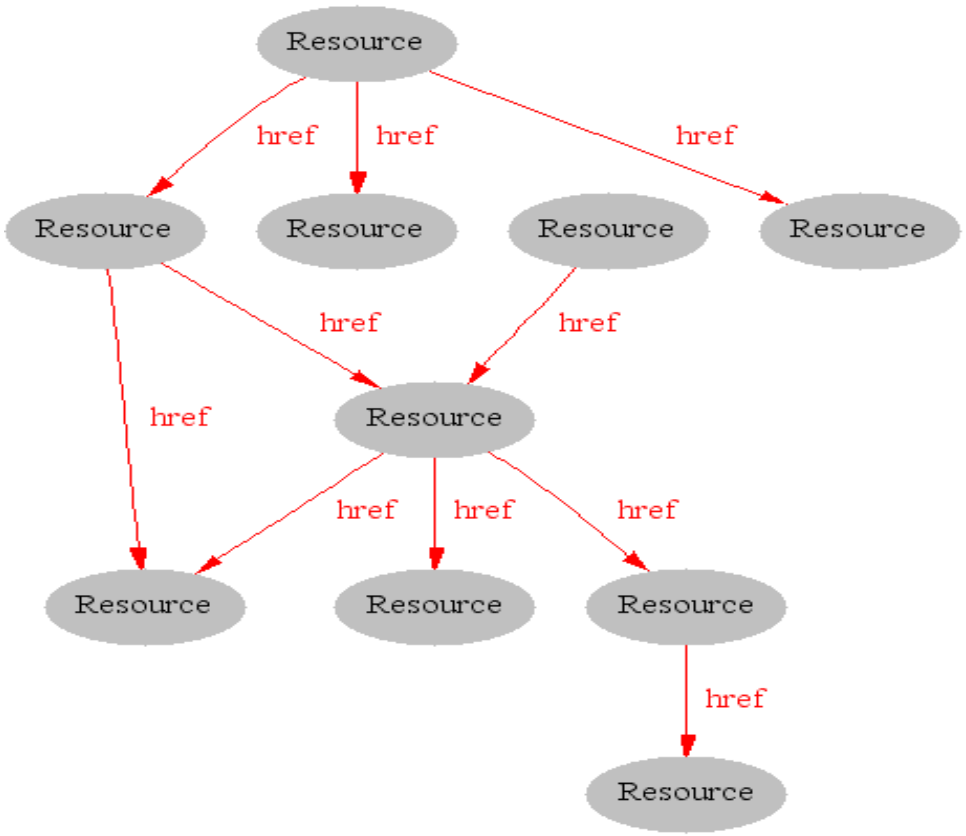
- ✓ 以内容为中心，以信息的发布、传输、分类、共享为目的的互联网应用称为Web1.0。

## 特点:

- ✓ 网站管理者负责制定网站内容，话语权是掌握在各大网站的手里
- ✓ 绝大多数网络用户只充当浏览者的角色

## 技术：

- ✓ hypertext (HTML)
- ✓ Web Browser
- ✓ HTTP




href：Hypertext Reference的缩写，超文本引用。标签的 href 属性用于指定超链接目标的URI。

# Social Web – Web 2.0

- Web2.0 是相对Web1.0 的新一类互联网应用的统称。Web2.0 更注重用户的交互作用，用户既是网站内容的浏览者，也是网站内容的制造者。
- http://en.wikipedia.org/wiki/Web_2.0
  - *“Web 2.0 … has … come to refer to what some people describe as a second phase of architecture and application development for the World Wide Web.”*
- The Web where “ordinary” users can meet, collaborate, and share using social software applications on the Web (tagged content, social bookmarking, AJAX, etc.)

# Web2.0：支撑技术

- P2P
  - ✓Peer to Peer，对等连接或对等网络，特征之一是弱化了服务器的作用，甚至取消服务器，任意两台PC互为服务器，同时又是客户机。
- Blog
  - ✓Web Blog，网络日志，是一种由个人管理、不定期张贴新的文章的网站。一个典型的博客结合了文字、图像、其他博客或网站的链接及其它与主题相关的媒体，能够让读者以互动的方式留下意见。

# Web2.0：支撑技术

- Wiki
  - ✓wee kee wee kee, 夏威夷语，“维基”或“维客”
  - ✓一种超文本系统
  - ✓在Web的基础上对Wiki文本进行浏览、创建和更改，而且创建、更改、发布的代价远比HTML文本小。
  - ✓还支持面向社群的协作式写作，为协作式写作提供必要帮助
  - ✓Wiki的写作者自然构成了一个社群，Wiki系统为这个社群提供简单的交流工具。
  - ✓与其它超文本系统相比，Wiki有使用方便及开放的特点，所以Wiki系统可以帮助我们在一个社群内共享某领域的知识。

# Web2.0：支撑技术

- XML
  - ✓eXtensible Markup Language，可扩展标记语言
  - ✓允许定义数量不限的标记来描述文档中的内容，允许嵌套的信息结构
  - ✓相对于HTML 主要用于描述Web 页的显示格式，XML 着重描述Web 页面的内容（XML 不会替代 HTML)
  - ✓XML 数据以纯文本格式进行存储，因此提供了一种独立于软件和硬件的数据存储方法。这让创建不同应用程序可以共享的数据变得更加容易。

# Web2.0：支撑技术

- Ajax
  - ✓Asynchronous JavaScript and XML，通过在用户和服务器之间添加一个中间层来实现用户操作与服务器响应的异步化。
  - ✓应用Ajax后，原本服务器承担的一些简单工作被转移到客户端，由客户端在闲置时完成。这样不但可以减轻服务器的工作压力，还可以实现节约带宽的双重目的。

# 万维网技术的局限性

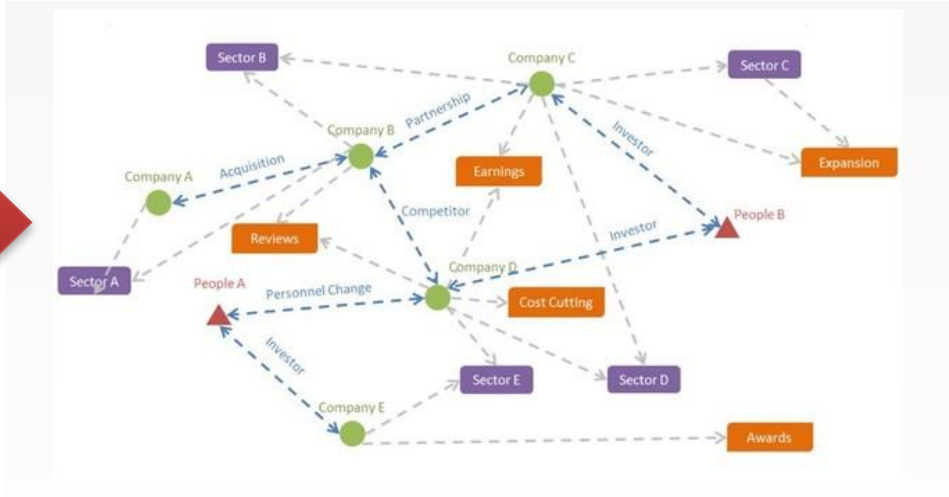
- 万维网现在采用的超文本标记语言HTML，是面向人的存储和共享信息的媒介，Web上的内容是提供给人而不是机器来理解和浏览的。由于Web内容没有采用形式化的表示方式，并且缺乏明确的语义信息，故而计算机“看到的”Web内容只是普通的二进制数据，对其内容无法进行识别。如果机器不能充分理解网页内容的含义，就无法实现Web内容的自动处理。

# 万维网技术的局限性(cont.)

- 万维网是按“网页的地址”，而**非“内容的语义”**来定位信息资源的（缺少语义联接），网上所有信息都是由不同的网站发布的，相同主题的信息分散在全球众多不同的服务器上，又缺少有效工具能将不同来源的相关信息综合起来，因此形成了一个个信息孤岛(其中有大量重复)，查找自己所需的信息就像大海捞针一样困难。

# 网络内容转化为知识

- What we need is to go from data to information, then from information to meaning: adding semantics to data and services is the solution!




网络内容　　知 识

# 语义网：Semantic Web

- 由万维网创始人的Tim Berners-Lee在1998年提出的对未来网络的一个设想，它是一种智能网络。

- 语义网就是以Web数据的内容（即数据的语义）为核心，用机器能够理解和处理的方式链接起来的海量分布式数据库。

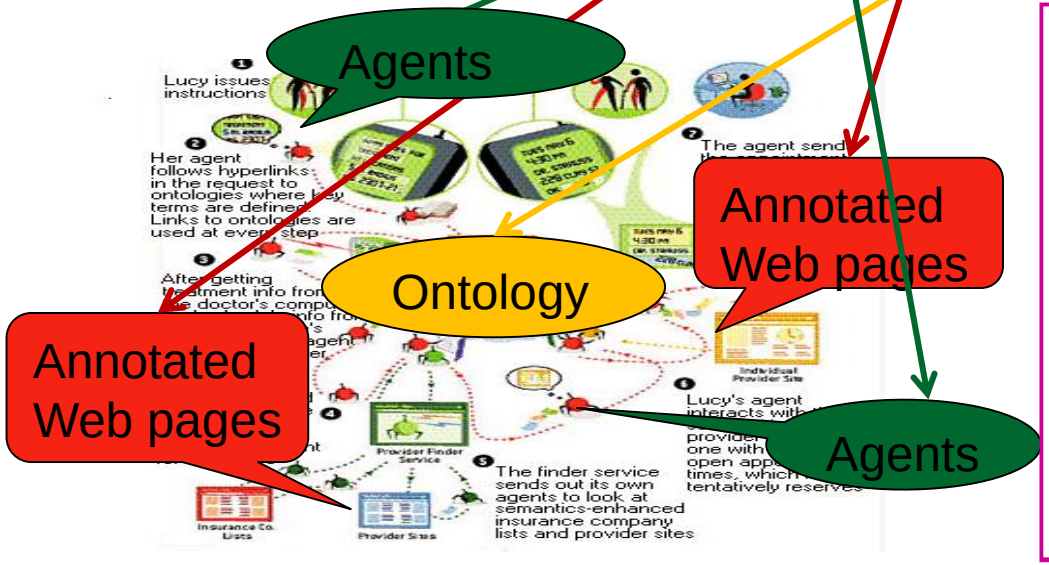
- 语义网是对现有Web增加了语义支持，它是现有万维网的延伸与变革，其目标是帮助机器在一定程度上理解Web信息的含义，使得高效的信息共享和机器智能协同成为可能。语义网将会为用户提供动态、主动的服务，从而更便于机器和机器、人和机器之间的对话及协同工作。

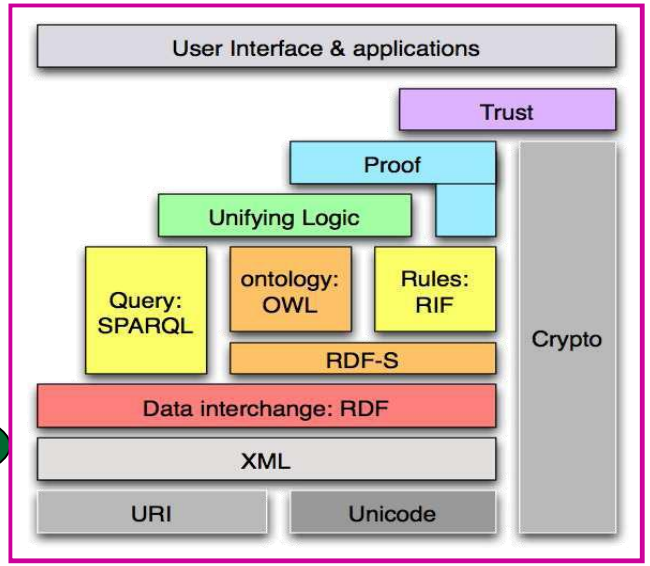
# 语义网

## ■定义

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation.

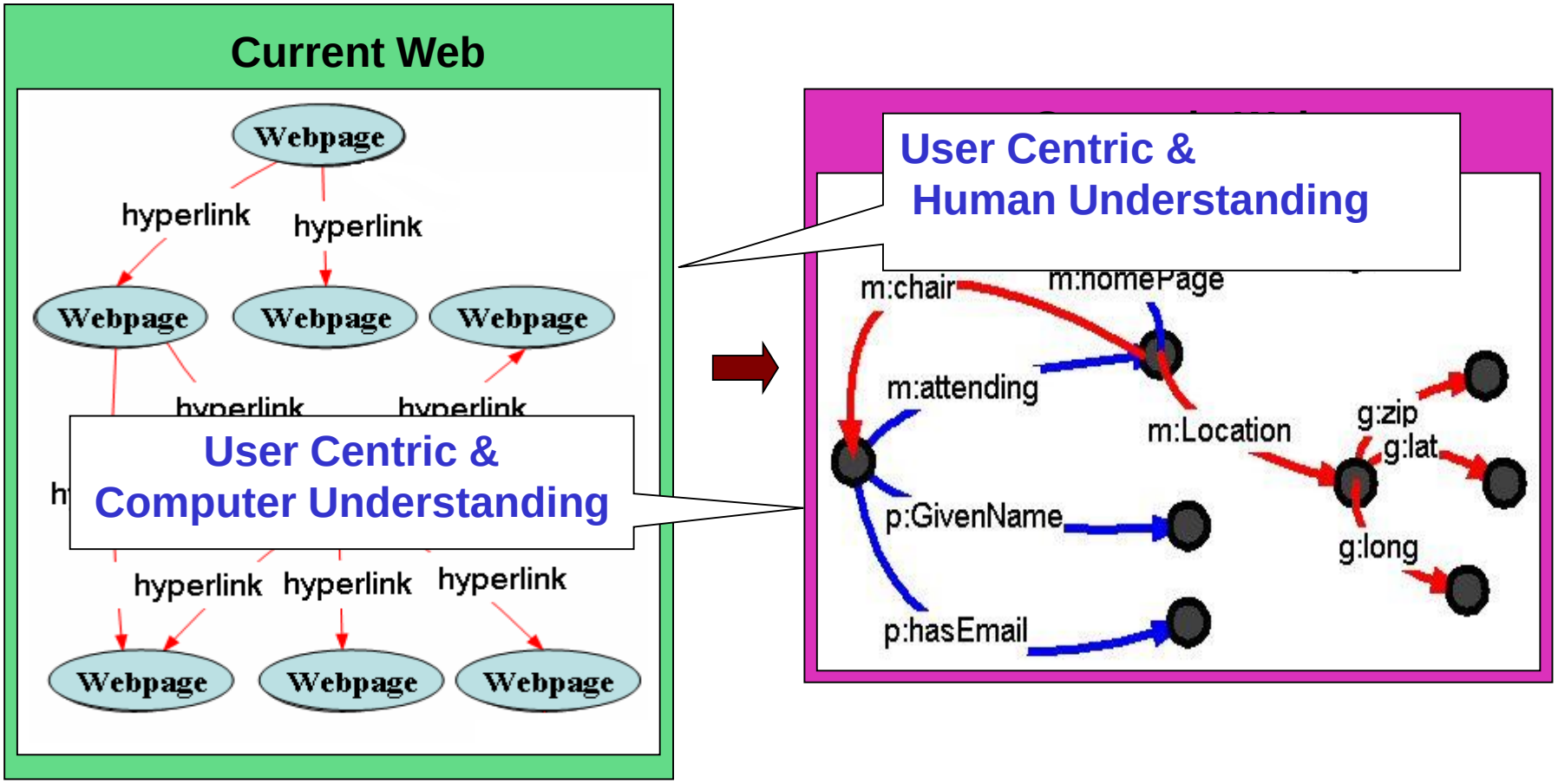
– Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001



# Semantic Web: Web 3.0

■动机



# 语义网的特征和目标

- 语义网不同于WWW，是WWW的扩展与延伸
- WWW的操作单元是文档，而语义网的操作单元是文档描述的语义内容
- WWW是面向人的网络服务，而语义网是面向机器的网络服务
- 语义网更利于计算机“理解与处理”，并具有一定的判断、推理能力

- 计算机理解海量网络资源的语义，从而将万维网中一个个现存的信息孤岛，连接成一个巨大的语义清晰的数据库。
- 在搜索数以万计的网页时，通过“智能代理”从中筛选出相关的有用信息。而不像现在的万维网，只给你罗列出数以万计的无用搜索结果。从而使人类用户从搜索相关网页的繁重劳动中解放出来。

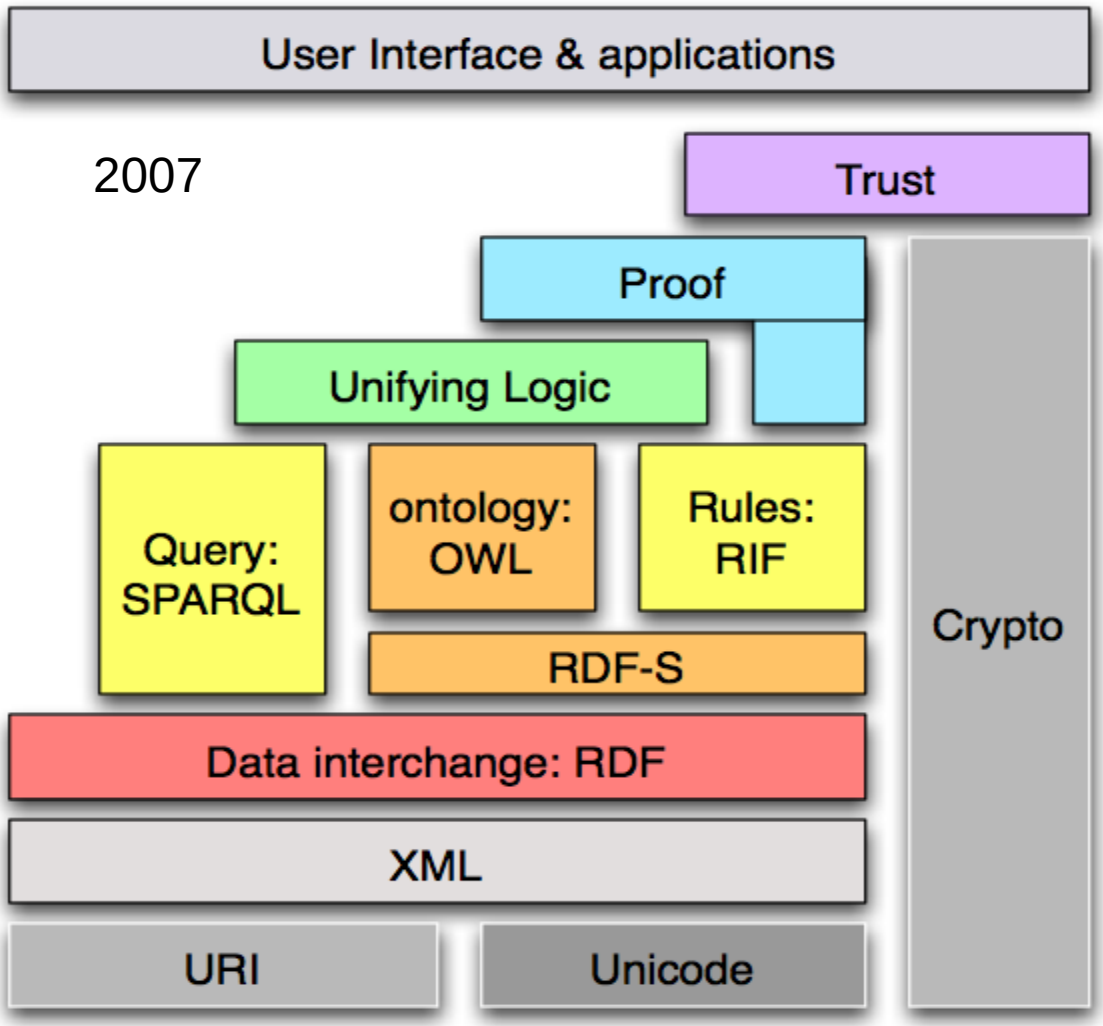
# 语义网的实现

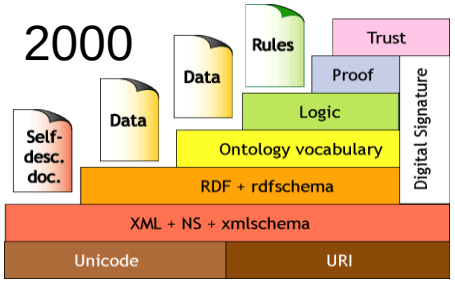
- 数据的语义标注和链接
  - 通过一套统一的数据标准对网络信息进行详细的语义标记，使得语义网能够精准的识别信息，区分信息的含义和作用
  - 基于可扩展标记语言XML和资源描述框架RDF实现
  - 通过亿万个体实现
- 具有语义分析能力的信息搜索引擎
  - 能够理解人类的自然语言，并且具有一定的推理和判断能力，从而提供符合用户需求的信息服务

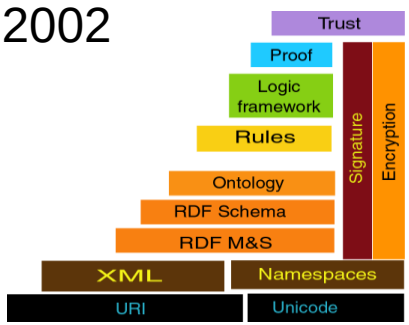
# 语义网的关键技术

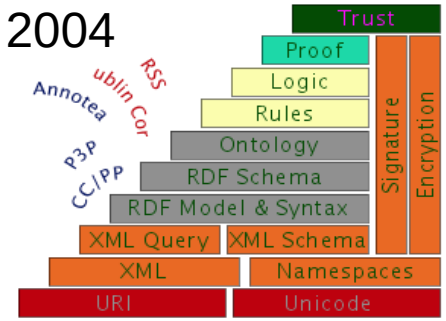
- 三大关键技术：XML、RDF和Ontology
  - 可扩展标记语言XML
    - 一种标准的元数据语法规范
  - 资源描述框架RDF
    - 一种标准的元数据语义描述规范
  - 本体Ontology
    - 一种描述客观世界的概念化规范

# Semantic Web的体系结构



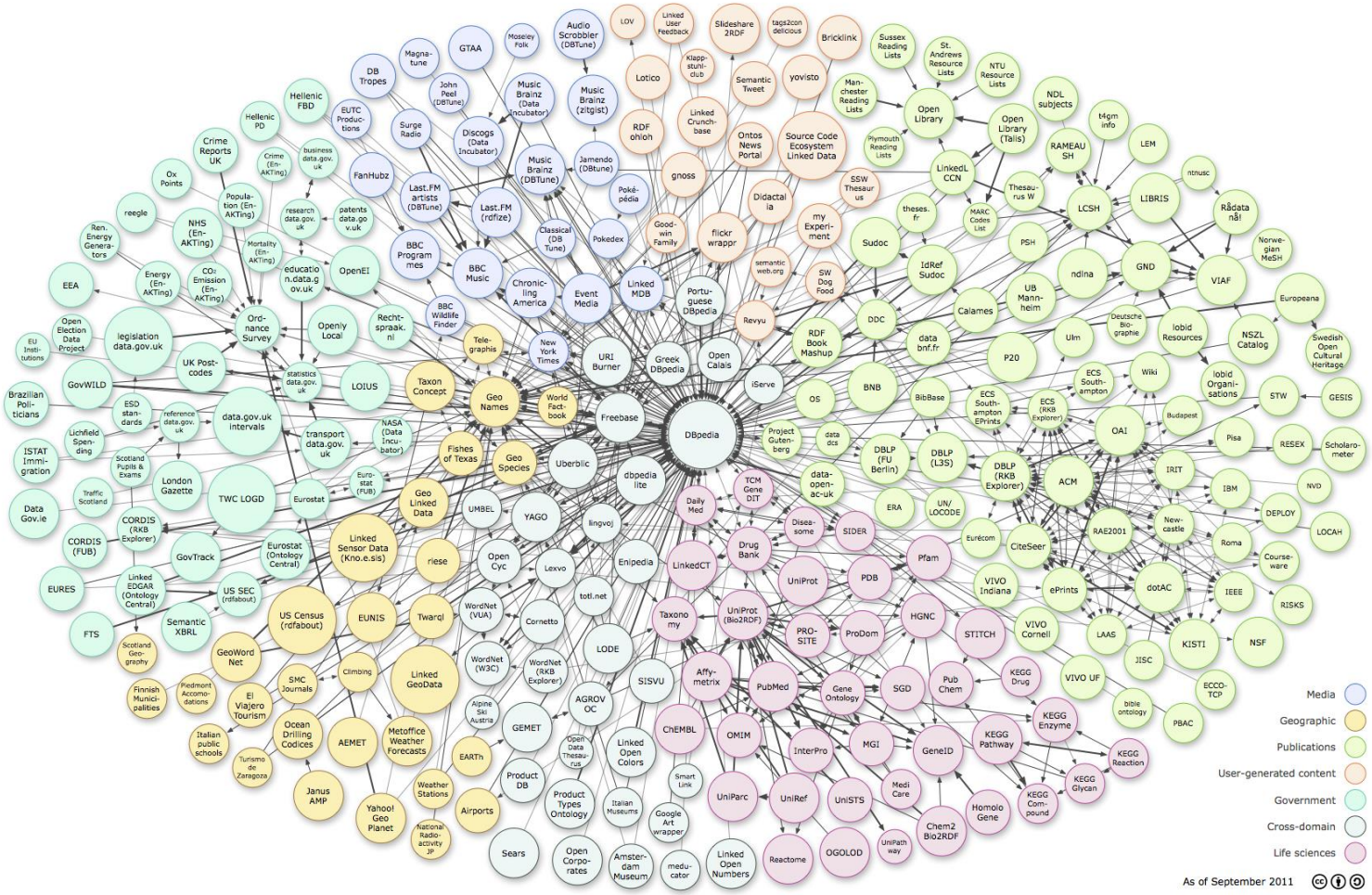




http://www.w3.org/2001/sw/

# Linked data in Sep. 2014

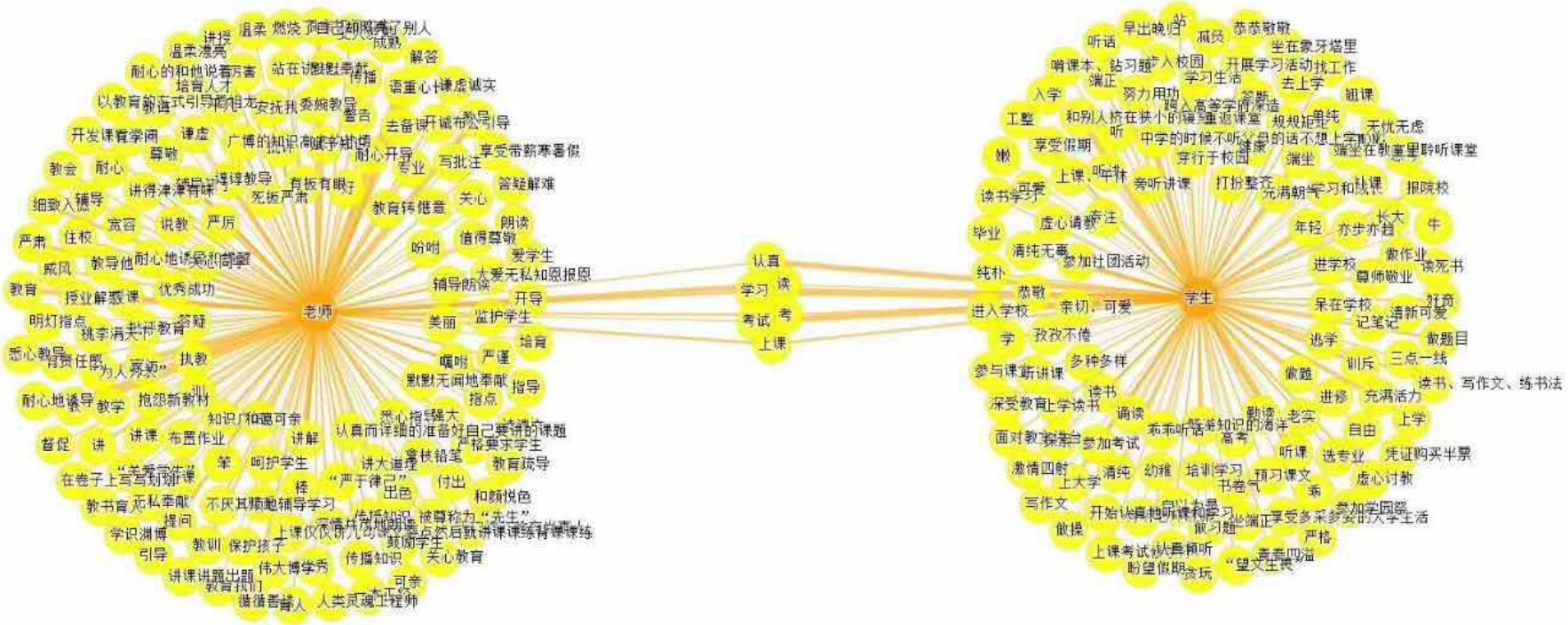



Media
Geographic
Publications
User-generated content
Government
Cross-domain
Life sciences

As of September 2011

# 知识图谱是Semantic Web的基件

- The Knowledge Graph is a system that understands facts about people, places and things and how these entities are all connected.



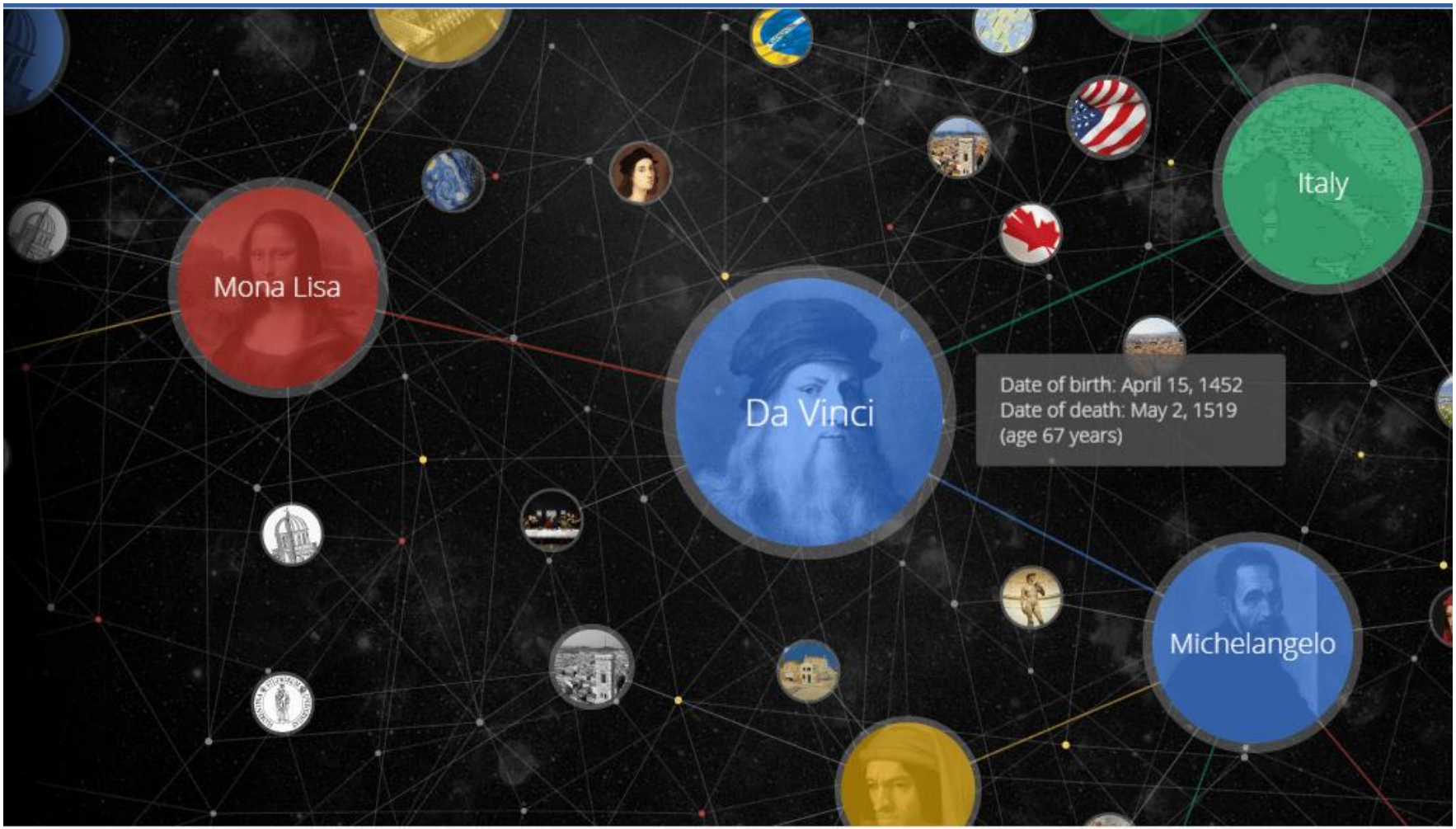
# Tim Berners-Lee获得图灵奖

万维网（World Wide Web）的发明人、麻省理工学院教授Tim Berners-Lee刚刚获得了由美国计算机协会（ACM）颁发的图灵奖（A.M. Turing Award），该奖项是全球范围内计算机科学领域最负盛名的大奖，通常被称为“诺贝尔计算机奖”。该奖项的100万美元奖金由Google提供的。

# 提纲

- 课程介绍
- 知识图谱发展历史
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用

# Google Knowledge Graph



# 知识组织：一些容易混淆的术语

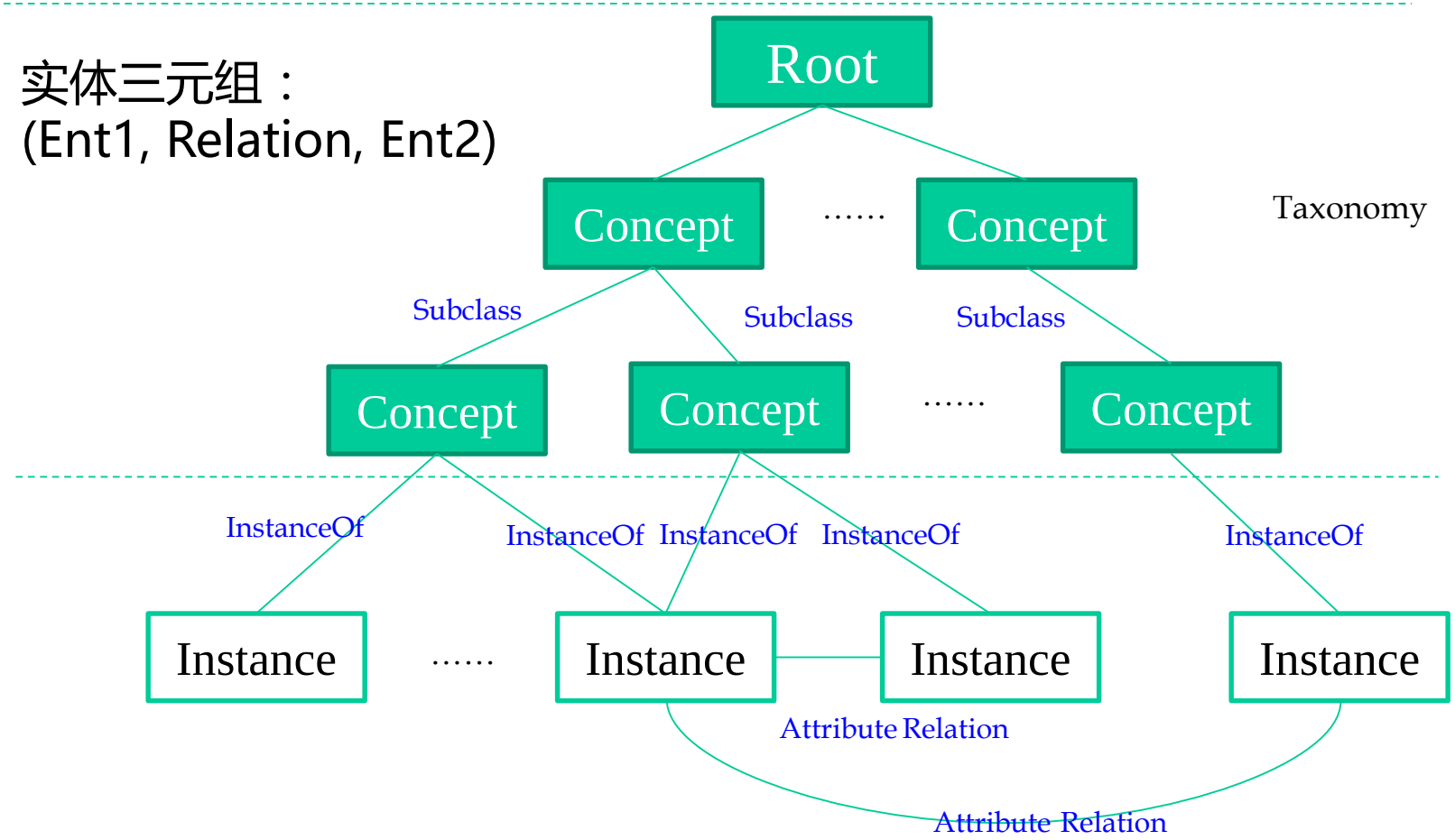
- Ontology vs. Knowledge Base
  - Ontology：共享概念化的规范，涉及概念、关系和公理三个要素
  - Knowledge：服从于ontology 控制的知识单元的载体
  - Ontology是蛋糕的模具，Knowledge Base是蛋糕
- 公理：Formal Ontology vs. Lightweight Ontology
  - Formal Ontology: 大量使用公理
  - Lightweight Ontology: 不用或很少使用公理

# 知识组织：一些容易混淆的术语

- Ontology
  - 树状结构，不同层节点之间是严格的IsA关系
  - 优点：可以适用于知识的推理
  - 缺点：无法表示概念的二义性（运动员：体育？人物？）
- Taxonomy
  - 树状结构，上下位节点之间非严格的IsA关系
  - 优点：可以表示概念的二义性（体育→运动员）
  - 缺点：不适用于推理，无法避免概念冗余（餐厅：美食？机构？地点？）
- Folksonomy
  - 类别标签，更加开放
  - 优点：能够涵盖更多的概念
  - 缺点：如何进行标签管理？
- 目前网络知识资源大多采用Folksonomy与Taxonomy结合的组织形式

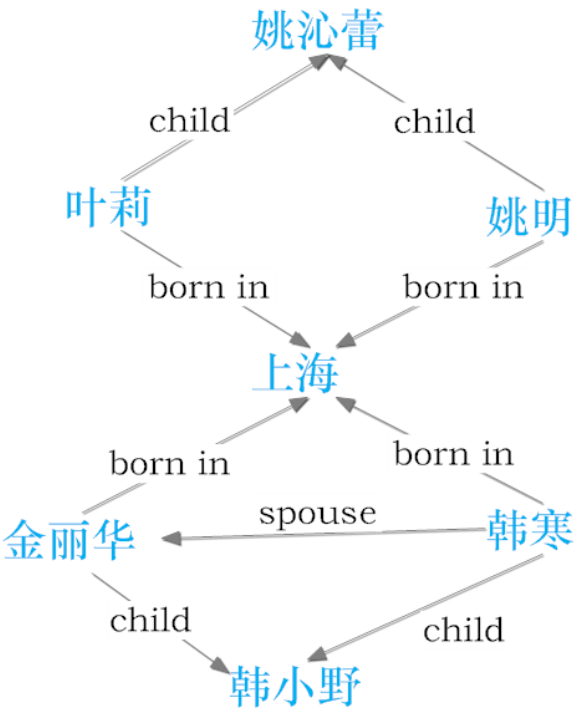
# 知识图谱的构成

实体三元组：
(Ent1, Relation, Ent2)



# 知识图谱形式化表示（符号表示）

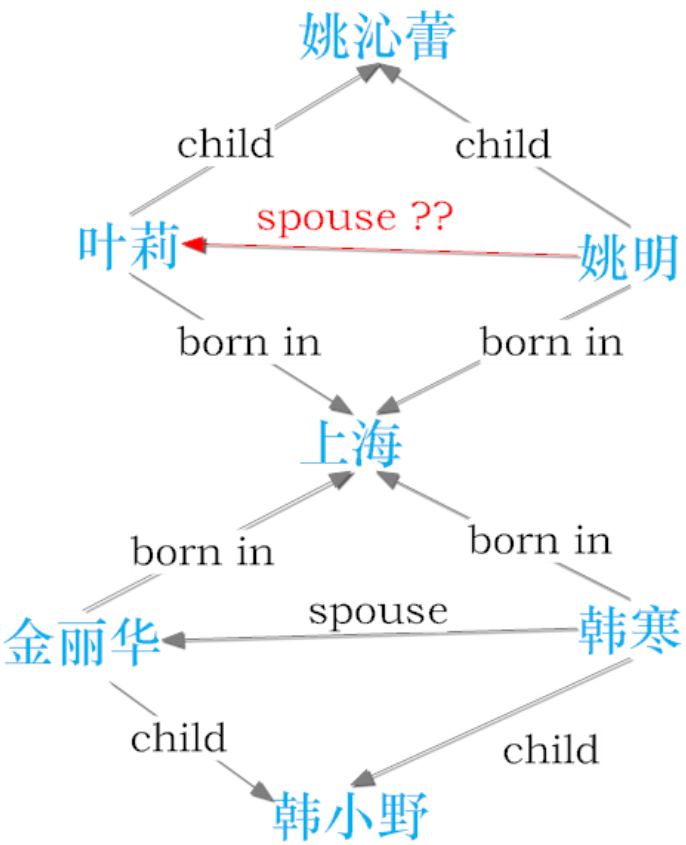
- 知识库是一个有向图
  - 多关系数据(multi-relational data)
  - 节点：实体/概念
  - 边：关系/属性
  - 关系事实 = *(head, relation, tail)*
    - head：头部实体
    - relation：关系/属性
    - tail：尾部实体

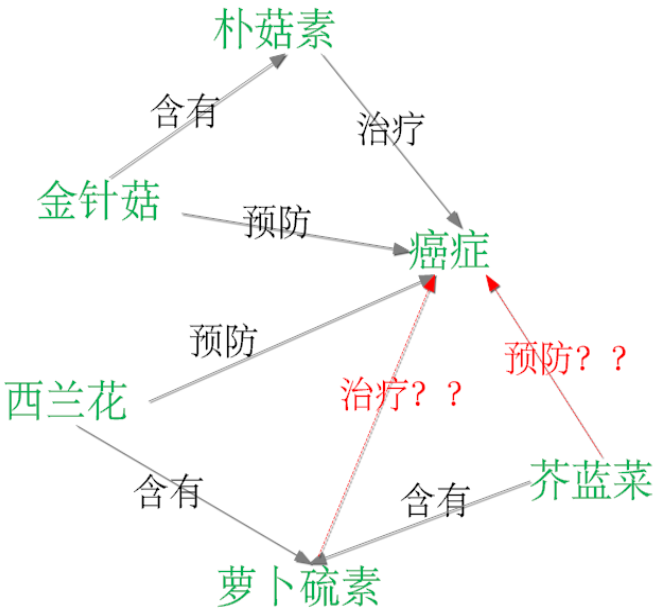



（姚明，born in，上海）

head　relation　tail

# 知识图谱形式化表示（符号表示）

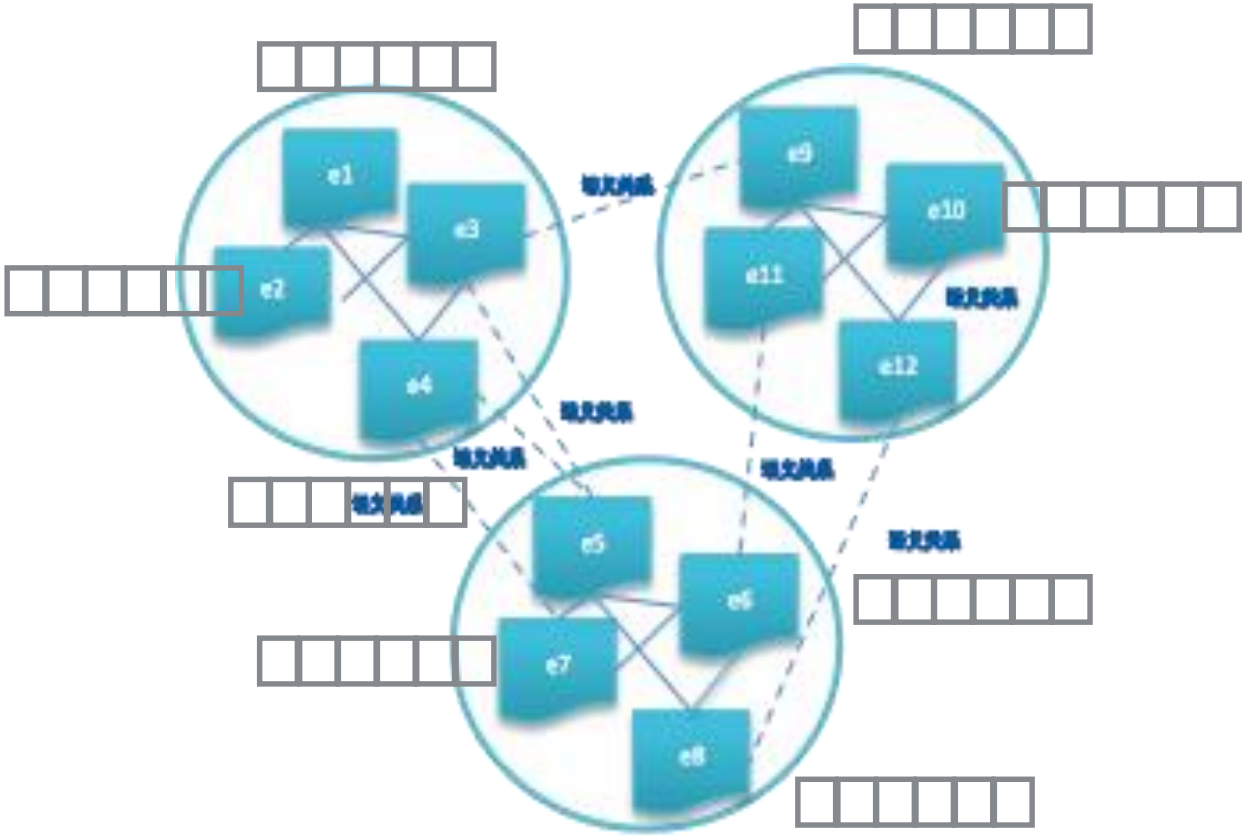



$\text{child}(A,B) \wedge \text{child}(A,C) \Rightarrow \text{spouse}(B,C)$

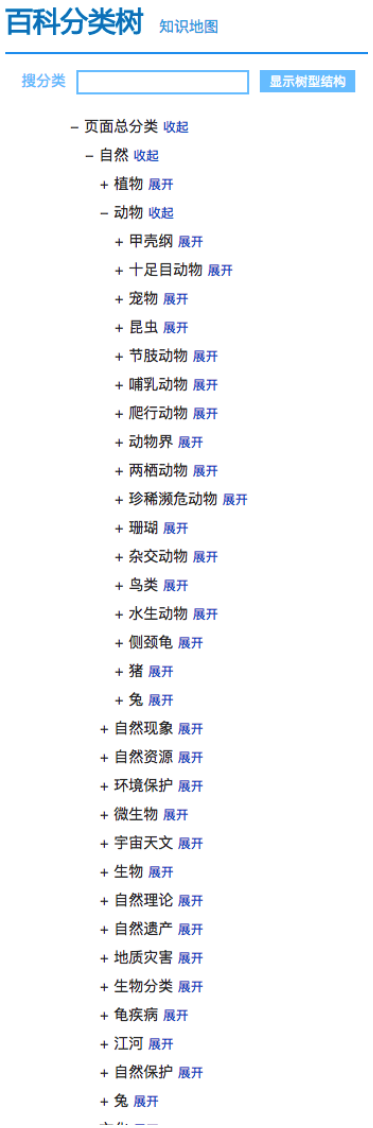
$\text{含有}(A,B) \wedge \text{治疗}(B,C) \Rightarrow \text{预防}(A,C)$

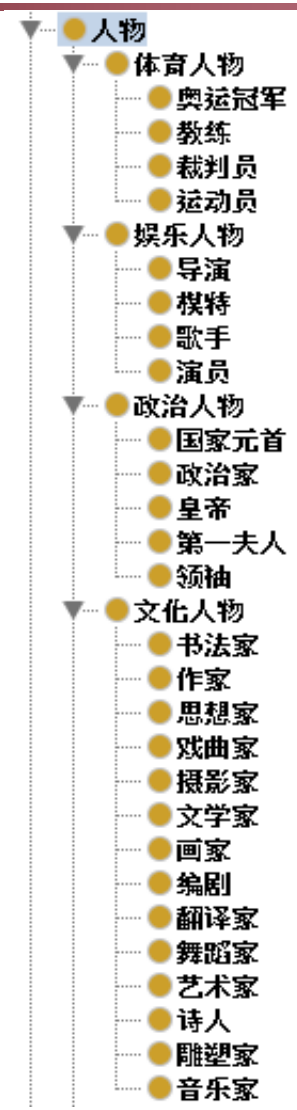
# 知识图谱的形式化表示（分布式表示）



# KG的基本概念

- Node：领域（Domain/Topic）



# KG的基本概念

- Node：概念（Concept）

# KG的基本概念

- Node：实例/实体（Entity/Objects/Instance）

**Yao Ming** (Q58590)

Chinese basketball player

edit

In more languages Configure

| Language | Label | Description | Also known as |
|---|---|---|---|
| English | Yao Ming | Chinese basketball player | |
| Chinese | 姚明 | 中国篮球运动员 | |
| Wu Chinese | No label defined | No description defined | |
| Cantonese | No label defined | No description defined | |

All entered languages

**Statements**

| | | |
|---|---|---|
| instance of | human<br>1 reference | edit<br>+ add |
| image | YaoMingonoffense2 crop.jpg<br>0 references | edit<br>+ add reference<br>+ add |
| sex or gender | male<br>3 references | edit<br>+ add |

# KG的基本概念

- Node：值（Value）
  - 实体（Entity）
    - （姚明，出生地，上海市）
  - 字符串（String）
    - （北京大学，学术传统，兼容并包、思想自由）
  - 数字（Number）
    - 平方公里：（北京市，面积， 1.641万）
    - 公斤：（姚明，体重，140公斤）
    - 米：（姚明，身高， 2.29米）
    - …
  - 时间（Date）
    - （姚明，出生年份，1981年）
  - 枚举（Enumerate）
    - （姚明，性别，男）
  - ……

# KG的基本概念

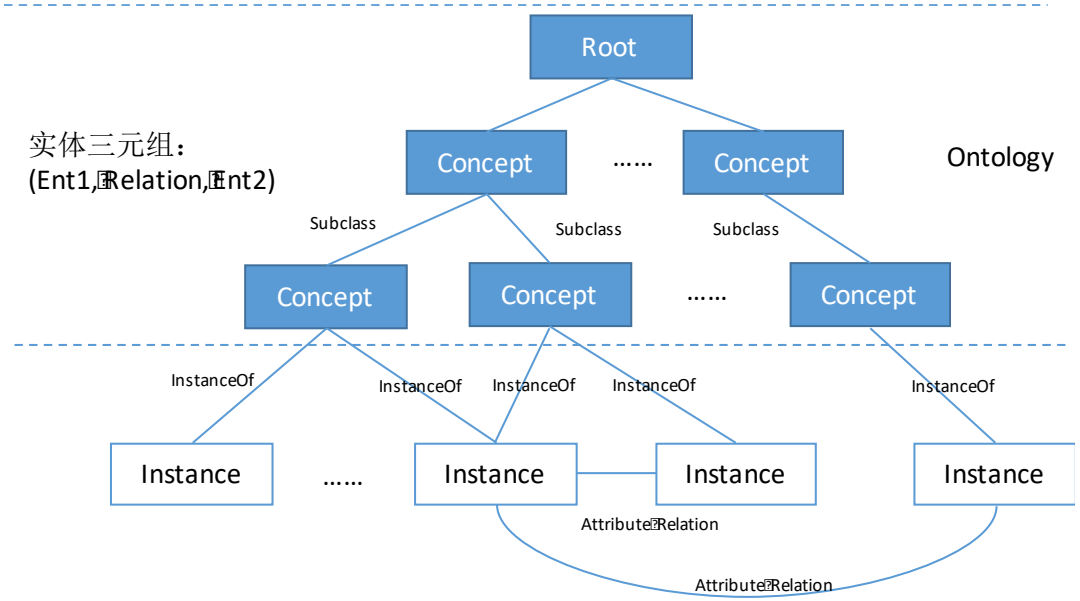
- 边：关系
  - Type：类型
  - Subclass：子类
  - Relation：关系
  - Property、Attribute：属性

(旺财， Is-A， 狗）

(狗， Is-A， 哺乳动物）

(旺财， 朋友， 小白）

(旺财， 颜色， 黄色）



# KG的基本概念

- 关系：Taxonomic Relation vs. Non-taxonomic Relation
  - Taxonomic Relation： is-a/Hypernym-Hyponym
  - Non-taxonomic Relation: 概念之间的相互作用
    - Meoronymy 部分整体
    - Attribute 属性
    - Thematic roles 论旨角色
    - Possession 领属
    - Casuality 因果
    - ……

# KG的基本概念

- Node：高阶三元组
  - 与时间、地点相关
    - （（美国，总统，特朗普），开始时间，2017）
  - 事件
    - Compound Value Type：组合值类型

A Compound Value Type is a Type within Freebase which is used to represent data where each entry consists of multiple fields. Compound value types, or CVT's are used in Freebase to represent **complex data**.

# Example of CVT

**Donald Trump** (Q22686)

| | | |
|---|---|---|
| spouse | Ivana Trump | |
| | start time | 7 April 1977 |
| | end time | 22 March 1992 |
| | 1 reference | |
| | reference URL | http://hollywoodlife.com/celeb/ivana-trump/ |
| | quote | Ivana married Donald Trump on April 7, 1977. (English) |
| | Melania Trump | |
| | start time | 22 January 2005 |
| | place of marriage | Mar-a-Lago |
| | 1 reference | |
| | reference URL | http://www.hollywoodreporter.com/features/trumps-wedding-melania-bill-hill-880088 |
| | quote | on Jan. 22, 2005, it was a different story. Trump married model Melania Knauss (English) |
| | Marla Maples | |
| | end time | 8 June 1999 |
| | start time | 19 December 1993 |
| | 0 references | |

# Example of CVT

Michael Jordan (Q41421)

member of sports team

North Carolina Tar Heels men's basketball — edit

| | |
|---|---|
| end time | 1984 |
| start time | 1981 |
| position played on team / speciality | shooting guard |
| | swingman |
| sport number | 23 |
| captain | *no value* |

0 references

+ add reference

Chicago Bulls — edit

| | |
|---|---|
| end time | 1993 |
| start time | 1984 |
| position played on team / speciality | shooting guard |
| sport number | 23 |
| captain | yes |

0 references

+ add reference

Scottsdale Scorpions — edit

| | |
|---|---|
| end time | 1994 |
| start time | 1994 |
| position played on team / speciality | outfielder |
| sport number | 35 |

0 references

+ add reference

# KG中的知识分类

## ■百科知识

标注系统　管理中心　导入体系　导入数据　浏览体系　共指列表　高级管理　你好：admin　修改日志　Log out

-root
- -创造性工作 [外文名,中文名]
  - +绘画 [别名,作者,尺寸,类型,年代,收藏单位]
  - +雕塑 [材料,高度,别名,作者,寓意,类型,年代]
  - +电视剧 [集数,导演,颜色,语言,编剧,地区,主演,别名,发行时间,获得荣誉,上映时间,制片人,类型,出品公司]
  - +音乐 [语言,地区,曲长,别名,发行时间,作曲者,类型]
  - +书籍 [isbn,语言,装帧,页数,开本,出版社,发行时间,作者,别名,字数,类型,价格]
  - +电影 [导演,颜色,语言,编剧,片长,主演,别名,发行公司,发行时间,获得荣誉,上映时间,imdb编码,制片人,分级,类型,出品公司,地区]
  - +软件应用 [语言,发行时间,开发者,编程语言,操作系统]
- -组织 [外文名,所在地,中文名,地址,别名,创建时间]
  - +教育组织 [类型]
  - +运动队 [主教练,代表队员,获得荣誉,所在联赛,运动项目]
  - +非政府组织 [创始人]
  - +政府机构 []
  - +公司 [注册资本,上市代码,公司口号,经营范围,法人代表,上市市场,证券简称,年盈利,总部所在地,宗旨理念,员工数,产品,创始人,总资产,行业,净利润,性质]
- -人物 [出生地,外文名,毕业院校,政党,职业,籍贯,去世日期,别名,信仰,国籍,中文名,体重,血型,星座,出生日期,性别,身高,相关事件,民族]
  - +网络人物 [代表作品,艺名,获得荣誉,主要成就]
  - +文化人物 [代表作品,获得荣誉,主要成就]
  - +娱乐人物 [艺名,获得荣誉,主要成就,经纪公司]
  - +政治人物 [获得荣誉,主要成就]
  - +虚拟人物 []
  - +体育人物 [运动项目,获得荣誉,主要成就]
  - +经济人物 [获得荣誉,主要成就]
  - +社会科学人物 [获得荣誉,主要成就]
  - +自然科学人物 [获得荣誉,主要成就]
- -地点 [外文名,所在地,面积,中文名,别名,位置]
  - +公共设施 []
  - +旅游景点 [主要景点,开园时间,闭园时间,邮政区码,地址,电话区码,门票,海拔,创建时间,分级,竣工日期]
  - +行政区域 [GDP,主要景点,民族,现任领导人,著名高校,下辖地区,方言,邮政区码,知名企业,车牌代码,火车站,电话区码,创建时间,机场,时区,人口,政府驻地,知名产业,特产,名人]
  - +地形地貌 [气候]

# 知识分类

## ■领域知识

```
股票:  tags [行业，地区，板块，股票种类]
{
    基本属性: {
        股票代码（ID）[String] (六位整形数字):
        股票种类      [String] (A B H N S股):
        股票简称      [String]
        股票英文简称  [String]:
        上市日期      [Date]   (1980.01.01-now):
        上市地点      [String]:
        上市板        [String]:
        交易币种      [String]:
        股票面值      [double] (>=0):
        摘牌日期      [Date]   (1980.01.01-now):
    }
```

```
发行属性: {
    公司名称:      [公司实体 Id]
    成立日期       [Date]:
    上市日期       [Date]:
    发行数量(万股)     [Double]
    发行价格(元)   [Double]:
    发行市盈率     [Double]:
    预计募资(万元)     [Double]:
    实际募资(万元)     [Double]:
    主承销商       [String]:
    上市保荐人     [String]:
}
```

# 知识分类

## ■事实性知识

- (乔布斯，CEO，苹果）
- (中华人民共和国，首都，北京）

## ■主观性知识

“我今年天让入手诺基亚5800，把玩不到24小时，目前感觉5800屏幕很好，操作也很方便，通话质量也不错，但是外形有些偏女性化，不适合男生。这些都是小问题，最主要的问题是电池不耐用，只能坚持一天，反正我觉得对不起这个价格。”

- 外形
- 电池

- 屏幕
- 操作
- 通话质量

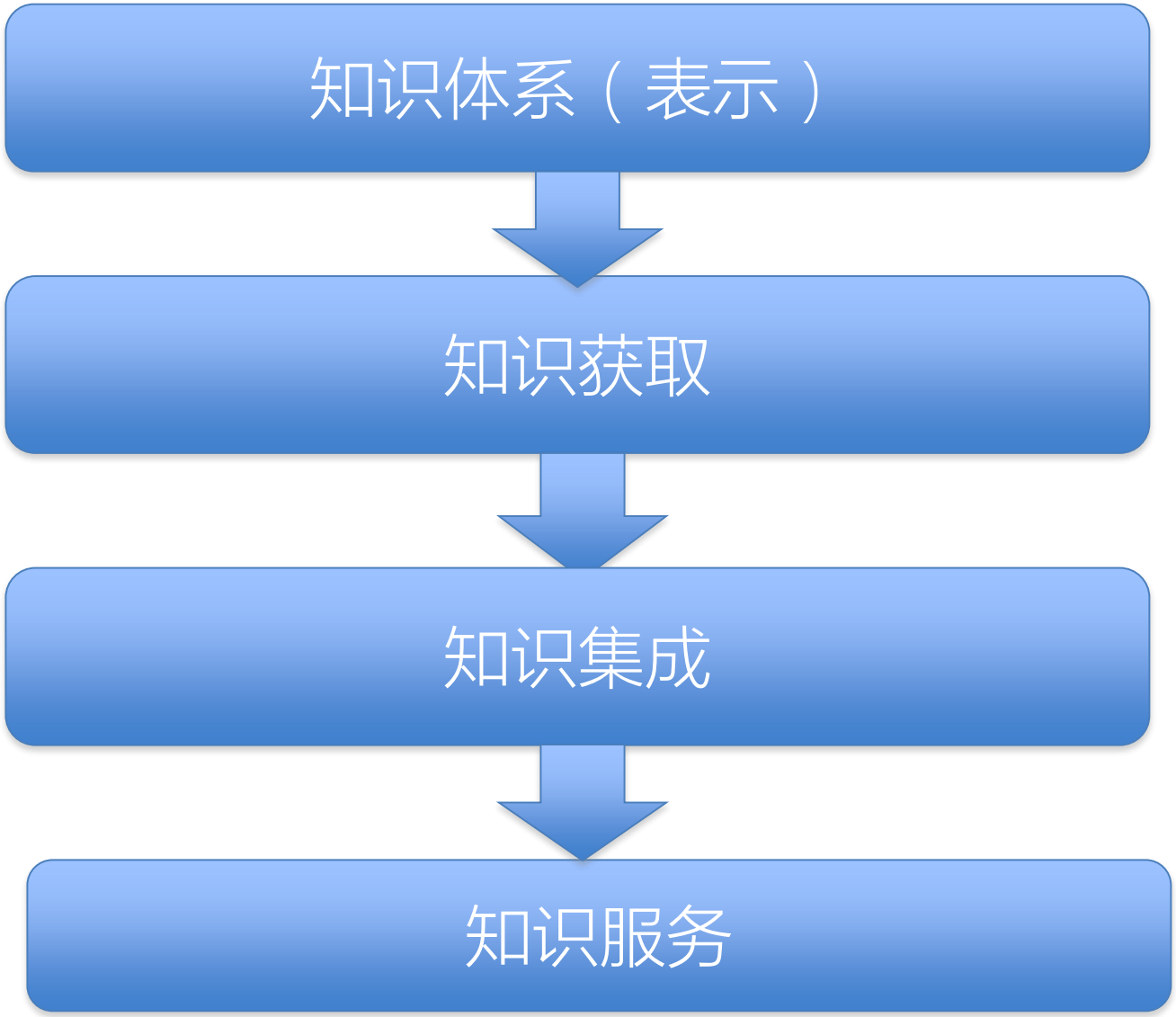
# 知识分类

- 场景知识
  - 订机票的步骤
  - 红烧肉的做法
- 语言知识
  - (乔丹，SameAs， 佐敦)
  - (乔丹，SameAs, Jordan)
  - (Microsoft， SameAs, MS)
  - (hasFounded, SameAs, isFounderof)
- 常识知识 (Common-sense Knowledge)
  - hasAbility(鸟，飞）， hasAbility(人，说话）
  - hasProperties(水，透明）， hasShape(球，圆的）
  - moreHeavy(大象，小狗）
  - Mother(x,y) ^ Brother(z,x) ➔ Uncle(z,y)

# 提纲

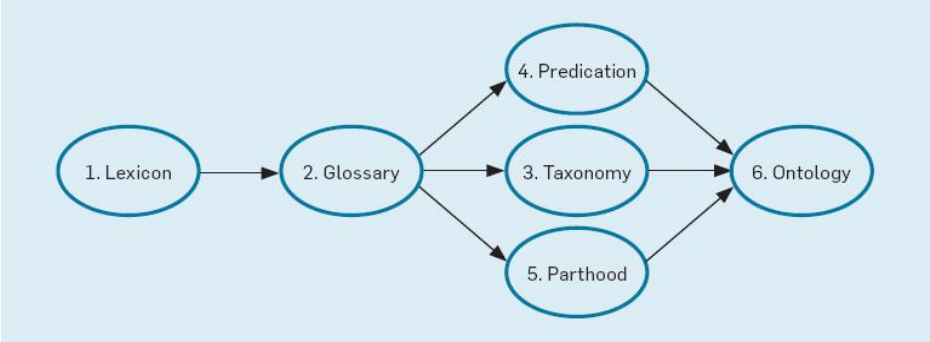
- 课程介绍
- 知识图谱发展历史
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用

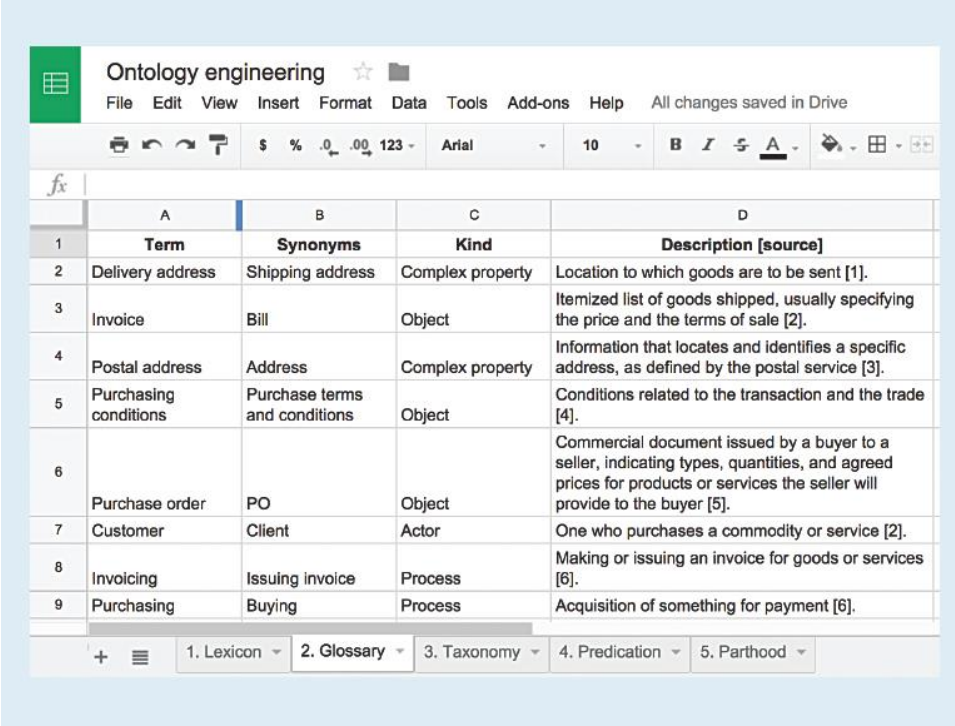
# 知识图谱中的四个关键问题



# 生命周期—领域知识建模

- 输入：
  - 领域 (医疗、金融...)
  - 应用场景
- 输出：领域知识本体
  - 领域实体类别体系
  - 实体属性
  - 领域语义关系
  - 语义关系之间的关系
- 关键技术：
  - Ontology Engineering



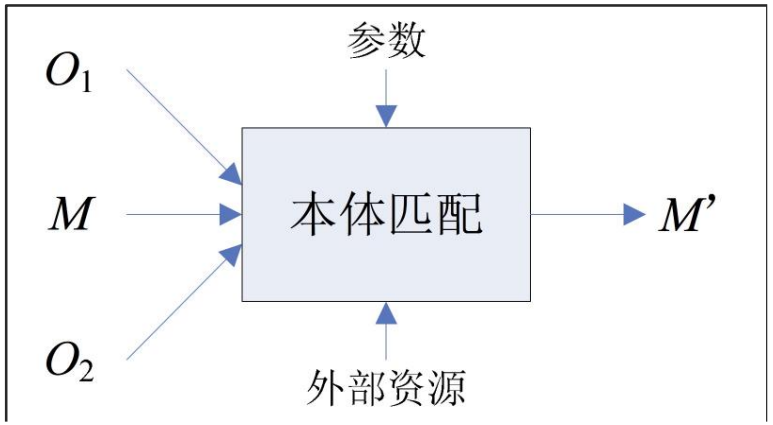

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Term | Synonyms | Kind | Description [source] |
| 2 | Delivery address | Shipping address | Complex property | Location to which goods are to be sent [1]. |
| 3 | Invoice | Bill | Object | Itemized list of goods shipped, usually specifying the price and the terms of sale [2]. |
| 4 | Postal address | Address | Complex property | Information that locates and identifies a specific address, as defined by the postal service [3]. |
| 5 | Purchasing conditions | Purchase terms and conditions | Object | Conditions related to the transaction and the trade [4]. |
| 6 | Purchase order | PO | Object | Commercial document issued by a buyer to a seller, indicating types, quantities, and agreed prices for products or services the seller will provide to the buyer [5]. |
| 7 | Customer | Client | Actor | One who purchases a commodity or service [2]. |
| 8 | Invoicing | Issuing invoice | Process | Making or issuing an invoice for goods or services [6]. |
| 9 | Purchasing | Buying | Process | Acquisition of something for payment [6]. |

# 生命周期—知识获取

- 输入：
  - 领域知识本体
  - 海量数据：文本、垂直站点、百科
- 输出：领域知识
  - 实体集合
  - 实体关系/属性
- 主要技术：
  - 信息抽取
  - 文本挖掘

# 生命周期—知识集成

- 输入：
  - 抽取出来的知识
  - 现有知识库
  - 知识本体

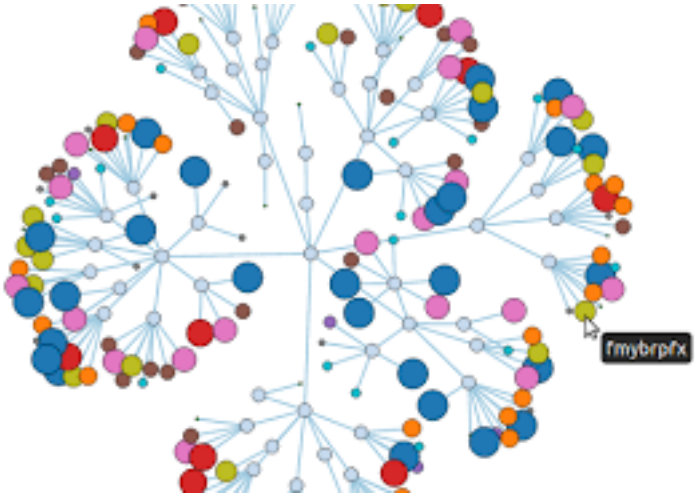



- 输出：
  - 知识置信度
  - 统一知识库

- 关键技术：
  - Ontology Matching
  - Entity Linking

# 生命周期—知识存储/查询/推理

- 输入：
  - 大规模知识库知识
- 输出：
  - 知识库存储/查询/推理服务
- 主要技术：
  - 知识表示
  - 知识查询语言
  - 存储/检索引擎
  - 推理引擎




SPARQL

# 知识图谱的生命周期

- 知识建模
  - 建模领域知识结构
- 知识获取
  - 获取领域内的事实知识
- 知识集成
  - 估计知识的可信度，将碎片知识组装成知识网络
- 知识存储/查询/应用
  - 提供高性能知识服务

# 提纲

- 课程介绍
- 知识图谱发展历史
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的现有应用

# 通用知识图谱

| 类别 | 名称 | 其他 |
| --- | --- | --- |
| 人工构建 | Cyc | http://www.cyc.com/platform/researchcyc |
| | WordNet | wordnet.princeton.edu |
| 基于维基百科 | DBPedia | dbpedia.org |
| | YAGO | yago-knowledge.org |
| | Freebase | freebase.com |
| | WikiTaxonomy | http://www.h-its.org/english/research/nlp/download/wikitaxonomy.php |
| | BabelNet | babelnet.org |
| 开放知识抽取 | KnowItAll | openie.cs.washington.edu |
| | NELL | rtw.ml.cmu.edu |
| | Probase | http://research.microsoft.com/en-us/projects/probase/ |
| 企业知识图谱 | 百度知心，搜狗知立方 | www.baidu.com, www.sougou.com |
| | Google KG，MS sotori | google.com, bing.com |

# 代表性知识图谱

- **人工构建知识图谱**
  - **WordNet**
  - CYC
- **基于Wikipedia的知识图谱**
  - Yago
  - DBPedia
  - Freebase
- **文本抽取知识图谱**
  - NELL

# WordNet

WordNet
A lexical database for English

- WordNet是什么?
  - 一部在线词典数据库系统，采用了与传统词典不同的方式，即按照词义而不是词形来组织
  - 词语被聚类成词义簇(synset)，词义之间通过语义关系连接成大的概念网络

- 1985年由普林斯顿大学认知科学实验室建立

**1978年Miller“自动化词典”**

**1985年WordNet开始实际运作**

**1986年Bienkowski用LISP语言写了Grinder的第一个版本**

**20世纪70—90年代添加词表并对词进行分类**

**1989年WordNet：词典浏览器->自足的词汇数据库**

**1991年WordNet1.0正式公布**

**WordNet 3.0**

# WordNet包含的知识

**描述的对象**

- compound（复合词）
- phrasal verb（短语动词）
- collocation（搭配词）
- idiomatic phrase（成语）
- word（单词）

**对象之间的语义关系**

- 同义反义关系（synonymy，antonymy）
- 上下位关系（hypernym, hyponymy, troponymy）
- 部分整体关系（entailment，meronymy）

**部分句法信息**

- 简单的动词基本句式信息（Verb Sentence Frames）

# WordNet的核心概念

- Synset： WordNet 将英语的名词、动词、形容词、和副词组织为Synsets，每一个Synset表示一个基本的词汇概念

- 概念关系
  - 同义关系
  - 反义关系
  - 上位关系
  - 下位关系
  - 整体关系（名词）
  - 部分关系（名词）
  - 蕴含关系（动词）
  - 因果关系（动词）
  - 近似关系（形容词）

**newspaper词义的上位synsets**

newspaper, paper
=> press, public press
=> print media
=> medium
=> instrumentality
=> artifact, artefact
=> whole, unit
=> object, physical object
=> physical entity
=> entity

# WordNet组织示例：上下位、同义

Concept hierarchy

{equipment}

Concept (“synset”)

Synonymous words for the concept

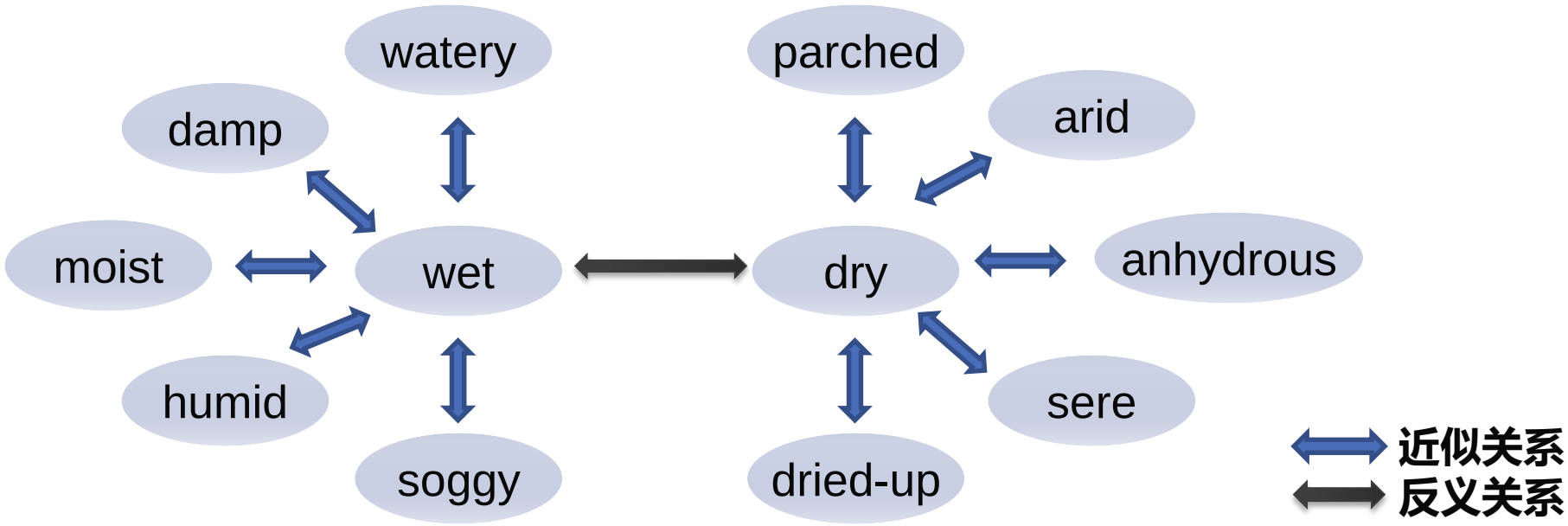
{ camera, photographic camera}

{ camera, television camera}

Gloss describing the concept

“television equipment consisting of a lens system that focuses an image on a photosensitive mosaic”

# WordNet组织示例：近义、反义




**基于反义、近义组织的形容词synset**

# WordNet的构建方法

- 人工构建+机器辅助（后续有很多自动构建技术研究）

Brown语料库

语料库

+

已有词表

→

WordNet
词汇

Laurence Urdang (1978)
《同义反义小词典》

Urdang(1978)
《 Rodale同义词词典》

Robert Chapmand (1977)
《罗杰斯同义词词林》

美国海军研究与发展中心的Fred Chang的词表，15%的重合词语

Ralph Grishman和他在纽约大学的同事的一个词表，包含39143个词，这个词表实际上包含在著名的COMLEX词典中。WordNet当时词表与该词表重合率为74%（1993年）。

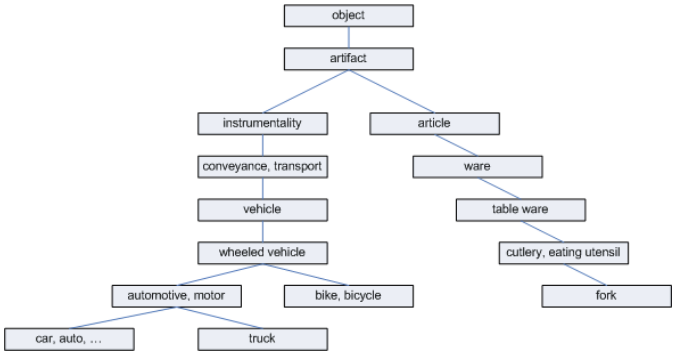
# WordNet规模

| POS | Unique Terms | Synsets | Total Word-Sense Pairs |
|---|---|---|---|
| Noun | 109195 | 75804 | 134716 |
| Verb | 11088 | 13214 | 24169 |
| Adjective | 21460 | 18576 | 31184 |
| Adverb | 4607 | 3629 | 5748 |
| Totals | 146350 | 111223 | 195817 |

# WordNet的应用

- WordNet在自然语言处理中被广泛应用
  - 作为词义消歧的目标知识库
  - 作为高质量的Taxonomy
  - 用于计算语义相似度

| Word 1 | Word 2 | lin | wup | path | remarks |
|---|---|---|---|---|---|
| genuineness | genuine | 0 | 0 | 0 | needs explanation |
| Valid | reality | 0 | 0 | 0 | needs explanation |
| Painter | paint | 0.15 | 0.62 | 0.09 | Good Enough |
| Really | fact | 0 | 0 | 0 | needs explanation |
| Real | reality | 0.1 | 0.3 | 0.09 | Good Enough |
| really | reality | 0 | 0 | 0 | needs explanation |
| paint | painting | 0.3 | 0.8 | 0.12 | Good Enough |




WordNet: a lexical database for English

GA Miller - Communications of the ACM, 1995 - dl.acm.org

Abstract Because meaningful sentences are composed of meaningful words, any system that hopes to process natural languages as people do must have information about words and their meanings. This information is traditionally provided through dictionaries, and

被引用次数：9323　相关文章　所有 34 个版本　引用　保存　更多

# WordNet综述

| | |
|---|---|
| Content | Adjectives, verbs, nouns and adverbs of the English language |
| Format | Visualization tool<br>data downloadable in Prolog-like format |
| Main strength | High quality lexicon for English |
| Technique | Manual |
| Size | Words: 155k |
| | Senses: 117k |
| | Word-sense pairs: 207k |
| License | Proprietary, free use |
| Reference | [Miller, Comm ACM 1995] |
| URL | http://wordnet.princeton.edu |

一个高质量英文电子词典和语言本体

# 代表性知识图谱

- **人工构建知识图谱**
  - WordNet
  - **CYC**
- **基于Wikipedia的知识图谱**
  - Yago
  - DBPedia
  - Freebase
- **文本抽取知识图谱**
  - NELL

# Cyc

- Cyc 是一个由Douglas Lenat 在1984年启动的人工智能项目，其目的是构建一个完整的、机器可使用的本体体系和人类常识知识库
  - 500万条知识
  - 50万概念
- OpenCyc 是其开放出来免费供大众使用的一部分知识
  - 24万概念
  - 200万条知识
- ResearchCyc是提供Research Liscence供研究使用的完整版

# Cyc构建方法--人手工构建

- 1986年, Doug Lenat估计整个Cyc需要包括25万条规则，耗费350人年
- 近年来，也开始使用自动构建方法，从自然语言中抽取知识
- 2008年开始，Cyc开始将其资源与Wikipedia、DBpedia和Freebase等资源开始建立Link

# 知识表示语言--CycL

- CycL: 自己设计的知识表示语言
  - 基于First Order Predicate Logic (FOPL)
  - 采用类似于LISP语言的语法
- Constants: 概念名字
  - Cyc，DougLenat，BaseKB，EnglishWord
- Variables
  - 以?开头
  - ?TYPE
- Predicates
  - isa， genls， comment
- Functions
  - (FruitFn AppleTree) → Apple

# 知识表示语言-CycL

- Formulas
  - 形式为(predicate arg1 arg2 ...)的表达式
  - 5种真值: {true, default true, false, default false, unknown}
  - 示例:
    - *(isa Dog BiologicalSpecies)*
    - *(genls Dog Carnivore)*
    - *(skillCapableOf LinusVanPelt PlayingAMusicalInstrument performedBy)*
- Logical connectors
  - not, and, or, implies
- Quantifiers
  - 任一, 存在

# Cyc中包含的知识

- Cyc中包含的大部分知识都是常识
  - Every tree is a plant(所有树都都是植物)
  - Plants die eventually（植物都会死）
- 核心概念：
  - Individuals: #$BillClinton, #$France
  - Collections:
    - #$Tree-ThePlant（包含了所有的树）
    - #$EquivalenceRelation(包括所有的等同关系)

# Cyc中包含的知识-Sentences

- 事实：通过CycL sentences 来表示
  - (#$isa #$BillClinton #$UnitedStatesPresident)
    - Bill Clinton is a President of the United States.
  - (#$genls #$Tree-ThePlant #$Plant)
    - All trees are plants.
- 规则: 包含变量(以?开头)的句子
  - 如果一个对象是SUBSET的成员，同时SUBSET是SUPERSET的子类，那么OBJ是SUPERSET的成员

```
(#$implies
    (#$and
      (#$isa ?OBJ ?SUBSET)
      (#$genls ?SUBSET ?SUPERSET))
    (#$isa ?OBJ ?SUPERSET))
```

# The Cyc Ontology and Content

Tens of thousands of Predicates

**between, owns, isa, beliefs**

Hundreds of thousands of Concepts



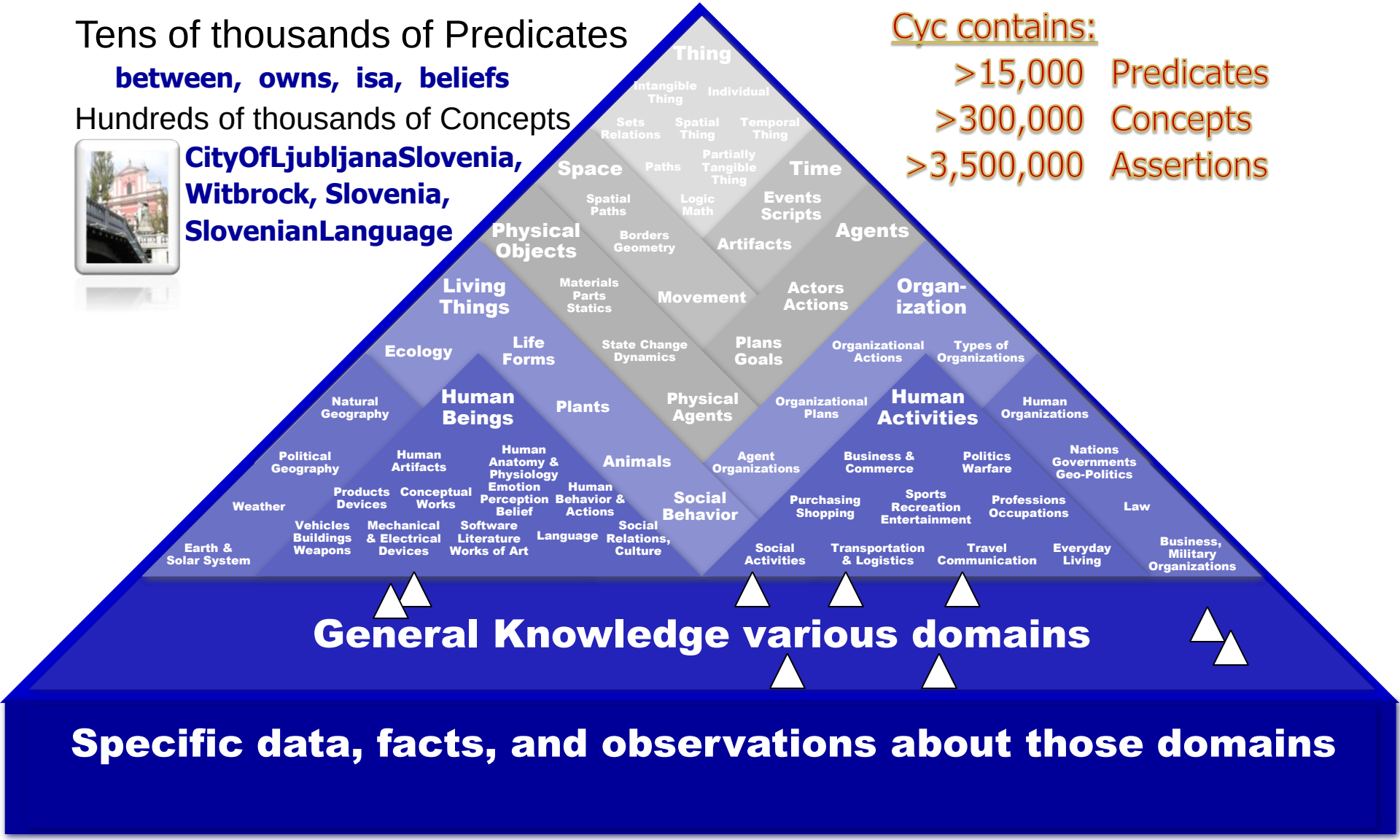
**CityOfLjubljanaSlovenia, Witbrock, Slovenia, SlovenianLanguage**

Cyc contains:

>15,000 Predicates

>300,000 Concepts

>3,500,000 Assertions



# 推理引擎与应用

- Cyc提供了非常多的推理引擎，支持演绎推理和归纳推理；同时也提供了扩展推理机制的模块
- 支持对自然语言的解析，对自然语言文本中的名词、动词等映射到概念和frame下，可以利用Cyc定义的规则进行推理

| | |
|---|---|
| English Words | 18,796 |
| Syntactic Frame Links | 23,336 |
| Single-word Denotation Mappings | 27,681 |
| Multi-word Phrase Denotation Mappings | 44,298 |
| Verbal Semantic Frame Links | 3,701 |
| Noun Semantic Frame Links | 2,578 |
| WordNet 2.0 Links | 11,322 |
| Names<br>(Includes chemical symbols,<br>person/place/organization names, acronyms, etc.) | 100,811 |
| Predicate-based Phrasal Links<br>(genTemplates for paraphrase) | 9,637 |

# 基于Cyc的自然语言解析示例

As of Feb (#$February). 24 (24), Air Force (#$UnitedStatesAirForce) officials (#$PublicOfficial #$OrganizationRepresentative) reported (#$RegisteringAComplaint #$Reporting) that personnel (#$Employee) in the area (#$Area 0 #$FieldOfStudy #$Region-Underspecified) numbered (#$Counting) close to 8,000 (8000). The 100 (100) aircraft (#$AirTransportationDevice) based (#$Base-Support #$MilitaryBase-Grounds #$BaseOfLandProtrusion #$NitrogenBase #$ChemicallyBasicSubstance) in Saudi Arabia (#$SaudiArabia) for patrols (#$Patrolling) over southern Iraq ((#$SouthernRegionFn #$Iraq)) has (#$possesses) seen (#$VisualPerception #$MeetingSomeone #$sees) the addition (#$DoingAddition) of two (2) dozen (12) F-15 (#$FighterPlane-F15) and F-16 fighter jets (#$FighterPlane-F16) to Bahrain (#$Bahrain-TheIsland #$Bahrain (#$CityNamedFn Bahrain #$Bahrain)). The Air Force (#$UnitedStatesAirForce) has (#$possesses) also authorized (#$GrantingPermission) the dispatch (#$SendingSomething) of 12 (12) F-117 (#$FighterPlane-F117) stealth (#$DodgeStealthCar) fighter jets (#$JetOfFluid #$JetPropelledAircraft) to Kuwait (#$CityOfKuwaitKuwait (#$ProperSubcollectionNamedFn-Ternary kuwait #$Individual 34057665-f4ed-11d9-9bea-0002b3a85b0b) #$Kuwait), three (3) B-1 bombers (#$B-1-Bomber) to Bahrain (#$Bahrain-TheIsland #$Bahrain (#$CityNamedFn Bahrain #$Bahrain)) and 14 (14) B-52 (#$B-52-Bomber) bombers (#$SubmarineSandwich #$BomberPlane #$Bomber) to the island (#$Island) of Diego Garcia. It also has (#$possesses) diverted (#$AmusingSomeone #$DivertingSomething) dozens (#$Dozens-Quant 12) of support (#$SupportingSomething #$ShowingSupportForSomeone (#$SubcollectionOfWithRelationFromTypeFn #$PartiallyTangible #$supportingObject #$SupportingSomething)) aircraft (#$AirTransportationDevice) to the region (#$TheRegion) for refueling (#$Refueling (#$MakingAvailableFn #$CombustibleFuelSubstance)),

# Cyc综述

一个人工撰写的常识知识库

| | Cyc |
|---|---|
| Content | Common sense knowledge, axioms |
| Main strength | Huge ontology, with tools |
| Technique | Manual |
| License | proprietary, OpenCyc is Apache License V2.0 |
| Entities | 500k |
| Assertions | 5m |
| Relations | 15k |
| Tools | Reasoner, NL tool |
| URL | http://cyc.com |
| References | [Lenat, Comm. ACM 1995] |

# 代表性知识图谱

- **人工构建知识图谱**
  - WordNet
  - CYC
- **基于Wikipedia的知识图谱**
  - Yago
  - DBPedia
  - Freebase
- **文本抽取知识图谱**
  - NELL

# Wikipedia

- 免费的在线百科全书
  - 2001年开始
  - crowdsource的方式构建
  - 目标：构建全世界最大的百科全书

基于Wikipedia的知识库

高质量数据源
500万概念
多语言
富含丰富语义结构的文档：
Infobox，table，list，category…

# Wikipedia: 文档结构

标题=概念

唐太宗

唐太宗李世民（598年1月28日－649年7月10日[1][2][3]），中国唐朝第二任皇帝。祖籍陇西郡成纪县（今甘肃省天水市秦安县北），生于陕西武功县，626年至649年在位。父亲是唐高祖李渊，母亲是窦皇后

概念文本描述

Infobox:以(属性，值)对形式呈现的信息表格

每个页面有多个类别，类别组成Taxonomy

- 分类：598年出生，649年逝世　唐朝皇帝

# 出发点

- 基于Wikipedia的知识库都基于几乎相同的思路：
  - **从Wikipedia丰富的半结构化信息中挖掘知识**
  - 包括：Infobox，Category，超链接, Table，List...
- 不同之处在于
  - 如何处理有歧义的属性映射
  - 如何构建知识库的Taxonomy

| | |
|---|---|
| 在世 | 598年1月28日 - 649年7月10日（52岁） |

| | |
|---|---|
| 出生 | 隋文帝仁寿四年604年 |

BirthDate（李世民，598年）
BirthDate（李元吉，604年）

# 这些知识库具有相同的模型

- 一个知识库包含一个集合的实体
  - 猫王、李世民、赵高、唐朝、分封制...
- 实体被划分到不同的类别中
  - 歌手(猫王)，皇帝(李世民), 朝代(唐朝), 制度(分封制)
- 类别通过上下位等关系相互关联
  - SubClassOf(歌手，人)， SubClassOf(皇帝，人)
- 实体和类别都通过属性和相互之间的关系来描述
  - BirthDate（李世民，598年）, Has(歌手，歌曲)
- 关系可以通过蕴含关系来进行推理
  - 歌曲→作品， 收购 → 持有

# DBPedia

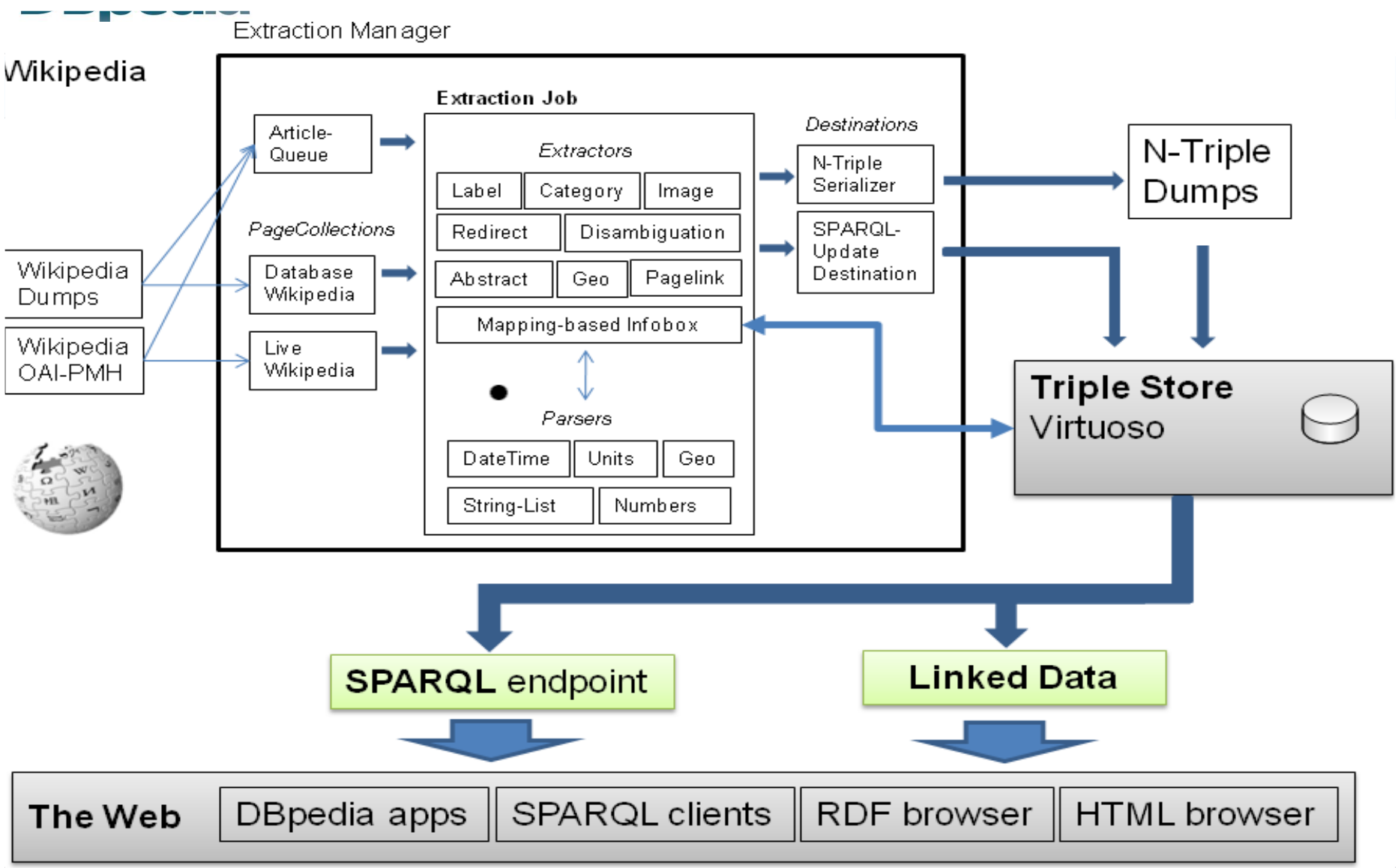
- 2007年开始，其主要目标是构建一个社区，通过社区成员来**定义和撰写准确的抽取模板，从维基百科中抽取结构信息，并将其发布到Web上**
- 社区通过人工的方式构建了Taxonomy
  - 280个类别
  - 覆盖约50%的维基百科实体

DBpedia

# 抽取方法

- DIEF - DBpedia Information Extraction Framework
  - 目标：抽取WIkipedia中的结构化信息
  - 方法：基于属性mapping的Infobox抽取，Raw Infobox Extraction, Feature Extraction, Statistical Extraction
  - 编程语言：Scala & Java
- DBPediaLive：持续保持与Wikipedia的同步
  - 2013年六月，英语维基百科有将近330万次编辑(每分钟越77次)

# 抽取框架图


Extraction Manager
Wikipedia
Extraction Job
Article-Queue
PageCollections
Database Wikipedia
Live Wikipedia
Wikipedia Dumps
Wikipedia OAI-PMH
Extractors
Label
Category
Image
Redirect
Disambiguation
Abstract
Geo
Pagelink
Mapping-based Infobox
Parsers
DateTime
Units
Geo
String-List
Numbers
Destinations
N-Triple Serializer
SPARQL-Update Destination
N-Triple Dumps
Triple Store
Virtuoso
SPARQL endpoint
Linked Data
The Web
DBpedia apps
SPARQL clients
RDF browser
HTML browser

# 抽取知识示例

## About: Harry Froboess

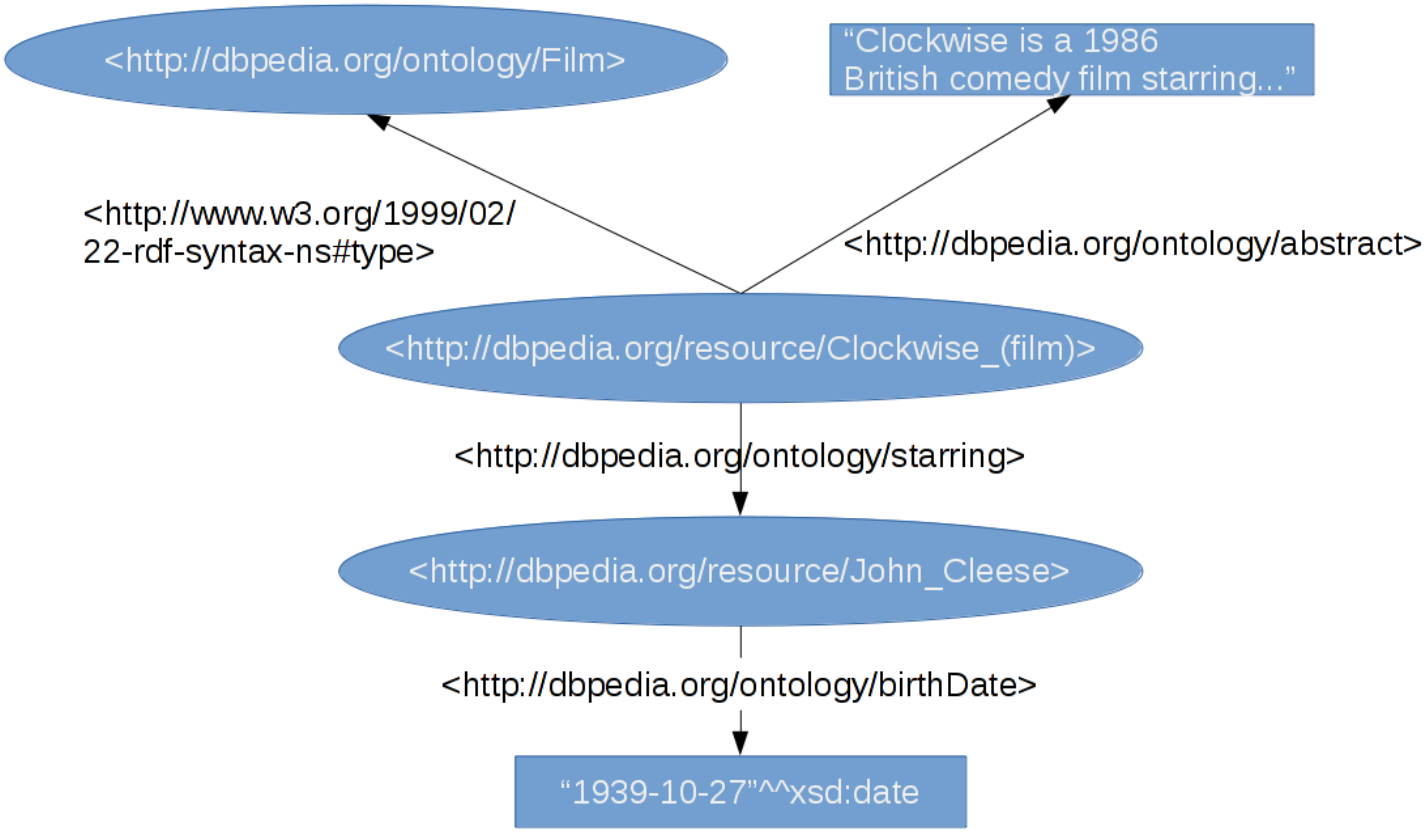
An Entity of Type : agent, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

Friedrich Harald August Froboess (1899–1985) born on 10.23.1899 in Dresden, Germany, was a German stunt diver, and high diver.

| Property | Value |
|---|---|
| dbpedia-owl:abstract | ▪ Friedrich Harald August Froboess (1899–1985) born on 10.23.1899 in Dresden, Germany, was a German stunt div |
| dbpedia-owl:alias | ▪ Froboess, Harry Arias |
| dbpedia-owl:birthDate | ▪ 1899-01-01 (xsd:date) |
| dbpedia-owl:birthName | ▪ Harry Arias Froboess |
| dbpedia-owl:birthPlace | ▪ dbpedia:Bern<br>▪ dbpedia:Dresden<br>▪ dbpedia:Germany<br>▪ dbpedia:Switzerland<br>▪ dbpedia:United_States |
| dbpedia-owl:birthYear | ▪ 0023-01-01 (xsd:date)<br>▪ 1899-01-01 (xsd:date) |
| dbpedia-owl:deathDate | ▪ 1899-01-01 (xsd:date) |
| dbpedia-owl:deathPlace | ▪ dbpedia:Switzerland |
| dbpedia-owl:deathYear | ▪ 0012-01-01 (xsd:date)<br>▪ 1899-01-01 (xsd:date) |
| dbpedia-owl:occupation | ▪ dbpedia:Diving<br>▪ dbpedia:Stunt_performer<br>▪ dbpedia:Harry_Froboess__1 |
| dbpedia-owl:spouse | ▪ dbpedia:Berlin<br>▪ dbpedia:Switzerland |

# 基于RDF表示抽取出来的知识

- 支持复杂的结构化query，SPARQL语言查询
- 支持与Web上其他数据集的链接和集成



# DBPedia综述

| | |
|---|---|
| Content | Entities of public interest |
| Format | RDF, API, SPARQL |
| Sources | Wikipedia, YAGO/WordNet |
| Main strengths | Focus on coverage,<br>interlinking with other data sets |
| Technique | Extraction from Wikipedia + manual supervision by the community |
| Size | Entities: 3.5m (in manual taxonomy: 1.7m)<br>Facts: 670m<br>Attributes: 9k (manually defined: 1k)<br>Manual Classes: 280 |
| License | CC-BY-SA & GNU FDL |

一个部分结构化的Wikipedia，
社区人工构建Taxonomy

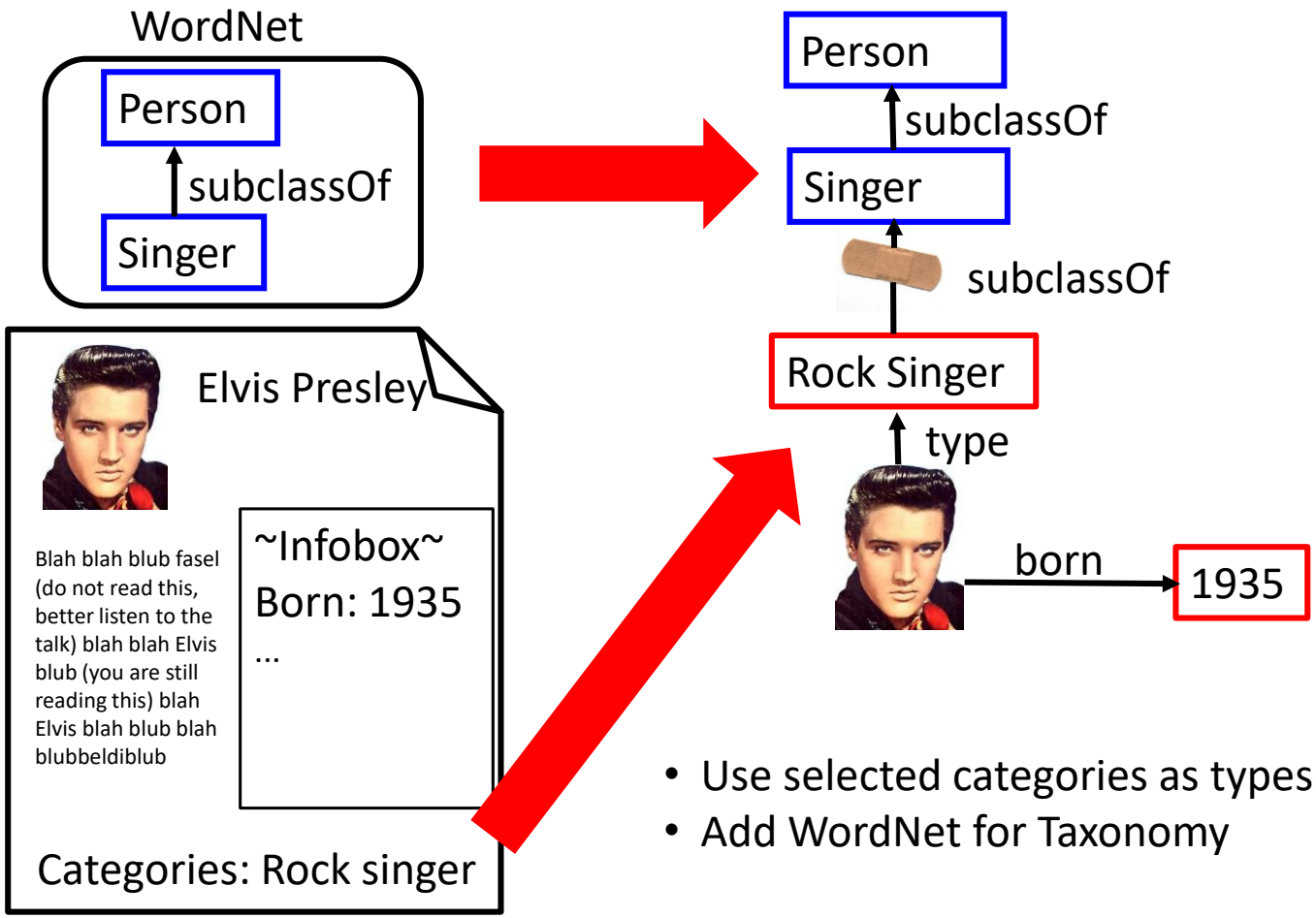
# Yago

- 德国马普研究所从2007年开始的一个项目
- 融合WordNet和Wikipedia
  - 从Wikipedia的结构中抽取信息：Infoboxes,类别...
  - 包括时间和地点标注
  - 人工采样评估
  - >1亿事实和100种关系

yago select knowledge = + PRINCETON UNIVERSITY WordNet A lexical database for English

# Yago Taxonomy构建

- 使用WordNet的Taxonomy作为基础
- 将Wikipedia中的类别加入到WordNet中




- Use selected categories as types
- Add WordNet for Taxonomy

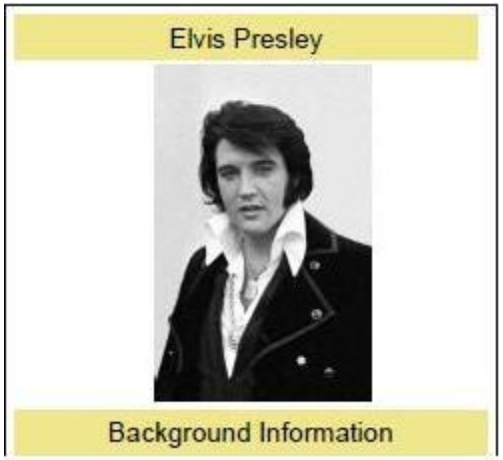
# Yago语义关系

- 人工定义了100多种语义关系
  - wasBornOnDate，locatedIn，hasPopulation

- 抽取方法：采用手写的规则抽取
  - Infobox Harvesting:信息框
  - Word-Level Techniques:重定向页
  - Category Harvesting:类别信息抽取
  - Type Extraction: 维基类别,WordNet类别

# 抽取示例

Infobox




Died: August 16, 1977

Attribute Map

| Attribute | Relation | Inverse | Manifold | Indirect |
|---|---|---|---|---|
| | | … | | |
| Died | diedOnDate | | | |
| | | … | | |

Relation Map

| Relation | Domain | Range |
|---|---|---|
| | … | |
| diedOnDate | person | yagoDate |
| | … | |

**Elvis Presley** **diedOnDate** **August 16, 1977**

# 抽取准确率

| Relation | Total Facts | Evaluated | Accuracy |
|---|---|---|---|
| actedIn | 12636 | 69 | 97.36% +- 2.64% |
| created | 225563 | 94 | 98.04%+-1.96% |
| graduatedFrom | 15583 | 57 | 96.84%+-3.16% |
| type | 8414398 | 208 | 97.68%+-1.83% |
| subClassOf | 367040 | 339 | 93.42%+-2.67% |
| diedIn | 28834 | 88 | 97.91%+-2.09% |

# Yago综述

| | |
|---|---|
| Content | Entities of public interest |
| Format | TSV, RDF, XML, N3, Web Interface |
| Sources | Wikipedia, WordNet, Geonames |
| Main strength | Focus on precision, geotemporal annotations, multilingual |
| Precision | 95% |
| Technique | Extraction from Wikipedia + matching with WordNet & Geonames + consistency checks |
| Size | Entities: 3 m (+ geonames -> 10m)<br>Facts: 120m (+geonames -> 460m)<br>Relations: 100, Classes: 200k, Languages: 200 |
| License | Creative Commons BY-SA |

一个部分结构化的Wikipedia，
Taxonomy：WordNet+Wikipedia Categories

# Freebase

- Metaweb公司2000年开始构建，2010年被Google收购
- 从Wikipedia和其他数据源（如IMDB、MusicBrainz）中导入知识
- 核心想法：
  - 在Wikipedia中，人们编辑文章
  - **在Freebase中，人们编辑结构化知识**

# 用户是Freebase知识构建的核心

- 编辑实体
  - 创建实体
  - 将实体分到类别
  - 增加/修改属性/关系
  - 上传图片
- 编辑Schema
  - 定义新类别
  - 定义类别的属性
- Review
  - 验证知识准确性
  - 投票
  - 删除错误知识
- DataGame
  - 寻找别名
  - 抽取事件日期
  - 使用Yahoo图片搜索加入图片

**用户在Freebase中的作用**

# Freebase综述

| | |
|---|---|
| Content | Entities with public information |
| Format | API, RDF |
| Construction | by the community<br>data import from public sources |
| Sources | Wikipedia, Libraries, WordNet, MusicBrainz… |
| Main strength | free and large |
| Size | Facts: several millions |
| | Entities: 20 m |

一个大规模协同构建知识库，
整合Wikipedia，WordNet以及多个百科知识库
目前归Google所有

# 代表性知识图谱

- **人工构建知识图谱**
  - WordNet
  - CYC
- **基于Wikipedia的知识图谱**
  - Yago
  - DBPedia
  - Freebase
- **文本抽取知识图谱**
  - NELL

# NELL

Read the Web
Research Project at Carnegie Mellon University

- 2009年开始的CMU项目
- 输入：
  - 初始本体（~800类别和关系）
  - 每个谓词的一些实例（~10-20个种子实例）
  - web(~10亿页面, ClueWeb)
  - 间歇性的人工干预
- 任务：
  - 24 x 7 持续运行(从2010年开始)
  - 每天
    - 抽取更多知识来补充给定本体
    - 学习如何更好的构建抽取模型
- 结果：超过9千万实例（不同置信度）

# 抽取结果示例

## Recently-Learned Facts

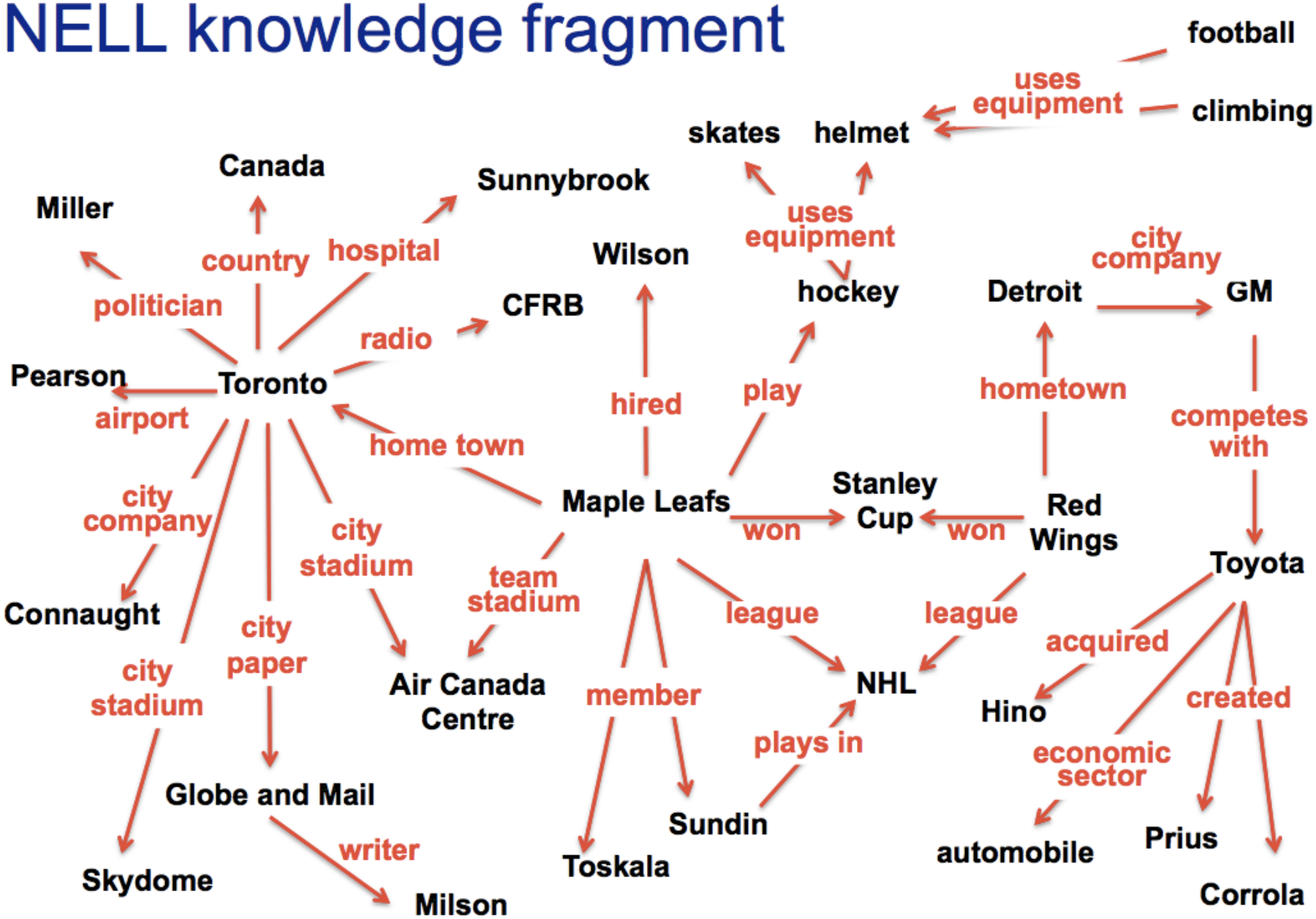
twitter

Refresh

| instance | iteration | date learned | confidence |
|---|---|---|---|
| rillito_river is a river | 1064 | 26-jun-2017 | 99.9 |
| james_dexter is a CEO | 1064 | 26-jun-2017 | 99.7 |
| richard_thompson is a Mexican person | 1064 | 26-jun-2017 | 100.0 |
| htc_droid_eris is a consumer electronic device | 1064 | 26-jun-2017 | 92.8 |
| radiant_snow is a weather phenomenon | 1066 | 15-jul-2017 | 99.8 |
| the_london_eye001 is a tourist attraction in the city london | 1069 | 03-aug-2017 | 100.0 |
| state_university is a sports team also known as michigan_state_university | 1067 | 21-jul-2017 | 100.0 |
| state_university is an organization also known as clemson | 1066 | 15-jul-2017 | 96.9 |
| irene_kirkaldy belongs to the religion seven_day_adventist | 1069 | 03-aug-2017 | 100.0 |
| kcal is a TV station in the city los_banos | 1069 | 03-aug-2017 | 100.0 |

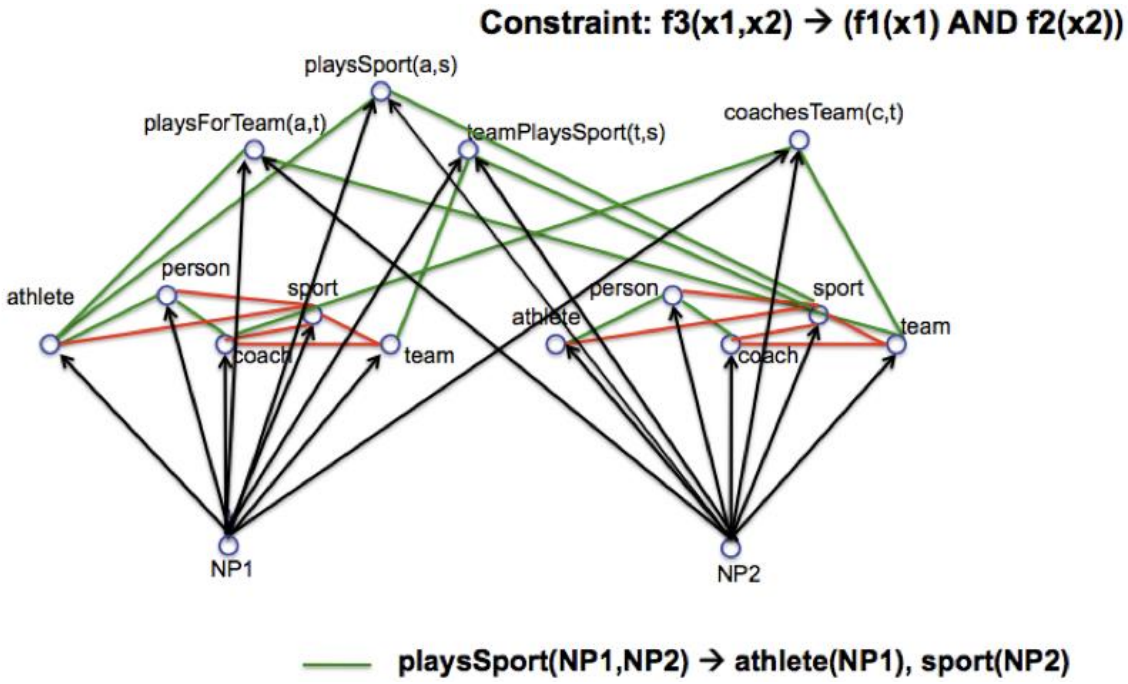
http://rtw.ml.cmu.edu

# NELL knowledge fragment



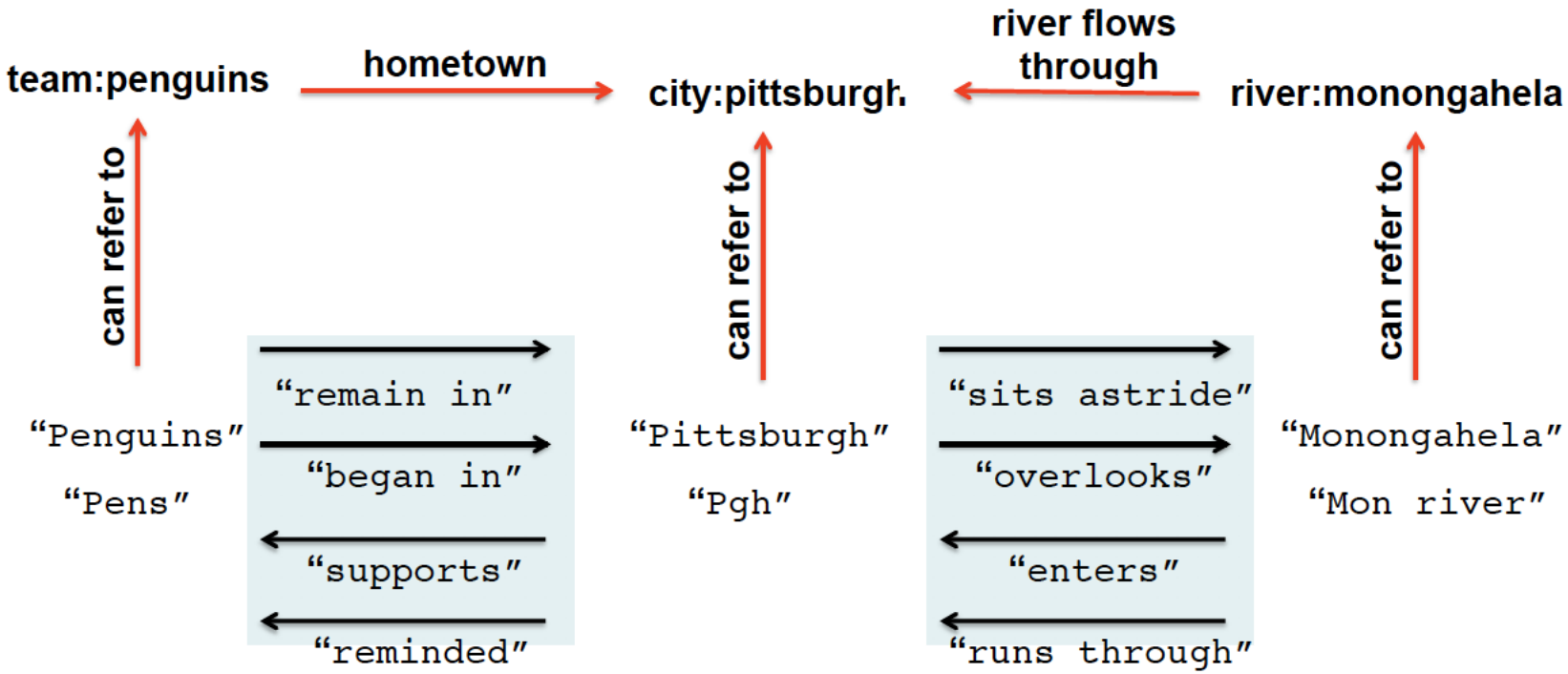
# 抽取步骤

- 把名词短语划分到给定类别
  - Entity Set Expansion: 基于Pattern的Bootstrapping
- 分类名词短语之间的语义关系
  - Coupling Learning



# 抽取步骤(cont.)

- 名词短语被映射到概念, 动词短语被映射到关系
  - Entity Linking, Entity Resolution

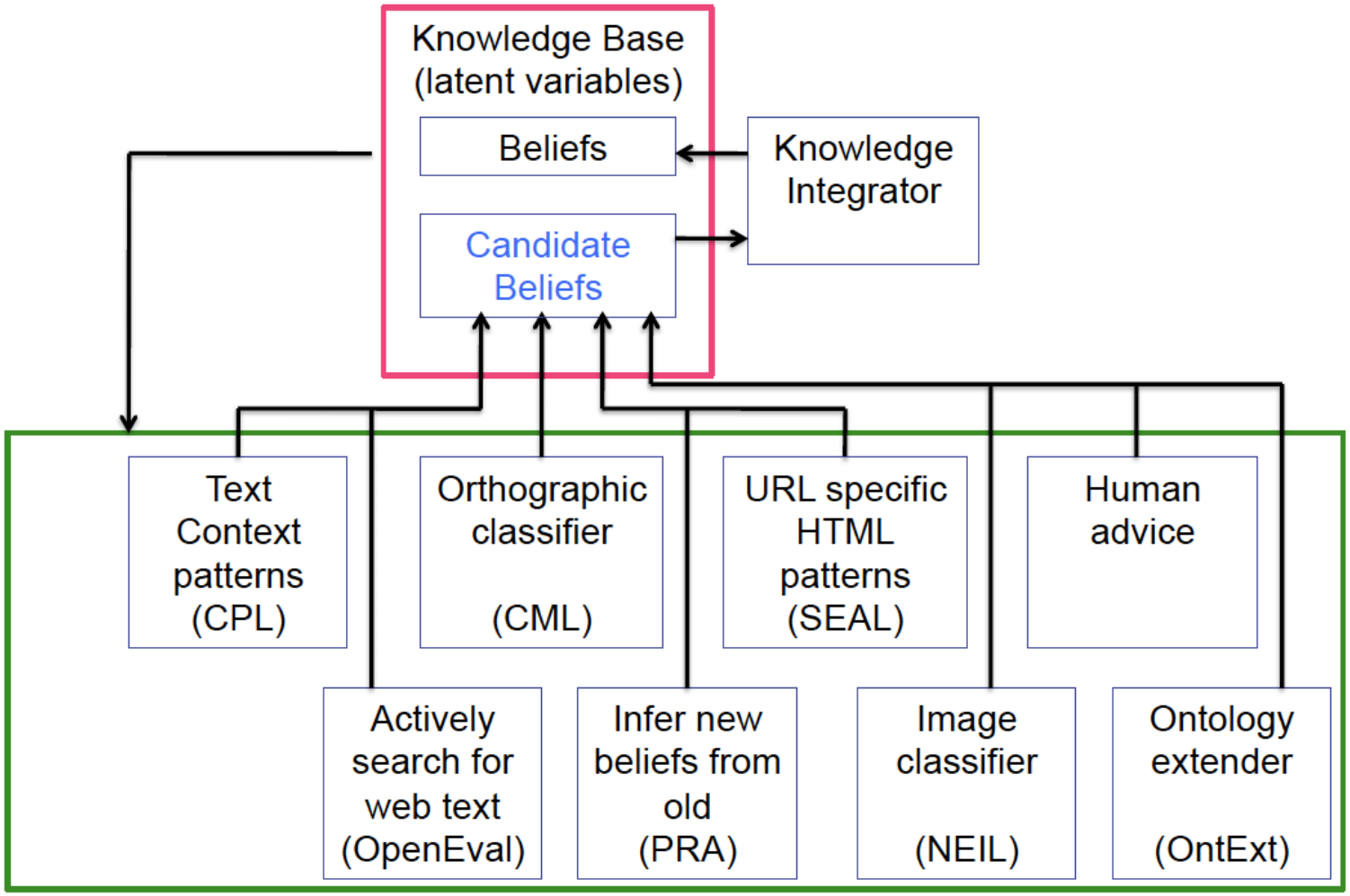


# 抽取步骤(cont.)

- 识别新的推理规则，用于发现新的关系实例
  - PathRank

```
0.95  athletePlaysSport(?x,basketball) :- athleteInLeague(?x,NBA)

0.93 athletePlaysSport(?x,?y) :-    athletePlaysForTeam(?x,?z)
                                    teamPlaysSport(?z,?y)

0.91  teamPlaysInLeague(?x,NHL) :- teamWonTrophy(?x,Stanley_Cup)

0.90 athleteInLeague(?x,?y):-   athletePlaysForTeam(?x,?z),
                                teamPlaysInLeague(?z,?y)

0.88 cityInState(?x,?y) :-  cityCapitalOfState(?x,?y),
                            cityInCountry(?y,USA)

0.62* newspaperInCity(?x,New_York) :-  companyEconomicSector(?x,media),
                                       generalizations(?x,blog)
```

# 完整的抽取框架


Knowledge Base
(latent variables)
Beliefs
Candidate
Beliefs
Knowledge
Integrator
Text
Context
patterns
(CPL)
Orthographic
classifier
(CML)
URL specific
HTML
patterns
(SEAL)
Human
advice
Actively
search for
web text
(OpenEval)
Infer new
beliefs from
old
(PRA)
Image
classifier
(NEIL)
Ontology
extender
(OntExt)

# NELL综述

| | |
|---|---|
| Content | Entities mentioned on Web pages |
| Format | TSV |
| Construction | by a perpetual extractor |
| Sources | The Web |
| Main strength | Not limited to a specific source |
| Size | Facts: 800k |
| | Categories & relations: 633 |
| Reference | [Carlson, AAAI 2010] |
| URL | http://rtw.ml.cmu.edu/ |

一个基于文本信息抽取技术持续不断更新的知识库

# 代表性知识图谱综述

- 人工构建知识图谱
  - WordNet： 英文电子词典
  - CYC：常识知识库
- 基于Wikipedia的知识图谱
  - Yago： Wikipedia+WordNet
  - DBPedia：基于社区抽取Wikipedia结构化信息
  - Freebase：知识编辑社区协同构建
- 文本抽取知识图谱
  - NELL：从文本中持续自动抽取海量知识
- 还有许多其他知识库没有覆盖：Wikidata、BabelNet、OpenMind...

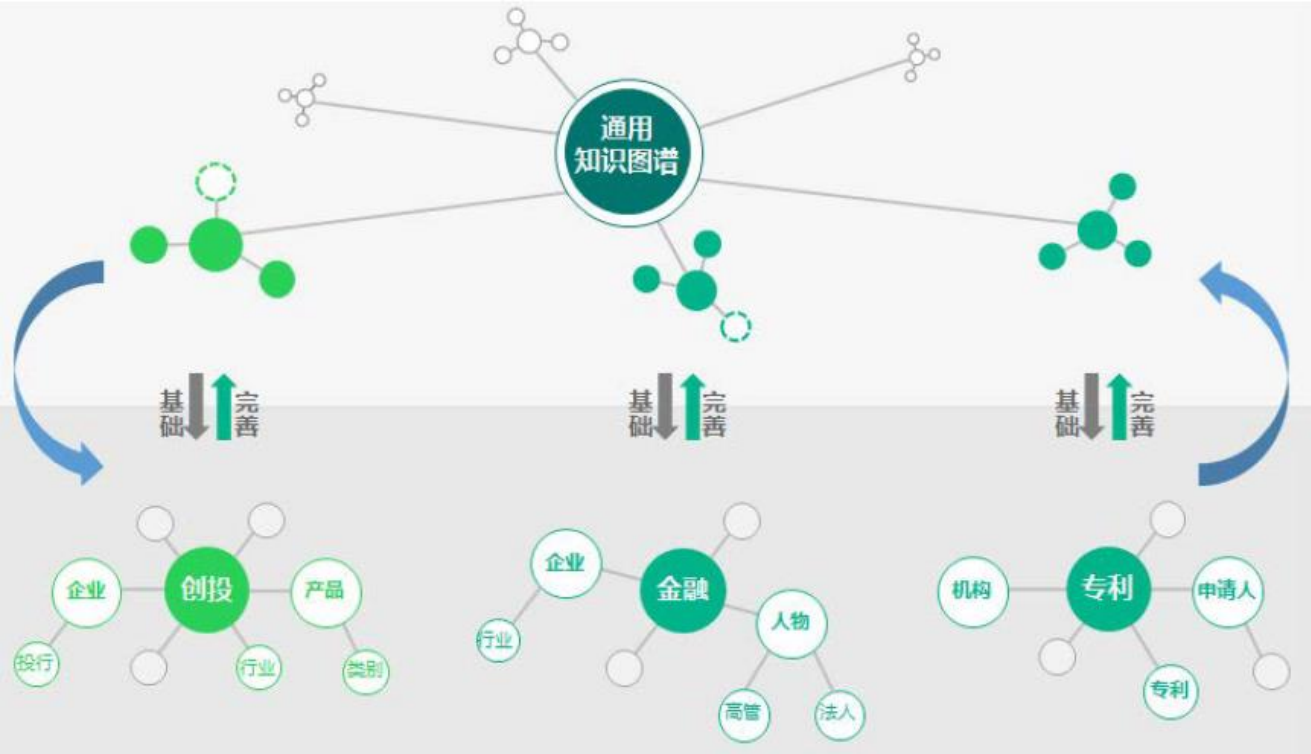
# 通用知识图谱 vs. 领域知识图谱

- 面向通用领域
- 以常识性知识为主
- 结构化的百科知识
- 强调知识的广度
- 使用者是普通用户

- 面向某一特定领域
- 基于行业数据构建
- 基于语义技术的行业知识库
- 强调知识的深度
- 潜在使用者是行业人员

# 通用知识图谱 + 领域知识图谱

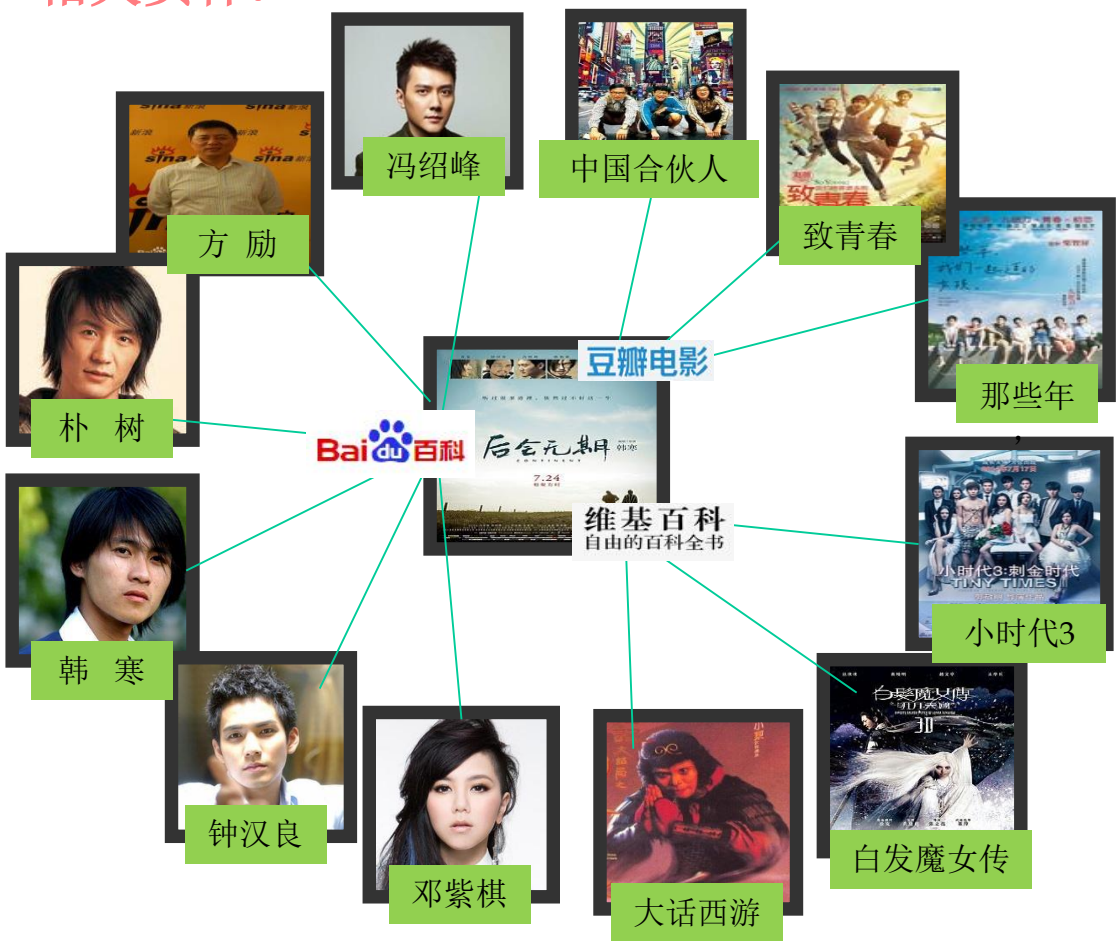
- 通用知识图谱的广度，领域知识图谱的深度，相互补充，形成更加完善的知识图谱
- 通用知识图谱中的知识，可以作为行业知识图谱构建的基础；而构建的行业知识图谱，再融合到通用知识图谱中。



# 影视领域知识图谱

- 影视知识融合

相关实体:




isA关系验证:

| | |
|---|---|
| 食人女 | isA |
| Baidu百科 | Film or TV? |
| Freebase | Film |
| 豆瓣电影 | Film |

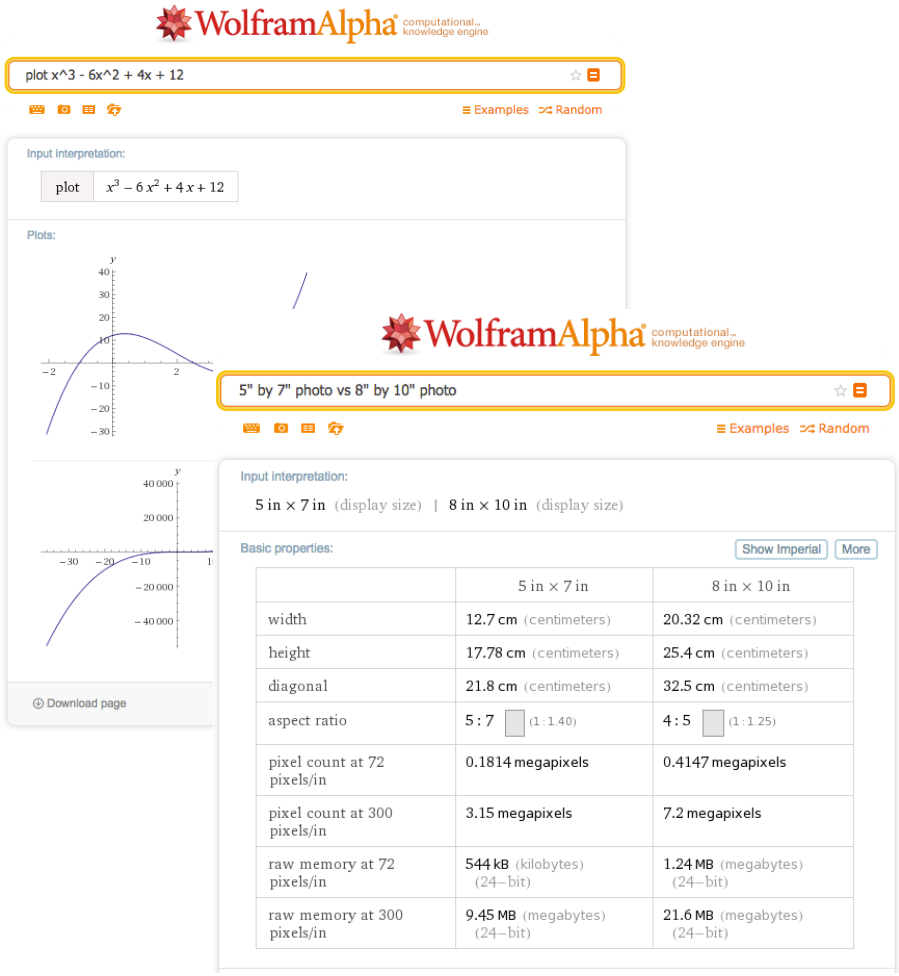
双语文字对齐:

movie:directed_by
*"Han Han"*@en,
*"韩寒"* @zh .
movie:genres
*"Comedy"*@en,
*"喜剧"* @zh .
movie:summary
*"The Continent is directed and written by Han Han…"@en,*
*"《后会无期》是一部由韩寒担任编剧及导演…"*@zh .

# 事件知识图谱

维基百科
自由的百科全书
首页
分类索引
特色内容
新闻动态
最近更改
随机条目
帮助
帮助
社区主页
方针与指引
互助客栈
知识问答
字词转换
IRC即时聊天
联系我们
关于维基百科
资助维基百科
工具
链入页面
相关更改
上传文件
特殊页面
打印版本
固定链接
页面 讨论 简体 汉 漢
阅读 编辑
2014年鲁甸地震 [编辑]
本文记述一项中国大陆新闻。随着事件发展，内容可能会快速更新。
除特别注明外，本文所有时间均以东八区时间（UTC+8）为准。
2014年鲁甸地震发生于北京时间（UTC+8）2014年8月3日16时30分10秒，地震规模为里氏6.5级[1]。震中位于中国云南省昭通市鲁甸县，震源距离地表12千米。震中在昭通市鲁甸县龙头山镇，距离昭通市区约49公里，昆明、重庆、成都、西安等地震感明显。[8]
目录 [隐藏]
1 背景
2 伤亡及损害
3 救援
4 分析
5 余震
6 各界反应
6.1 两岸三地
6.2 国际
7 捐赠
7.1 政府及社会组织
7.2 企业
7.3 个人
8 纪念
9 相关事件
10 参考资料
11 外部链接
主题
属性
infobox
2014年鲁甸地震
主震
日期 2014年8月3日 16:30:10 (UTC+8)[1]
震中 27.1°N 103.3°E[1]
规模 6.5 $M_L$[1] 6.1 $M_w$[2]
最大烈度 IX
深度 12 千米[1]
次数 763 次余震[3] (截止8月6日 20:00)
最大规模 4.2 $M_L$ 余震[3] (截止8月6日 20:00)
破坏
受影响地区 中国云南省昭通市鲁甸县、巧家县、曲靖市会泽县[1]
伤亡人数 死亡：617 (截止8月8日 15:00)[4]
受伤：3,143 (截止8月8日 15:00)[4]
重伤：? (截止8月8日 15:00)[4]
失踪：112 (截止8月8日 15:00)[4]
转移安置：22.97万 (截止8月8日 15:00)[4]
受灾：108.84万 (截止8月8日 15:00)[4]

# 计算知识图谱




http://www.wolframalpha.com/

# 跨语言知识图谱

BabelNet 1.1.1

A very large multilingual ontology

**Meaning:** actor[1] • **ID:** bn:00001176n • **Type:** Concept [details] [explore]

**Senses:**

**actor**[1], histrion[1], player[3], thespian[1], role player[2]

joueur, musicien, agent, comédien, acteur, personnage, ηθοποιός, θεατρίνος, שַׂחְקָן, attore, interprete, istrione, アクター, 俳優, 出演者, 名優, 役者, 男優, actor, actor dramático, actuación, artista, comediante, ejecutor teatral, histrionista, histrión, intérprete

**Actor**, ممثل, 演員, Acteur, Schauspieler, Ηθοποιός, שחקן, Attore, 俳優, Актёр, Actuación

American actress, Movie actress, Joyu, Synthetic actor, Film actress, Craig Watkinson, Film Actor, Theatre actor, Theatre actress, Stage actress, Actors and Actresses, Stage actor, Television actor, Actors, Actors and actresses, Film actor, Stage actors, Player (dating), American actor, Actresses, Actor & Actress, ممثل, تلفزي, الممثلين, الممثل, ممثلة, 男演员, 女演員, 戏子, 女演员, 男演員, 演员, Actrice futur, Jeu de l'acteur, Comédien, Comédiens, Actrice, Jeu dramatique, Comédienne, Comedien, Acteurs, Schauspielkunst, Fernsehdarsteller, Schauspielerin, Bühnenschauspieler, Theaterschauspielerin, Theaterschauspieler, Aktrice, Fernsehschauspieler, Schauspielerei, שחקן תיאטרון, שחקנית טלוויזיה, שחקן קולנוע, שחקנית קולנוע, שחקן טלוויזיה, כוכב קולנוע, תספיאן, Attrice (spettacolo), Attore (spettacolo), Attrice, Attori, 舞台女優, 男優, 実写俳優, 舞台俳優, 旅役者, 俳優業, トップスター, 役者, 映画俳優, Киноартист, Киноактриса, Киноактёр, Киноартистка, Актер, Киноактер, Актриса, Actuacion, Actríz, Actores, Actor, Actuar, Actor de cine, Actriz

**actor**, player, acteur, Schauspielerin, Darsteller, Schauspieler, Darstellerin, attore, 俳優, actriz, actris, actor

[w] **actor**, ممثل, acteur, actrice, Schauspieler, Schauspielerin, ὑποκριτής, שחקן, שחקנית, अभिनेता, अभिनेत्री, attore, attrice, 俳優, 役者, 女優, актёр, актриса, артист, артистка, actor, actriz

actor / actress, ممثل, 演員, acteur, Schauspieler, ηθοποιός, שחקן, अभिनेता, attore, 俳優, актёр, actor / actriz

الممثلة الاميركية, فاعل, 美国女演员, 戏剧演员, 电视演员, 演员, 电影演员, 演员, 美国演员, actrice américaine, actrice de cinéma, acteur de théâtre, acteur de télévision, acteurs, acteur, actrices, joueur, acteur américain, us-amerikanische schauspielerin, filmschauspielerin, schauspielerin, theaterschauspieler, fernsehschauspieler, schauspieler, schauspielerinnen, spieler,

**演员/Actor**

http://babelnet.org/

DBpedia

yago select knowledge

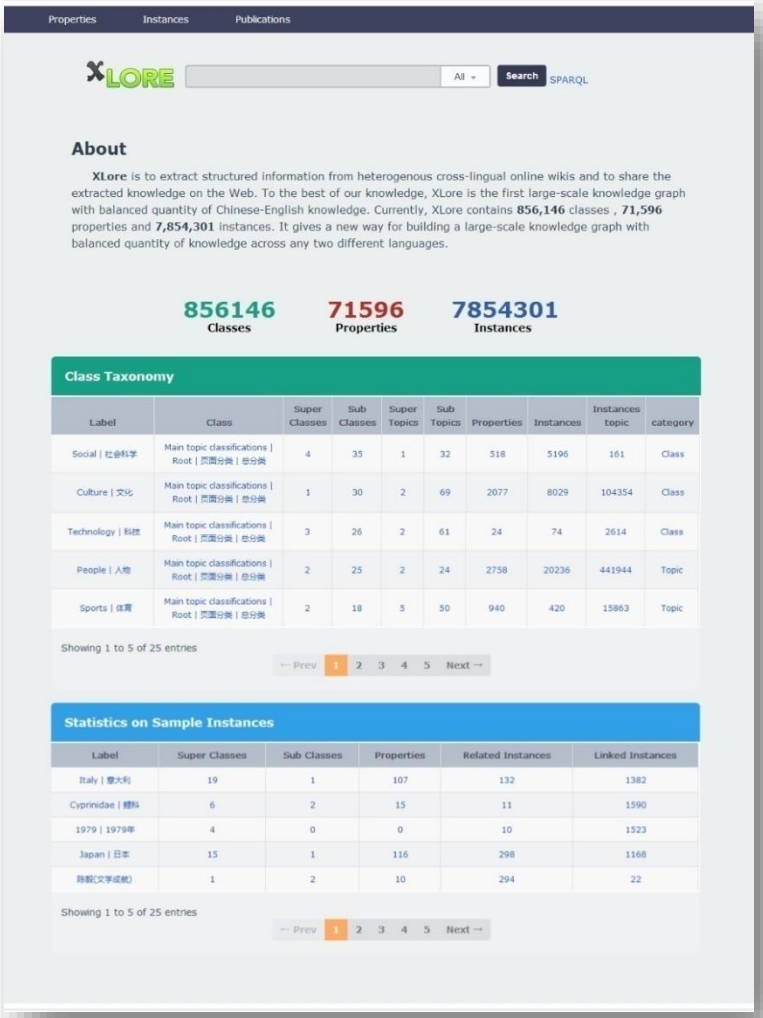
Freebase

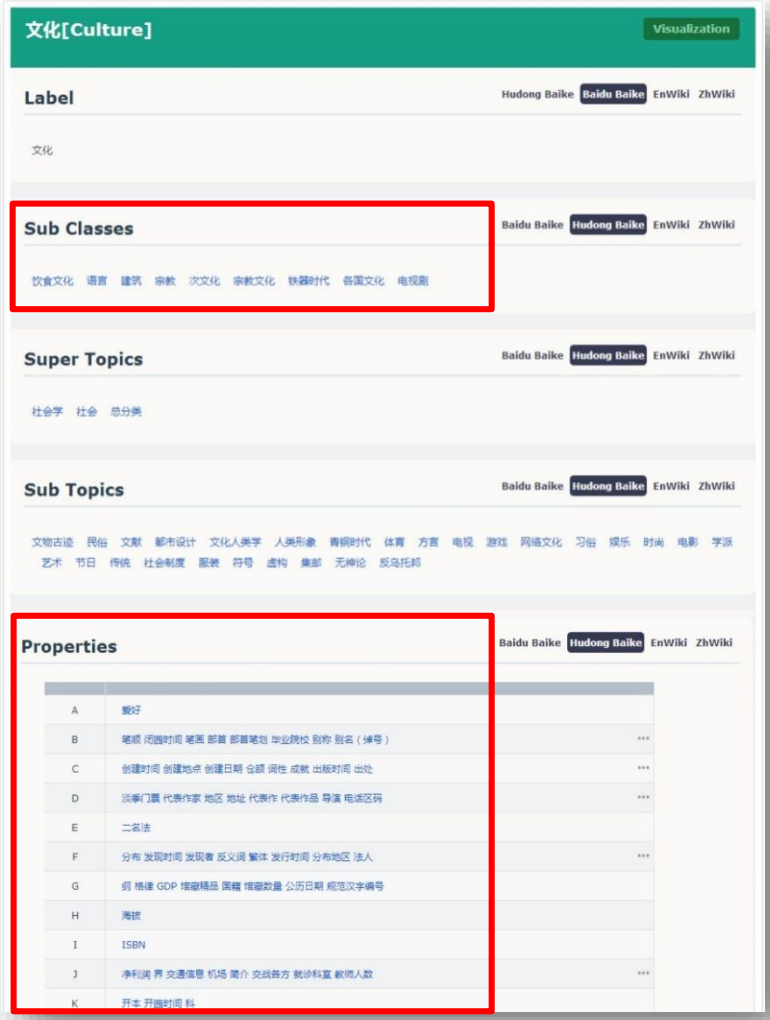
Google KG

EuroWordNet

XLORE

# 跨语言知识图谱XLORE



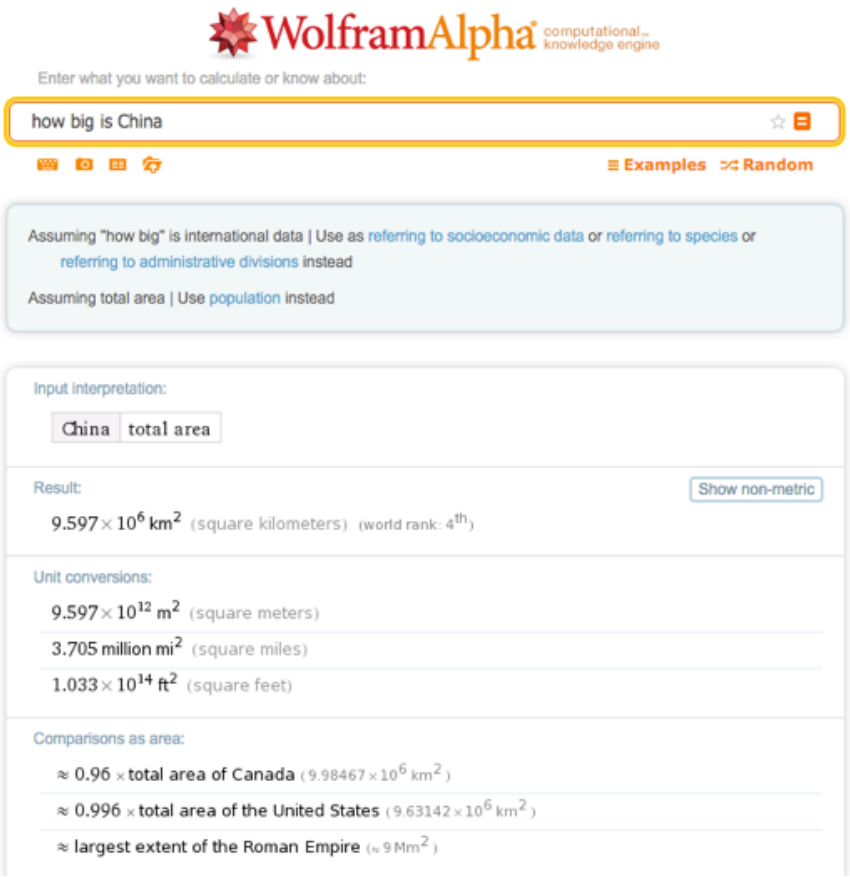




http://xlore.org/

# 提纲

- 课程介绍
- 知识图谱发展历史
- 知识图谱基本概念
- 知识图谱的生命周期
- 代表性知识图谱
- 知识图谱的应用

# 问答


WolframAlpha computational knowledge engine
Enter what you want to calculate or know about:
how big is China
Examples
Random
Assuming "how big" is international data | Use as referring to socioeconomic data or referring to species or referring to administrative divisions instead
Assuming total area | Use population instead
Input interpretation:
China total area
Result:
Show non-metric
9.597 × 10^6 km^2 (square kilometers) (world rank: 4th)
Unit conversions:
9.597 × 10^12 m^2 (square meters)
3.705 million mi^2 (square miles)
1.033 × 10^14 ft^2 (square feet)
Comparisons as area:
≈ 0.96 × total area of Canada (9.98467 × 10^6 km^2)
≈ 0.996 × total area of the United States (9.63142 × 10^6 km^2)
≈ largest extent of the Roman Empire (≈ 9 Mm^2)

姚明个子有多少
网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»
百度为您找到相关结果约3,030,000个
搜索工具
姚明身高:
226cm
姚明，1980年生于上海市徐汇区，祖籍吴江震泽。中国篮球运动员。1998年4月，他入选王非执教的国家队，开始篮球生涯。2002年，他以状元秀身份被NBA的休斯敦火箭队选中。20... 详情>>
来自百度百科 | 报错

# 精准语义搜索

奥巴马 百度一下

网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»

百度为您找到相关结果约86,500,000个 搜索工具

奥巴马_百度百科

姓名：贝拉克·侯赛因·奥巴马
生日：1961年8月4日 职业：政治家、律师、总统
简介：贝拉克·侯赛因·奥巴马（Barack Hussein Obama），1961年8月4日出生，美国民主党籍政治家，第44任美国总统，为美...
人物经历 执政表现 主要作品 家庭生活 人物评价 更多>>
查看"奥巴马"全部4个含义>>
baike.baidu.com/

奥巴马的最新相关信息

奥巴马卸任后当NBA球队老板? 白宫发言人:在讨论

ESPN消息,美国总统奥巴马即将卸任,而他未来要做的事情似乎已经确定好了,那就是当一支NBA球队的老板。 奥巴马卸任后当NBA球队老板? 白宫发言人:在讨...
网易体育 49分钟前

奥巴马欲成NBA球队老板 总统助奇才抢杜兰... 腾讯体育 1小时前
奥巴马的算盘: 英国留欧利于美国外交 搜狐财经 2小时前
奥巴马称朝鲜仍是特别威胁 决定对朝制裁... 网易军事 18小时前
作为最亲密的盟友,奥巴马对英国"脱欧"... 新浪财经 18小时前

奥巴马_百度图片

image.baidu.com 查看全部767,024张图片

历年时代周刊年度风云人物 展开

普京
俄罗斯铁腕总统

克林顿
曾任美国总统

小布什
美国第43任总统

比尔·盖茨
微软公司创始人前首富

肯尼迪
美国第35任总统

罗斯福
美国蝉联4届的总统

斯大林
苏联共产党中央总书记

里根
演员出身的美国总统

美国民主党员 展开

希拉里
美国第67任国务卿

拜登
美国现任副总统

卡罗琳·肯尼迪
被美国人称全国的宝贝

克里
美国第68任国务卿

# 精准语义搜索

王健林 下载报告

介绍： 王健林，男，1954年10月24日出生于四川省广元市，1989年起担任大连万达集团股份有限公司董事长。1970年入伍，1986年毕业于辽宁大学，同年7月进入大连市西岗区人民政府任办公室主任，1989年进入房地产行业，1993年起担任大连万达集团股份有限公司董事长、总裁。是中共十七大代表、第十一届全国政协常委、第十一届全国工商联副主席，兼任... 详细

他的所有商业角色　他的所有企业

| 角色 | 企业 | 省份地区 | 开业日期 | 注册资本 | 经营状态 |
|---|---|---|---|---|---|
| | 大连万达集团股份有限公司 | 辽宁 | 1992-09-28 | 100000 万人民币 | 存续（在营、开业、在册） |
| | 北京中住联华科技发展有限公司 | 北京 | 2001-04-25 | 120 万元 | 吊销 |
| | 北京环球高尔夫俱乐部有限公司 | 北京 | 1993-04-10 | 1120万元美元 | 吊销，未注销 |
| | 大连万达商业开发有限公司 | 辽宁 | 1994-12-12 | 0 万 | 吊销，未注销 |
| | 大连合兴投资有限公司 | 辽宁 | 2007-04-27 | 7860 万人民币 | 存续（在营、开业、在册） |
| | 大连万达通信有限公司 | 辽宁 | 1998-01-23 | 15000 万 | 吊销，未注销 |
| | 大连万达酒业有限公司 | 辽宁 | 1998-03-30 | 240 万 | 其他 |
| | 大连万达中心大厦有限公司 | 辽宁 | 1995-04-29 | 10000 万人民币 | 吊销，未注销 |
| | 四川万达房地产有限公司（转成都市锦江区） | 四川 | 1999-07-22 | 2000万人民币 | 存续（在营、开业、在册） |
| | 大连宏大物业管理有限公司 | 辽宁 | 1995-04-11 | 1000 万人民币 | 吊销，未注销 |
| | 大连天兴大酒店有限公司 | 辽宁 | 1997-01-20 | 100 万 | 吊销，未注销 |

# 关系搜索

beta
人立方
关系搜索
renlifang.msra.cn
六度秀
分享
帮助
周杰伦
奥巴马
六度搜索
周杰伦
迪克
切尼
奥巴马
点击上方人物，自定义关系链
希拉里
赖斯
汤姆
侯佩岑
布什
切尼
拉姆斯菲尔德
范特西
周杰伦
迪克
奥巴马
墨菲
爱德华兹
蔡依林
考夫曼
希拉里·克林顿

# 分类浏览

Baidu音乐
请输入歌名、歌词、歌手或专辑
薛之谦 林忆莲《我不.. 李宗盛 五月天 梁静茹 莫文蔚 陈梓童 孙盛希《Betw.. 周杰伦 张杰
首页 歌单 动态 NEW 歌手 分类 榜单 MV 演出
标签
热门
新歌 热歌 中国好声音 经典老歌 电视剧 广场舞 热
欧美 热 轻音乐 热 DJ 舞曲 80后 网络歌曲 热 劲爆
儿歌 纯音乐 粤语 民歌 钢琴曲 萨克斯
古典音乐 对唱 佛教音乐 成名曲 草原歌曲
心情
伤感 热 激情 安静 舒服 热 甜蜜 励志
寂寞 想念 浪漫 怀念 喜悦 深情
美好 怀旧 轻松
风格
小清新 热 DJ 舞曲 纯净 唯美 轻音乐 舒缓
劲爆 热 慢摇 民歌 热 青春 好听 交响乐
乐播
笑话段子 相声曲艺 热 脱口秀 母婴儿童 小说读物 综艺娱乐
都市情感 商业财经 教育 健康 新闻时事 科技
生活 社会文化 英语播客

音乐分类 > 纯净
纯净 共1000首歌 收听该分类电台 收听音乐人热播电台
相关无损大碟推荐 换一换
柴科夫斯基作品精选
浪漫主义钢琴诗人·肖邦盘点
Beethoven: Overtures
Mozart: Le nozze di Figaro - Highlights
全部 播放选中歌曲 加入播放列表 收藏 批量下载
01 心有灵犀 熊天平 《雪候鸟》
02 心有灵犀 熊天平 《雪候鸟》
03 有一个人 齐豫 《齐豫个人中文..》
04 心有灵犀 熊天平 《熊心万丈》
05 夏天的风 元卫觉醒 《艾回20年200曲..》
06 可不可以爱 何炅 《可以爱》
07 Eternity Kelly Sweet 《We Are One》
08 Dreamer Sophie Zelmani 《Time To Kill》
09 Ayo Technology Milow 《Milow》
10 A Little Love 冯曦妤 《A Little Love》
11 我不想忘记你 郭静 《我不想忘记你》
审批文号：WJ214775
12 陪你到世界的终结 棉花糖 《小飞行》
13 小飞行 棉花糖 《小飞行》
14 多余的解释 许嵩 《自定义》
15 想念 许哲佩 《气球》

# 推荐

战狼2 百度一下

网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»

百度为您找到相关结果约5,830,000个 搜索工具

战狼2_百度百科

关注点 TA说：深度解读功与过

《战狼Ⅱ》是吴京执导的动作军事电影，由吴京、弗兰克·格里罗、吴刚、张翰、卢靖姗、淳于珊珊、丁海峰等主演。该片于2017年7月27日在中国内地上映 。影片讲述了脱下军装的冷锋被卷入了一场非洲国家的叛乱，本来能够安全撤离的他无法忘记军...

剧情简介 演职员表 角色介绍 音乐原声 幕后花絮 更多>>

baike.baidu.com/ - 

战狼2的最新相关信息

战狼2依然无敌 其他片即便是"炮灰"也各有特色

上周五,《心理罪》《侠盗联盟》《蛟珠传》集体进入电影市场,对票房强势的《战狼2》形成围剿。但姜还是老的辣!从上周末的市场走向来看,《战狼2》依然...

新浪 17分钟前

| | | |
|---|---|---|
| 最全票房信息都在这里!战狼2票房将超6亿... | 人民网重庆站 | 37分钟前 |
| 《战狼2》助力国产片票房逆袭 大批国片火... | 电影网 | 45分钟前 |
| 专访《战狼2》航拍主飞手:辛苦与压力并存 | 华夏经纬 | 4小时前 |
| 外媒:《战狼2》触动中国观众的民族情结,... | 观察者网 | 6小时前 |

战狼2吧_百度贴吧

关注用户：10万人 累计发贴：52万

精品贴(21)

| | | |
|---|---|---|
| ★【战狼2】??【BD1280】[全]完 整出了。在线观... | 点击：2402 | 回复：9 |
| 战狼2是我贡献过票房的电影 | 点击：1334 | 回复：5 |
| 战狼2一出，我就是元老了，哈哈 | 点击：488 | 回复：57 |

查看更多战狼2吧的内容>>

相关影视作品 收起

叶问4
甄子丹主演的动作电影

印囧
徐峥编导公路喜剧片

火蓝刀锋2之雄鹰展翅
杨志刚主演当代军旅剧

使徒行者2
苏万聪执导警匪电视剧

僵尸世界大战2

大闹东海

复仇者联盟3

敢死队4

环太平洋2

金刚狼3

杀破狼2

极限特工3

孤狼特别突击队

利刃出击

拆弹专家

第一滴血5

# 推理

梁启超的儿子的老婆的情人的父亲 搜狗搜索

网页 论坛 知识 新闻 博客 百科 更多 什么是分类搜索 找到约 173,657 条结果

**梁启超的儿子的老婆的情人的父亲：**

徐申如

**推理说明：** 梁启超的儿子是梁思成。梁思成的妻子是林徽因。林徽因的情人是徐志摩。徐志摩的父亲是徐申如

梁启超的儿子的老婆的情人的老婆-读书-DoNews.COM
2004年6月15日... 作者 帖子主题： 知道是谁不？ 作者 帖子主题： RE：梁启超的儿子的老婆的情人的老婆 【（shengfang）回复（cool）的大作】 陆小慢 作者 帖子主题： RE:...
donewsIT门户 - home.donews.com/.../477746.html - 2004-6-15 - 快照 - 预览

梁启超的儿子的老婆的情人的父亲 最佳答案 搜狗知识搜索
梁启超的儿子的老婆的情人的老婆是谁 - 已解决 搜搜问问 2007-11-25
答：梁启超的儿子呢是中国的著名建筑师梁思成 梁思成的老婆呢叫林薇茵看过《人间四月天》的人应该知道啊 那么林薇茵的情人呢就是大名鼎鼎的徐志摩啦 那么徐志摩的老婆...
梁启超的儿子的老婆的情人的老婆是谁？？？ - 已解决 搜搜问问 2011-3-4
梁启超的儿子的太太的情人的太太分别是谁 - 已解决 搜搜问问 2007-12-9
梁启超的儿子的妻子的情人的老婆是谁？ 百度知道 2006-10-4

**梁启超**

梁启超(1873.2.23—1929.1.19)，生于广东新会。1894年，梁启超提倡变法，并于上海主撰《时务报》，著《变法通议》，刊布报端，启发国人之革新思想。与谭嗣同..
相关阅读

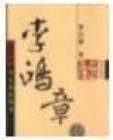
**出生：** 1873-02-23 / 广东新会
**逝世：** 1929-01-19
**妻子：** 李蕙仙 (正室) / 王桂荃 (老婆)
**人物关系：** 梁宝瑛 (父亲) / 梁思达 (儿子) / 梁思忠 (儿子) / 梁思懿 (女儿) / 梁思成 (儿子)
**个人名言：** 享受工作的同时享受生活

**著作** 更多>>

中国近三百年学..

中国历史研究法

新大陆游记



李鸿章传

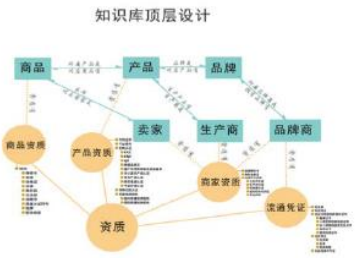
清代学术概论

# 领域知识图谱的应用

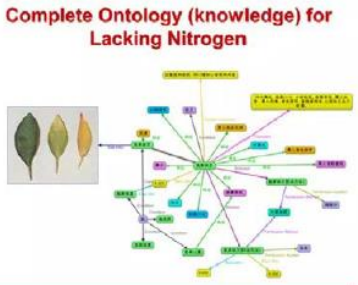
金融证券

生物医疗

图书情报

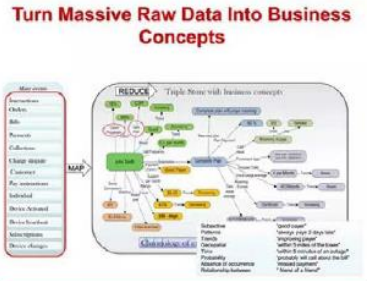



电商

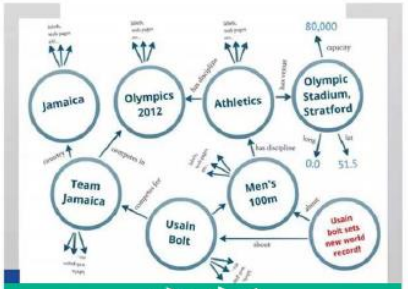



农业

政府




电信




出版

# 金融证券领域

## 企业风险评估

- 基于企业的基础信息、投资关系、诉讼、失信等多维度关联数据，利用图计算等方法构建科学、严谨的企业风险评估体系，有效规避潜在的经营风险与资金风险。

### 用户群体

银行、担保、投行、政府……

### 应用环节

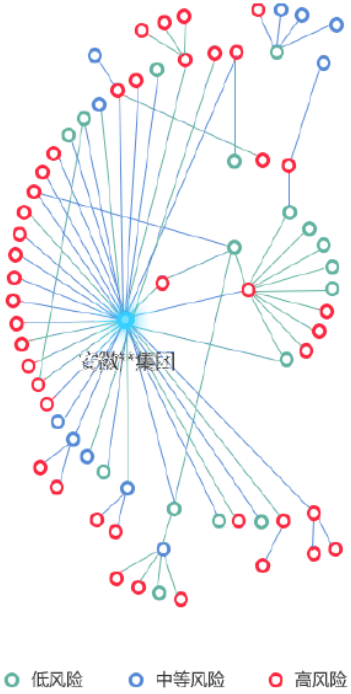
客户资源分类管理

信贷前期风险评估

采购企业风险审核

招投标企业资质评级

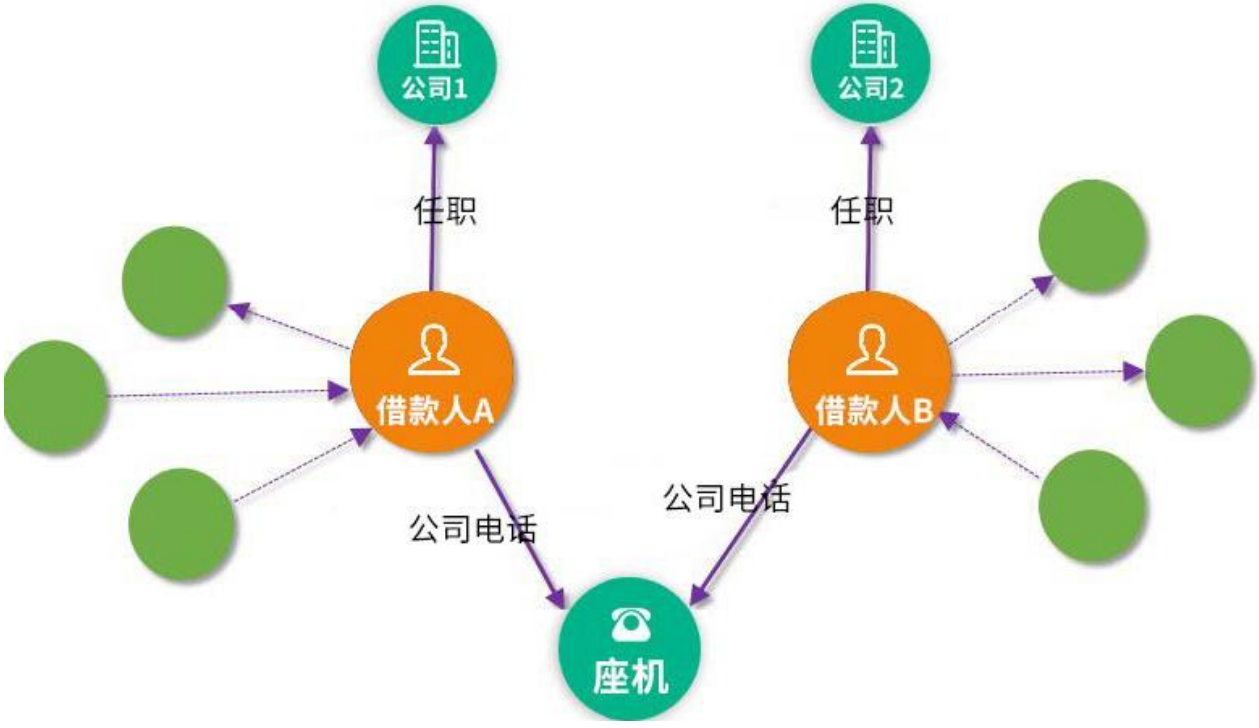
……



# 金融证券领域

## 反欺诈

- 不一致性验证可以用来判断一个借款人的欺诈风险，类似交叉验证。比如借款人A和借款人B填写的是同一个公司电话，但借款人A填写的公司和借款人B填写的公司完全不一样，这就成了一个风险点，需要审核人员格外的注意。

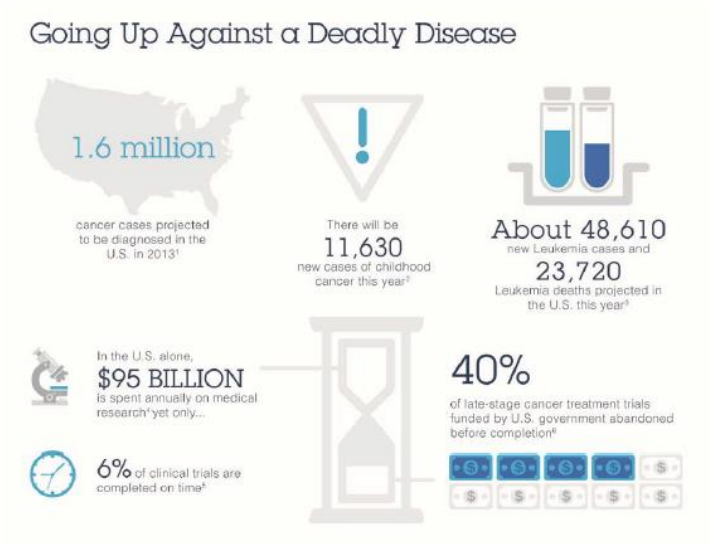


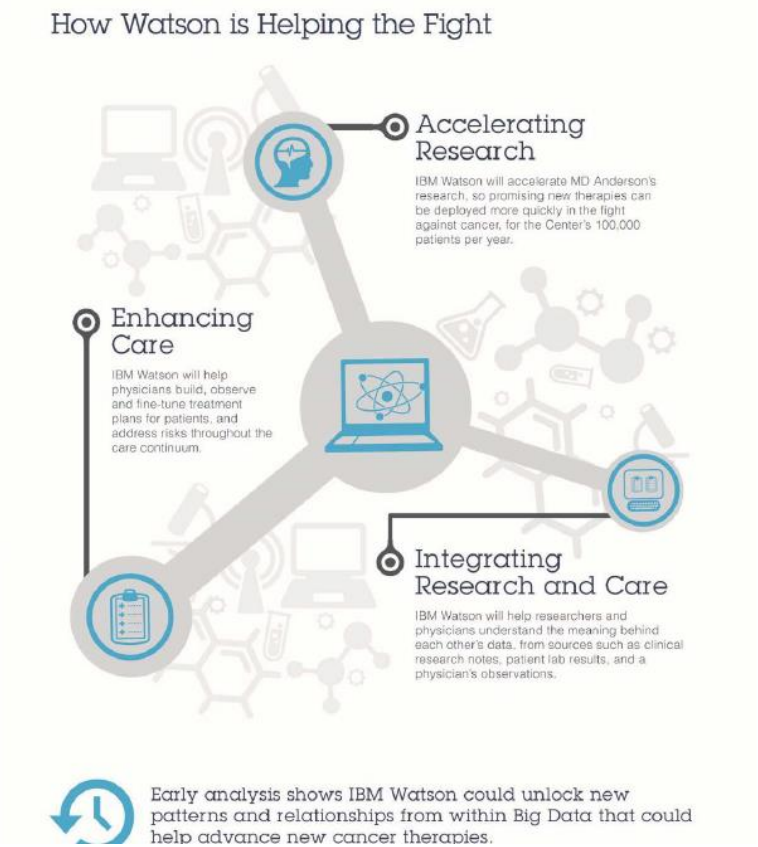
# 医疗领域

- 中医药知识服务平台
  - 辅助中医专家厘清学术发展脉络，浏览中医知识，发现知识点之间的联系。

# 医疗领域

- Watson辅助诊断与治疗
  - 安德森癌症中心联合IBM Watson 开展终结癌症的任务



# 医疗领域

- Open PHACTS 新药物发现
  - 欧盟重大联合攻关项目：面向药物研发的开放数据访问平台开发，其核心技术就是采用语义技术为有关研究人员提供高效的数据访问技术环境的支持。

Open PHACTS

Bringing together **pharmacological data resources** in an integrated, interoperable infrastructure

**Explore.**

**Researchers** can use Open PHACTS to access vast amounts of pharmacological data, all from a single, simple interface

**Build.**

**Developers** get free access to the Open PHACTS API, to query the pharmacological data resources in our integrated triple store

**Join.**

**Members** of the Open PHACTS Foundation get prioritised access to data, support and updates, as well as training opportunities

# 当前KG的限制和不足

- 领域限制
  - 一些知识库侧重于语言: WordNet,BabelNet
  - 一些知识库侧重于Schema： ConceptNet, UMBEL
  - 一些知识库侧重于Fact: DBPedia, Yago
- 对时空属性的建模
  - 对动态性的实体，如Event建模不足
  - Yago 3在一定程度上考虑时间和地理属性
- 自动构建
  - 自动构建是维护和保持KG质量和覆盖的核心技术
- 与LOD的集成
  - 缺乏Schema之间的alignment
  - 往往只用到底层的表达能力，OWL的高级功能很少涉及

# KG展望

- 新的知识表示模型
  - Ontology engineering已经被用了超过15年
- 新类型的知识图谱
  - 不再围绕实体和关系的存储
  - 如Event-centric KG
- 知识图谱自动构建技术
  - 在Freebase中，71%的人没有出生日期
  - 新技术：Distant Supervision, KG embedding, 知识集成（如Google的Knowledge Vault）

# 教师简介

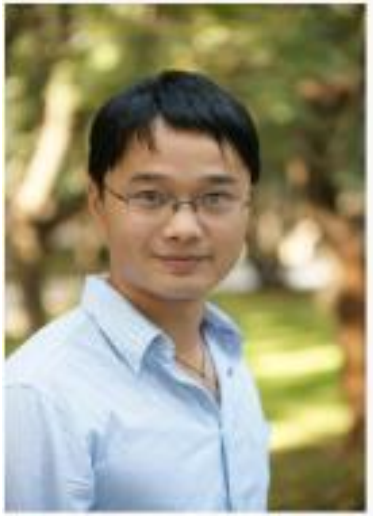
## ■赵 军

中国科学院自动化研究所研究员，博士生导师。中国科学院大学岗位教授。长期从事自然语言处理、知识图谱和问答系统方向的研究，承担国家自然科学基金重点课题、973计划、863计划等多个重要科研项目，在IEEE TKDE、ACL、SIGIR、IJCAI、AAAI、COLING、EMNLP、CIKM、WWW等国际顶级期刊和会议上发表论文80多篇，曾获国际计算语言学大会COLING 2014最佳论文奖；主持研发的推荐系统获得2011 SIGKDD CUP第二名（2/1792）；主持研发了汉语分析、知识图谱、知识问答等软件工具和平台，在中国大百科全书出版社、华为、蚂蚁金服等得到应用，曾获CCF-腾讯犀牛鸟基金优秀专利奖和华为重点专利奖。

# 教师简介

## 刘 康

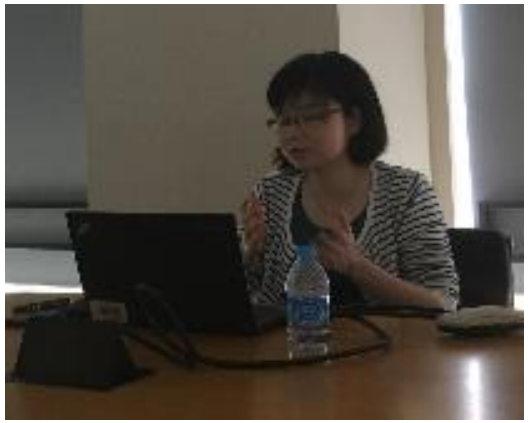
中科院自动化所模式识别国家重点实验室副研究员。研究领域包括信息抽取、网络挖掘、问答系统等，同时也涉及模式识别与机器学习方面的基础研究。在自然语言处理、知识工程等领域国际重要会议和期刊发表论文四十余篇（如TKDE、ACL、IJCAI、EMNLP、COLING、CIKM等），获得KDD CUP 2011 Track2 全球亚军，COLING 2014最佳论文奖，首届“CCF-腾讯犀牛鸟基金卓越奖”、2014年度中国中文信息学会“钱伟长中文信息处理科学技术奖-汉王青年创新一等奖”、2015、2017 Google Focused Research Award等。



# 教师简介

## ■王 泉

中科院信息工程研究所第二研究室副研究员。研究方向为机器学习、文本挖掘、自然语言处理，现主要研究知识的自动获取、表示和推理等关键技术及其在文本处理领域中的应用。在SIGIR、IJCAI、ACL、EMNLP、ACM TOIS、IEEE TKDE等国际顶级会议和期刊上发表多篇学术论文。主持国家自然科学基金青年基金等项目。曾于2011年获微软学者奖，2014年获中国中文信息学会优秀博士学位论文提名，入选2014年中国科学院信息工程研究所优秀青年人才计划，入选2015年微软亚洲研究院青年教师铸星计划。



# 助教简介

## ■陈玉博

陈玉博，中科院自动化所模式识别国家重点实验室助理研究员，2017年获得中国科学院大学工学博士学位，研究方向为事件抽取、知识图谱和自然语言处理。在ACL、EMNLP、WWW等国际重要会议发表论文多篇。获得2016年全国计算语言学大会最佳论文奖、2017年全国知识图谱与语义计算大会最佳论文奖。北京市优秀博士毕业生。参与国家自然科学基金重点项目、973计划、863计划以及多项企业（华为、讯飞、Docomo等）合作科研项目的研发。

# 课程安排

- **第1章 课程介绍及知识图谱基础 （3学时）赵军**
  - 课程简介
  - 知识图谱概述
  - 已有知识图谱介绍
- **第2章 机器学习基础 （4学时）刘康**
  - 机器学习基础
  - 深度学习基础
  - 典型深度学习模型（一）
  - 典型深度学习模型（二）
- **第3章 知识表示方法 （3学时）王泉**
  - 基于符号的知识表示方法概述
  - 语义网络
  - 基于分布式的知识表示方法

# 课程安排

- **第4章 知识图谱框架 （3学时）刘康**
  - 知识框架的学习
  - 知识框架对齐
  - 研讨：知识融合
- **第5章 实体识别（3学时）赵军**
  - 命名实体识别（一）
  - 命名实体识别（二）
  - 实体扩展
- **第6章 实体消歧（3学时）赵军**
  - 实体消歧概述
  - 基于无监督的实体消歧
  - 基于知识库链接的实体消歧
- **第7章 关系抽取（3学时）赵军**

# 课程安排

- 限定域关系抽取
- 开放域关系抽取（一）
- 开放域关系抽取（二）

## 第8章 研讨课：事件抽取（3学时）赵军

- 事件知识表示
- 事件知识抽取
- 基于事件的故事概要抽取

## 第9章 知识的存储和检索（3学时）王泉

- 基于关系数据库的知识图谱存储
- 基于图数据库的知识图谱存储
- 知识图谱的检索

## 第10章 知识推理（3学时）王泉

- 推理概述
- 归纳推理

# 课程安排

- 演绎推理
- 常识推理
- 基于表示学习的推理

**第11章 研讨课：知识图谱构建（3学时）赵军**

- 领域知识图谱框架构建
- 半结构化文本中的知识抽取
- 非结构化文本中的知识抽取

**第12章 知识图谱应用1（3学时）刘康**

- 问答系统
- 基于语义解析的知识库问答
- 基于检索的知识库问答

**第13章 知识图谱应用2（3学时）刘康**

- 基于深度神经网络的端到端知识库问答
- 基于知识图谱的对话系统

# 考核方式

- 演绎推理
- 常识推理
- 基于表示学习的推理

**第11章 研讨课：知识图谱构建（3学时）赵军**

- 领域知识图谱框架构建
- 半结构化文本中的知识抽取
- 非结构化文本中的知识抽取

**第12章 知识图谱应用1（3学时）刘康**

- 问答系统
- 基于语义解析的知识库问答
- 基于检索的知识库问答

**第13章 知识图谱应用2（3学时）刘康**

- 基于深度神经网络的端到端知识库问答
- 基于知识图谱的对话系统

# 参考

- 李涓子、刘康、韩先培、王昊奋等老师的工作

谢谢

jzhao@nlpr.ia.ac.cn

# 机器学习基础

刘康

中国科学院自动化研究所

# 研讨课安排

- 研讨课
  - 第4章 知识图谱框架 （3学时）刘康（10月10日）
  - 第8章 研讨课：事件抽取（3学时）赵军（11月7日）
  - 第11章 研讨课：知识图谱构建（3学时）赵军（11月28日）

- 形式
  - 分组汇报（每组：6-8人，15分钟汇报一篇相关的文章）
  - 候选Topic：
    - 知识融合(1-6组，内容可以参考OAEI评测相关文章)
    - 事件抽取（7-12组，内容可以包括事件抽取；事件关系预测；事件框架生成；事件预测）
    - 知识图谱构建（13-18组，内容可以包括实体识别；实体消歧；关系抽取）

# 研讨课安排

- 分组办法
  - 有道云协作：http://163.fm/4dMfuC4t
  - 请组长进群后填写本组相关信息
  - 信息填写截止时间：2017年9月20日晚8：00（周三）
  - 文章下载地址：http://www.aclweb.org/anthology/　只要Topic属于上述候选Topic即可，也可以从其他期刊或者会议下载
  - **编号规则**：请大家**按编辑先后顺序**编写自己的小组序号，先到先得，例如：第一个编辑的小组可以选择1-18中的任意编号，第二个小组可以选择除了第一个小组以外的其它编号，如果后填写的小组看到自己的小组号码或者文章题目和已经填写好的小组冲突，请后面的小组更换为不冲突的编号和文章。
  - **切记：这个协作笔记有操作记录，请大家只编辑自己小组的信息，不要更改其余小组的信息！！！！！！**

# 参考书籍

- 1. Knowledge Representation and Reasoning (Ronald J.Brachman, Hector J.Levesque)
- 2. Artificial Intelligence: A Modern Approach (Stuart Russell, Peter Norvig)
- 3. A Semantic Web Primer（Grigoris Antoniou等）
- 4. Speech and Language Processing（Daniel Jurafsky，James H.Martin）

# 目录

- 机器学习基础理论与概念
- 神经网络与深度学习基础
- 卷积神经网络
- 循环神经网络

# 机器学习

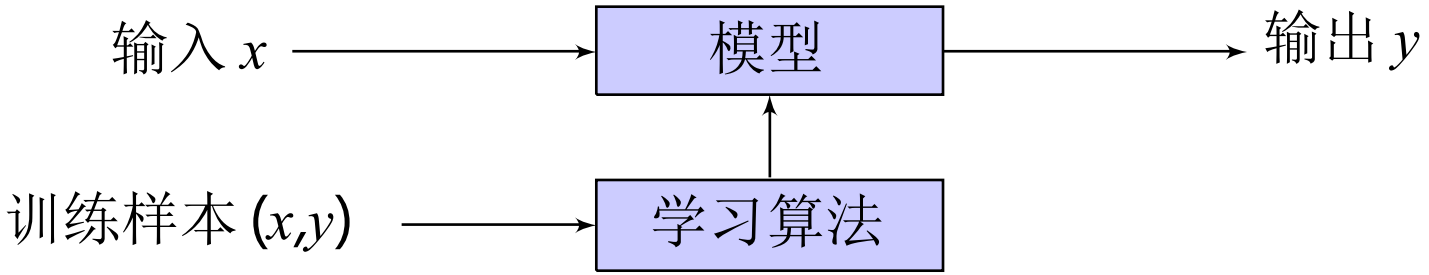
- 机器学习(Machine Learning, ML)是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能

- 机器学习是人工智能的一个分支,其目的在于使得机器可以根据数据进行自动学习,通过算法使得机器能从大量历史数据中学习规律从而对新的样本做决策

- 它目前是人工智能的核心，是使计算机具有智能的根本途径，其应用遍及人工智能的各个领域，它主要使用归纳、综合而不是演绎

# 机器学习

机器学习主要是研究如何使计算机从给定的数据中学习规律，即从观测数据（样本）中寻找规律，并利用学习到的规律（模型）对未知或无法观测的数据进行预测。目前，主流的机器学习算法是基于统计的方法，也叫统计机器学习



# 机器学习概要

**训练数据：** $(x_i, y_i), 1 \le i \le m$

**模型：**

线性方法：$y = f(x) = w^T x + b$

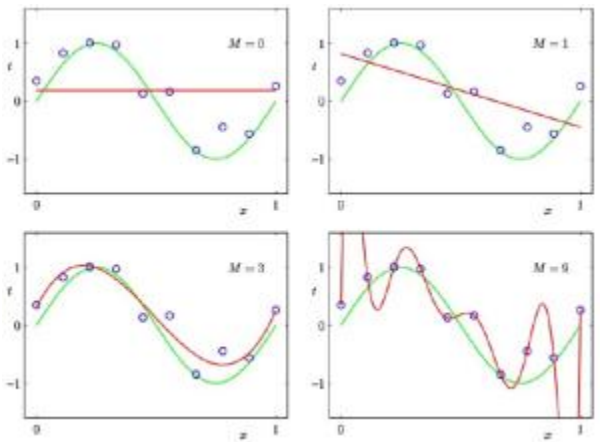
非线性方法：神经网络

**优化：**

损失函数：$L(y, f(x))$

经验风险最小化：$Q(\theta) = \frac{1}{m} \cdot \sum_{i=1}^{m} L(y_i, f(x_i, \theta))$

正则化：$\|\theta\|^2$



奥卡姆剃刀原则

**优化目标函数：** $Q(\theta) + \lambda \|\theta\|^2$

# 机器学习

狭义地讲，机器学习是给定一些训练样本 $(x_i, y_i), 1 \le i \le N$（其中 $x_i$ 是 输入，$y_i$ 是需要预测的目标），让计算机自动寻找一个决策函数 $f(\cdot)$ 来建立 $x$ 和 $y$ 之间的关系。

$$\hat{y} = f(\Phi(x),\ \theta)$$

这里，$\hat{y}$是模型输出, $\theta$为决策函数的参数，$\Phi(x)$表示样本 x 对应的特征表示。
因为 x 不一定都是数值型的输入，因此需要通过 $\Phi(x)$ 将 x 转换为数值型的输入。

# 损失函数

在机器学习算法中，一般定义一个**损失函数** $L(y, f(x, \theta))$，在所有的训练样本上来评价决策函数的好坏（风险）。

$$R(\theta)=\frac{1}{N}\sum_{i=1}^{N}L\left(y^{(i)},f\left(x^{(i)},\theta\right)\right)$$

风险函数 R(θ ) 是在已知的训练样本（经验数据）上计算得来的，因此被称之为经验风险。参数的求解其实就是寻求一组参数，使得经验风险函数达到最小值，就是我们常说的经验风险最小化原则（Empirical Risk Minimization）

$$\theta^{*}=\arg\min_{\theta} R(\theta)$$

# 损失函数

如何度量错误的程度。

0-1 损失函数

$$L\left(y, f\left(x,\theta\right)\right)=\begin{cases}0 & if \quad y=f\left(x,\theta\right)\\ 1 & if \quad y\neq f\left(x,\theta\right)\end{cases}$$
$$=l\left(y\neq f\left(x,\theta\right)\right)$$

平方损失函数

$$L\left(y,\hat{y}\right)=\left(y-f\left(x,\theta\right)\right)^2$$

# 损失函数

交叉熵损失函数 对于分类问题，模型输出$f(x, \theta)$ 为每个类 $y$ 的条件概率。

假设 $y \in \{1,\cdots,C\}$，模型预测样本属于第 i 个类的条件概率$P(y = i \mid x) = f_i(x,\theta)$，则 $f(x,\theta)$ 满足

$$f_i(x,\theta) \in [0,1], \quad \sum_{i=1}^{C} f_i(x,\theta) = 1$$

$f_y(x,\theta)$ 可以看作对于所标注类别 $y$ 的似然函数。参数可以直接用最大似然估计来优化。考虑到计算问题，我们经常使用最小化负对数似然，即**负对数似然损失函数**（Negative Log Likelihood function）。

$$L(y, f(x,\theta)) = -\log f_y(x,\theta)$$

# 损失函数

如果我们用 one-hot 向量$y$来表示目标类别 $c$，其中只有 $y_c = 1$，其余向量元素都为 0。
则目标函数可以写为：

$$L\left(y, f\left(x,\theta\right)\right) = -\sum_{i=1}^{C} y_i \log f_i\left(x,\theta\right)$$

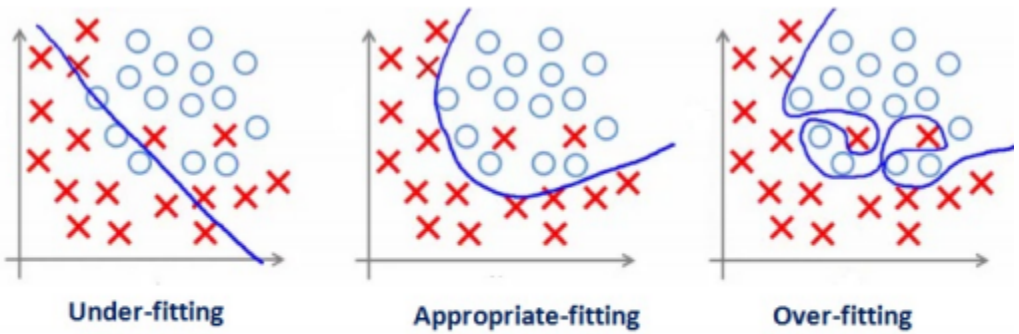
$y_i$是所标注真实类别的分布，上式恰好是交叉熵的形式。因此，损失函数也称之为交叉熵损失函数（Cross Entropy Loss function）。

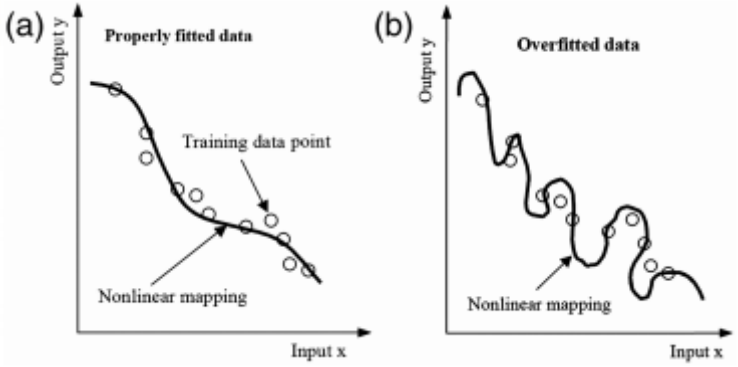
# 损失函数

Hinge 损失函数 对于两类分类问题，假设 $y$ 和 $f(x,\theta)$ 的取值为 $\{-1,+1\}$。Hinge 损失函数（Hinge Loss Function）的定义如下：

$$
\begin{aligned}
L\left(y, f\left(x,\theta\right)\right) &= \max\left(0, 1 - yf\left(x,\theta\right)\right) \\
&= \left|1 - yf\left(x,\theta\right)\right|_{+}
\end{aligned}
$$

# 过拟合 overfitting



# 结构风险最小化原则

为了解决过拟合问题，一般在经验风险最小化的原则上加参数的**正则化**（Regularization），也叫**结构风险最小化原则**（Structure Risk Minimization）。

$$\theta^* = \arg\min_{\theta} R(\theta) + \lambda \|\theta\|^2$$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L\left(y^{(i)}, f\left(x^{(i)}, \theta\right)\right) + \lambda \|\theta\|^2$$

**正则化项**

$\lambda$ 用来控制正则化的强度，正则化项也可以使用其它函数，比如 $L1$ 范数

# 学习

在机器学习问题中，我们需要学习到参数 $\theta$，使得风险函数最小化。

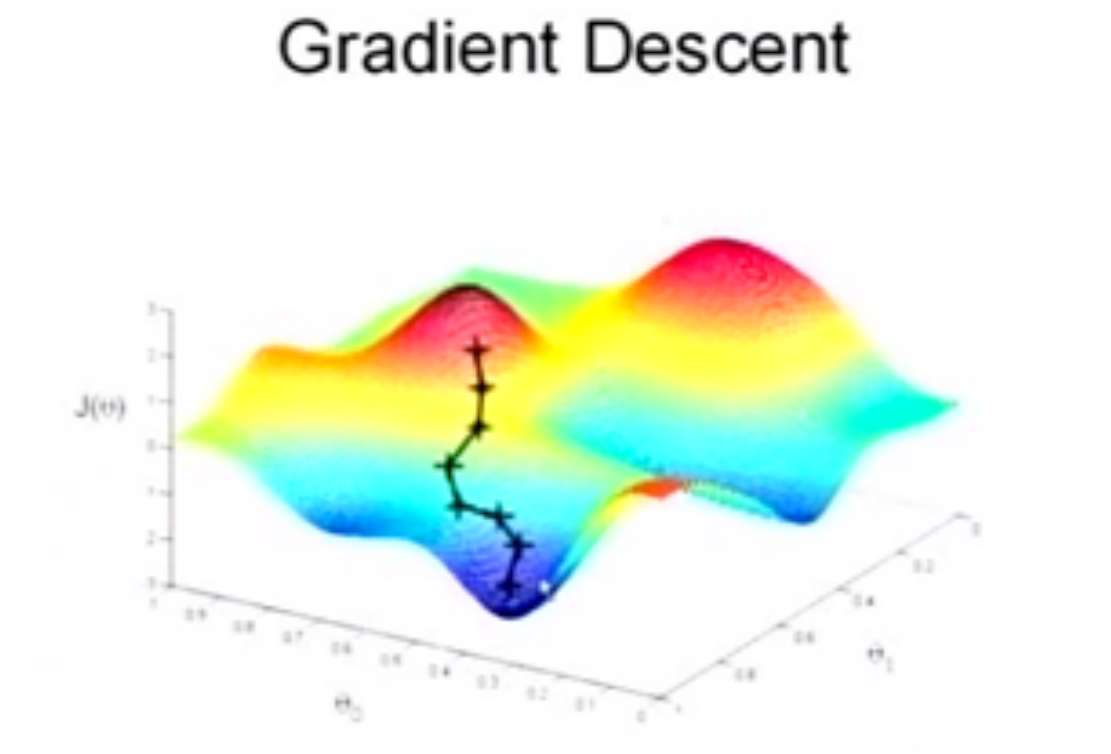
$$\theta^* = \arg\min_{\theta} R(\theta_t)$$
$$= \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} L\left(y^{(i)}, f\left(x^{(i)}, \theta\right)\right)$$

如果用梯度下降法进行参数学习，

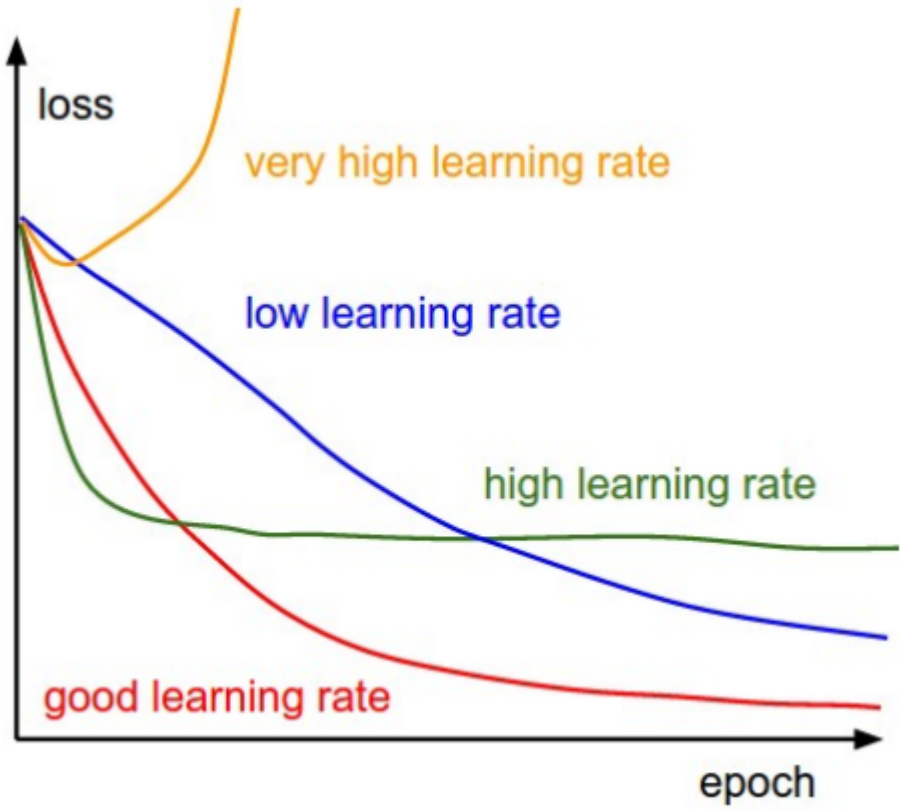
$$a_{t+1} = a_t - \lambda \frac{\partial R(\theta)}{\partial \theta_t}$$
$$= a_t - \lambda \sum_{i=1}^{N} \frac{\partial R\left(\theta_t; x^{(i)}, y^{(i)}\right)}{\partial \theta}$$

搜索步长 $\lambda$ 在机器学习中也叫作学习率（Learning Rate）。

# 梯度下降法



# 学习率


loss
very high learning rate
low learning rate
high learning rate
good learning rate
epoch

# 学习率设置：自适应法

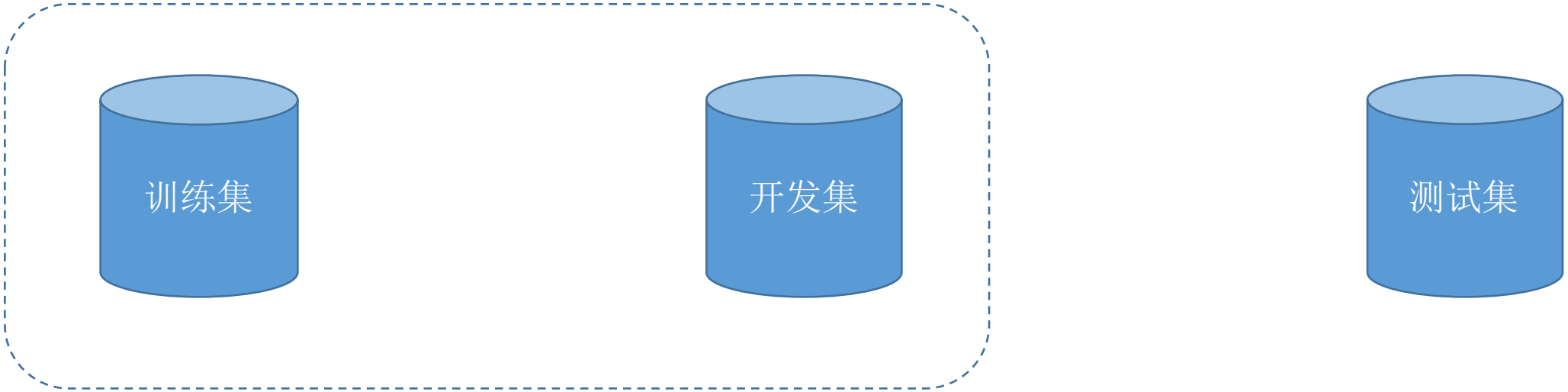
AdaGrad（Adaptive Gradient）算法是借鉴 $L_2$ 正则化的思想。在第 $t$次迭代时，

$$\theta_t = \theta_{t-1} - \frac{\rho}{\sqrt{\sum_{\tau=1}^{t} g_\tau^2}} g_t$$

其中，$\rho$ 是初始的学习率，$g_\tau \in R^{|\theta|}$ 是第 $\tau$ 次迭代时的梯度。随着迭代次数的增加，梯度逐渐缩小

# 开发集

在梯度下降训练的过程中，由于过拟合的原因，在训练样本上收敛的参数，并不一定在测试集上最优。因此，我们使用一个**验证集**（ValidationDataset）（也叫**开发集**（Development Dataset））来测试每一次迭代的参数在验证集上是否最优。如果在验证集上的错误率不再下降，就停止迭代。如果没有验证集，可以在训练集上进行**交叉验证**



# 机器学习问题类型

回归（Regression） $y$ 是连续值（实数或连续整数），$f(x)$ 的输出也是连续值。这种类型的问题就是回归问题。对于所有已知或未知的 $(x, y)$，使得 $f(x,\theta)$ 和 $y$ 尽可能地一致。损函数通常定义为平方误差。

分类（Classification） $y$ 是离散的类别标记（符号），就是分类问题。损失函数有一般用 0-1 损失函数或负对数似然函数等。在分类问题中，通过学习得到的决策函数 $f(x,\theta)$ 也叫分类器。

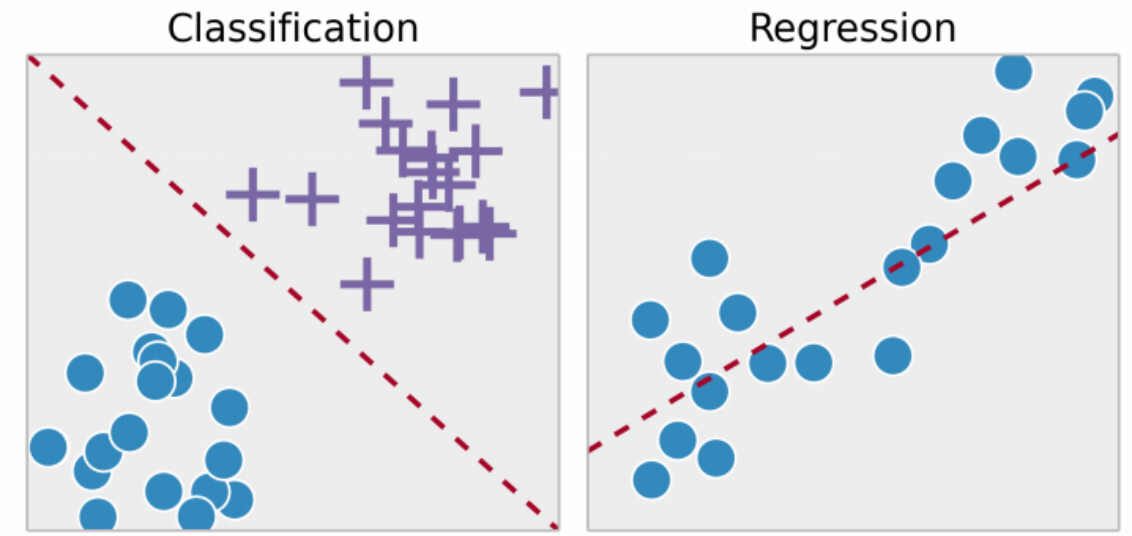
# 机器学习算法类型

**有监督学习（Supervised Learning）**是利用一组已知输入 $x$ 和输出 $y$ 的数据来学习模型的参数，使得模型预测的输出标记和真实标记尽可能的一致

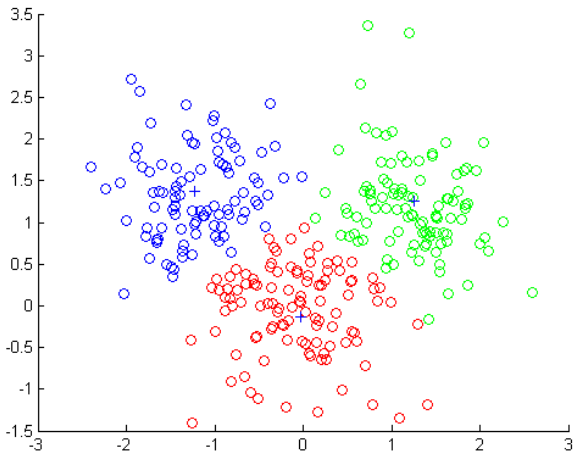
**半监督学习（Semi-Supervised Learning）**是利用少量已知输入 $x$ 和输出 $y$ 的数据以及未标注的样本，来学习模型的参数

**无监督学习（Unsupervised Learning）** 用来学习的数据不包含标注信息，需要学习算法自动学习到一些有价值的信息，例如**聚类**(Clustering）

# 分类、回归和聚类




Clustering



# 机器学习分类

- 主动学习（Active Learning）
- 集成学习（Ensemble Learning）
- 迁移学习（Transfer Learning）
- 多任务学习（Multi-task Learning）
- 强化学习（Reinforcement Learning）
- 终生学习（Life-long Learning）
- 课程学习（Curriculum Learning）
- 零样本学习 (One/zero shot Learning)
- ……

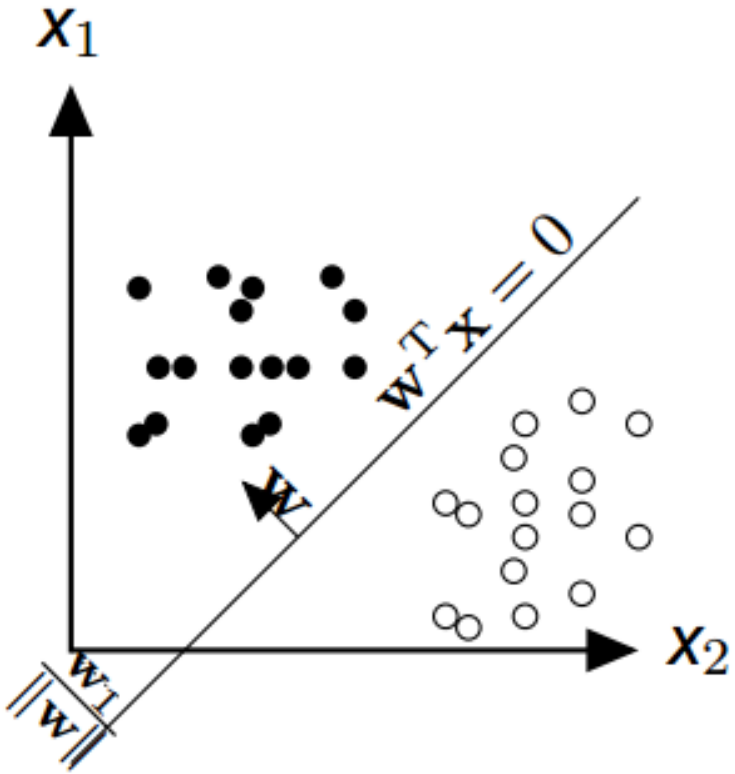
# 特征表示：特征选择、特征抽取

- 主成分分析
- 流形学习
- 核方法
- 主题模型
- 度量学习
- Embedding
- ....

# 线性分类

线性分类是机器学习中最常见并且应用最广泛的一种分类器。

$$\hat{y} = \begin{cases} 1 & if \quad w^T x > 0 \\ 0 & if \quad w^T x \le 0 \end{cases} = l\left(w^T x\right)$$



# Logistic Regression

**Logistic 回归**

我们定义目标类别 $y = 1$ 的后验概率为：

$$P(y = 1|x) = \sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)}$$

其中，$\sigma(\cdot)$ 为 logistic 函数，x 和 w 为增广的输入向量和权重向量。

$y = 0$ 的后验概率为

$$P(y = 0|x) = 1 - P(y = 1|x) = \frac{\exp(-w^T x)}{1 + \exp(-w^T x)}$$

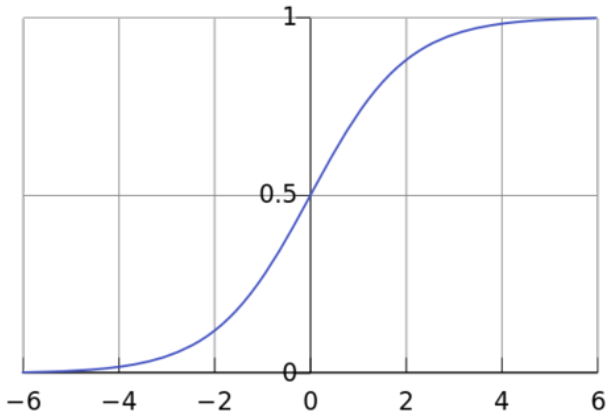
# Logistic 函数

logistic 函数经常用来将一个实数空间的数映射到 (0,1) 区间，记为 $\sigma(x)$

$$\sigma(x)=\frac{1}{1+e^{-x}}$$

其导数为

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

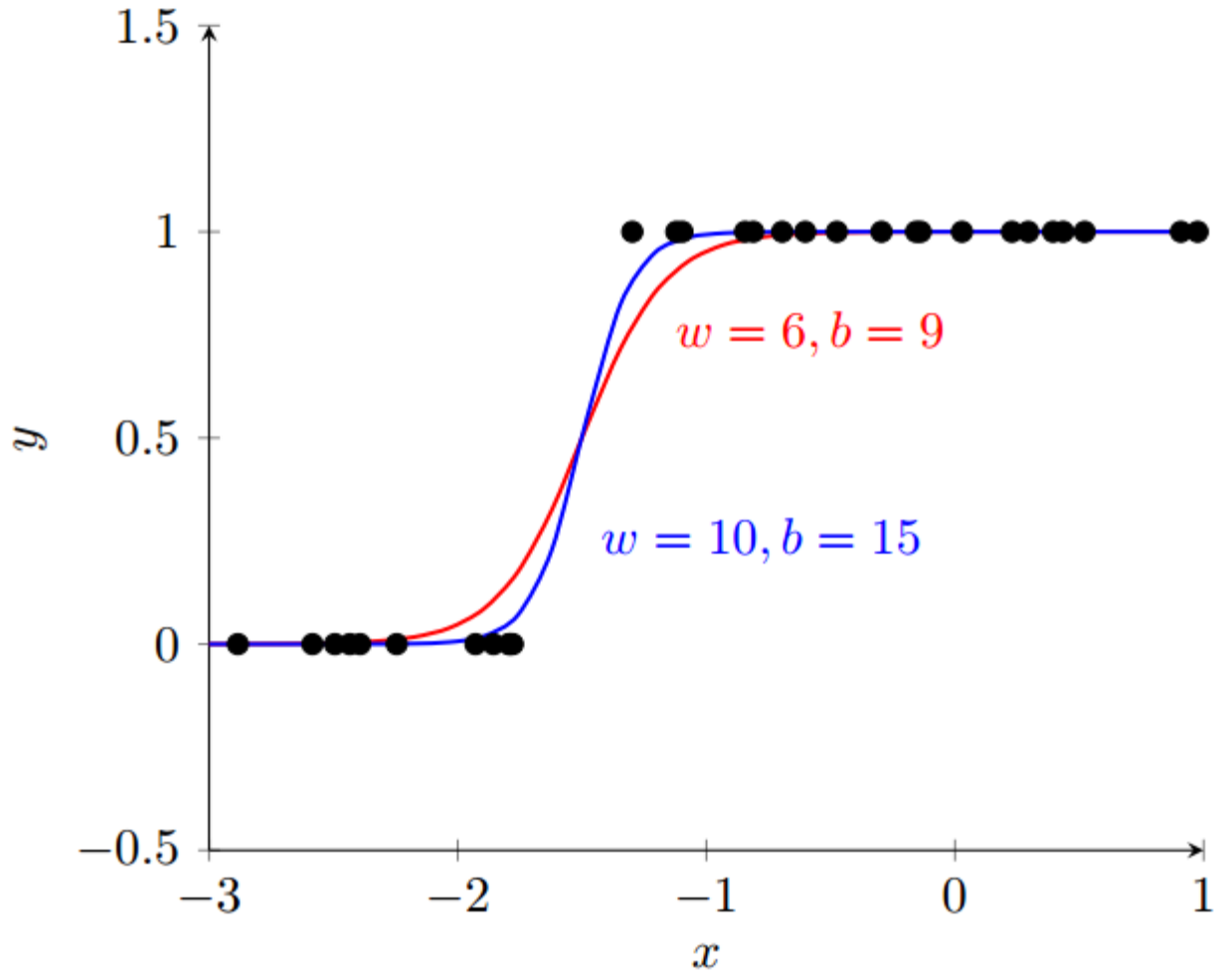



当输入为 $K$ 维向量 $x = [x_1, \cdots, x_K]^T$ 时，其导数为

$$\sigma'(x) = \mathrm{diag}(\sigma(x) \odot (1 - \sigma(x)))$$

# Logistic Regression

$y = \sigma(w\,x + b)$



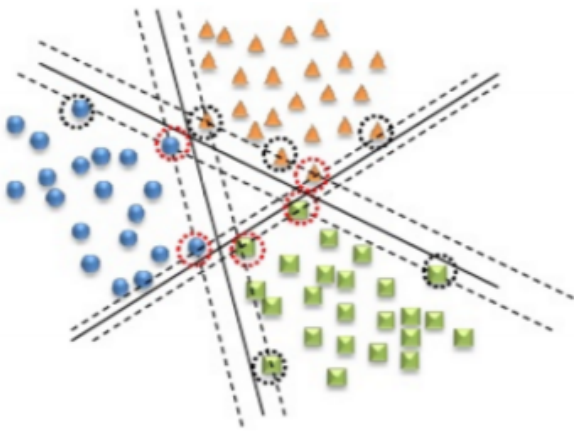
# 多类线性分类

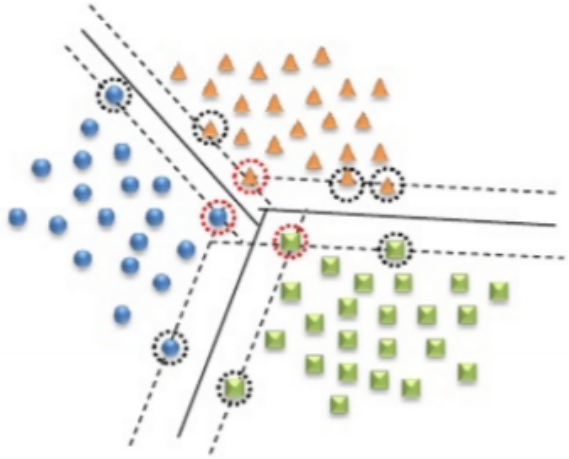
对于多类分类问题（假设类别数为 $C(C > 2)$），一般有两种多类转两类的转换方式：

- 把多类分类问题转换为 C 个两类分类问题，构建 C 个一对多的分类器。每个两类分类问题都是把某一类和其他类用一个超平面分开

- 把多类分类问题转换为 C(C – 1)/2 个两类分类问题，构建C(C – 1)/2 个两两分类器。每个两类分类问题都是把 C 类中某两类用一个超平面分开。

缺陷：一起区域中，点的类别是不能区分确定的





# 多类线性分类

为了避免上述缺陷，可以使用一个更加有效的决策规则，直接建立多类线性分类器。假设 $y = \{1,\cdots,C\}$ 共 $C$ 个类别，首先定义 $C$ 个判别函数：

$$f_c(x) = w_c^T x, \qquad c = 1,\cdots,C$$

这里$w_c$为类 $c$ 的权重向量。对于空间中的一个点 x，如果存在类别 c，对于所有的其他类别 $\tilde{c}(w_c^T x \neq c)$ 都满足$f_c(x) > f_{\tilde{c}}(x)$，那么 x 属于类别 c。相应的分类函数可以表示为：

$$\hat{y} = \arg\max_{c=1}^{C} w_c^T x$$

# Softmax 回归

SoftMax 回归是 Logistic 回归的多类推广。

我们定义目标类别 $y = c$ 的后验概率为：

$$P\left(y=c \mid x\right)=soft\max\left(w_c^T x\right)=\frac{\exp\left(w_c^T x\right)}{\sum_{i=1}^{C}\exp\left(w_i^T x\right)}$$

# 评价方法

常见的评价标准有正确率、准确率、召回率和 F 值等。

给定测试集 $T = (x_1,y_1),\cdots,(x_N,y_N)$，对于所有的 $y_i \in \{\omega_1,\cdots,\omega_C\}$。
假设分类结果为 $Y = \widehat{y_1},\cdots,\widehat{y_N}$。
则**正确率**（Accuracy，Correct Rate）为：

$$Acc = \frac{\sum_{i=1}^{N} \left| y_i = \hat{y}_i \right|}{N}$$

其中，$|\cdot|$ 为指示函数
和正确率相对应的就是**错误率**（Error Rate）。

$$Err = \frac{\sum_{i=1}^{N} \left| y_i \neq \hat{y}_i \right|}{N}$$

正确率是平均的整体性能。

# 评价方法

在很多情况下，我们需要对每个类都进行性能估计，这就需要计算准确率和召回率。正确率和召回率是广泛用于信息检索和统计学分类领域的两个度量值，在机器学习的评价中也被大量使用。

**准确率**（Precision，P），是识别出的个体总数中正确识别的个体总数的比例。对于类 $c$ 来说，

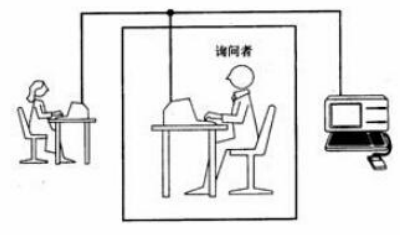
$$P_c = \frac{\sum_{\substack{i=1 \\ \hat{y}_i=c}}^{N} \left| y_i = \hat{y}_i \right|}{\sum_{\substack{i=1 \\ \hat{y}_i=c}}^{N} 1}$$

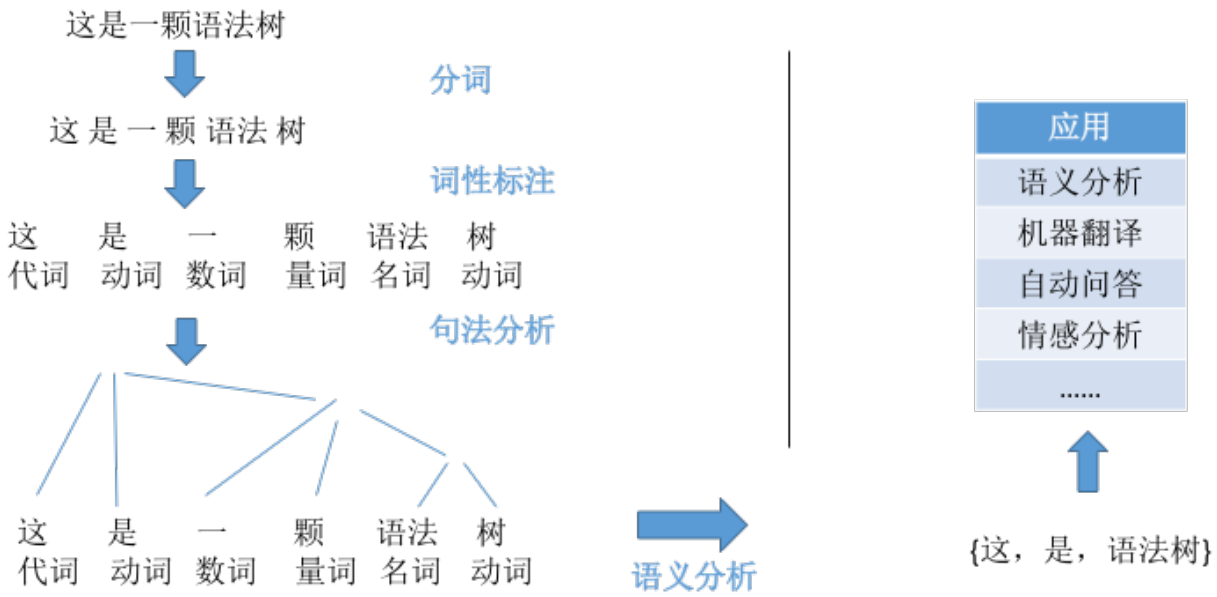
**召回率**（Recall，R），也叫查全率，是测试集中存在的个体总数中正确识别的个体总数的比例。

$$R_c = \frac{\sum_{\substack{i=1 \\ y_i=c}}^{N} \left| y_i = \hat{y}_i \right|}{\sum_{\substack{i=1 \\ y_i=c}}^{N} 1}$$

# 自然语言处理

- 让机器理解自然语言



# 自然语言难点

- 分词
  - 中国人为了实现自己的梦想
    - 中国/ 人为/ 了/ 实现/ 自己/ 的/ 梦想
    - 中国人/ 为了/ 实现/ 自己/ 的/ 梦想
    - 中/ 国人/ 为了/ 实现/ 自己/ 的/ 梦想
  - 其他例子
    - “大学生”、“研究生物”、“从小学起”、“为人民工作”、“中国产品质量”、“部分居民生活水平”、“南京市长江大桥”等等

- 指代消解
  - 我们把香蕉给猴子，因为它们饿了
  - 我们把香蕉给猴子，因为它们熟透了

# 自然语言难点

## 更困难的例子

冬天，能穿多少穿多少；夏天，能穿多少穿多少。
剩女的原因：一是谁都看不上，二是谁都看不上。
单身的来由：原来是喜欢一个人，现在是喜欢一个人。
女致电男友：地铁站见。如果你到了我还没到，你就等着吧。如果我到了你还没到，你就等着吧！！

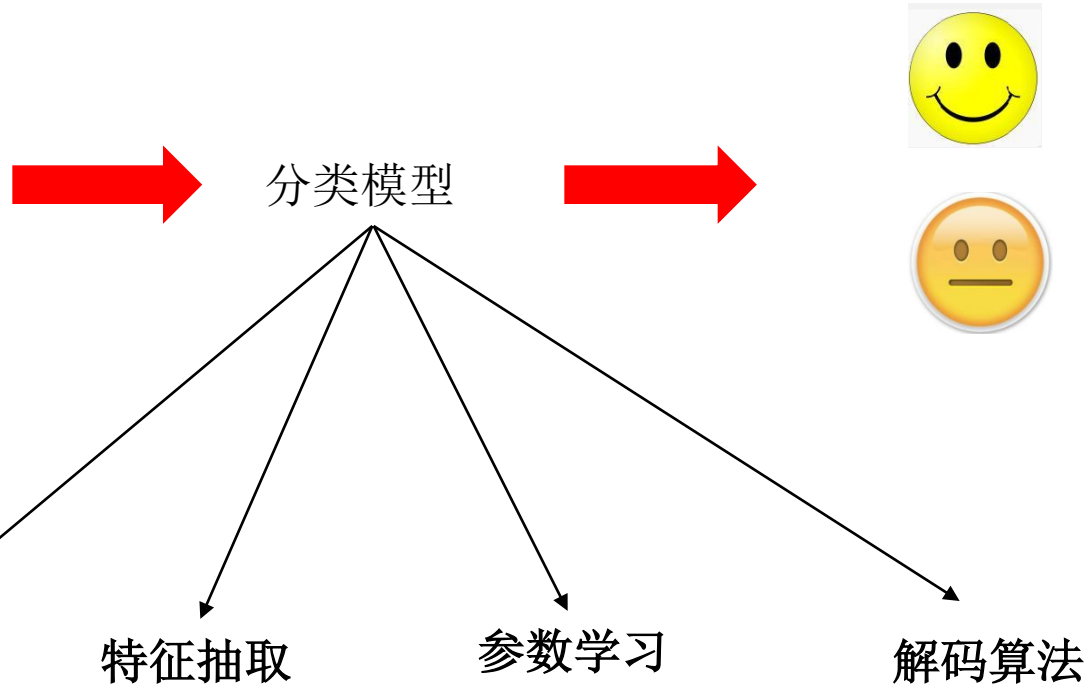
大舅去二舅家找三舅说四舅被五舅骗去六舅家偷七舅放在八舅柜子里九舅借十舅发给十一舅工资的1000元

问：
1、究竟谁是小偷？
2、钱本来是谁的？

# 情感分类：分类问题

这个电影很好看。

这个电影太糟糕了。



# 情感分类

*诺基亚5800屏幕很好，操作也很方便，通话质量也不错，*

| | |
|---|---|
| *诺基亚* | *1* |
| *5800* | *1* |
| *屏幕* | *1* |
| *很好* | *1* |
| *操作* | *1* |
| *也* | *2* |
| *很* | *1* |
| *方便* | *1* |
| *通话* | *1* |
| *质量* | *1* |
| *不错* | *1* |

*(1,0,0,1,0,1,1.....,0,1)*

分类器

# 中文分词：分类问题

| 自 | ： | 然 | ： | 语 | ： | 言 | ： | 处 | ： | 理 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | | 1 | | 0 | | 1 | | 0 | |

| 窗口大小 | 样本$x$ | 类别标签$y$ |
|---|---|---|
| 2 | “自:然” | 0 |
| | “然:语” | 1 |
| | “语:言” | 0 |
| 4 | “自然:语言” | 1 |
| | “然语 :言处” | 0 |
| | “语言:处理” | 1 |

| | |
|---|---|
| 单字符串特征 | $x_{-2}y_0, x_{-1}y_0, x_0y_0, x_1y_0, x_2y_0$ |
| 双字符串特征 | $x_{-1}x_0y_0, x_0x_1y_0, x_{-1}x_1y_0,$ |
| 三字符串特征 | $x_{-1}x_0x_1y_0$ |
| 马氏链特征 | $y_{-1}y_0$ |

[000010001000100011001]

1/0

# 中文分词：分类问题

上海/ 计划/ 到/ 本/ 世纪/ 末/ 实现/ 人均/ 国内/ 生产/ 总值/ 五千美元/ 。/

上/B 海/E 计/B 划/E 到/S 本/S 世/B 纪/E 末/S 实/B 现/E 人/B 均/E 国/B
内/E 生/B 产/E 总/B 值/E 五/B 千/M 美/M 元/E 。/S

B:词的开始字符
M:词的中间字符
E:词的结尾字符
S:单字符词

# 深度学习和神经网络基础

# Machine Learning

Representation

Objective

Optimization

# Traditional Machine Learning

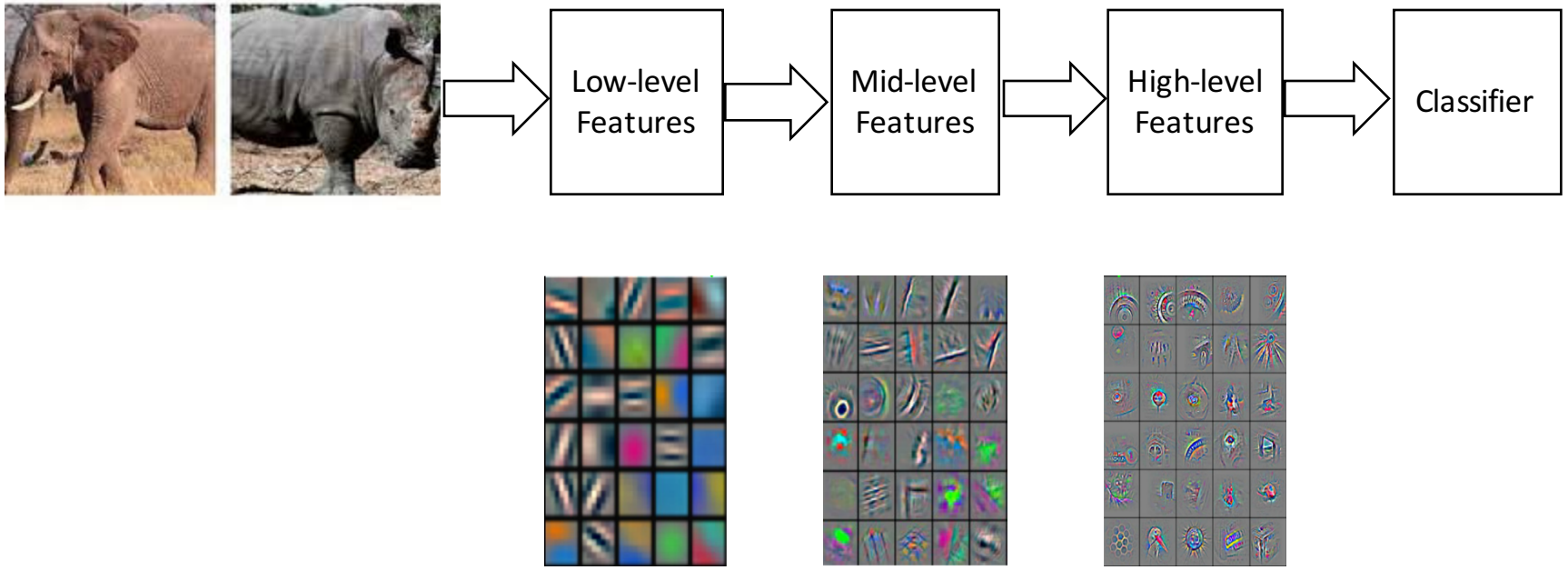
- 人工特征工程+分类器

特征抽取
（Segmentation、PCA、Shape）

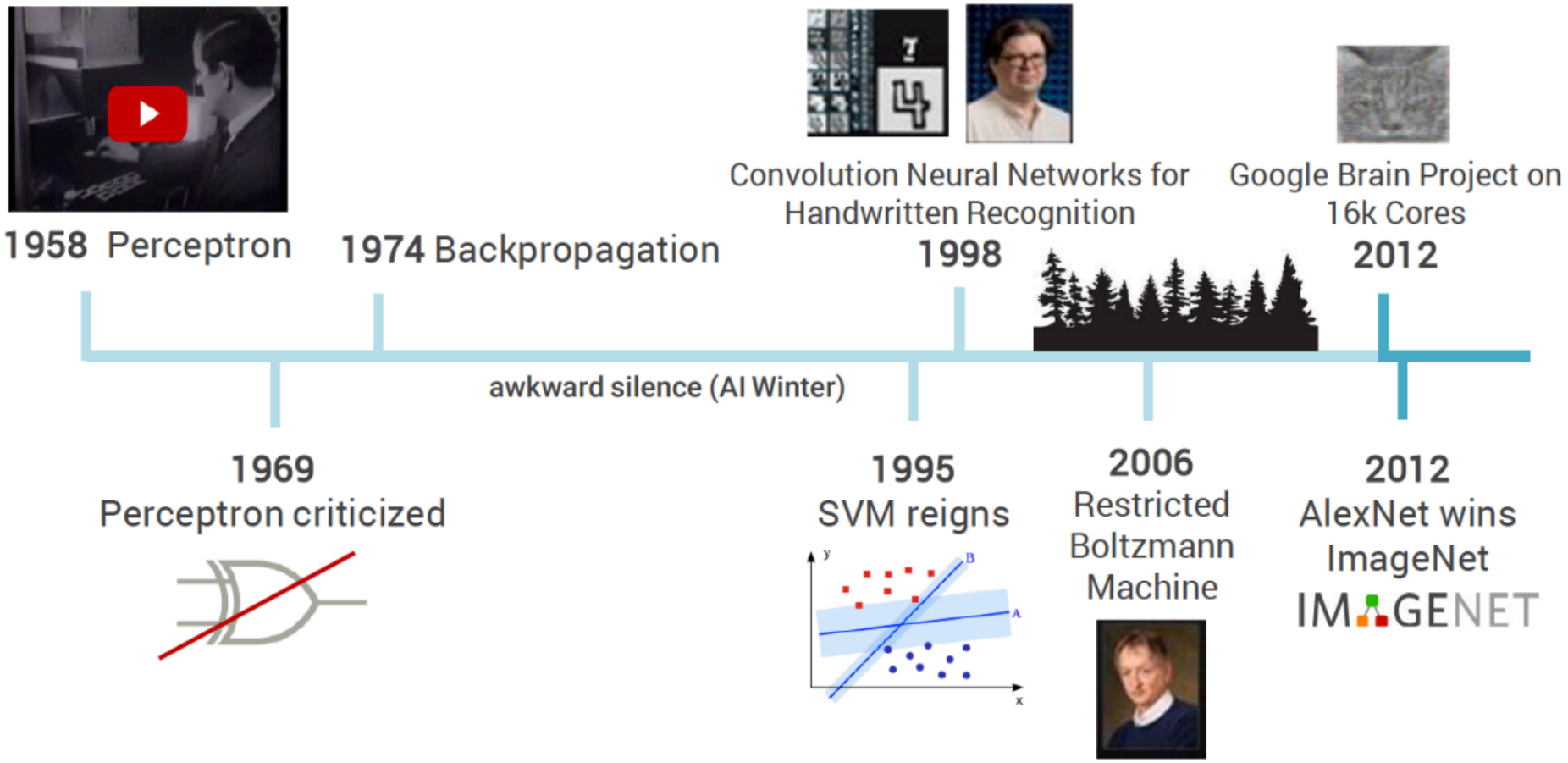
分类器
（SVM、NB、Maximum Entropy、CRF）

# Deep Learning

- 自动学习多尺度的特征表示



# History


1958 Perceptron
1974 Backpropagation
Convolution Neural Networks for Handwritten Recognition
1998
Google Brain Project on 16k Cores
2012
awkward silence (AI Winter)
1969
Perceptron criticized
1995
SVM reigns
2006
Restricted Boltzmann Machine
2012
AlexNet wins ImageNet
IMAGENET

# 目录

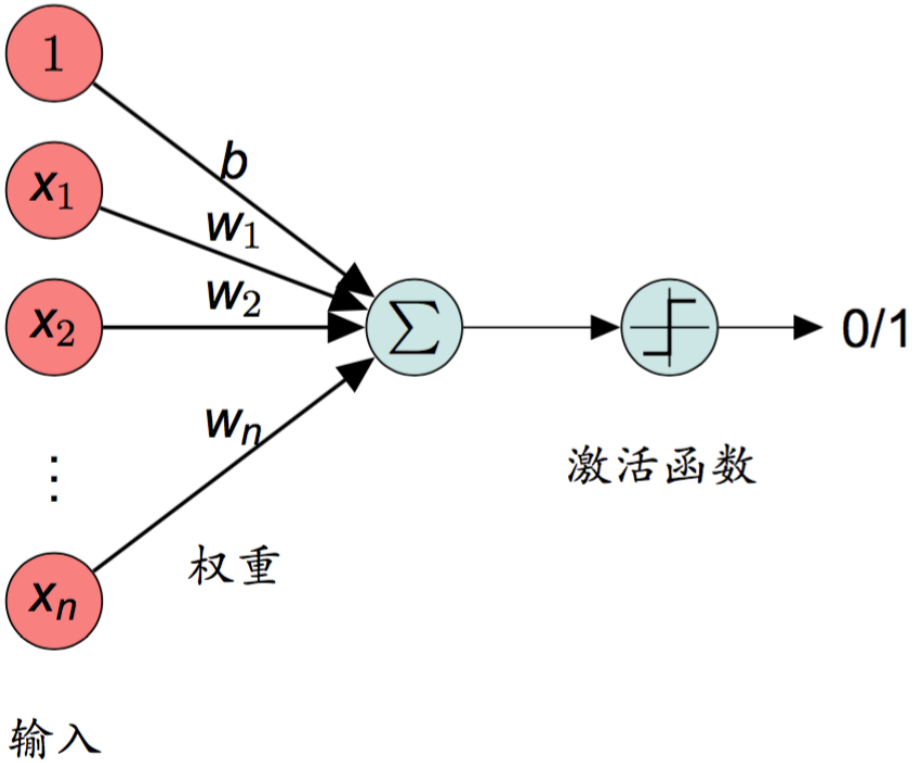
- 感知机
- 前馈神经网络

# 感知机 Perceptron

# 感知机 Perception

- 感知器是对生物神经细胞的简单数学模拟,是最简单的人工神经网络，只有一个神经元。 感知器也可以看出是线性分类器的一个经典学习算法。
- 细胞体(Soma)中的神经细胞膜上有各种受体和离子通道,胞膜的受体可与相应的化学物质神经递质结合,引起离子通透性及膜内外电位差发生改变,产生相应的生理活动:兴奋或抑制。 细胞突起是由细胞体延伸出来的细长部分,又可分为**树突和轴突**。
  - 树突(Dendrite)可以接受刺激并将兴奋传入细胞体。每个神经元可以有一或多个树突。
  - 轴突 (Axons) 可以把兴奋从胞体传送到另一个神经元或其他组织。 每个神经元只有一个轴突。
- 抑制与兴奋
  - 神经 细胞的状态取决于从其它的神经细胞收到的输入信号量,及突触的强度(抑制或加强)。当信号量总和超过了某个阈值时,细胞体就会兴奋,产生电脉冲。电脉冲沿着轴突并通过突触传递到其它神经元。

# 感知机




$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

# 感知机

- 给定输入 $x=(x_1,x_2,x_3,\cdots x_n)$

$$y=f(x)=sign(w\bullet x+b)$$

$$sign(x)=\begin{cases}+1, & x\geq 0\\ -1, & x<0\end{cases}$$

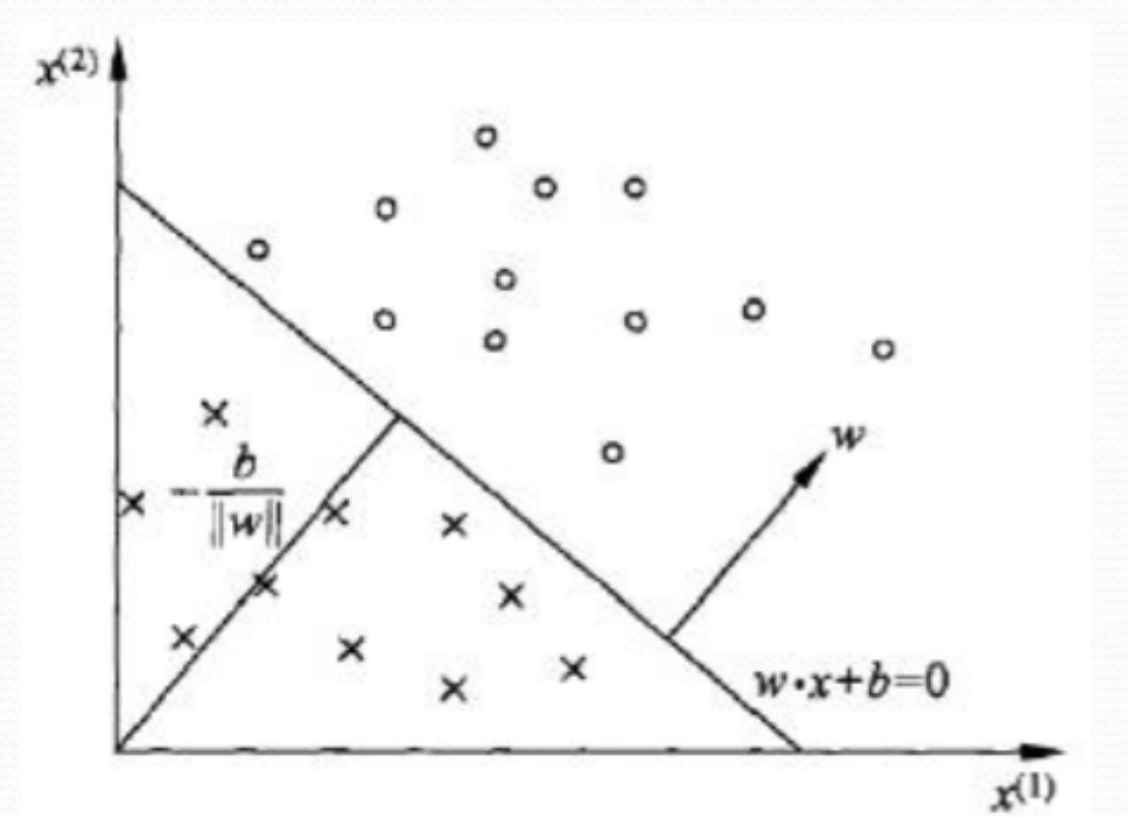
$$y=f(x)=sign(w\bullet x)$$

$$x=(b,x_1,x_2,x_3,\cdots x_n)$$

$$w=\{1,w_1,w_2,w_3,\cdots w_n\}$$

# 感知机几何解释

$$w \cdot x + b = 0$$



# 感知机参数学习

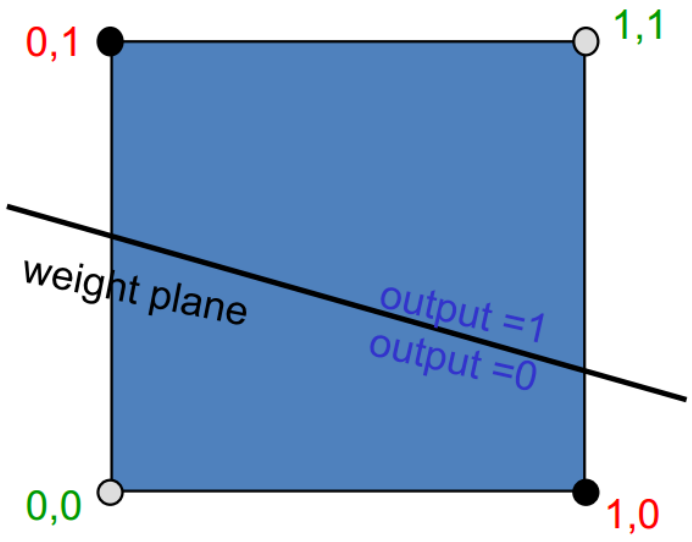
$$L(w,b)=\frac{1}{2}\sum_{x_{\mathrm{i}}\in M}(y_i-(w\cdot x_{\mathrm{i}}+b))^2 \qquad \widetilde{y_i}=w\cdot x_{\mathrm{i}}+b$$

$$\frac{\partial L(w,b)}{\partial w}=-\sum_{x_{\mathrm{i}}\in M}(y_i-\widetilde{y_i})x_{\mathrm{i}} \qquad \frac{\partial L(w,b)}{\partial b}=-\sum_{x_{\mathrm{i}}\in M}(y_i-\widetilde{y_i})$$

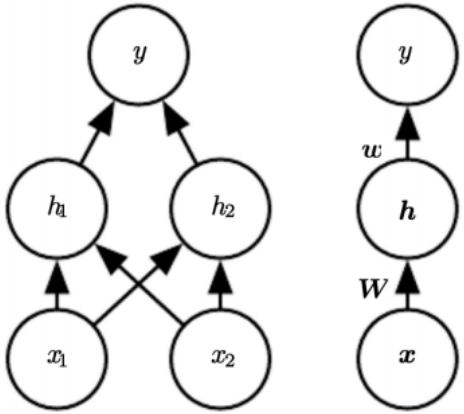
$$w\leftarrow w+\lambda\sum_{x_{\mathrm{i}}\in M}(y_i-\widetilde{y_i})x_{\mathrm{i}} \qquad b\leftarrow b+\lambda\sum_{x_{\mathrm{i}}\in M}(y_i-\widetilde{y_i})$$

# The Problems of Perception

- 异或问题 1969, Minsky and Papert



# Non-linear Hidden Layer

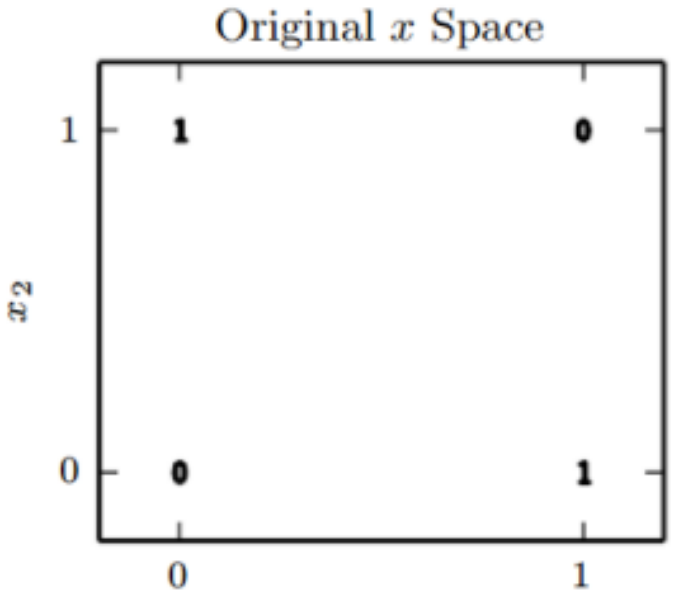



$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

$$b = 0.$$




$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b$$

# Example

- [0,0]
  - $\max(0, \left(\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0\\-1\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}$
  - $\begin{bmatrix}1\\-2\end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix}0\\0\end{bmatrix} + 0 = 0$
- [1,0]
  - $\max(0, \left(\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}$
  - $\begin{bmatrix}1\\-2\end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix}1\\0\end{bmatrix} + 0 = 1$
- [0,1]
  - $\max(0, \left(\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}$
  - $\begin{bmatrix}1\\-2\end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix}1\\0\end{bmatrix} + 0 = 1$
- [1,1]
  - $\max(0, \left(\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0\\-1\end{bmatrix}\right) = \begin{bmatrix}2\\1\end{bmatrix}$
  - $\begin{bmatrix}1\\-2\end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix}2\\1\end{bmatrix} + 0 = 0$

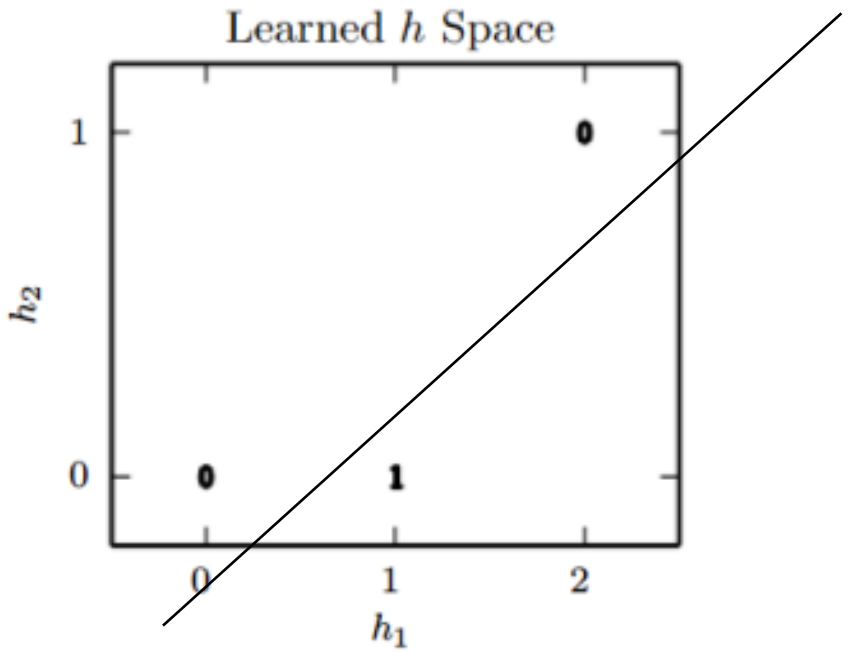
$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b$$

$$\boldsymbol{W} = \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix},$$

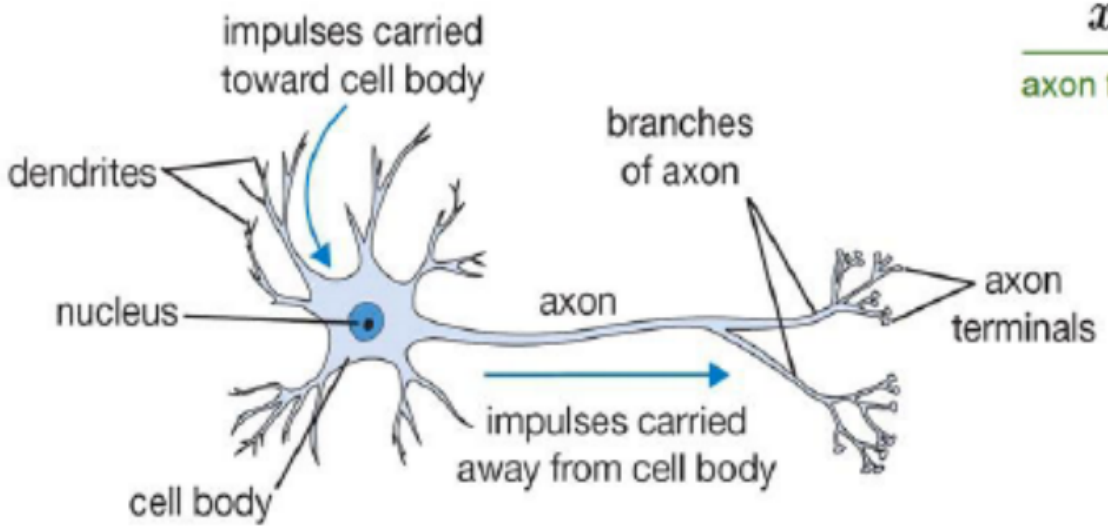
$$\boldsymbol{c} = \begin{bmatrix}0\\-1\end{bmatrix},$$

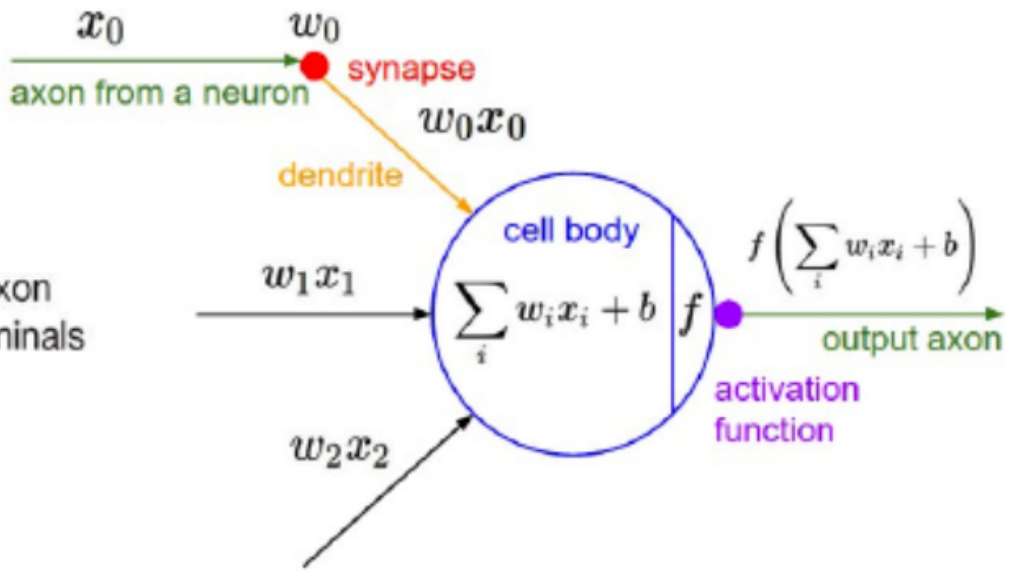
$$\boldsymbol{w} = \begin{bmatrix}1\\-2\end{bmatrix},$$

$$b = 0.$$



# Non-linear Neurons

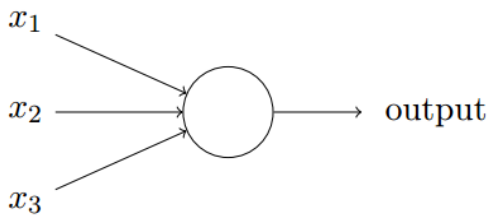



$$z = w \cdot x + b$$

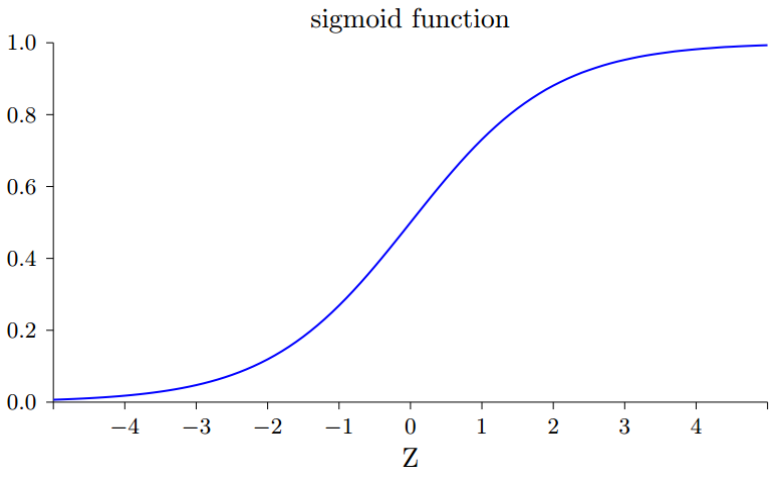
$$y = f(z)$$

# Sigmoid




Sigmoid Function

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-\sum_j w_j\, x_j - b)}$$

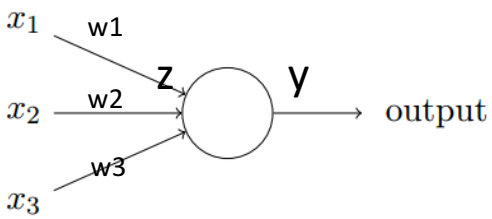


# Training

$$E = \frac{1}{2} \sum_{output} (t - y)^2$$

$$y = \frac{1}{1+e^{-z}}$$

$$z = \sum_i w_i x_i$$




$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial z} \frac{\partial z}{\partial w_i}$$

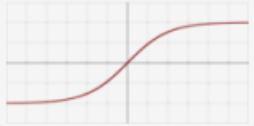
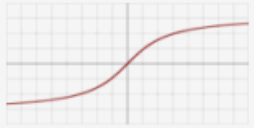
$$\frac{\partial E}{\partial y} = -(t - y)$$

$$\frac{\partial y}{\partial z} = y(1 - y)$$

$$\frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial E}{\partial w_i} = -(t - y)\, y(1 - y) x_i$$

# Other Activation Functions

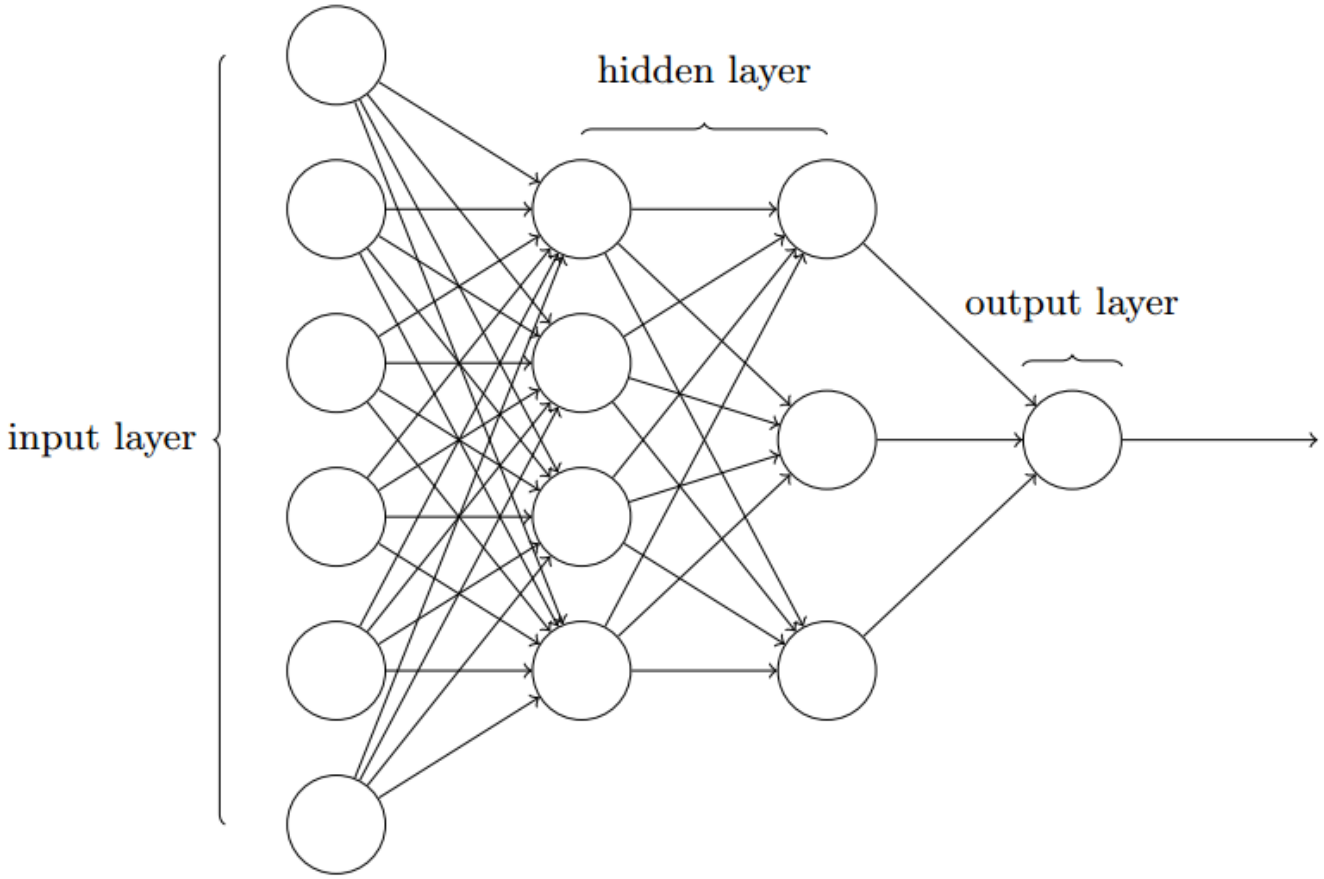
| | | | |
|---|---|---|---|
| Logistic (a.k.a Soft step) | | $f(x) = \frac{1}{1+e^{-x}}$ | $f'(x) = f(x)(1-f(x))$ |
| TanH |  | $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$ | $f'(x) = 1 - f(x)^2$ |
| ArcTan |  | $f(x) = \tan^{-1}(x)$ | $f'(x) = \frac{1}{x^2+1}$ |
| Rectified Linear Unit (ReLU)[7] | | $f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$ |
| Parameteric Rectified Linear Unit (PReLU)[8] | | $f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$ |

rectifier($x$) = max(0, $x$)

rectifier 函数被认为有生物上的解释性。神经科学家发现神经元具有单侧抑制、宽兴奋边界、稀疏激活性等特性。采用 rectifier 函数的单元也叫作正线性单元(rectified linear unit, ReLU)

# 前馈神经网络

# 前馈神经网络 Feed Forward Neural Networks



# 前馈计算

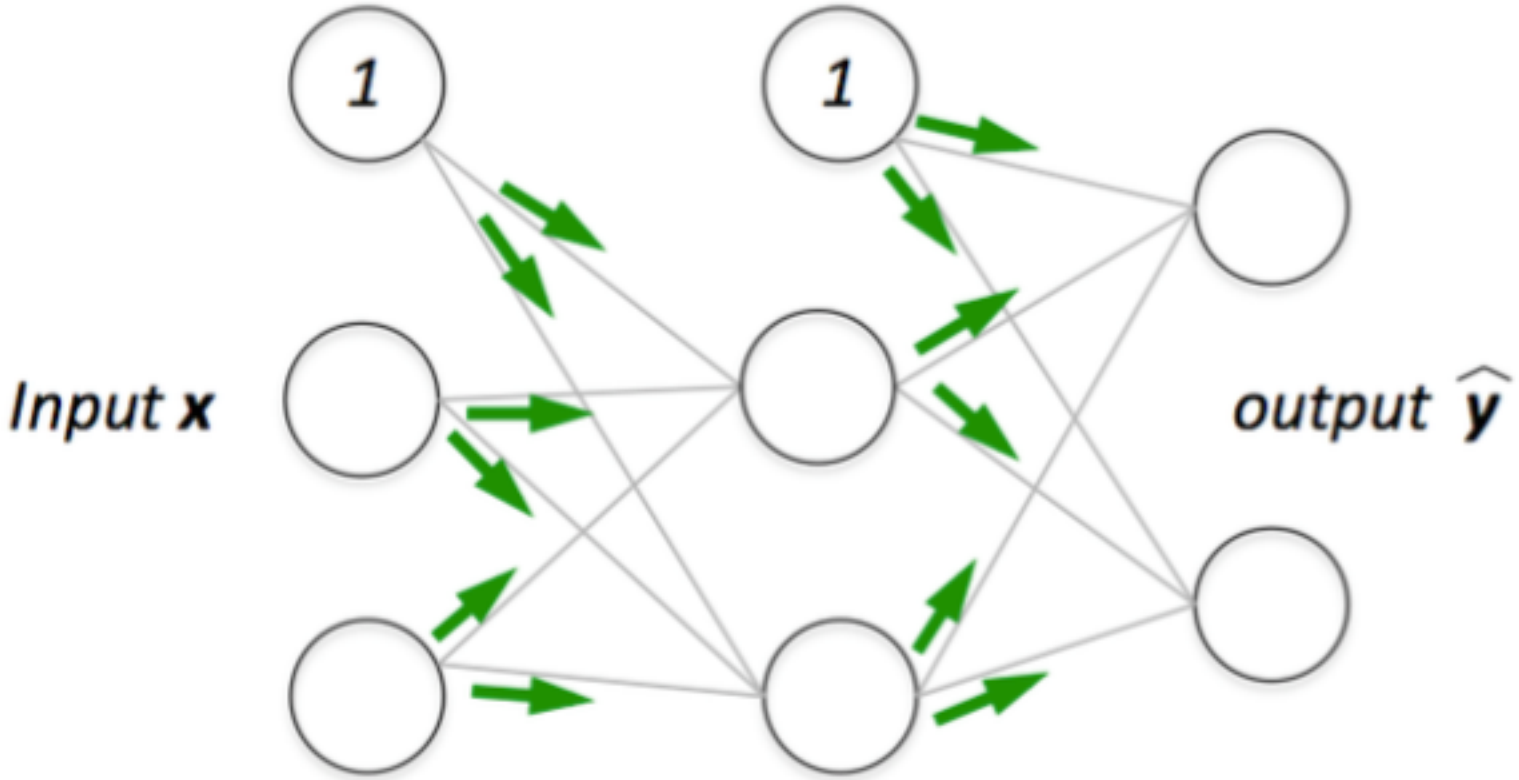
- $L$表示神经网络的层数
- $n^l$表示第$l$层神经元的个数
- $f_l(z)$表示第$l$层的激活函数
- $\mathrm{w}^l$表示第$l$层的权重
- $\mathrm{b}^l$表示第$l$层神经元的偏置
- $\mathrm{z}^l$表示第$l$层神经元的状态
- $\mathrm{y}^l$表示第$l$层神经元的输出

$$\mathrm{z}^l = \mathrm{w}^l \cdot \mathrm{y}^{l-1} + \mathrm{b}^l$$

$$\mathrm{y}^l = f_l(\mathrm{z}^l)$$

$$\mathrm{z}^l = \mathrm{w}^l \cdot f_l(\mathrm{z}^{l-1}) + \mathrm{b}^l$$

# 前馈计算

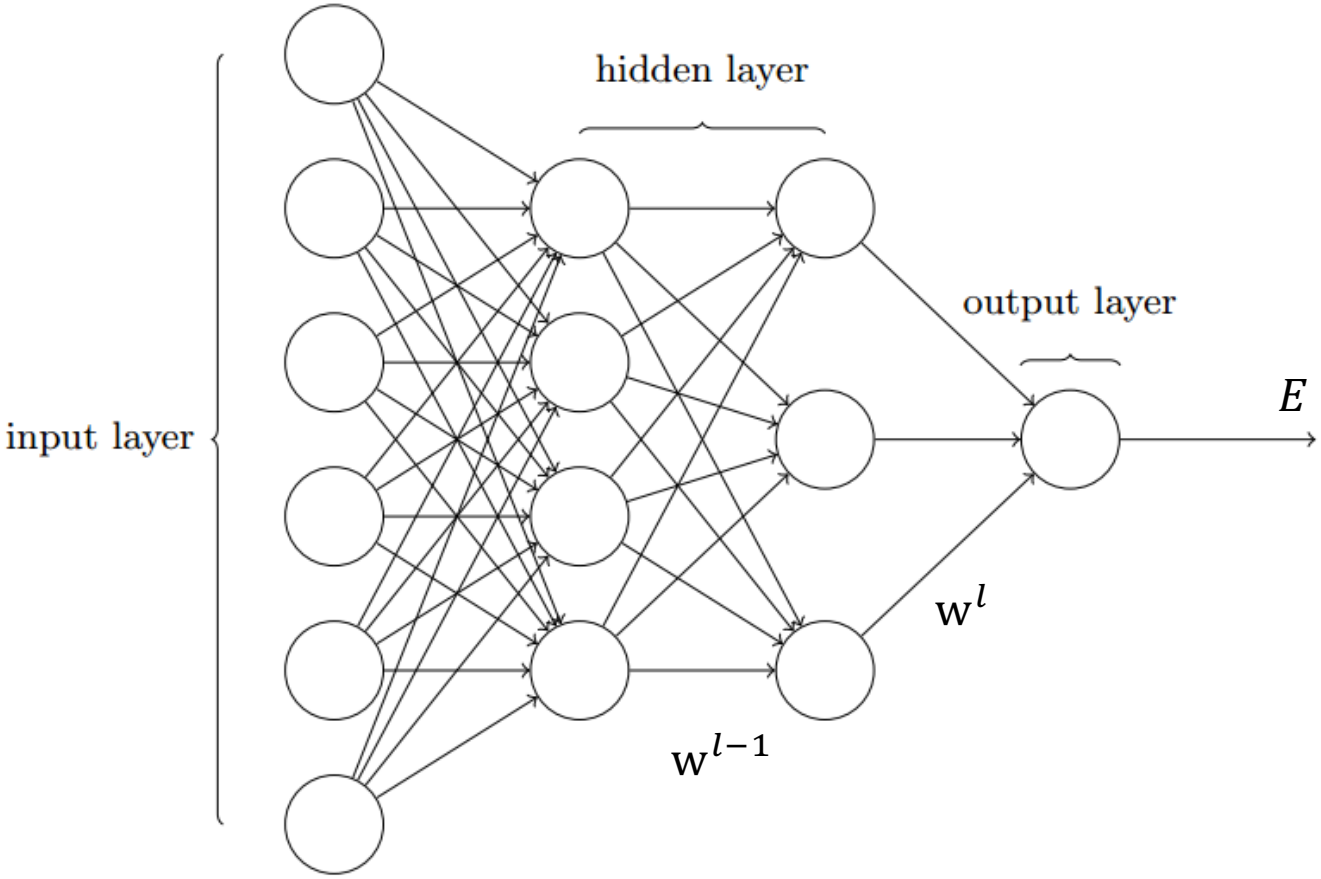



$$\mathrm{x} \rightarrow \mathrm{z}^1 \rightarrow f_1(\mathrm{z}^1) \rightarrow \mathrm{z}^2 \rightarrow f_2(\mathrm{z}^2) \rightarrow \ldots\ldots \rightarrow \mathrm{z}^l \rightarrow f_l(\mathrm{z}^l) \rightarrow \ldots\ldots \rightarrow \mathrm{z}^L \rightarrow f_L(\mathrm{z}^L) \rightarrow y$$

# 训练：梯度下降法

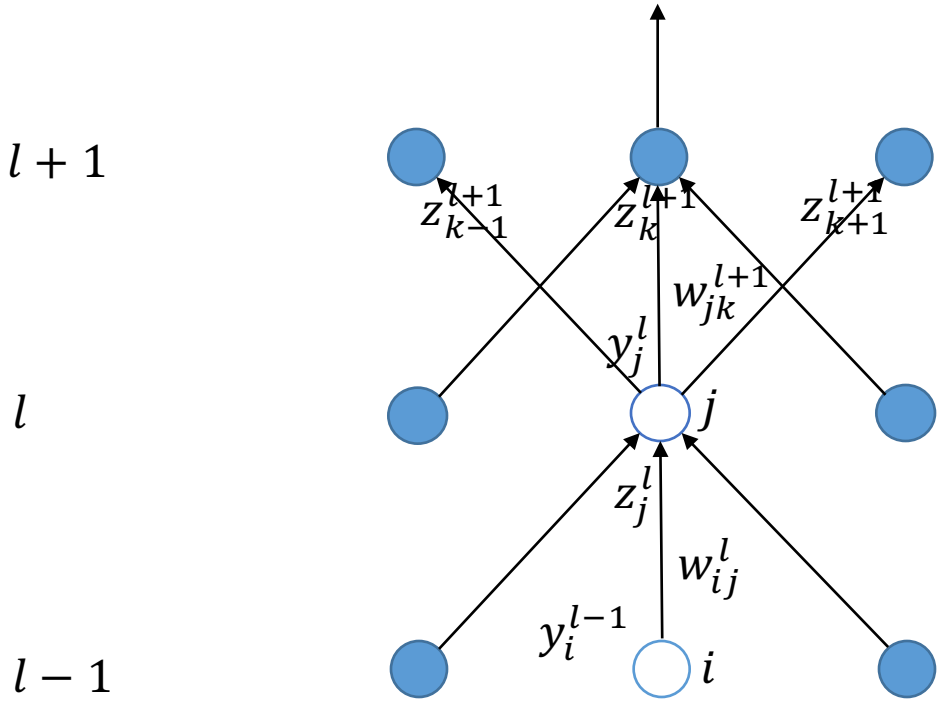
$$E = \frac{1}{2}\sum_{output}(t_j - y_j)^2$$

$$w_{ij} \leftarrow w_{ij} - \lambda\frac{\partial E}{\partial w_{ij}}$$




$$E = \frac{1}{2}\sum_{output}(t_j - y_j)^2$$

# 训练：Error Back Propagating




$$E = \frac{1}{2}\sum_{output}(t_j - y_j^L)^2$$

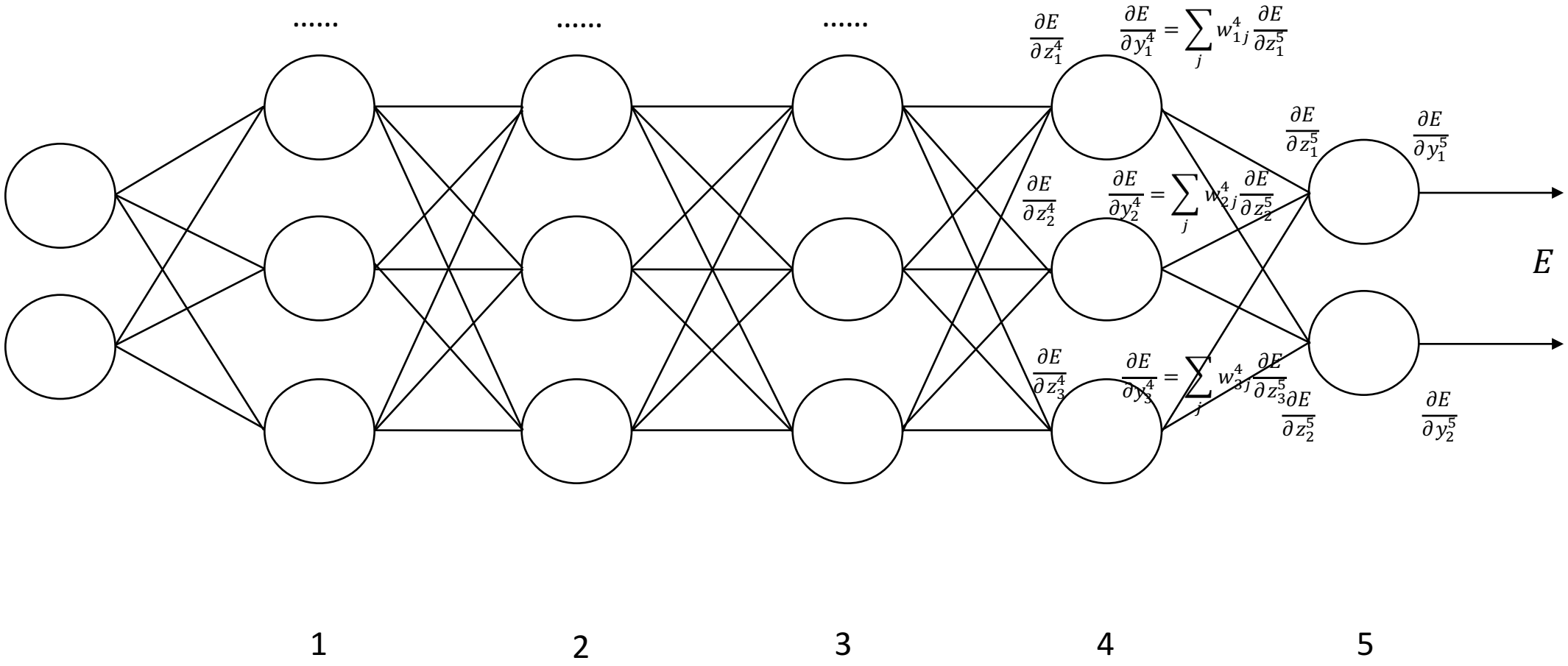
$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial z_j^l}{\partial w_{ij}^l}\frac{\partial E}{\partial z_j^l} = \frac{\partial E}{\partial z_j^l} y_i^{l-1}$$

$$\frac{\partial E}{\partial z_j^l} = \frac{\partial y_j^l}{\partial z_j^l}\frac{\partial E}{\partial y_j^l} = \frac{\partial E}{\partial y_j^l} y_j^l(1 - y_j^l)$$

$$\frac{\partial E}{\partial y_j^l} = \sum_k \frac{\partial z_k^{l+1}}{\partial y_j^l}\frac{\partial E}{\partial z_k^{l+1}} = \sum_k w_{jk}^{l+1}\frac{\partial E}{\partial z_k^{l+1}}$$

$$y = \frac{1}{1+e^{-z}} \qquad z = \sum_i w_i y_i$$

# 反向错误传播


.....
.....
.....
$\frac{\partial E}{\partial z_1^4}$
$\frac{\partial E}{\partial y_1^4} = \sum_j w_{1j}^4 \frac{\partial E}{\partial z_1^5}$
$\frac{\partial E}{\partial z_2^4}$
$\frac{\partial E}{\partial y_2^4} = \sum_j w_{2j}^4 \frac{\partial E}{\partial z_2^5}$
$\frac{\partial E}{\partial z_3^4}$
$\frac{\partial E}{\partial y_3^4} = \sum_j w_{3j}^4 \frac{\partial E}{\partial z_3^5}$
$\frac{\partial E}{\partial z_1^5}$
$\frac{\partial E}{\partial y_1^5}$
$\frac{\partial E}{\partial z_2^5}$
$\frac{\partial E}{\partial y_2^5}$
$E$
1
2
3
4
5

# 反向传播

- 在第$l$层神经元上的误差：$\delta^l = \frac{\partial E}{\partial z^l}$
- 各层神经元上的误差
  - $\delta^L = \frac{\partial E}{\partial y^L} \odot f'(z^L)$
  - $\delta^l = \left(w^{l+1}\delta^{l+1}\right) \odot f'\left(z^l\right)$

  - $\frac{\partial E}{\partial b^l} = \delta^l$
  - $\frac{\partial E}{\partial w^l} = y^{l-1}\delta^l$

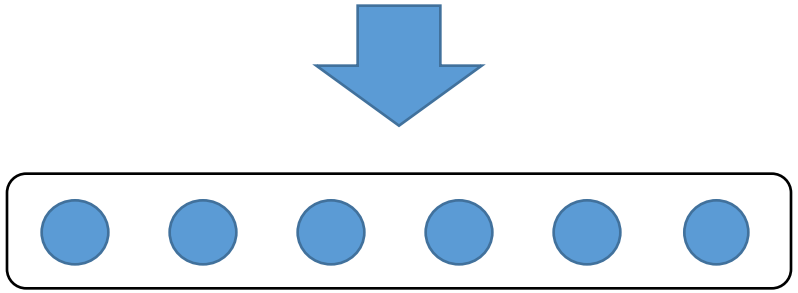
# Back Propagation Algorithm

- Input x
- Feed Forward
  - For each layer $l = 2,3,\dots,L$, compute $z^l = w^l y^{l-1}$ and $y^l = f(z^l)$
- Output Error
  - Compute $\delta^L = \nabla_y E \odot f'(z^l)$
- Back Propagating
  - For each $l = L-1, L-2, \dots, 2$ compute $\delta^l = ((w^{l+1})\delta^{l+1}) \odot f'(z^l)$
- Output
  - The gradient of the cost function is given by $\frac{\partial E}{\partial w^l} = y^{l-1}\delta^l$

# 卷积神经网络
# Convolutional Neural Network

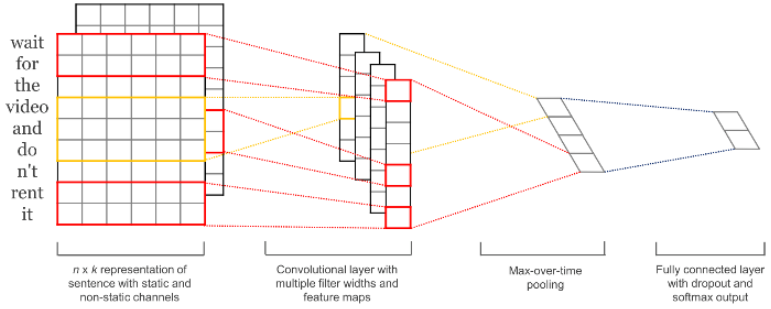
# Sequence Modeling

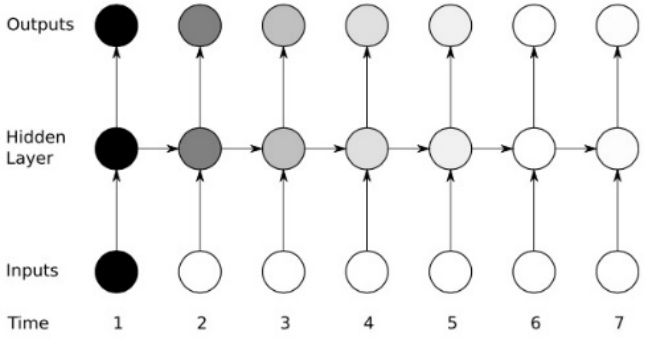
I want to play game with that little girl.

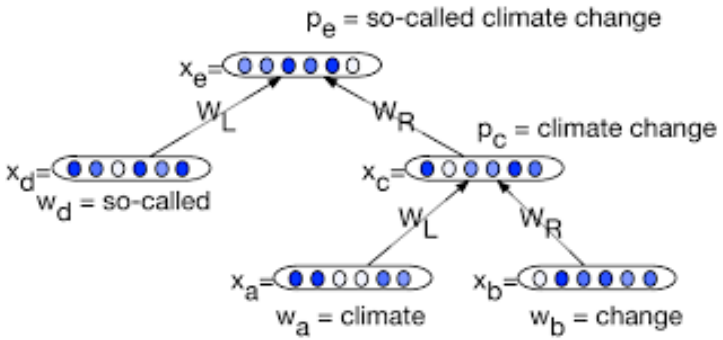


# NN for sentence modeling

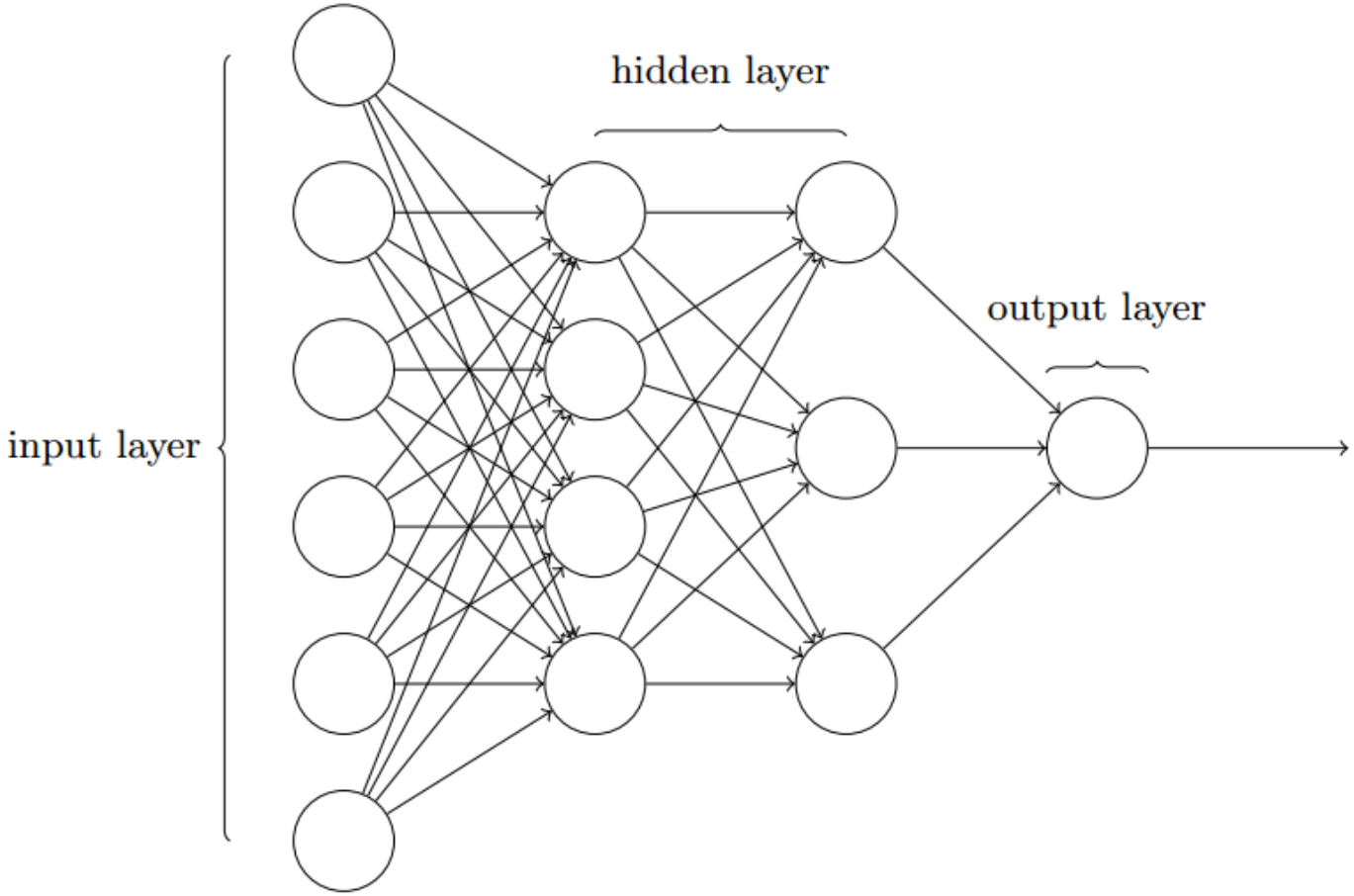
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Recursive Neural Network (RNN)



# Feed Forward Neural Networks



# 全连接

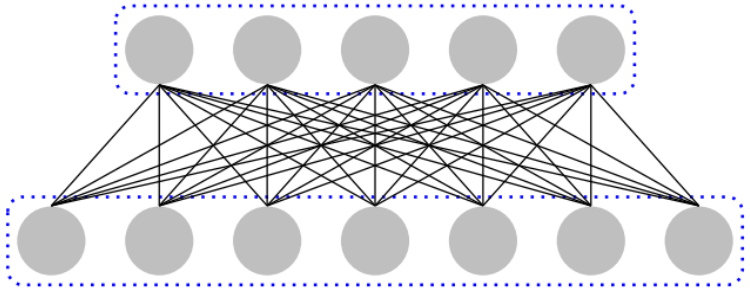
- $n^l$表示第$l$层神经元的个数
- $L$表示神经网络的层数
- 参数个数：$\prod_{l=1}^{L} n^l$
- 权重矩阵参数非常多，训练效率低下
- 数据不足时，欠学习

# CNN

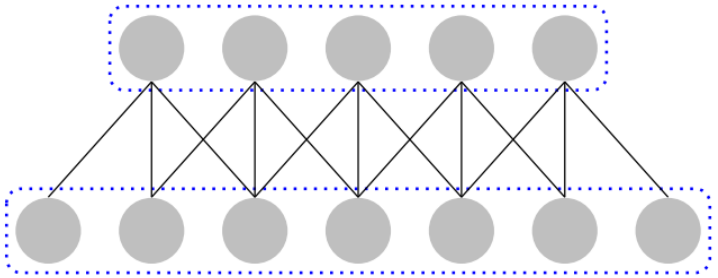
- Convolutional Neural Network是一种前馈神经网络。卷积神经网络是受生物学上感受野(Receptive Field) 的机制而提出的。一个神经元的感受野是指特定区域，只有这个区域内的刺激才能够激活该神经元。
  - 局部链接
  - 权值共享
  - 采样

- 具有平移、缩放和扭曲不变性

# 全连接 vs. 卷积



全连接

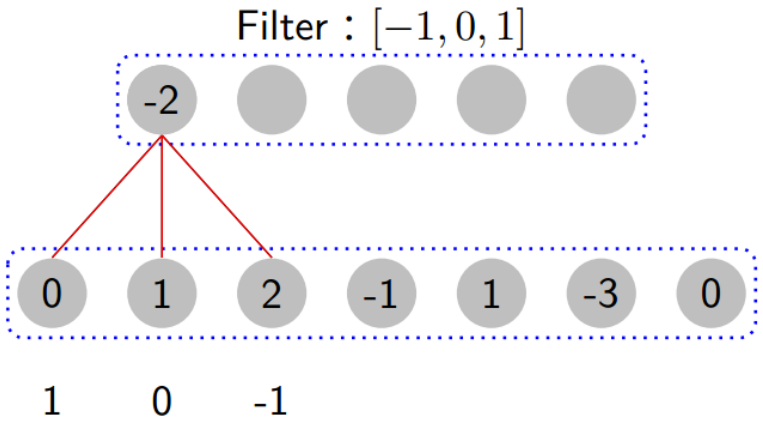


CNN

# 一维卷积

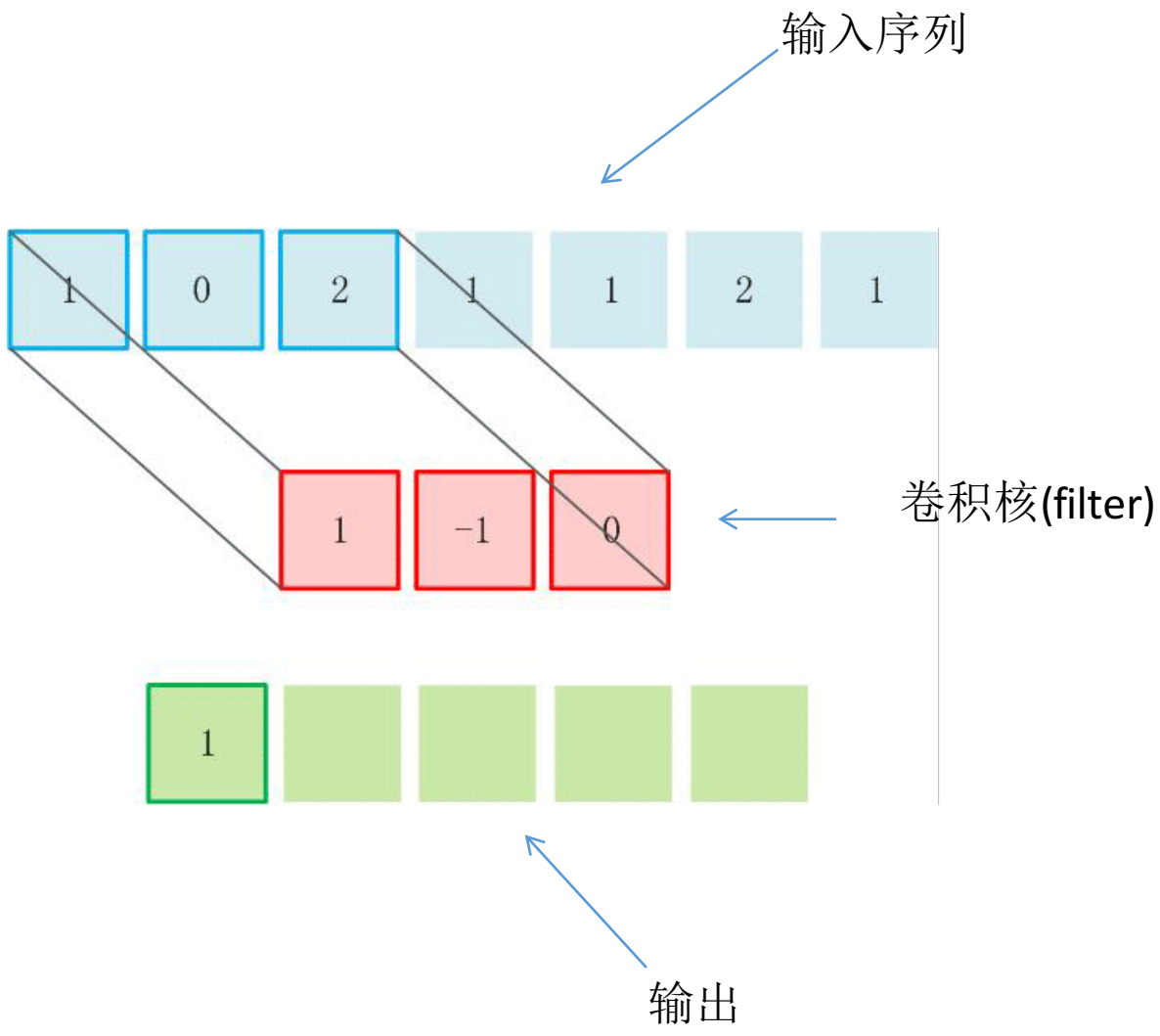
- 信号x，信号长度n
- 滤波器f，滤波器长度m

$$y_t = \sum_{k=1}^{n} f_k x_{t-k+1}$$



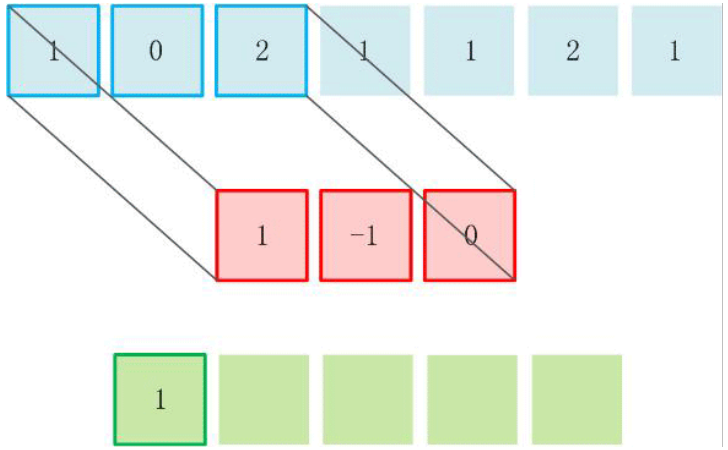

当$f_k = \frac{1}{m}$时，相当于移动平均

# Example

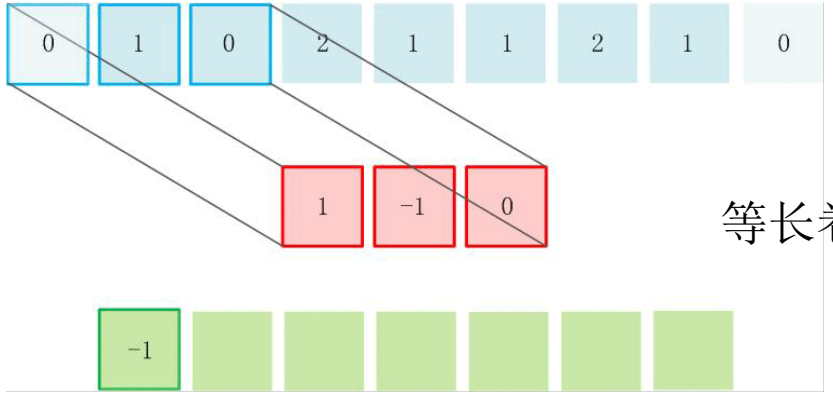


# 卷积类型

- 窄卷积
  - 信号两端不补0
  - 输出信号长度为n-m+1
- 宽卷积
  - 信号两端补0
  - 输出信号长度为n+m-1
- 等长卷积
  - 信号两端补0
  - 输出信号长度为n



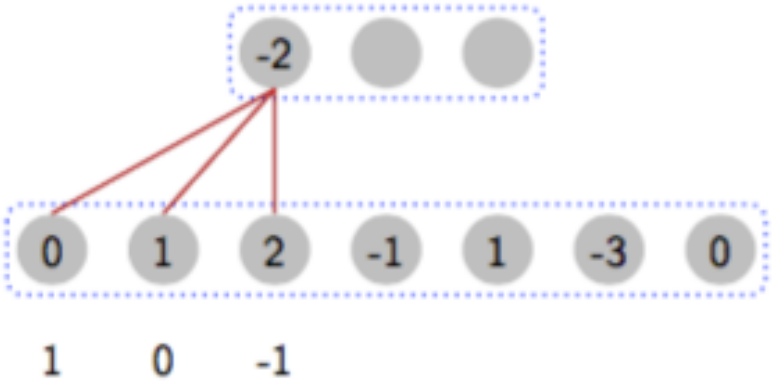

窄卷积




等长卷积

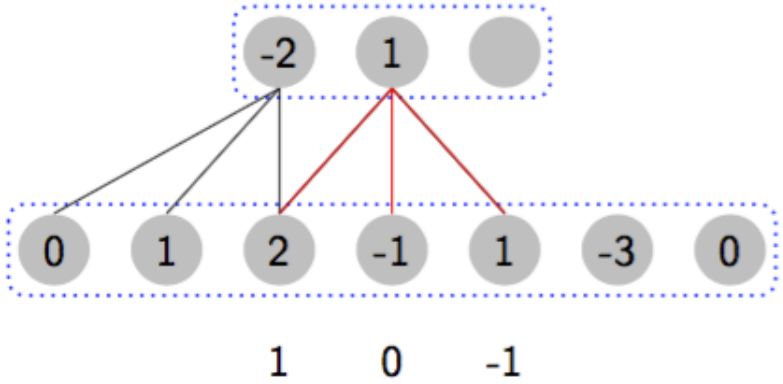
# 滤波器步长

- 步长=2



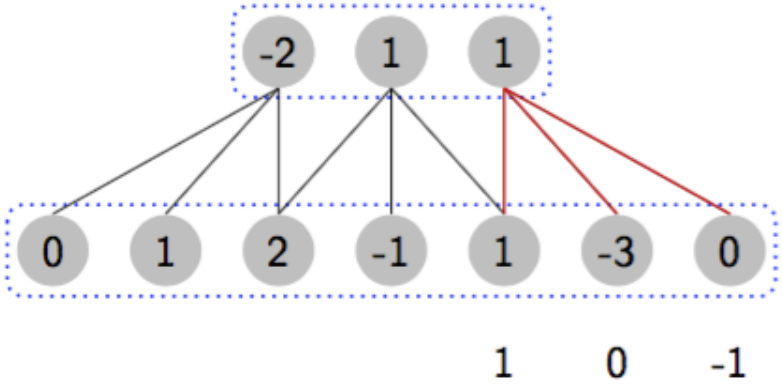
# 滤波器步长

- 步长=2



# 滤波器步长

- 步长=2



# 二维卷积

- 信号x，信号长度M*N
- 滤波器f，滤波器长度m*n
- 在图像中，卷积意味着区域内像素的加权平均

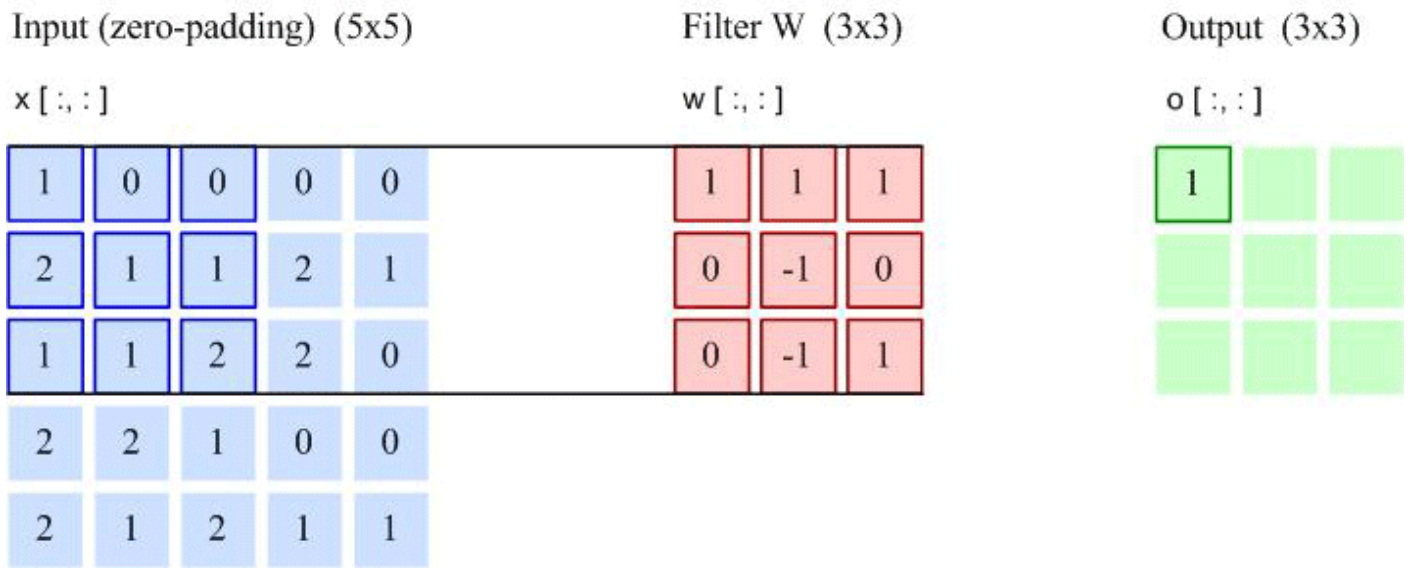
$$y_{ij} = \sum_{u=1}^{m} \sum_{v=1}^{n} f_{uv} x_{i-u+1,j-v+1}$$

当$f_{uv} = \frac{1}{mn}$时，相当于平均

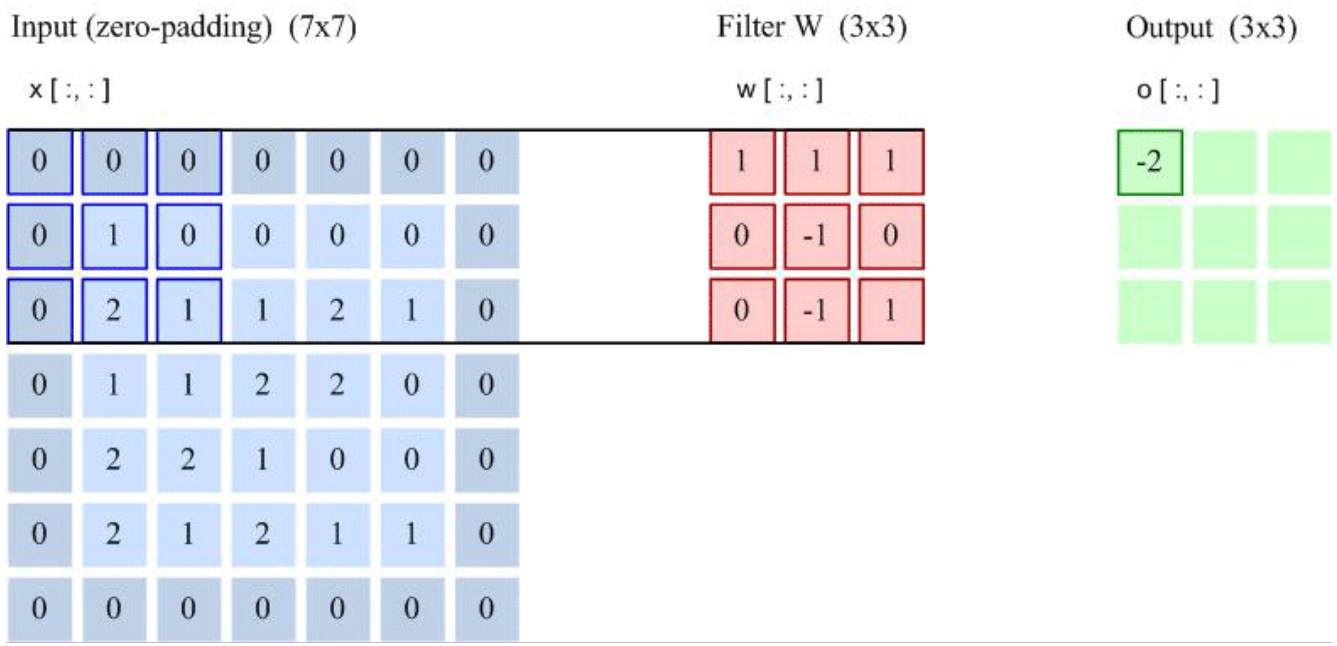
# 卷积类型

- 窄卷积
  - 信号四周不补0
  - 输出信号长度为M-m+1*N-n+1
- 宽卷积
  - 信号四周补0
  - 输出信号长度为M+m-1 *N+n-1
- 等长卷积
  - 信号四周补0
  - 输出信号长度为M*N

# Examples


Input (zero-padding) (5x5)
x [ :, : ]
1 0 0 0 0
2 1 1 2 1
1 1 2 2 0
2 2 1 0 0
2 1 2 1 1
Filter W (3x3)
w [ :, : ]
1 1 1
0 -1 0
0 -1 1
Output (3x3)
o [ :, : ]
1

# Examples

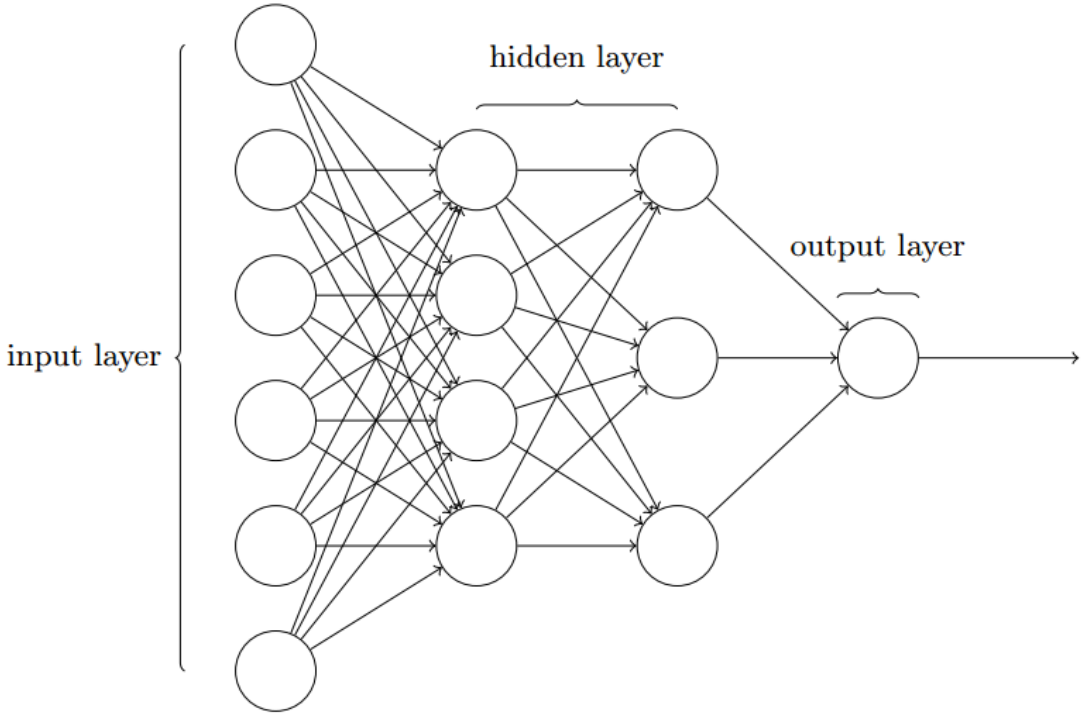

Input (zero-padding) (7x7)
x [ :, : ]
0 0 0 0 0 0 0
0 1 0 0 0 0 0
0 2 1 1 2 1 0
0 1 1 2 2 0 0
0 2 2 1 0 0 0
0 2 1 2 1 1 0
0 0 0 0 0 0 0
Filter W (3x3)
w [ :, : ]
1 1 1
0 -1 0
0 -1 1
Output (3x3)
o [ :, : ]
-2

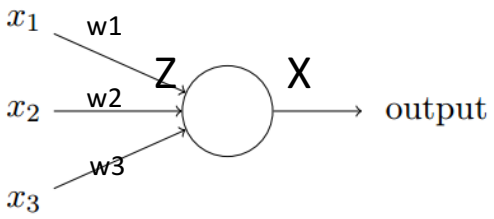
# 两维卷积实例

| | | | | | |
|---|---|---|---|---|---|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ | | | | |
| Edge detection | $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ | | Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ | |
| | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ | | Box blur (normalized) | $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ | |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ | | Gaussian blur (approximation) | $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ | |

# 卷积层

- 全连接前馈神经网络




$$X_i^{(l)} = f\left(\sum_j^{n^{(l)}} w_j^{(l)} X_j^{(l-1)} + b_i^{(l)}\right)$$

$$X_i^{(l)} = f(z_i^{(l-1)}) \qquad z_i^{(l-1)} = \sum_i w_i^{(l)} x_i^{(l-1)} + b_i^{(l)}$$

# 卷积层

- 第 $l$层的每一个神经元都只和第 $l-1$ 层的一个局部窗口内的神经元相连，构成一个局部连接网络。

$$X_i^{(l)} = f(\sum_j^m w_j^{(l)} X_{i-l+m}^{(l-1)} + b^{(l)})$$
$$= f(w^{(l)} \cdot X_{i-l+m}^{(l-1)} + b^{(l)})$$

- 其中，$w^{(l)} \in \mathbb{R}^m$ 为$m$维的滤波器，则有

$$X^{(l)} = f(w^{(l)} \otimes X_{i-l+m}^{(l-1)} + b^{(l)})$$

- **权值共享**：在卷积层里，我们只需要$m+1$个参数。第 $l+1$层的神经元个数不是任意选择的，而是满足$n^{(l+1)} = n^{(l)} - m + 1$

# 二维卷积层

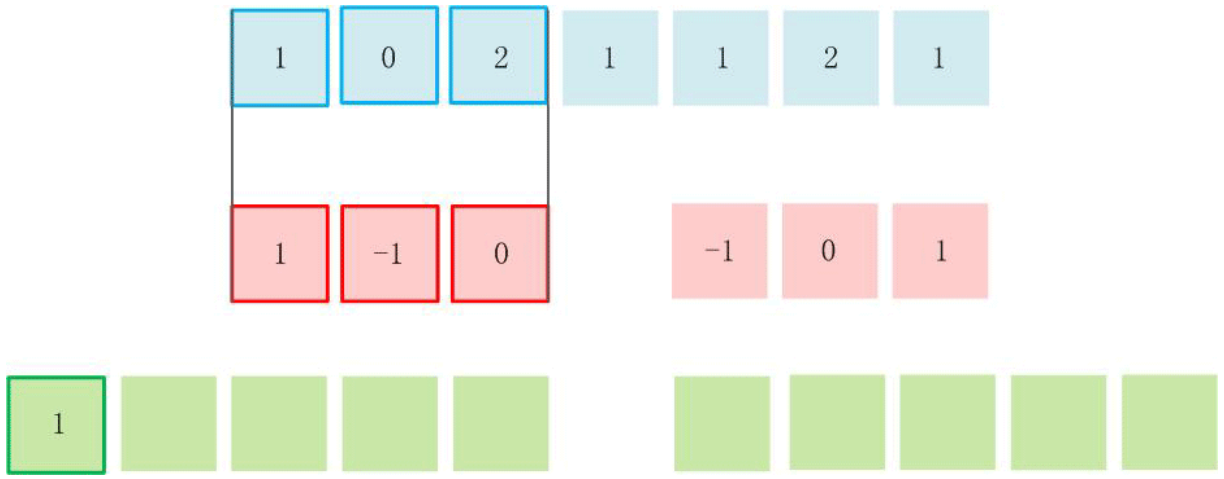
- 假设 $x^l \in \mathbb{R}^{(w_l \times h_l)}$ 和 $x^{(l-1)} \in \mathbb{R}^{(w_{l-1} \times h_{l-1})}$分别是第$l$层和第 $l-1$层的神经元活性。$x^l$ 的每一个元素为：

$$X_{s,t}^{(l)} = f\left(\sum_{i=1}^{u}\sum_{j=1}^{v} w_{i,j}^{(l)} X_{s-i+u,t-j+v}^{(l-1)} + b^{(l)}\right)$$

$$X^{(l)} = f(w^{(l)} \otimes X^{(l-1)} + b^{(l)})$$

- 其中，$W^{(l)} \in \mathbb{R}^{u \times v}$为两维的滤波器， $b^{(l)}$为偏置矩阵。
- 第 $l-1$ 层的神经元个数为 $w_l \times h_l$，并且$w_l = w_{l-1} - u + 1, h_l = h_{l-1} - v + 1$

# 两个filters



# 特征映射 Feature Map

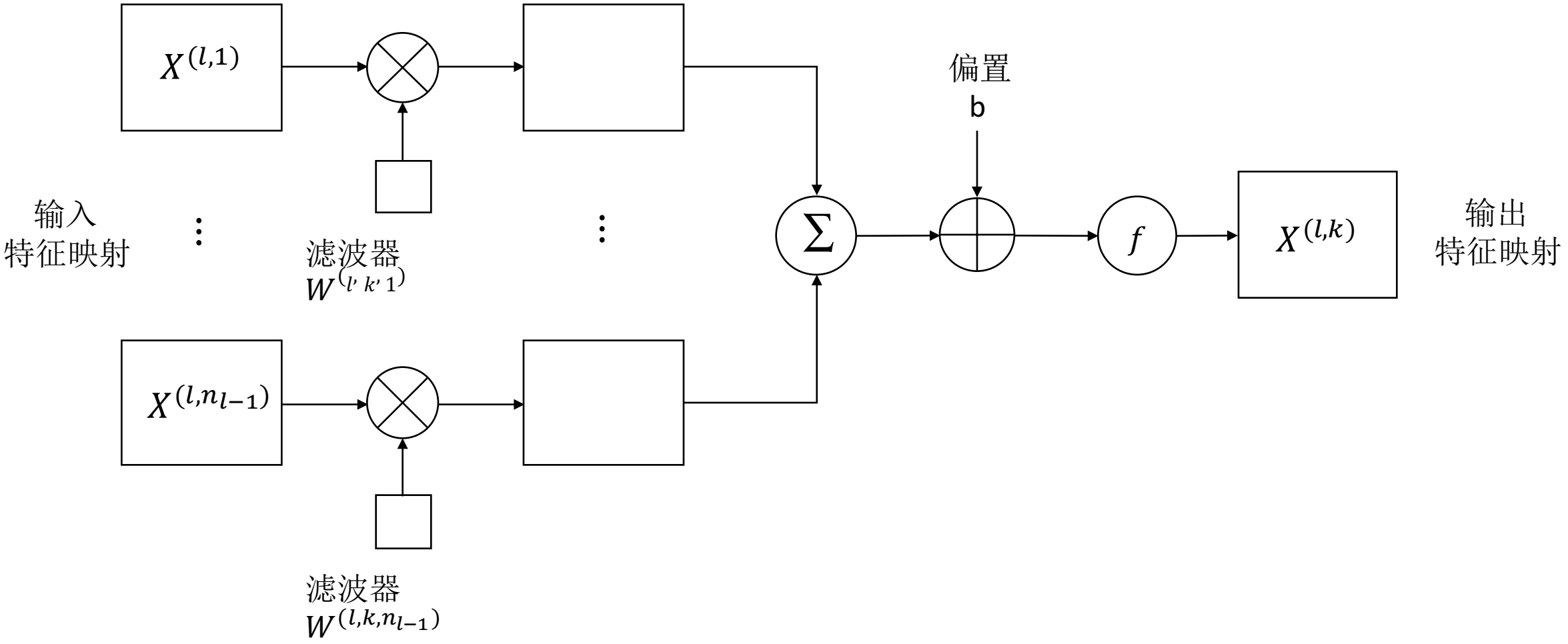
- 为了增强卷积层的表示能力，我们可以使用$K$个不同的滤波器来得到$K$组输出。每一组输出都共享一个滤波器。如果我们把滤波器看成一个特征提取器，每一组输出都可以看成是输入图像经过一个特征抽取后得到的特征。因此，在卷积神经网络中每一组输出也叫作一组**特征映射**（Feature Map）

- 假设第 $l-1$ 层的特征映射组数为 $n_{l-1}$，每组特征映射的大小为 $m_{l-1} = w_{l-1} \times h_{l-1}$。第 $l-1$ 层的总神经元数：$n_{l-1} \times m_{l-1}$。第 $l$层的特征映射组数为 $n_l$。如果假设第 $l$层的每一组特征映射 $X^{(l,k)}$的输入为第 $l-1$ 层的所有组特征映射。第 $l$ 层的第 k 组特征映射$X^{(l,k)}$为：

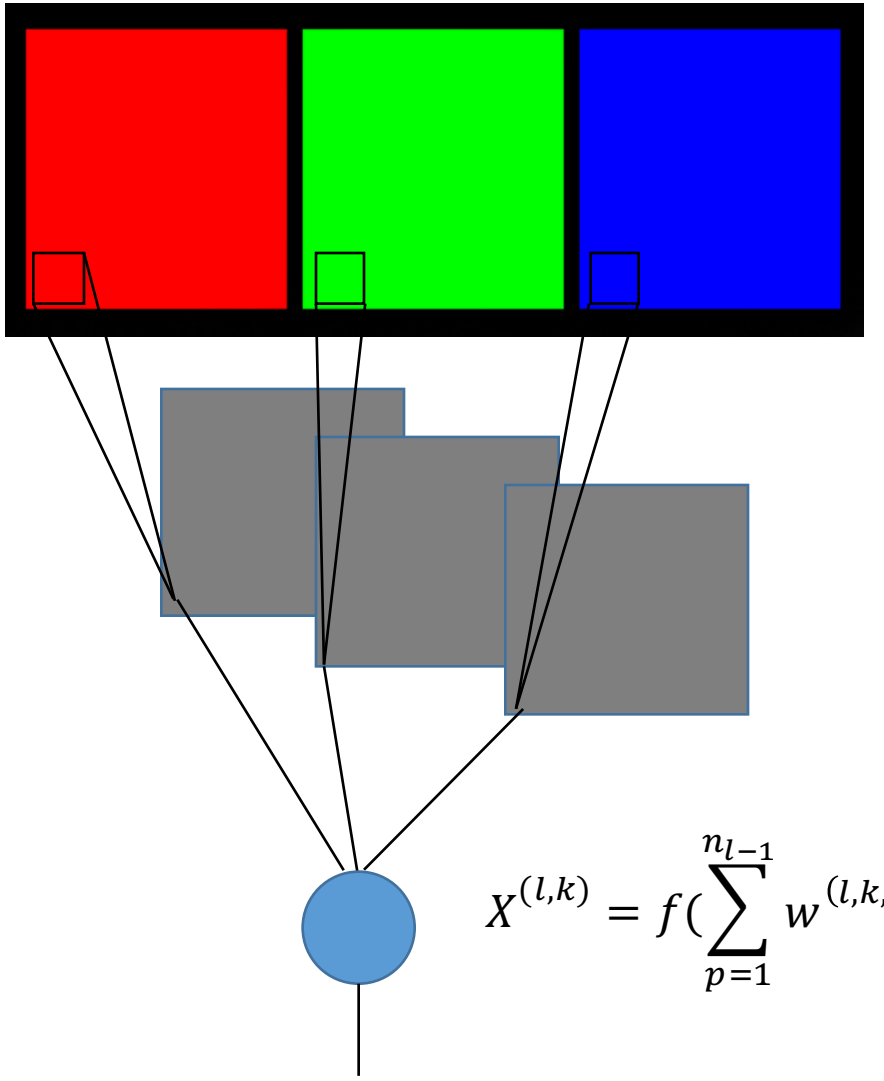
$$X^{(l,k)} = f\left(\sum_{p=1}^{n_{l-1}} w^{(l,k,p)} \otimes X^{(l-1,p)} + b^{(l,k)}\right)$$

- 其中，$W^{(l,k,p)}$ 表示第 $l-1$ 层的第 $p$ 组特征向量到第 $l$ 层的第 $k$ 组特征映射所需的滤波器

# 两维卷积层的映射关系



# 两维卷积层示例



$$X^{(l,k)} = f\left(\sum_{p=1}^{n_{l-1}} w^{(l,k,p)} \otimes X^{(l-1,p)} + b^{(l,k)}\right)$$

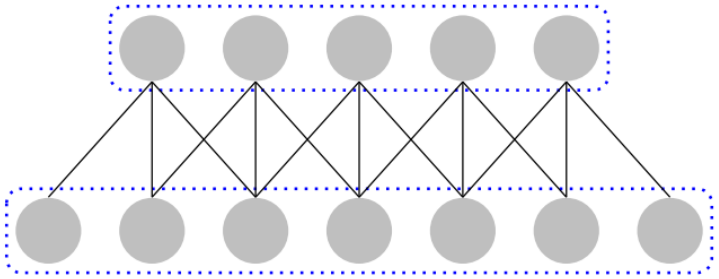
# 连接表

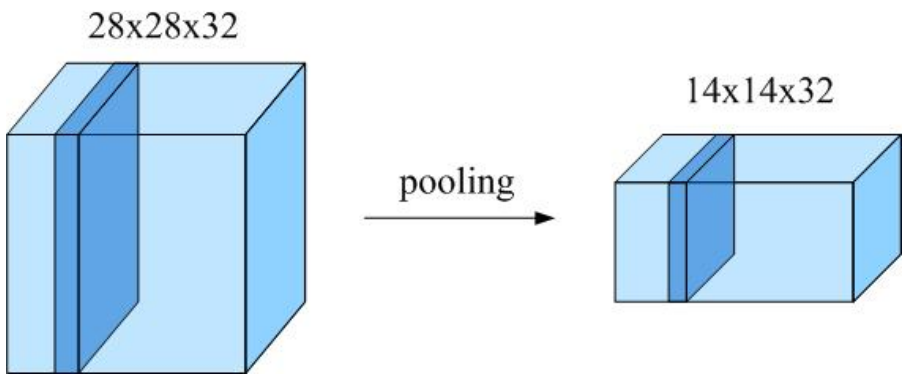
- 第$l$ 层的每一组特征映射都依赖于第$l$ 层的所有特征映射，相当于不同层的特征映射之间是全连接的关系。实际上，这种全连接关系不是必须的。我们可以让第$l$ 层的每一组特征映射都依赖于前一层的少数几组特征映射。这样，我们定义一个**连接表**$T$来描述不同层的特征映射之间的连接关系。如果第$l$层的第 $k$ 组特征映射依赖于前一层的第 $p$ 组特征映射，则 $T_{p,k} = 1$，否则为 0

$$X^{(l,k)} = f\left(\sum_{\substack{p=1 \\ T_{p,k}=1}}^{n_{l-1}} w^{(l,k,p)} \otimes X^{(l-1,p)} + b^{(l,k)}\right)$$

- 这样，假如连接表$T$的非零个数为$K$，，那每个滤波器的大小为$u \times v$，共需要$K + u \times v + n_l$参数

# 子采样层






- 卷积层虽然可以显著减少连接的个数，但是每一个特征映射的神经元个数并没有显著减少
- 高维数据→过拟合
- 降低维度：Pooling、Subsampling
- 特征选择、特征抽取

# 子采样层

- 对于卷积层得到的一个特征映射$X^{(l)}$，我们可以将$X^{(l)}$划分为很多区域$R_k, k = 1, \cdots, K$ 。区域$R_k$可以重叠，也可以不重叠，
- 则采样层输出有：

$$X^{(l+1)} = f(w^{(l+1)} \cdot down(R_k) + b^{(l+1)})$$

- 其中，$w^{(l+1)}$和 $b^{(l+1)}$分别是可训练的权重和偏置参数

$$X^{(l+1)} = f(w^{(l+1)} \cdot down(X^{(l)}) + b^{(l+1)})$$

$down(X^{(l)})$是指子采样后的特征映射

# 子采样层

- 最大值采样（Maximum Pooling）

$$pool_{max}\left(R_k\right) = \max_{i \in R_k} a_i$$

- 最小值采样（Minimum Pooling）

$$pool_{min}\left(R_k\right) = \min_{i \in R_k} a_i$$

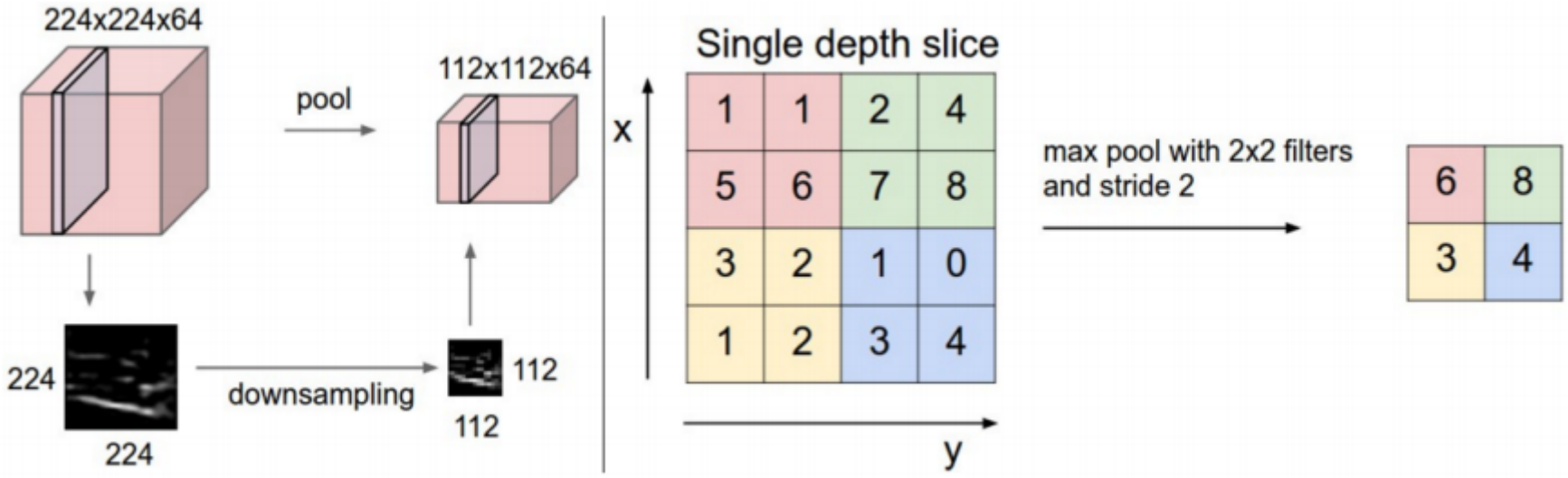
- 平均值（Average Pooling）

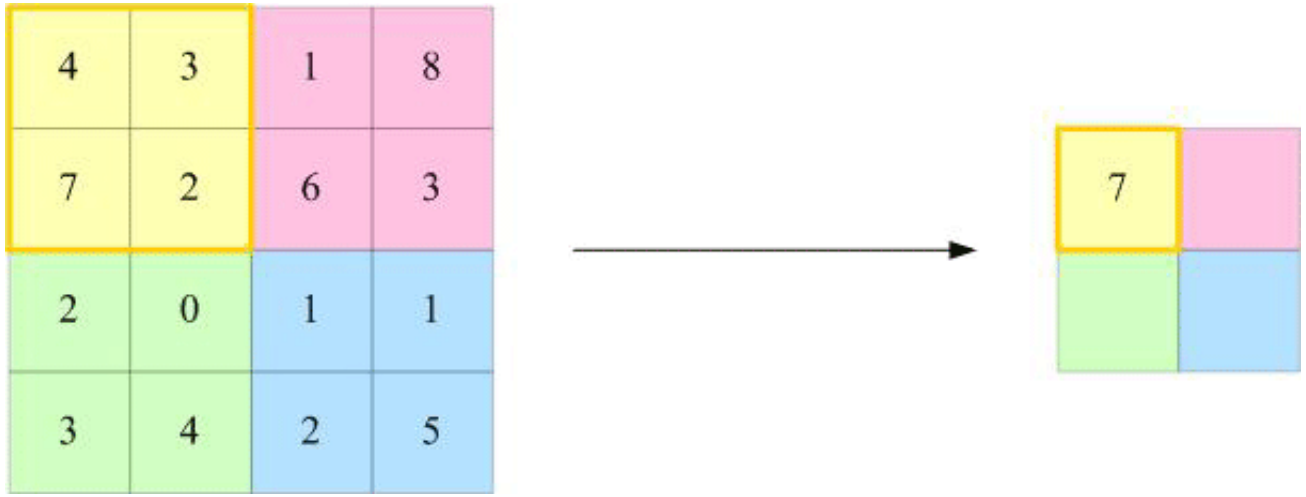
$$pool_{ave}\left(R_k\right) = \frac{1}{|R_k|} \sum_{i \in R_k}^{|R_k|} a_i$$

- TopK采样（Average Pooling）

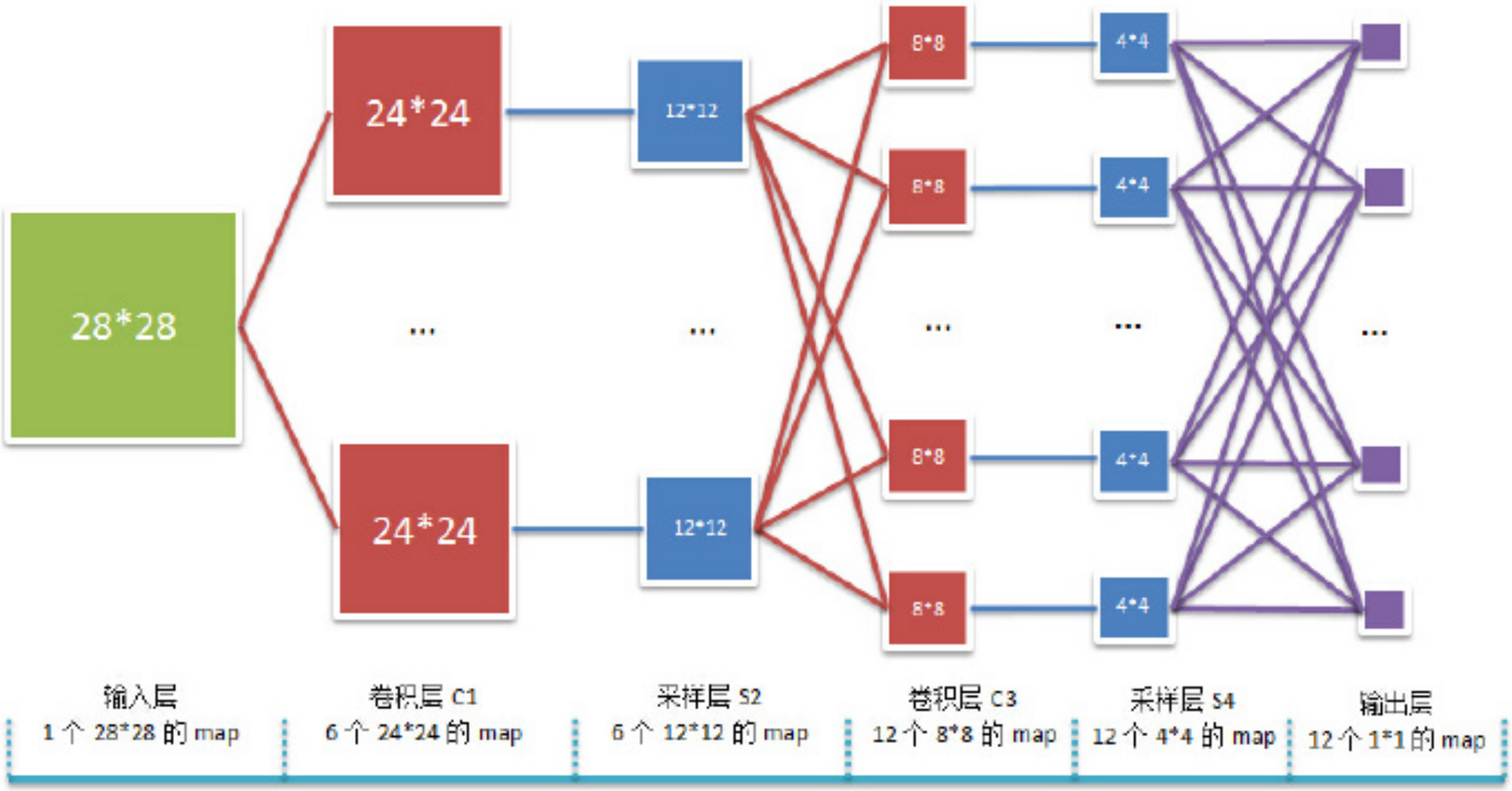
$$pool_{k}\left(R_k\right) = \operatorname*{topk}_{i \in R_k} a_i$$

# 子采样层示例


224x224x64
pool
112x112x64
224
224
downsampling
112
112
Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

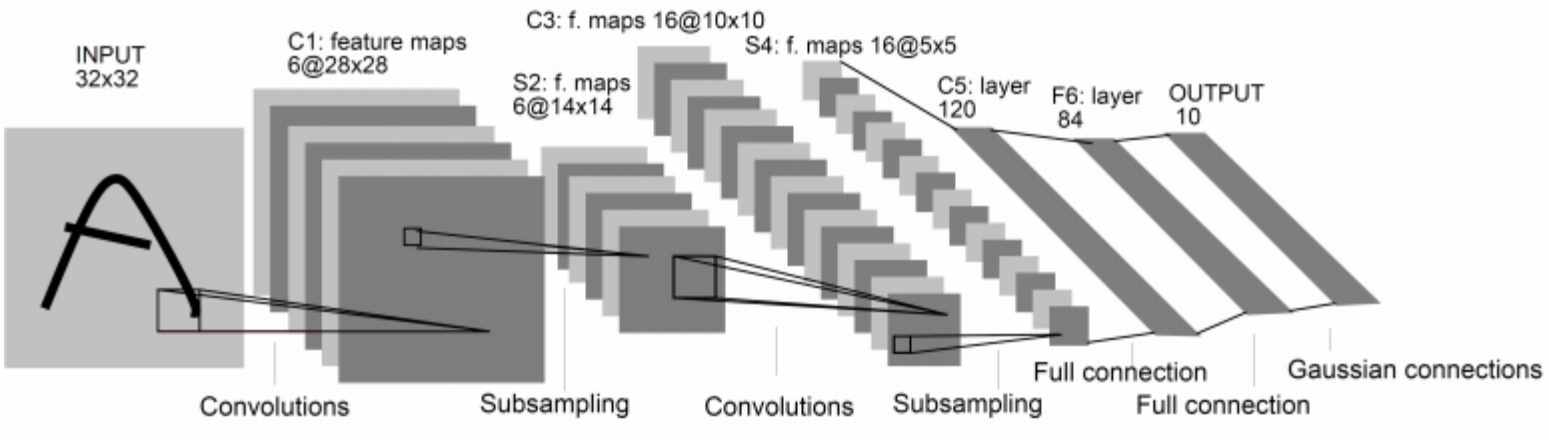

4 3 1 8
7 2 6 3
2 0 1 1
3 4 2 5
7

# Example


28*28
24*24
24*24
12*12
12*12
8*8
8*8
8*8
8*8
4*4
4*4
4*4
4*4
输入层
1 个 28*28 的 map
卷积层 C1
6 个 24*24 的 map
采样层 S2
6 个 12*12 的 map
卷积层 C3
12 个 8*8 的 map
采样层 S4
12 个 4*4 的 map
输出层
12 个 1*1 的 map

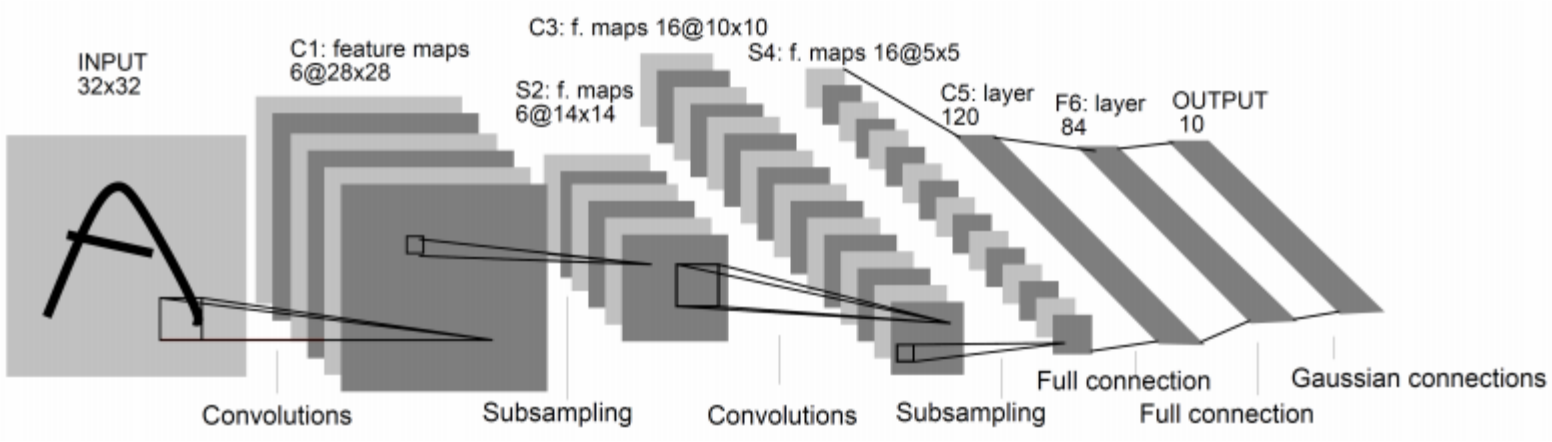
# CNN在图像处理中的应用

LeNet-5 虽然提出时间比较早，但是是一个非常成功的神经网络模型。基于 LeNet-5 的手写数字识别系统在 90 年代被美国很多银行使用，用来识别支票上面的手写数字。LeNet-5共有 7 层。




LeNet-5 网络结构

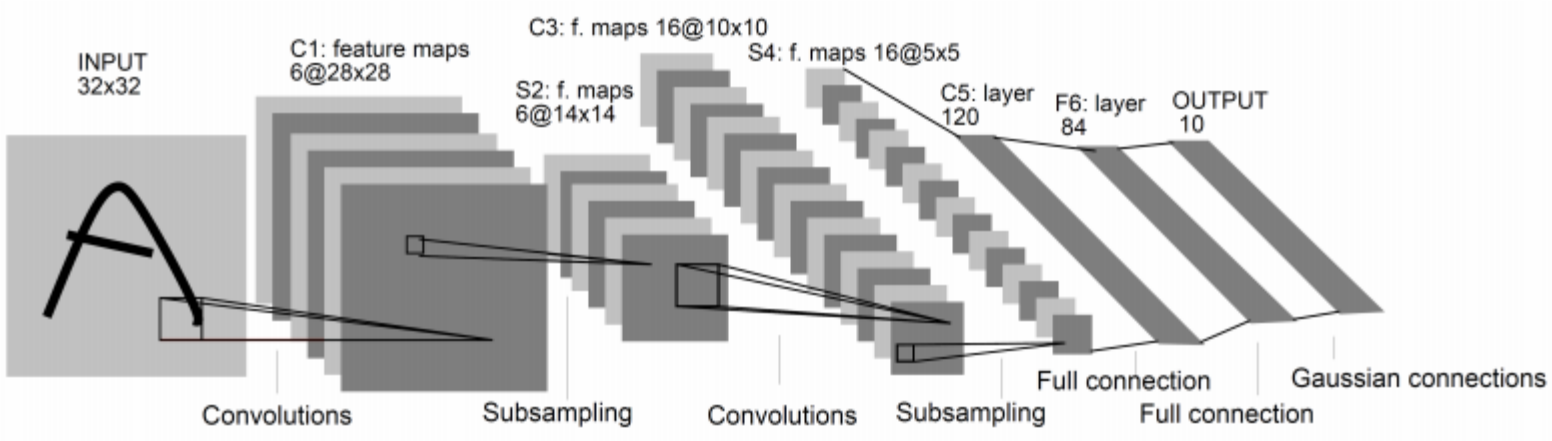
# LeNet-5




LeNet-5 网络结构

- 输入层：输入图像大小为 32 × 32 = 1024。
- C1 层：这一层是卷积层。滤波器的大小是 5×5 = 25，共有 6 个滤波器。得到 6 组大小为 28 × 28 = 784 的特征映射。因此，C1 层的神经元个数为 6 × 784 = 4,704。可训练参数个数为6 × 25 + 6 = 156。连接数为 156 × 784 = 122,304

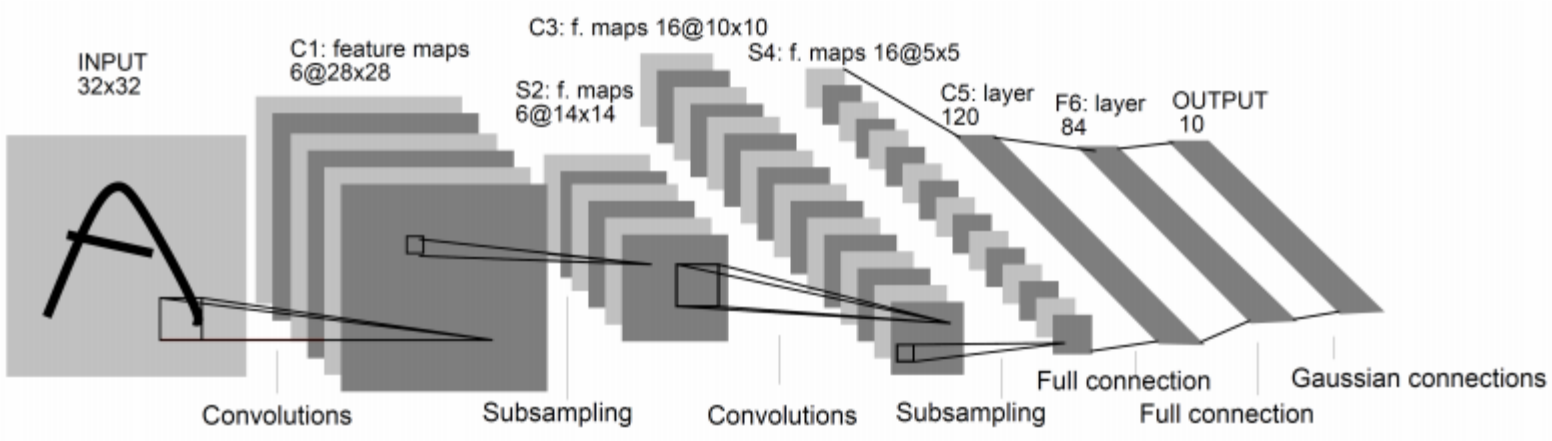
# LeNet-5




LeNet-5 网络结构

- S2 层：这一层为子采样层。由 C1 层每组特征映射中的 2×2 邻域点次采样为 1 个点，也就是 4 个数的平均。这一层的神经元个数为14× 14 = 196。可训练参数个数为 6× (1 + 1) = 12。连接数为6×196×(4 + 1) = 122,304

# LeNet-5




LeNet-5 网络结构

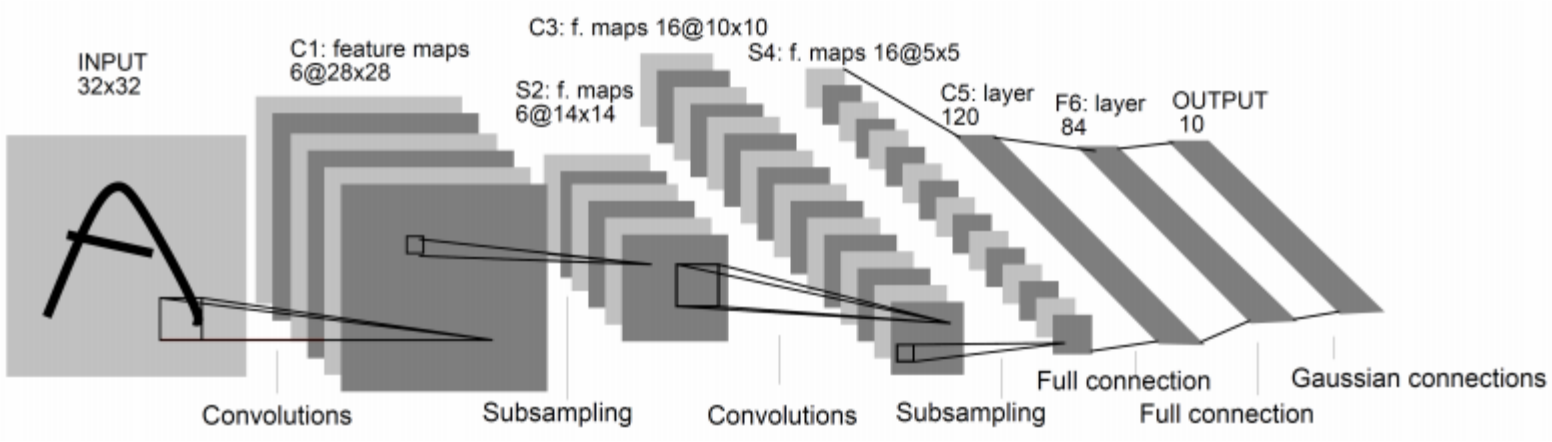
- C3 层：这一层是卷积层。由于 S2 层也有多组特征映射，需要一个连接表来定义不同层特征映射之间的依赖关系。LeNet-5 的连接表如图所示，共有60个滤波器，大小是5×5=25。
- 得到 16 组大小为 10×10 = 100 的特征映射。C3 层的神经元个数为 16×100 = 1,600。可训练参数个数为 60×25+16 = 1,516。连接数为 1,516×100 = 151,600。

# LeNet-5

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | X | | | | X | X | X | | | X | X | X | X | | X | X |
| 1 | X | X | | | | X | X | X | | | X | X | X | X | | X |
| 2 | X | X | X | | | | X | X | X | | | X | | X | X | X |
| 3 | | X | X | X | | | X | X | X | X | | | X | | X | X |
| 4 | | | X | X | X | | | X | X | X | X | | X | X | | X |
| 5 | | | | X | X | X | | | X | X | X | X | | X | X | X |

LeNet-5 中C3层的连接表

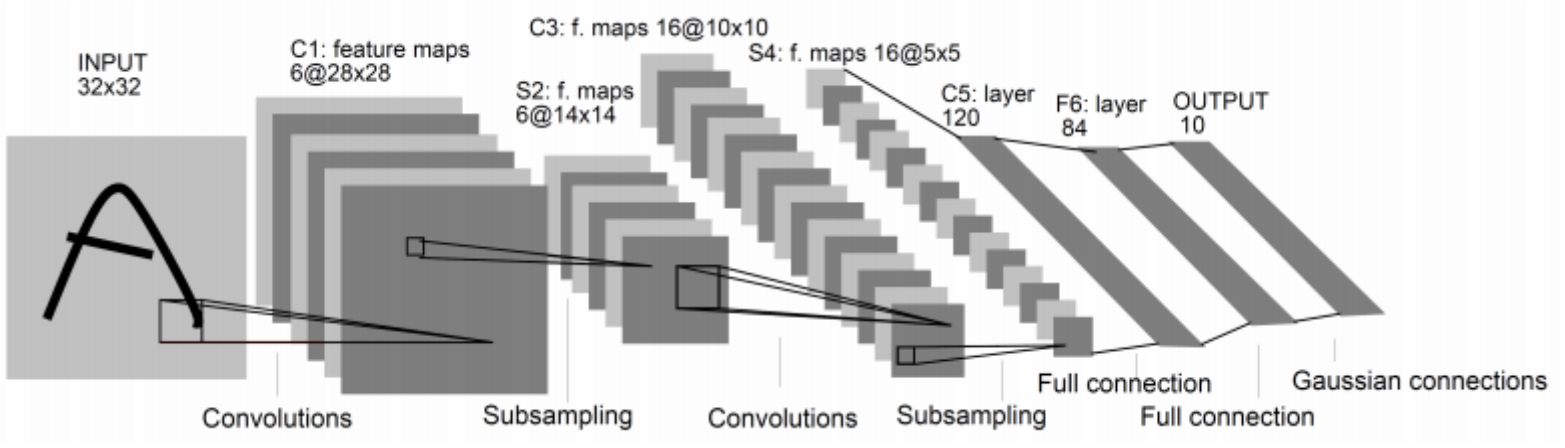
# LeNet-5




LeNet-5 网络结构

- S4 层：这一层是一个子采样层，由 2×2 邻域点次采样为 1 个点，得到 16 组 5×5 大小的特征映射。可训练参数个数为 16×2 = 32。连接数为 16×(4 +1) = 80。

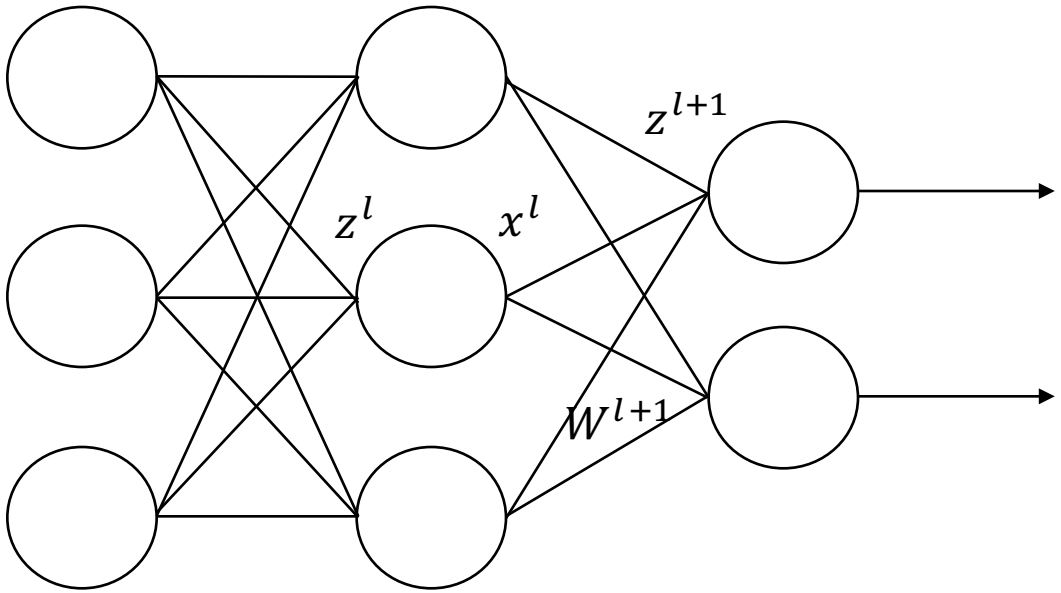
# LeNet-5




- C5 层：是一个卷积层，得到 120 组大小为 1×1 的特征映射。每个特征映射与 S4 层的全部特征映射相连。有120×16 = 1,920 个滤波器，大小是5×5 = 25。C5层的神经元个数为120，可训练参数个数为 1,920×25+120 = 48,120。连接数为120×(16×25+1) = 48,120
- F6 层：是一个全连接层，有 84 个神经元，可训练参数个数为84×(120+1) = 10,164。连接数和可训练参数个数相同，为10,164
- 输出层：输出层由 10 个欧氏径向基函数（Radial Basis Function，RBF）函数

# 梯度计算

全连接前馈神经网络




$J$

$$\frac{\partial J}{\partial z^l} = \frac{\partial x^l}{\partial z^l}\frac{\partial J}{\partial x^l} = f'(z^l) \odot (W^{l+1})^T \frac{\partial J}{\partial z^{l+1}}$$

$$\delta^l = f'(z^l) \odot (W^{l+1})^T \delta^{l+1}$$

# 卷积层的梯度

- 假定卷积层为 $l$ 层，子采样层为 $l+1$ 层。因为子采样层是下采样操作，$l+1$ 层的一个神经元的误差项 $\delta$ 对应于卷积层（上一层）的相应特征映射的一个区域。$l$ 层的第 $k$ 个特征映射中的每个神经元都有一条边和 $l+1$层的第 $k$ 个特征映射中的一个神经元相连。

$$X^{(l+1)} = f(w^{(l+1)} \cdot down(X^{(l)}) + b^{(l+1)})$$

- 求第$l$层的一个特征映射的误差项 $\delta^{(l,k)}$，需要将 $l+1$层对应特征映射的误差项 $\delta^{(l+1,k)}$进行上采样操作（和第 $l$ 层的大小一样），再和 $l$ 层特征映射的激活值偏导数逐元素相乘，再乘上权重$w^{(l+1,k)}$，就得到了 $\delta^{(l,k)}$：

$$\begin{aligned}\delta^{(l,k)} &= f'(z^l) \odot (W^{(l+1,k)})^T \delta^{(l+1,k)} \\ &= f'(z^l) \odot (\mathrm{up}((W^{(l+1,k)})^T \delta^{(l+1,k)}) \\ &= (W^{(l+1,k)})^T (f'(z^l) \odot \mathrm{up}(\delta^{(l+1,k)}))\end{aligned}$$

其中，up 为上采用函数（Upsampling）。

# 卷积层的梯度

- 损失函数关于第 $l$ 层的第$k$个特征映射神经元滤波器$W_{i,j}^{(l,k,p)}$的梯度

$$\frac{\partial J}{\partial W^{(l,k,p)}} = \sum_{s=1}^{w_i}\sum_{t=1}^{h_i}(X_{s-i+u,t-j+v}^{(l-1,p)} \cdot (\delta^{(l,k)})_{s,t})$$

$$=\sum_{s=1}^{w_i}\sum_{t=1}^{h_i}(X_{u-i-s,v-j-t}^{(l-1,p)} \cdot rot180((\delta^{(l,k)}))_{s,t})$$

$$\frac{\partial J}{\partial W^{(l,k,p)}} = \frac{\partial X_{s,t}^{(l,k)}}{\partial W^{(l,k,p)}}\delta^{(l,k)}$$

- 损失函数关于第 $l$ 层的第 $k$ 个特征映射的偏置 $b^{(l)}$ 的梯度可以写为：

$$\frac{\partial J}{\partial b^{(l,k)}} = \sum_{i,j}(\delta^{(l,k)})_{i,j}$$

$$\frac{\partial J}{\partial b^{(l,k)}} = \frac{\partial X_{s,t}^{(l,k)}}{\partial b^{(l,k)}}\delta^{(l,k)}$$

# 子采样层的梯度

- 我们假定子采样层为 $l$ 层，$l+1$ 层为卷积层。

$$X^{(l+1,k)} = f\left(\sum_{\substack{p=1 \\ T_{p,k}=1}}^{n_{l-1}} w^{(l+1,k,p)} \otimes X^{(l,p)} + b^{(l+1,k)}\right) \qquad Z^{(l+1,k)} = \sum_{\substack{p=1 \\ T_{p,k}=1}}^{n_{l-1}} w^{(l+1,k,p)} \otimes X^{(l,p)} + b^{(l+1,k)}$$

- 第 $l$ 层的第 $k$ 个特征映射的误差项 $\delta^{(l,k)}$

$$\begin{aligned}\delta^{(l,k)} &= f'\left(z^l\right) \odot \left(W^{(l+1,k)}\right)^T \delta^{(l+1,k)} \\ &= f'\left(z^l\right) \odot \left(\textstyle\sum_p \delta^{(l+1,p)} \otimes rot180\left(W^{(l,k,p)}\right)\right)\end{aligned}$$

# 子采样层的梯度

- 损失函数关于第 $l$ 层的第 $k$ 个特征映射神经元滤波器 $W^{(l,k,p)}$ 的梯度

$$\frac{\partial J}{\partial W^{(l,k)}} = \sum_{i,j}(down(X^{(l-1,k)}) \cdot \delta^{(l,k)})_{i,j}$$

$$\frac{\partial J}{\partial W^{(l,k)}} = \frac{\partial X^{(l,k)}}{\partial W^{(l,k)}}\delta^{(l,k)}$$

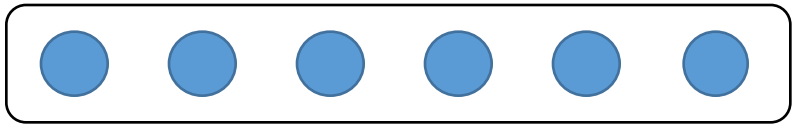
- 损失函数关于第 $l$ 层的第 $k$ 个特征映射的偏置 $b^{(l)}$ 的梯度

$$\frac{\partial J}{\partial W^{(l,k)}} = \sum_{i,j}(\delta^{(l,k)})_{i,j}$$

$$\frac{\partial J}{\partial b^{(l,k)}} = \frac{\partial X^{(l,k)}}{\partial b^{(l,k)}}\delta^{(l,k)}$$

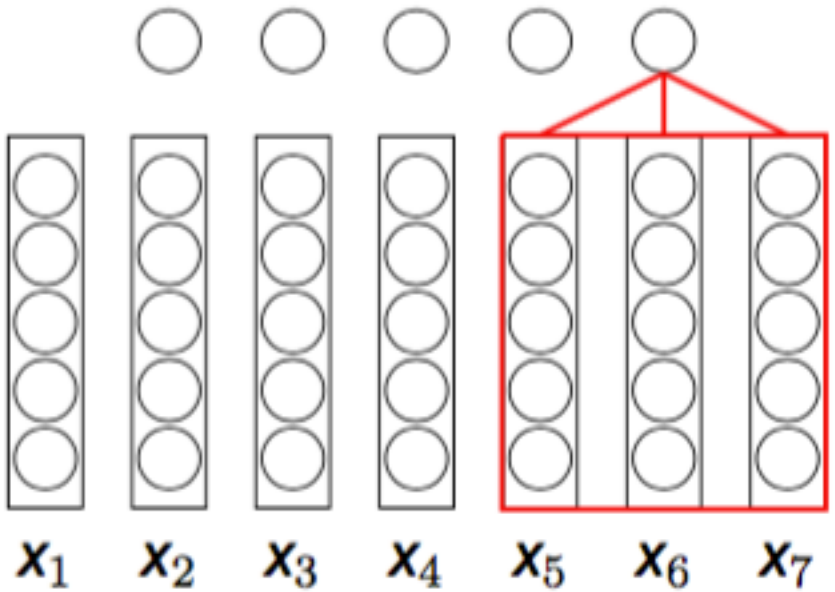
# CNN在自然语言处理中的应用

I want to play with that girl.

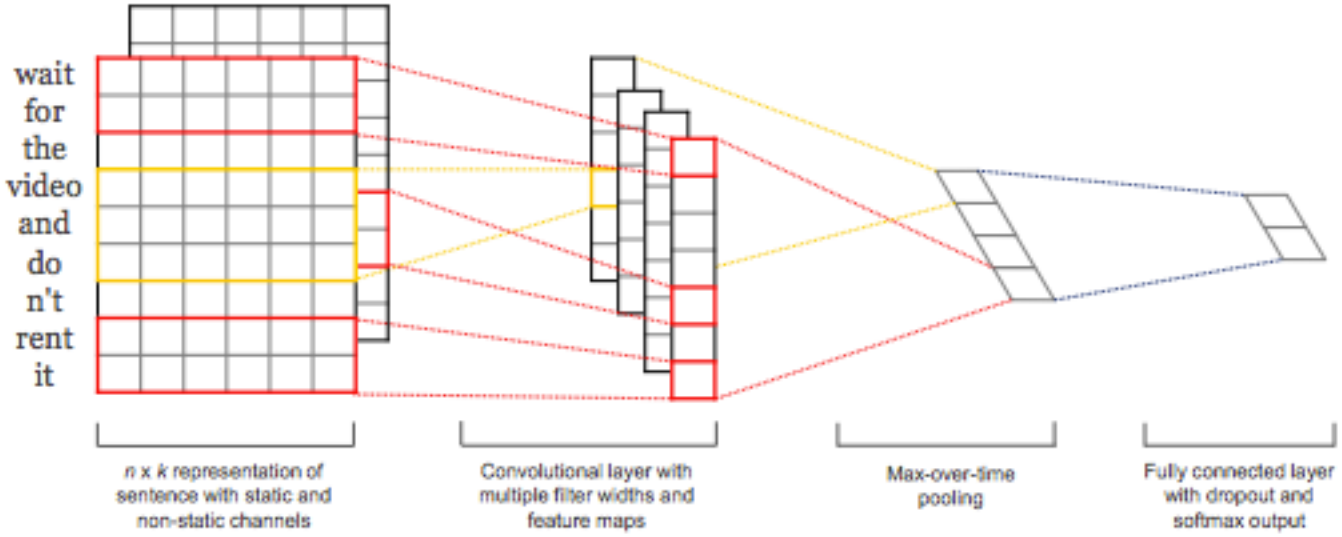


# Sentence Modeling

lookup

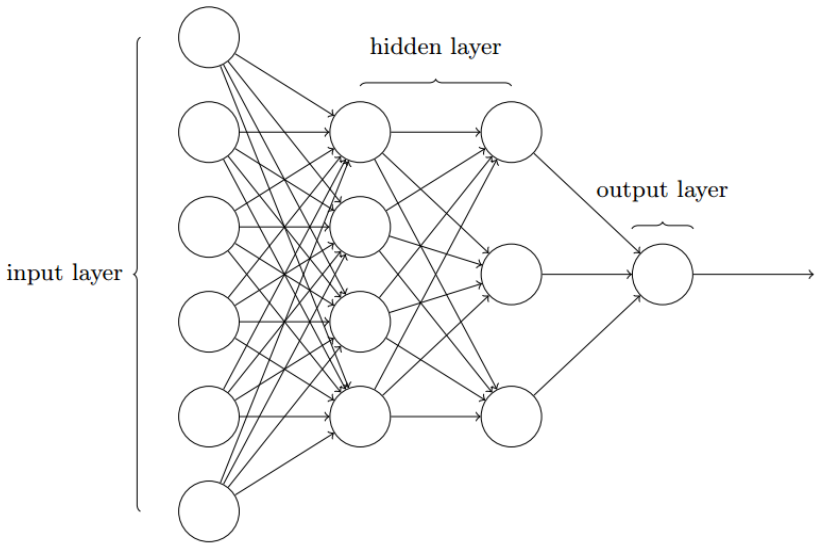


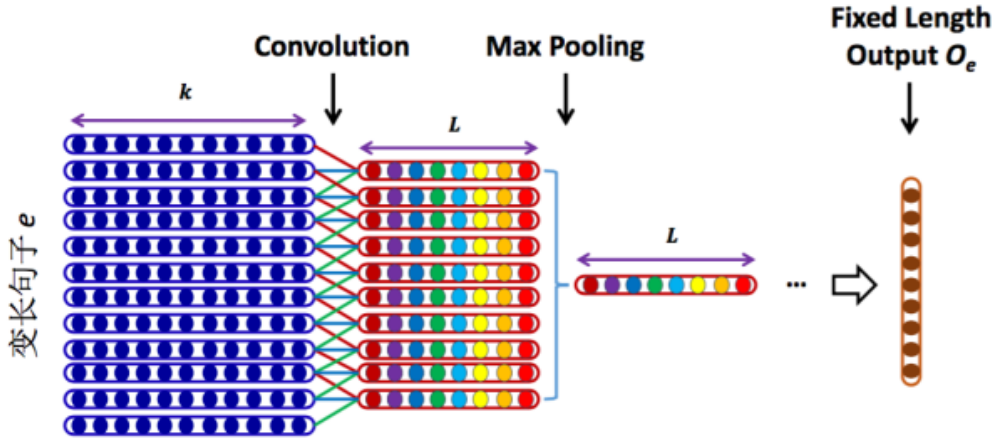
# 文本分类


wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

# 循环神经网络
# Recurrent Neural Network

# 前馈、卷积神经网络




- 连接存在层与层之间，每层的节点之间是无连接的
- 输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据
- 假设每次输入都是独立的，也就是说每次网络的输出只依赖于当前的输入

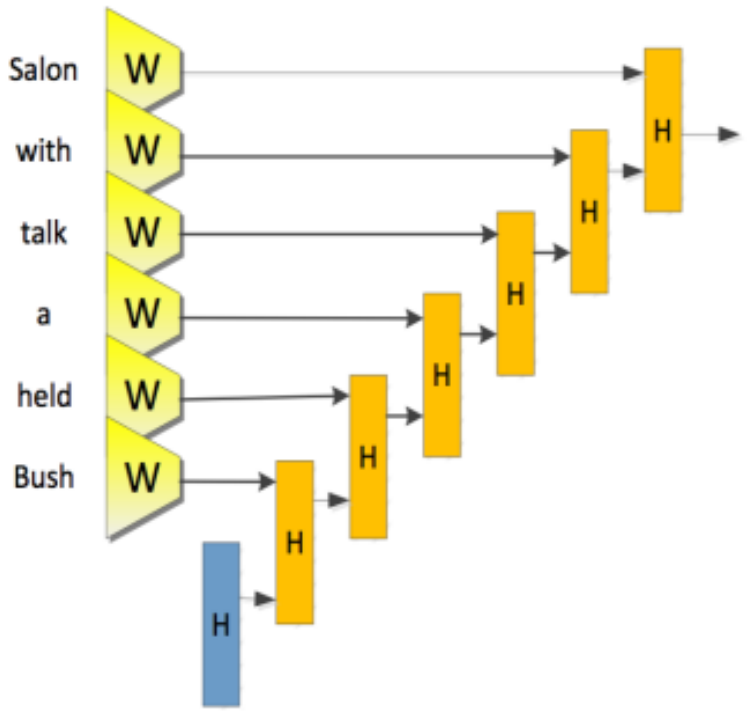
# 各种处理任务

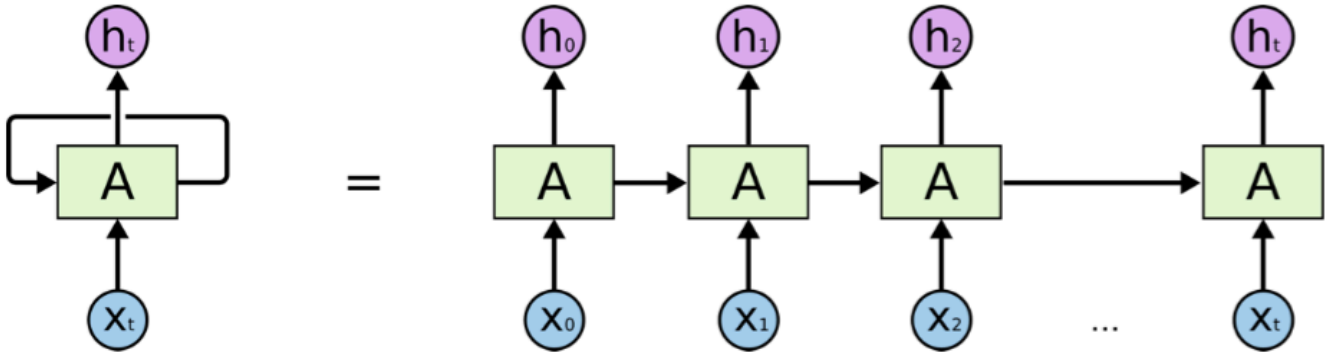
- 变长输入
  - 不同大小的图片
  - 时长不一的视频
  - 长短不同的句子
  - 序列长度不同的对话
  - …
- 相互依赖
  - 前言要搭后语
  - 动画片由连续的图片组成
  - 字词搭配
  - …

# 循环神经网络

- **循环神经网络（ Recurrent Neural Network， RNN）**
  - 前馈、卷积神经网络的输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据。
  - 假设每次输入都是独立的，也就是说每次网络的输出只依赖于当前的输入。
- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的序列。

# 循环神经网络



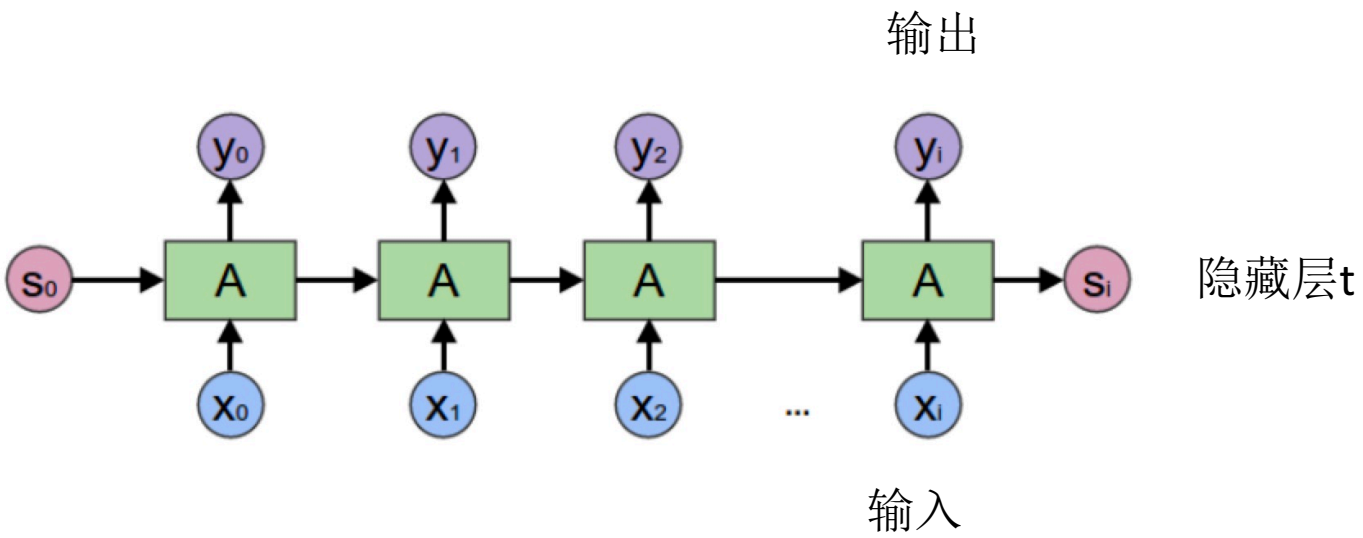



# 循环神经网络

- 给定一个输入序列$\boldsymbol{x} = (x_1, \ldots, x_t, \ldots, x_n)$，循环神经网络通过下面公式更新带反馈边的隐藏层的活性值 $h_t$，即抽象表示：

$$h_t = \begin{cases} 0 & t = 0 \\ f(h_{t-1}, x_t) & otherwise \end{cases}$$

# 循环神经网络的示例



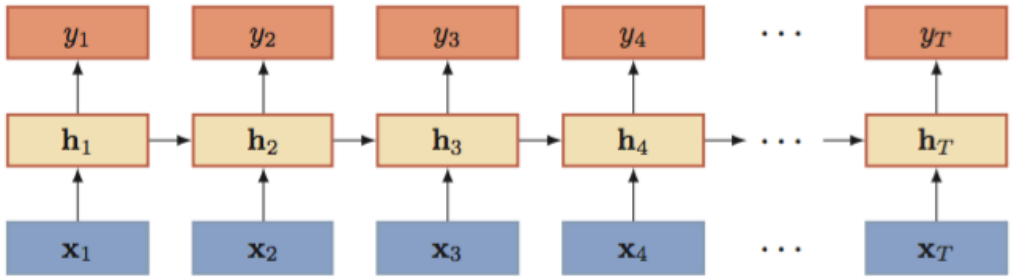
# 简单循环网络

- 假设时刻$t$时，输入为$x_t$，隐层状态为$h_t$。 $h_t$不仅和当前时刻的输入相关，也和上一个时刻的隐层状态$h_{t-1}$相关。

- 一般我们使用如下函数：

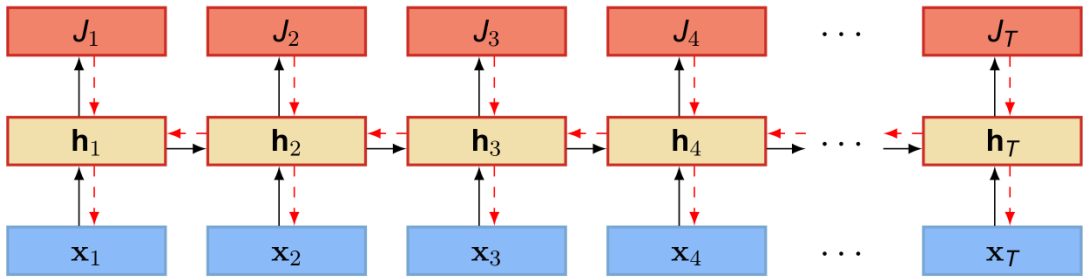
$$\boldsymbol{h}_t = f(\boldsymbol{U}\boldsymbol{h}_{\boldsymbol{t-1}} + \boldsymbol{W}\boldsymbol{x}_{\boldsymbol{t}} + b) \qquad \boldsymbol{y}_t = softmax(V\boldsymbol{h}_t)$$

这里，$f$是非线性函数，可以为 sigmod函数或 tanh 函数。



# 梯度计算

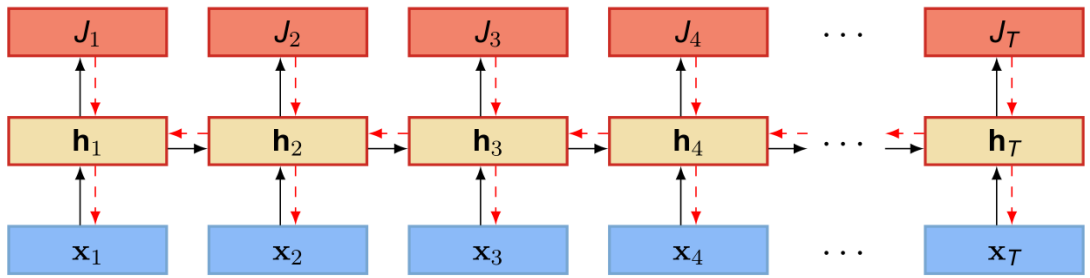
- 循环神经网络的参数训练可以通过随时间进行反向传播（Backpropagation Through Time，BPTT）算法。




P. J. Werbos. “Backpropagation through time: what it does and how to do it”. In: Proceedings of the IEEE 10 (1990).

# 梯度

- 假设循环神经网络在每个时刻 t 都有一个监督信息，损失为 $J_t$ 。则整个序列的损失为 $J = \sum_{t=1}^{T} J_t$ 。
  - $J_t = -t_t log y_t$
  - $J = -\sum_{t=1}^{T} t_t log y_t$




- 损失 $J$ 关于 V的梯度为

$$\frac{\partial J}{\partial V} = \frac{\partial}{\partial V} \sum_{t=1}^{T} J_t = \sum_{t=1}^{T} \frac{\partial J_t}{\partial V}$$

$$\frac{\partial J_t}{\partial V} = \frac{\partial J_t}{\partial y_t} \frac{\partial y_t}{\partial V} = \frac{\partial J_t}{\partial y_t} \frac{\partial y_t}{\partial z_t} \frac{\partial z_t}{\partial V}$$

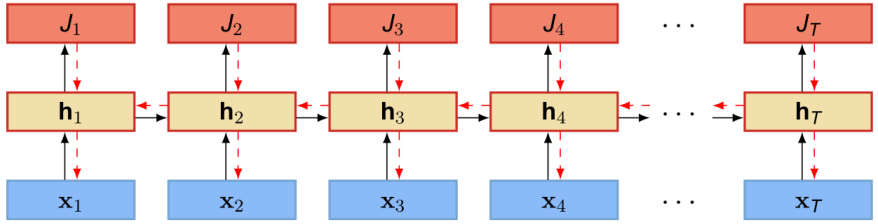
$$y_t = softmax(z_t)$$
$$z_t = V h_t$$

# 梯度

- 损失 $J$ 关于 U 的梯度为：

$$\frac{\partial J}{\partial U} = \frac{\partial}{\partial U}\sum_{t=1}^{T} J_t = \sum_{t=1}^{T}\frac{\partial J_t}{\partial U}$$

$$= \sum_{t=1}^{T}\frac{\partial h_t}{\partial U}\frac{\partial J_t}{\partial h_t}$$




- 其中，$\boldsymbol{h}_t$ 是关于**U**和$\boldsymbol{h}_{t-1}$的函数，而$\boldsymbol{h}_{t-1}$又是关于**U**和$\boldsymbol{h}_{t-2}$的函数。

# 梯度

- 用链式法则得到

$$\frac{\partial J}{\partial U} = \sum_{t=1}^{T} \sum_{k=1}^{t} \frac{\partial h_k}{\partial U} \frac{\partial h_t}{\partial h_k} \frac{\partial y_t}{\partial h_t} \frac{\partial J_t}{\partial y_t}$$

$$h_t = f(Uh_{t-1} + Wx_t + b)$$

- 其中

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}}$$

$$= \prod_{i=k+1}^{t} U^T diag[f`(h_{i-1})]$$

# 梯度

- 对$U$的梯度为

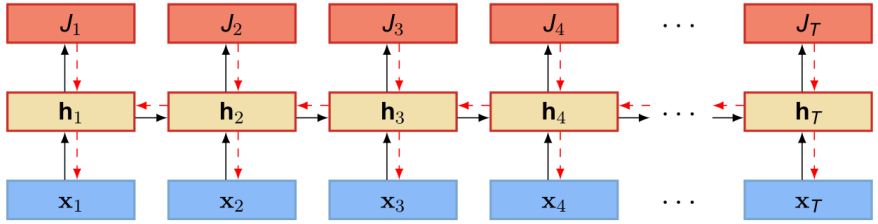
$$\frac{\partial J}{\partial U}=\sum_{t=1}^{T}\sum_{k=1}^{t}\frac{\partial h_k}{\partial U}\left(\prod_{i=k+1}^{t}U^T diag[f`(h_{i-1})]\right)\frac{\partial y_t}{\partial h_t}\frac{\partial J_t}{\partial y_t}$$

# 梯度

- 损失 $J$ 关于 W 的梯度为：

$$\frac{\partial J}{\partial W} = \frac{\partial}{\partial W}\sum_{t=1}^{T} J_t = \sum_{t=1}^{T}\frac{\partial J_t}{\partial W}$$

$$= \sum_{t=1}^{T}\frac{\partial h_t}{\partial W}\frac{\partial J_t}{\partial h_t}$$



# 梯度

- 用链式法则得到

$$\frac{\partial J}{\partial W} = \sum_{t=1}^{T} \sum_{k=1}^{t} \frac{\partial h_k}{\partial W} \frac{\partial h_t}{\partial h_k} \frac{\partial y_t}{\partial h_t} \frac{\partial J_t}{\partial y_t}$$

$$h_t = f(Uh_{t-1} + Wx_t + b)$$

- 其中，

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}}$$

$$= \prod_{i=k+1}^{t} U^T diag[f`(h_{i-1})]$$

# 梯度

- 对$W$的梯度为

$$\frac{\partial J}{\partial W}=\sum_{t=1}^{T}\sum_{k=1}^{t}\frac{\partial h_k}{\partial W}\left(\prod_{i=k+1}^{t}U^T diag[f`(h_{i-1})]\right)\frac{\partial y_t}{\partial h_t}\frac{\partial J_t}{\partial y_t}$$

# 长期依赖问题/梯度消失问题

- 我们定义 $\gamma = ||\boldsymbol{U}^T \boldsymbol{diag}(f'(\boldsymbol{h}_{i-1}))||$，则在上面公式中的括号里面为 $\gamma^{t-k}$ 。如果$\gamma > 1$，当 t – k → ∞ 时，$\gamma^{t-k} \to \infty$，会造成系统不稳定，也就是所谓的梯度爆炸**梯度爆炸**问题；相反，如果$\gamma < 1$，当 t – k → ∞ 时，$\gamma^{t-k} \to 0$，会出现和深度前馈神经网络类似的**梯度消失**问题。

- 因此，虽然简单循环网络从理论上可以建立长时间间隔的状态之间的依赖关系（Long-Term Dependencies），但是由于梯度爆炸或消失问题，实际上只能学习到短周期的依赖关系。这就是所谓的**长期依赖问题**。

# 改进方案

- 为了避免梯度爆炸或消失问题，关键是使得 $\boldsymbol{U}^T \boldsymbol{diag}(f'(\boldsymbol{h}_{i-1}))= 1$。一种方式就是选取合适的参数，同时使用非饱和的激活函数。但这样的方式需要很多人工经验，同时限制了模型的广泛应用。

- 还有一种方式就是改变模型，比如让$\boldsymbol{U}$=1，同时使用 $f'(\boldsymbol{h}_{i-1})= 1$。

$$\boldsymbol{h}_t = \cancel{f(}\cancel{\boldsymbol{U}}\boldsymbol{h}_{\boldsymbol{t-1}} + \boldsymbol{W}\boldsymbol{x}_{\boldsymbol{t}} + b\cancel{)}$$

$$\boldsymbol{h}_t = \boldsymbol{h}_{\boldsymbol{t-1}} + \boldsymbol{W}\boldsymbol{g}(\boldsymbol{x}_{\boldsymbol{t}}) + b$$

- 但这样的形式，丢失了神经元在反馈边上的非线性激活的性质。

# 改进方案

- 一个更加有效的改进是引入一个新的状态 $\boldsymbol{c}_t$专门来进行线性的反馈传递，同时在 $\boldsymbol{c}_t$ 的信息非线性传递给 $\boldsymbol{h}_t$ 。

$$c_t = c_{t-1} + Wg(x_t)$$

$$h_t = \tanh(c_t)$$

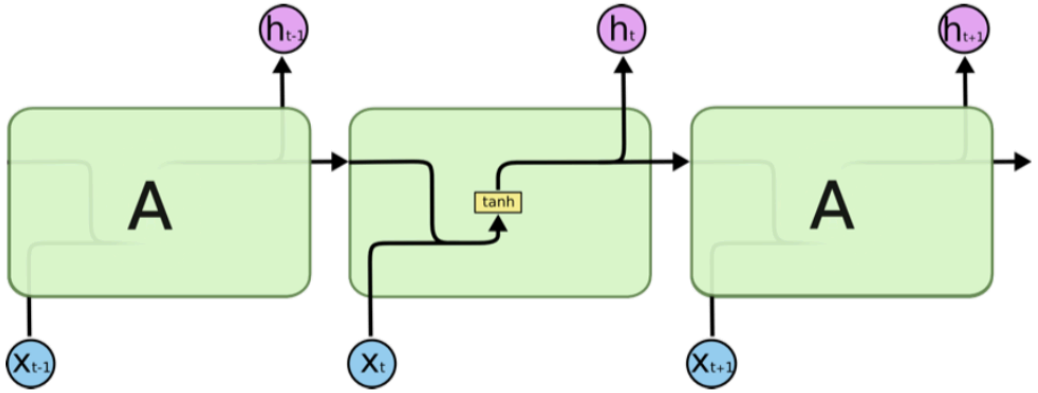
- 但是，这样依然存在一定的问题。因为 $\boldsymbol{c}_t$和$\boldsymbol{c}_{t-1}$ 是线性关系，同时不断累积 $\boldsymbol{x}_t$ 的信息，会使得 $\boldsymbol{c}_t$变得越来越大。

# 长短时记忆神经网络：LSTM

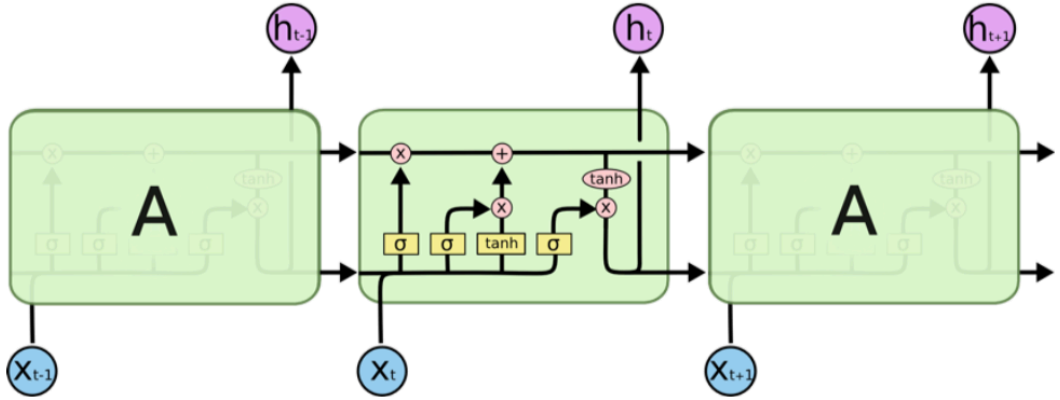
- 长短时记忆神经网络（Long Short-Term Memory Neural Network，LSTM）是RNN的一个变体，可以有效地解决简单循环神经网络的梯度爆炸或消失问题。

- LSTM 模型的关键是引入了一组记忆单元（Memory Units），允许网络可以学习何时遗忘历史信息，何时用新信息更新记忆单元。在时刻 t 时，记忆单元 $\mathbf{c}_t$ 记录了到当前时刻为止的所有历史信息，并受三个“门”控制：输入门 $\mathbf{i}_t$，遗忘门$\mathbf{f}_t$和输出门$\mathbf{o}_t$。三个门的元素的值在 [0, 1] 之间。

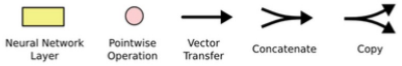
# Long Short Term Memory (LSTM)



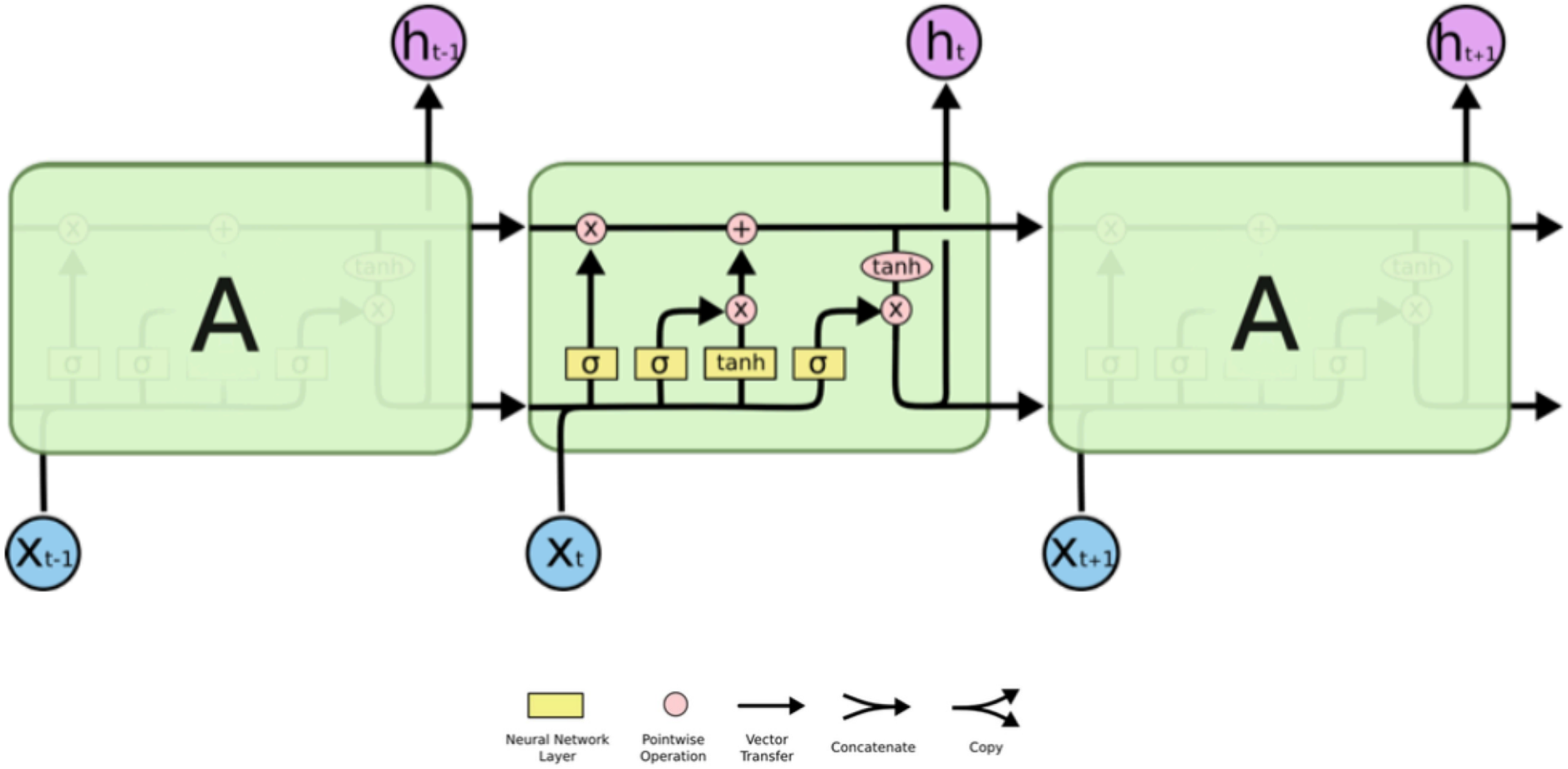

General RNN




LSTM

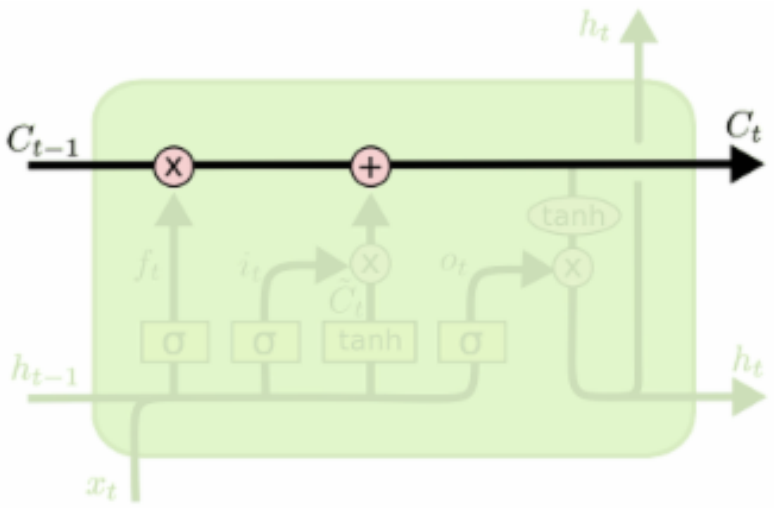


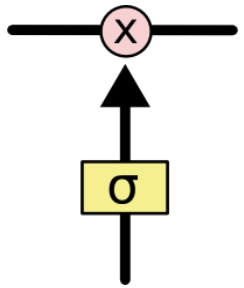
# Long Short Term Memory (LSTM)



# LSTM

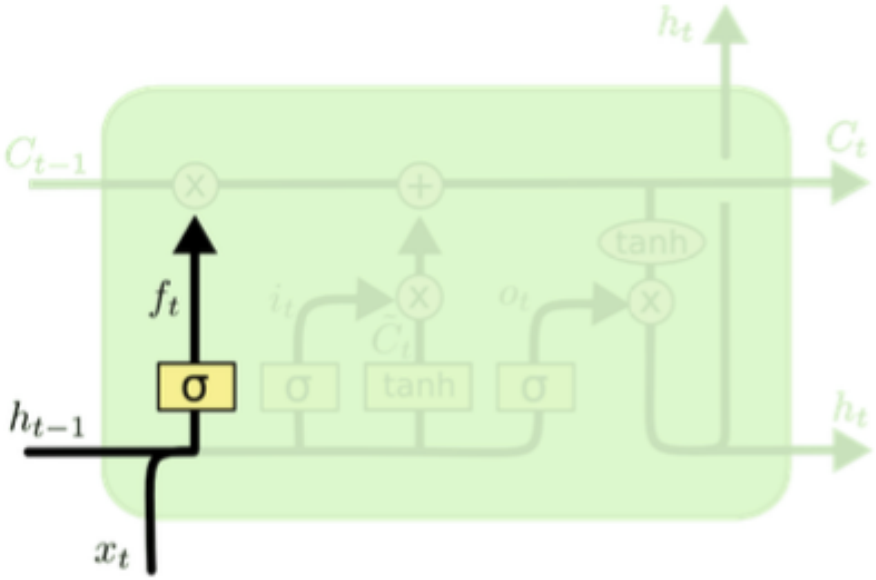
- 核心：记忆（细胞状态）



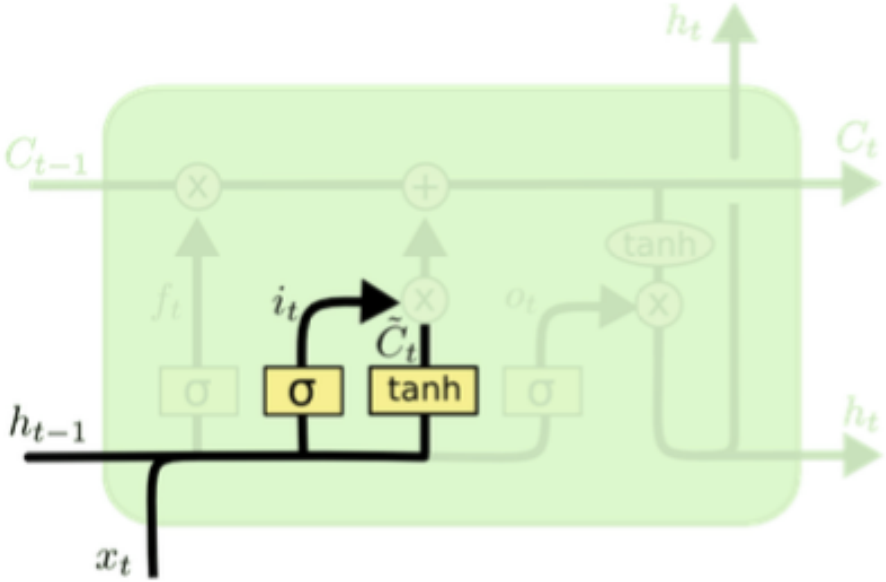

门
Sigmoid：1 or 0
Pointwise乘法

# Forget Gate



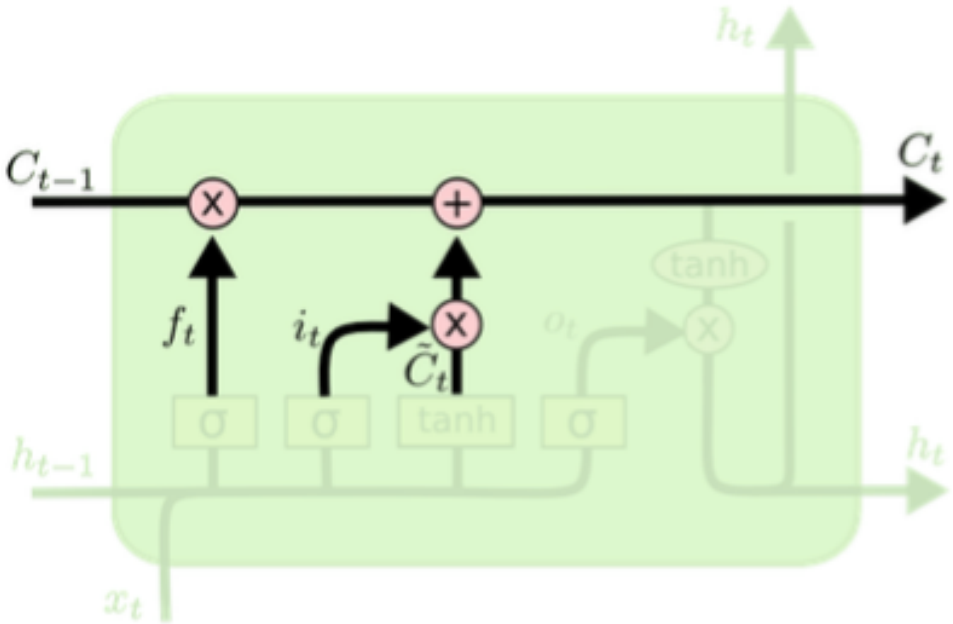

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

# Input Gate



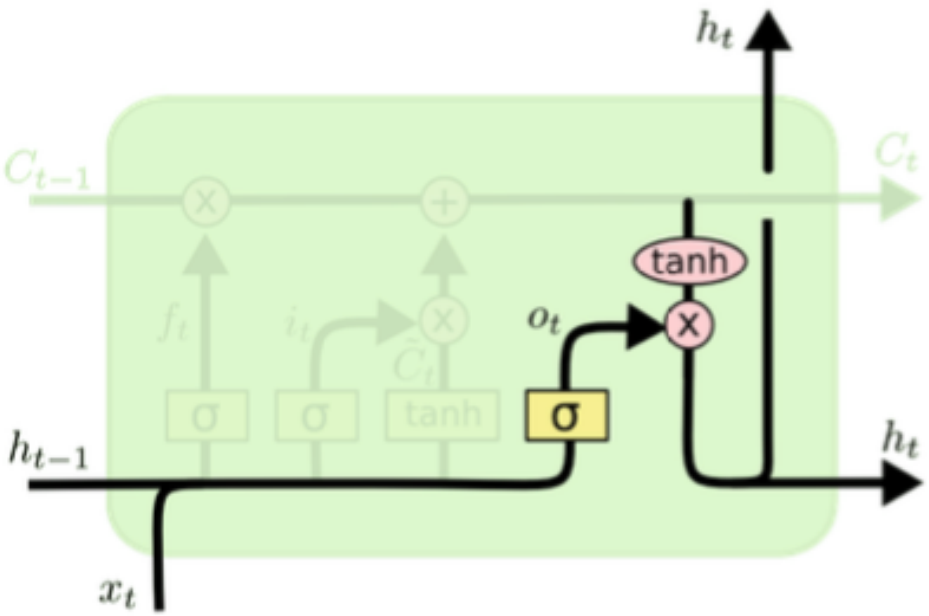

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

# Update Memory




$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

# Output Gate




$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

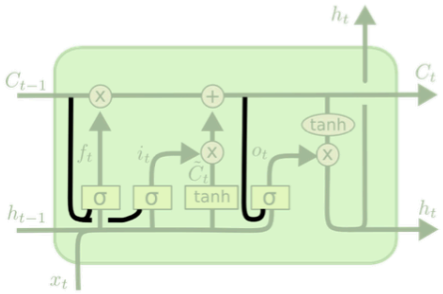
# Long Short Term Memory (LSTM)

- 在时刻 t 时 LSTM 的更新方式如下：

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + V_i\mathbf{c}_{t-1}),\\
\mathbf{f}_t &= \sigma(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + V_f\mathbf{c}_{t-1}),\\
\mathbf{o}_t &= \sigma(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + V_o\mathbf{c}_t),\\
\tilde{\mathbf{c}}_t &= \tanh(W_c\mathbf{x}_t + U_c\mathbf{h}_{t-1}),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t,\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t),
\end{aligned}
$$

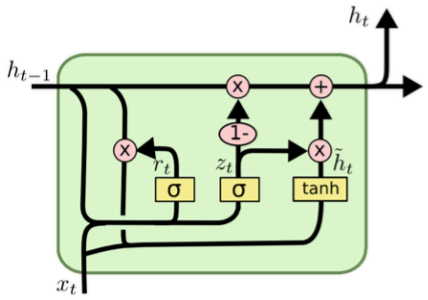
这里，$\mathbf{x}_t$是当前时刻的输入，σ 是 logistic 函数，$\mathbf{V}_i$ ，$\mathbf{V}_f$，$\mathbf{V}_o$是对角矩阵。
遗忘门 $\mathbf{f}_t$控制每一个内存单元需要遗忘多少信息
输入门 $\mathbf{i}_t$控制每一个内存单元加入多少新的信息
输出门$\mathbf{o}_t$控制每一个内存单元输出多少信息。

# LSTM变种




$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o\right)$$

Peephole




$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

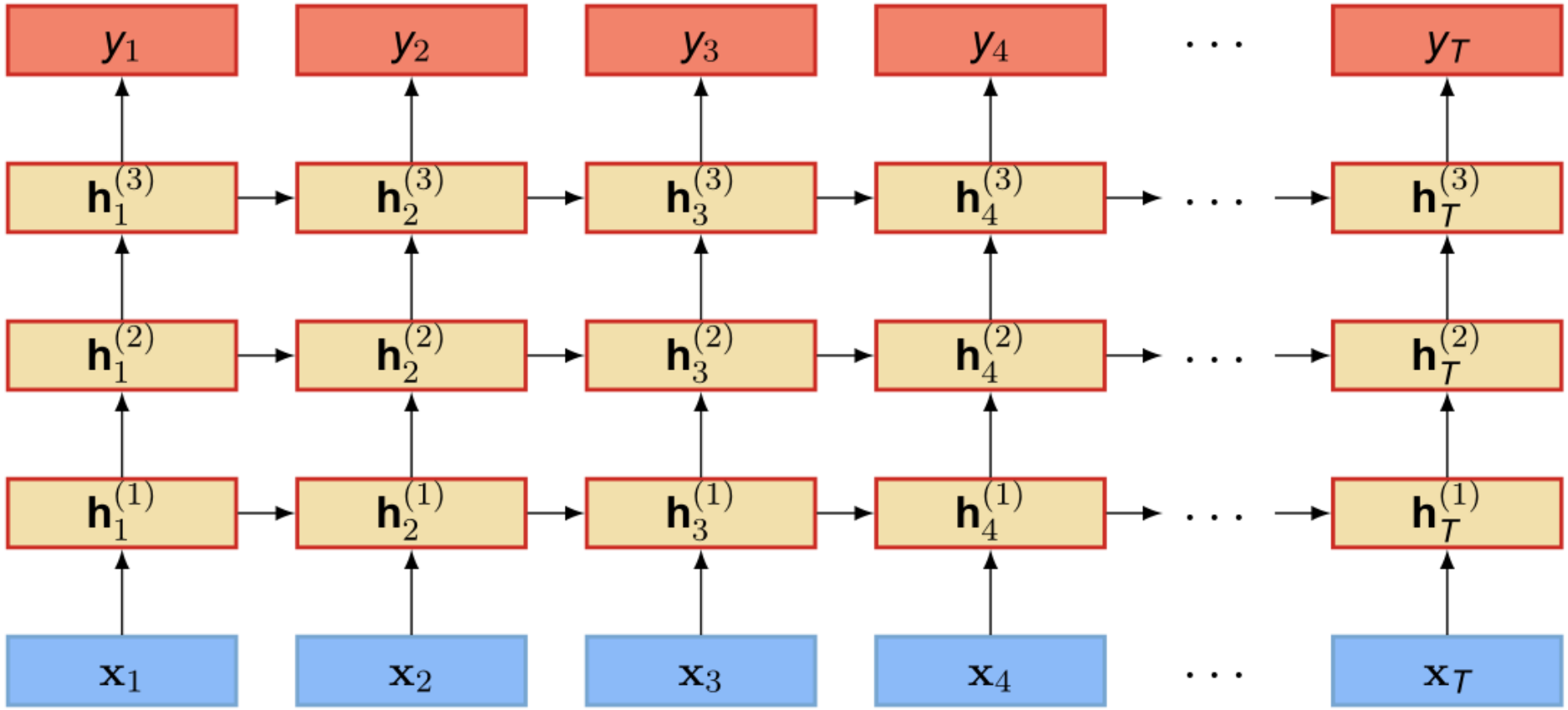
GRU

# 门限循环单元：GRU

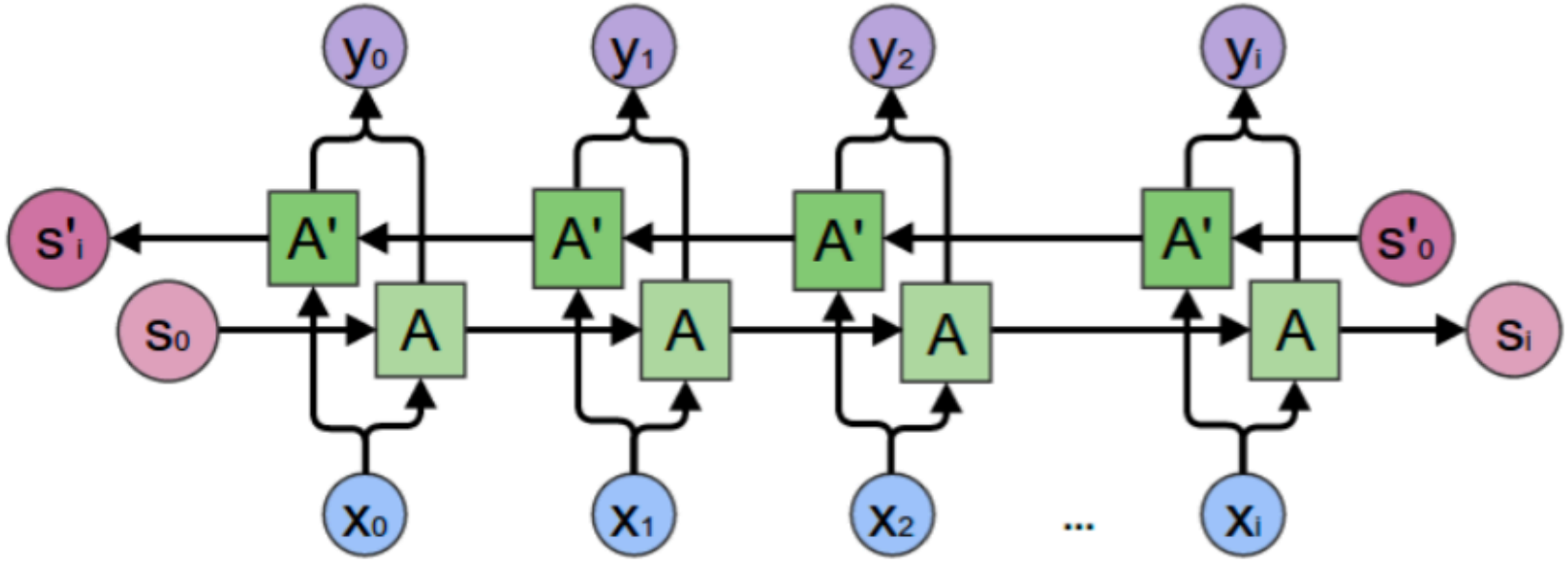
- 门限循环单元（Gated Recurrent Unit，GRU）是一种 LSTM 的简化版本。GRU 将输入门与和遗忘门合并成一个门：更新门（Update Gate），同时还合并了记忆单元$c_t$和神经元活性$h_t$。GRU 模型：更新门 **z** 和重置门 **r**
  - 更新门 **z**：用来控制当前的状态需要遗忘多少历史信息和接受多少新信息
  - 重置门 **r**：用来控制候选状态中有多少信息是从历史信息中得到

$$
\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}) \\
\mathbf{z}_t &= \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})), \\
\mathbf{h}_t &= \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t,
\end{aligned}
$$

# 堆叠(Stack)循环神经网络

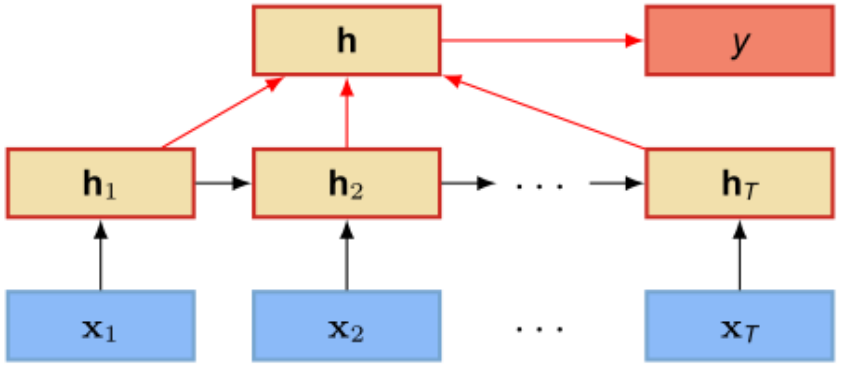


# 双向循环神经网络

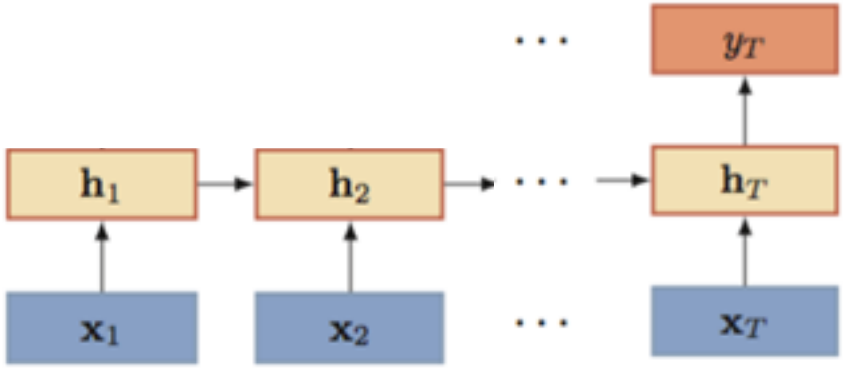


# 输出：序列到类别

- 输入为序列，输出为类别。比如在文本分类中，输入数据为单词的序列，输出为该文本的类别。



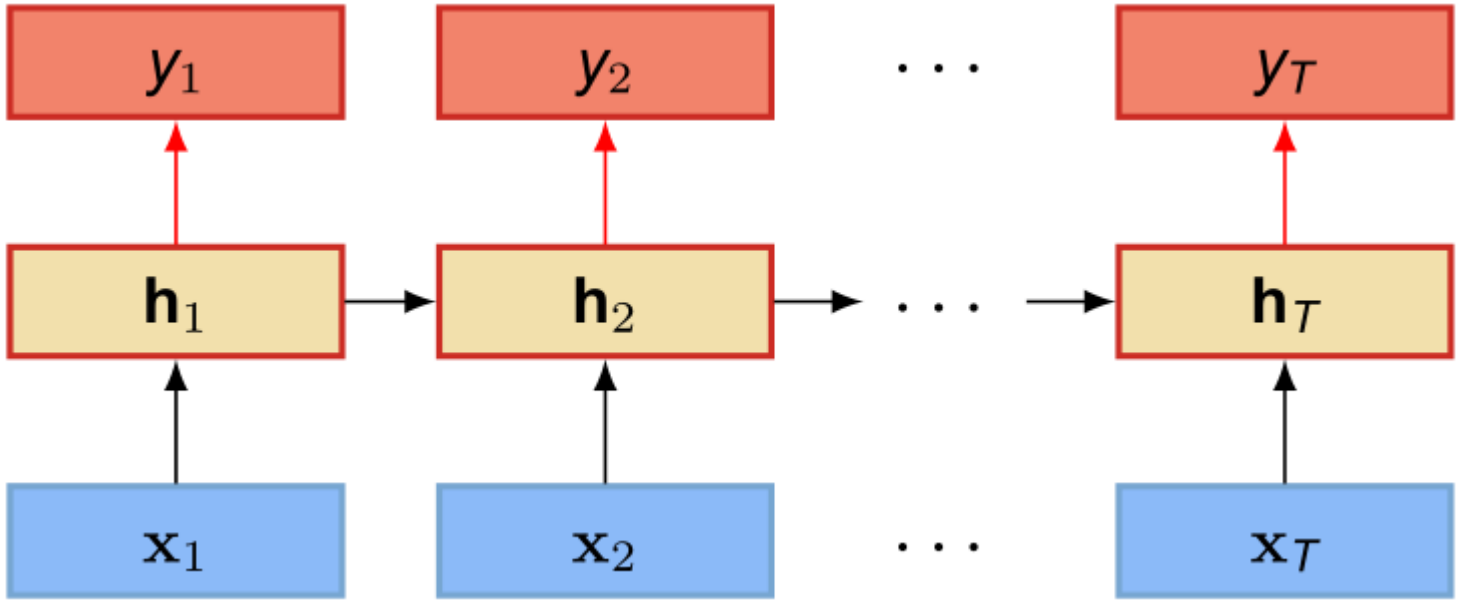

平均




不平均

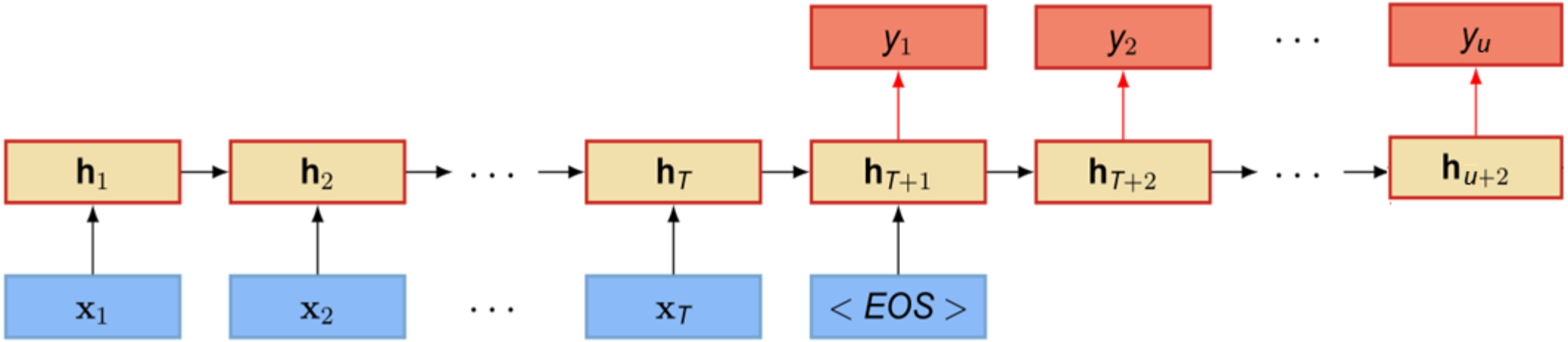
# 输出：同步序列到序列

- 输入和输出同步，即每一时刻都有输入和输出。比如在序列标注问题，每个时刻的输入都需要有一个输出。输入序列和输出序列的长度相同。



# 输出：异步序列到序列

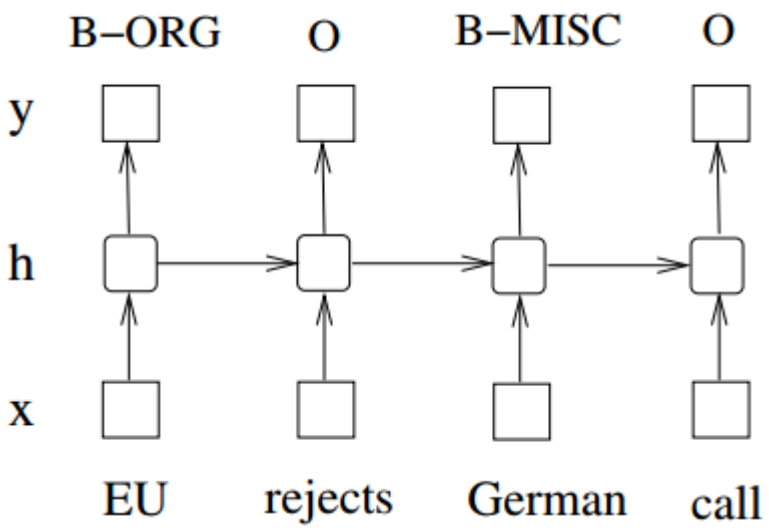
- 输入和输出不需要有严格的对应关系。比如在机器翻译中，输入为源语言的单词序列，输出为目标语言的单词序列。输入和输出序列并不需要保持相同的长度。

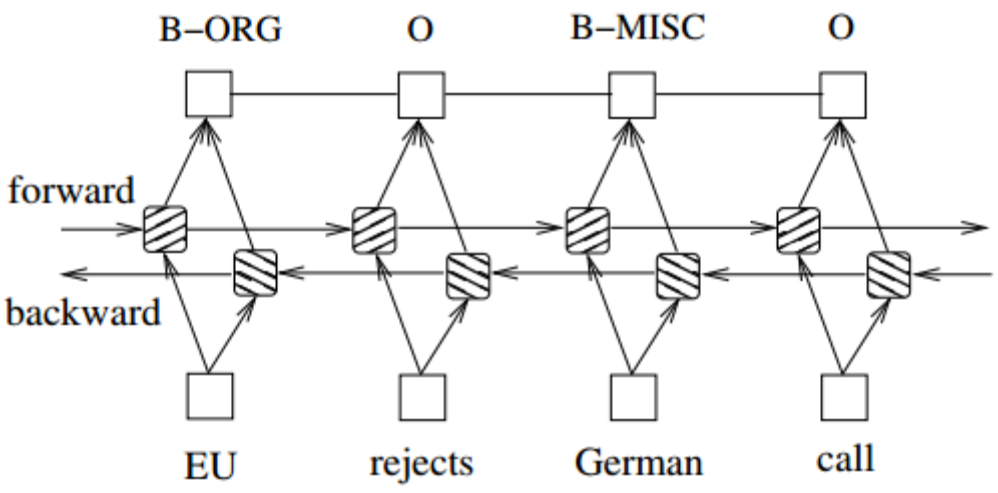


# 循环神经网络在NLP中的常见任务和应用

- 基础NLP任务：语言模型、文本表示、词法分析、句法分析
- NLP应用：文本分类、情感分析、机器翻译、自动问答、自动摘要等

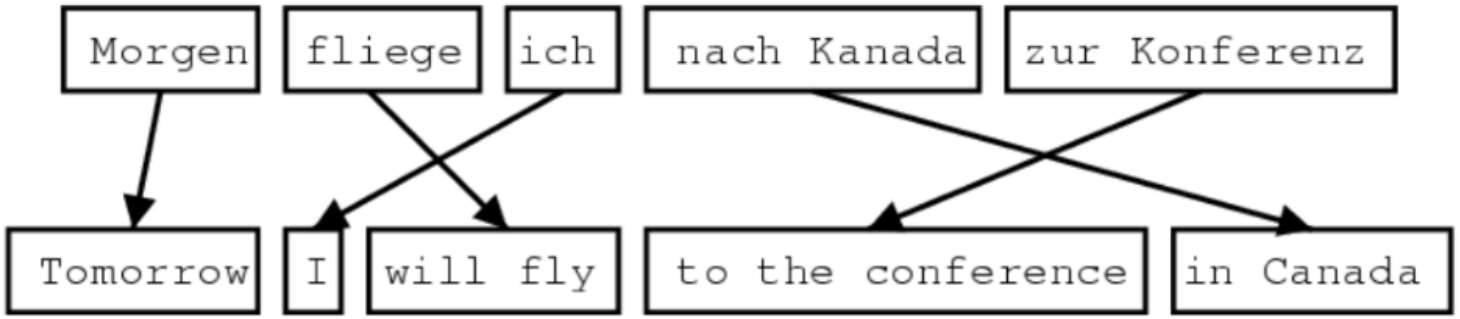
# 序列标注：NER


B–ORG
O
B–MISC
O
y
h
x
EU
rejects
German
call

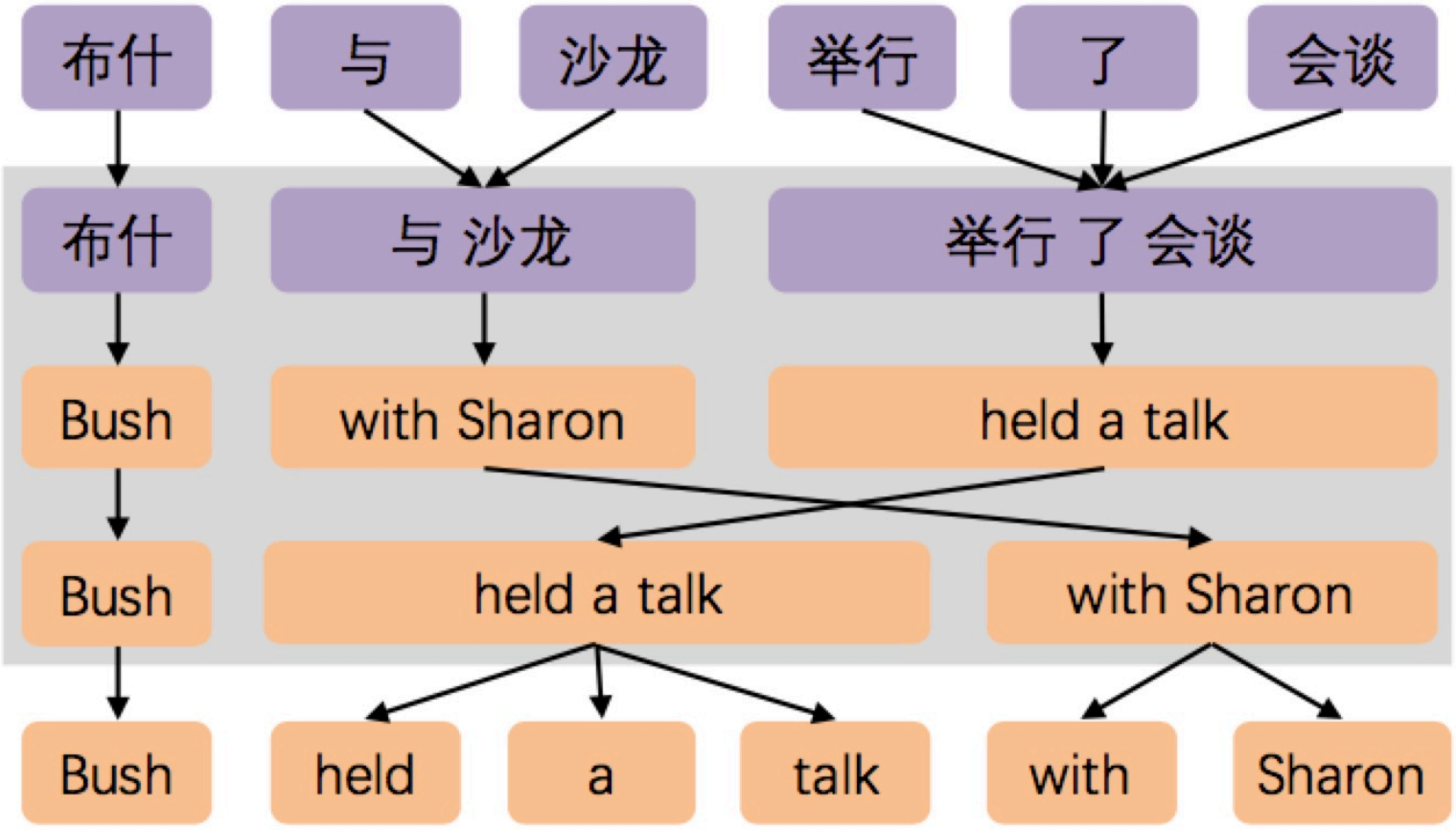

B–ORG
O
B–MISC
O
forward
backward
EU
rejects
German
call

# 机器翻译：传统统计机器翻译

- 源语言：f
- 目标语言：e
- 模型：$\hat{e} = \text{argmax}_e p(e|f) = \text{argmax}_e\ p(f|e)\ p(e)$
  - $p(f|e)$: 翻译模型
  - $p(e)$: 语言模型



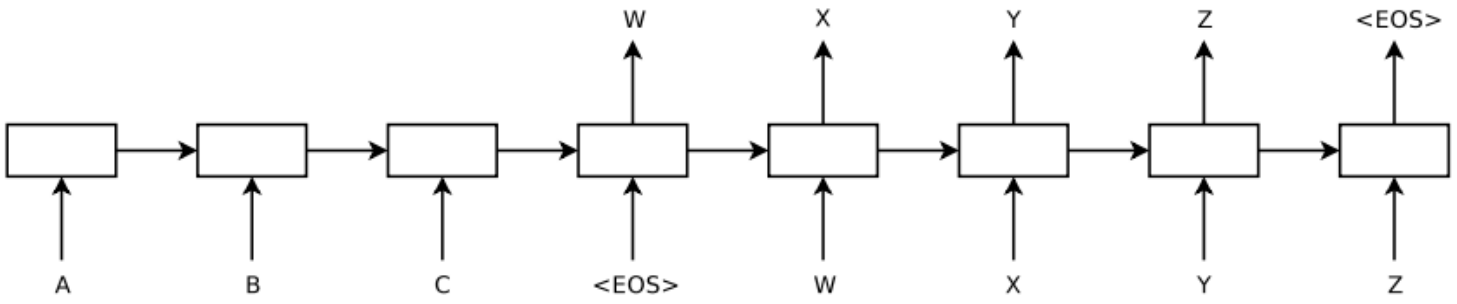
# 机器翻译：基于短语的统计机器翻译




(Koehn et al., 2003)

# 基于序列到序列的机器翻译

- 完全基于神经网络的机器翻译模型
  - 一个 RNN 用来编码：读入源句子（变长向量），转换成一个固定的上下文向量
  - 另一个 RNN 用来解码：给定上下文和之前预测的词，预测下一个翻译的词

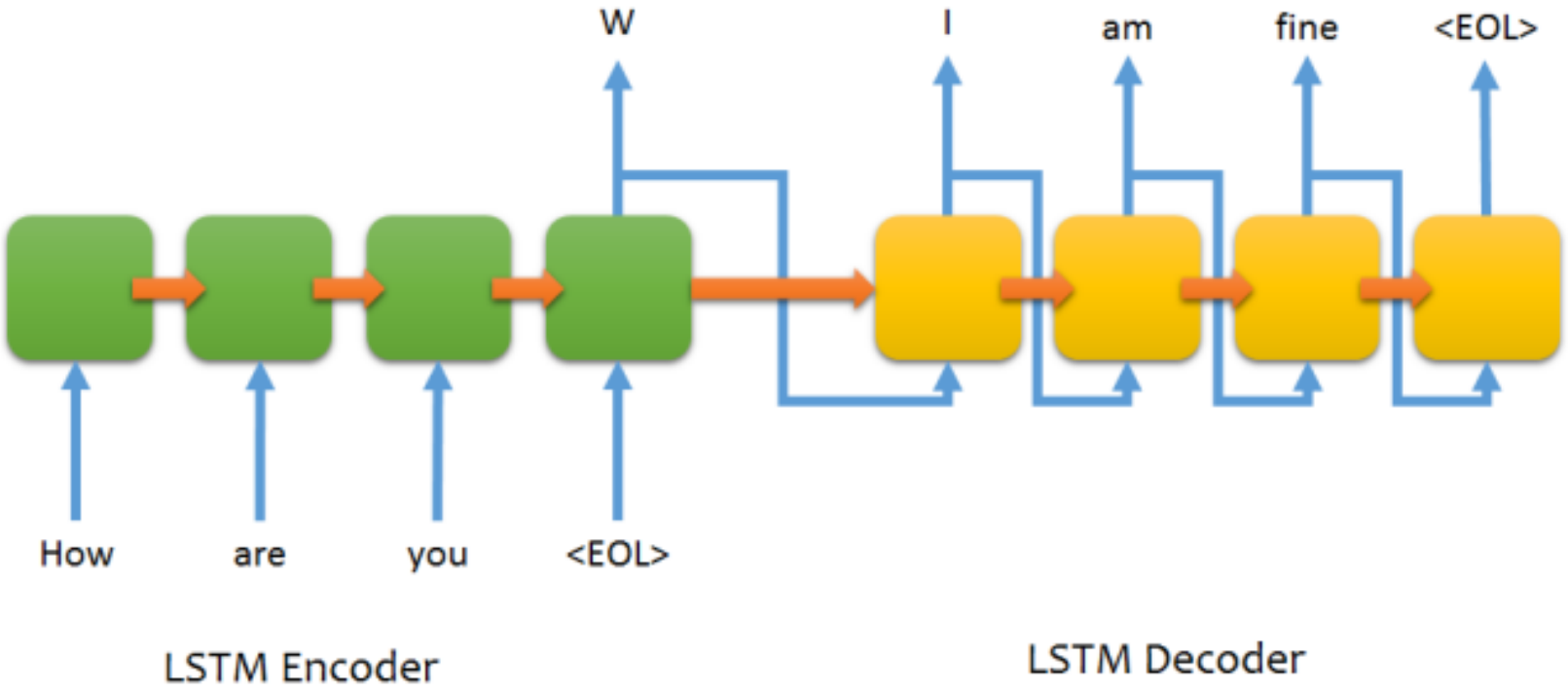



seq2seq

encoder-decoder

https://arxiv.org/pdf/1409.3215v3.pdf

# 对话系统：聊天机器人



# 其他应用

- 看图说话
- 代码生成
- 自动作诗
- …

- 自动摘要
- 阅读理解
- 关键词预测
- …

# 总结

- 机器学习基础
- 感知机和前馈神经网络
- 卷积神经网络
- 循环神经网络

# 谢谢

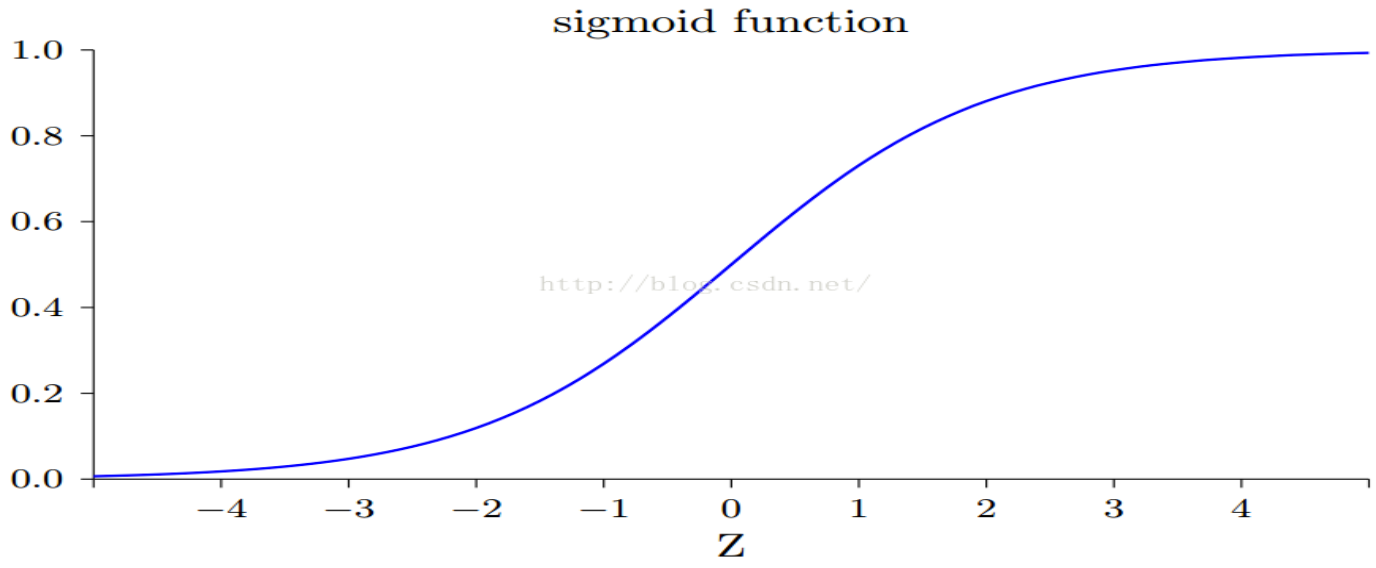
# 附录

# 交叉熵损失函数

- 二次损失函数

$$E = \frac{1}{2}\sum_{output}(t_j - y_j)^2 \qquad y = \sigma(z) = \frac{1}{1+e^{-z}} \qquad z = \sum_i w_i x_i$$

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w} = (t-y)\left(\frac{\partial}{\partial z}\frac{1}{1+e^{-z}}\right)x$$



# 交叉熵损失函数

- 快速更新的损失函数

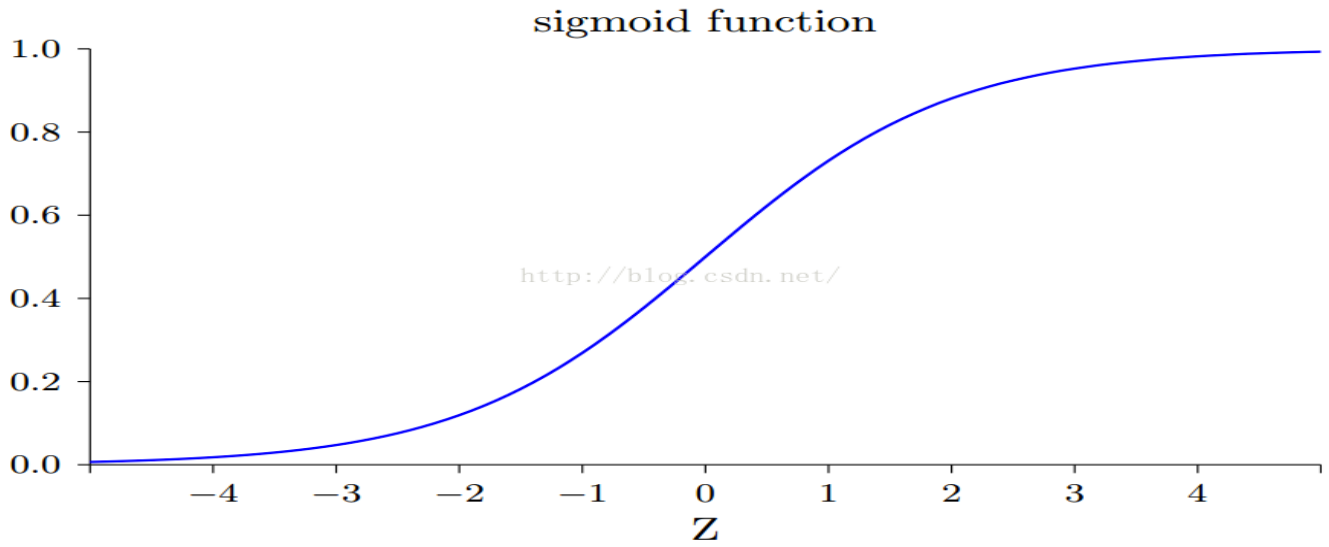
$$E = -\frac{1}{n}\sum_{x} tln(y) + (1-t)\ln(1-y)$$

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w}$$

$$= -\frac{1}{n}\sum_{x}\left(\frac{t}{\sigma(z)} - \frac{1-t}{1-\sigma(z)}\right)\frac{\partial\sigma(z)}{\partial z}x$$

$$= -\frac{1}{n}\sum_{x}\left(\frac{\frac{\partial\sigma(z)}{\partial z}x}{\sigma(z)(1-\sigma(z))}\right)(t-\sigma(z))$$

$$= -\frac{1}{n}\sum_{x} x(t-\sigma(z))$$



# Stochastic Gradient Descent (SGD)

- Gradient Descent
  - $\theta \leftarrow \theta + \lambda \nabla_\theta L$
- Stochastic Gradient Descent(SGD)
  - Online GD (=1), Minibatch SGD (>1)
  - 每次迭代随机选择一个样本计算梯度，并更新参数

$$w_{t+1} \leftarrow w_t - \eta \frac{\partial R(w_t, x_t, y_t)}{\partial w}$$

# Mini-batch SGD

批量梯度下降和随机梯度下降的折中
每次（随机）选取m个样本进行参数的更新

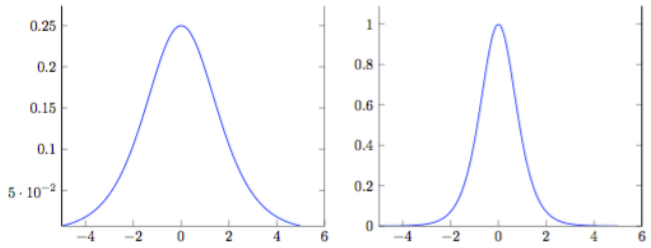
**Algorithm 8.1** Stochastic gradient descent (SGD) update at training iteration $k$

**Require:** Learning rate $\epsilon_k$.
**Require:** Initial parameter $\boldsymbol{\theta}$
  **while** stopping criterion not met **do**
    Sample a minibatch of $m$ examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow +\frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \hat{\boldsymbol{g}}$
  **end while**

# 梯度消失问题

- 在神经网络中误差反向传播的迭代公式为

$$\delta^l = (w^{l+1}\delta^{l+1}) \odot f'(z^l)$$

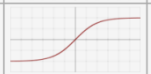



(a) logistic 函数的导数　(b) tanh 函数的导数

- 误差从输出层反向传播时,在每一层都要乘以该层的激活函数的导数
- Sigmod函数、ReLU函数、tanh函数

sigmod $\dfrac{\partial y}{\partial z} = y(1-y) \in [0,0.25]$

tanh $\dfrac{\partial y}{\partial z} = 1 - y^2 \in [0,1]$

| | | | |
|---|---|---|---|
| Logistic (a.k.a Soft step) | | $f(x) = \frac{1}{1+e^{-x}}$ | $f'(x) = f(x)(1-f(x))$ |
| TanH |  | $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$ | $f'(x) = 1 - f(x)^2$ |
| ArcTan | | $f(x) = \tan^{-1}(x)$ | $f'(x) = \frac{1}{x^2+1}$ |
| Rectified Linear Unit (ReLU)[7] | | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ |
| Parameteric Rectified Linear Unit (PReLU)[8] | | $f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ |

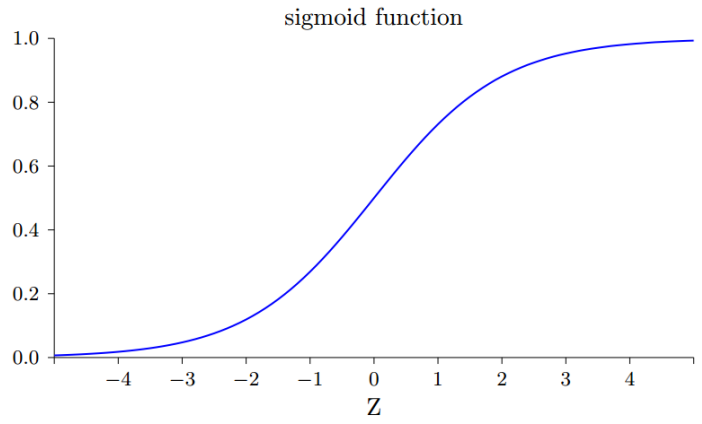
# 权重初始化

- 权重不能全部设为0
  - 每个神经元都有着相同的输出，在反向传播时具有相同的梯度
- 小随机数初始化
  - w = np.random.randn(1000)
  - 问题：一个神经网络的层中的权重值很小，那么在反向传播的时候就会计算出非常小的梯度（因为梯度与权重值是成比例的）。这就会很大程度上减小反向传播中的“梯度信号”

$$Var(z) = \sum_{i=1}^{n} Var(w_i x_i) \qquad z = \sum_{i=1}^{n} w_i x_i$$

$$Var(z) = \sum_{i=1}^{n} Var(x_i) Var(w_i), if\ E(w_i) = E(x_i) = 0$$

$$Var(z) = nVar(w)Var(x) \qquad Var(w) = 1/n$$



# 权重初始化：校准方差

- 解决方案：校准方差，除以1/sqrt(n)
  - `w = np.random.randn(n) / np.sqrt(n)`
  - ReLU:`w = np.random.randn(n) * np.sqrt(2/n)`

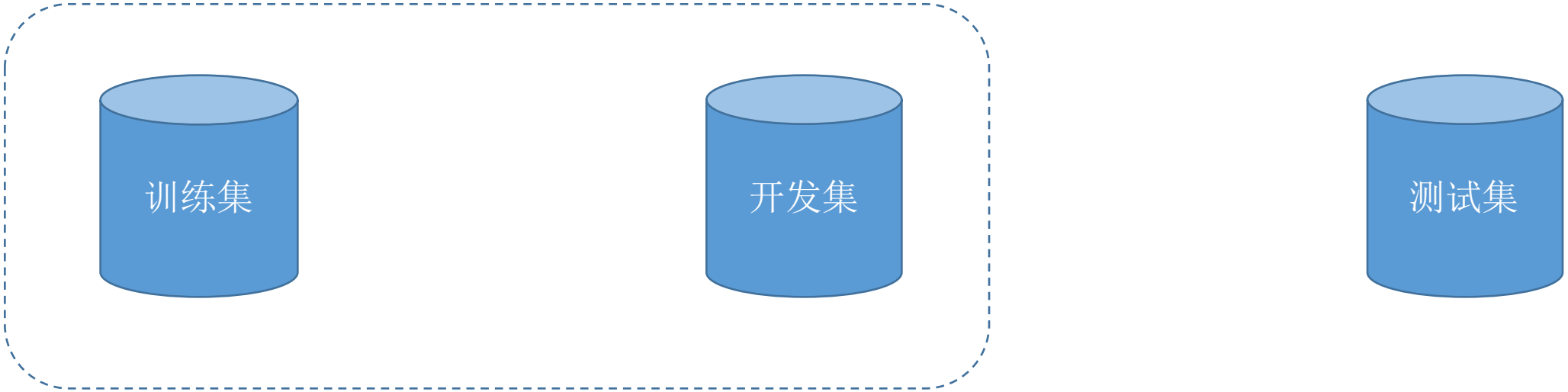
- b的初始化
  - 通常设为0
  - 有些设为如0.01的小值
  - 这样做能让所有的ReLU单元一开始就激活，这样就能保存并传播一些梯度

# 数据预处理

- 原始的训练数据中,每一维特征的来源以及度量单位不同,会造成这些特征值的分布范围往往差异很大。
  - 标准归一化:将每一个维特征都处理为符合标准正态分布(均值为 0,标准差为 1)
  - 缩放归一化:通过缩放将特征取值范围归一到 [0, 1] 或 [−1, 1]
  - 白化:使用 PCA (Principal Component Analysis) 去除掉各个成分之间的相关性

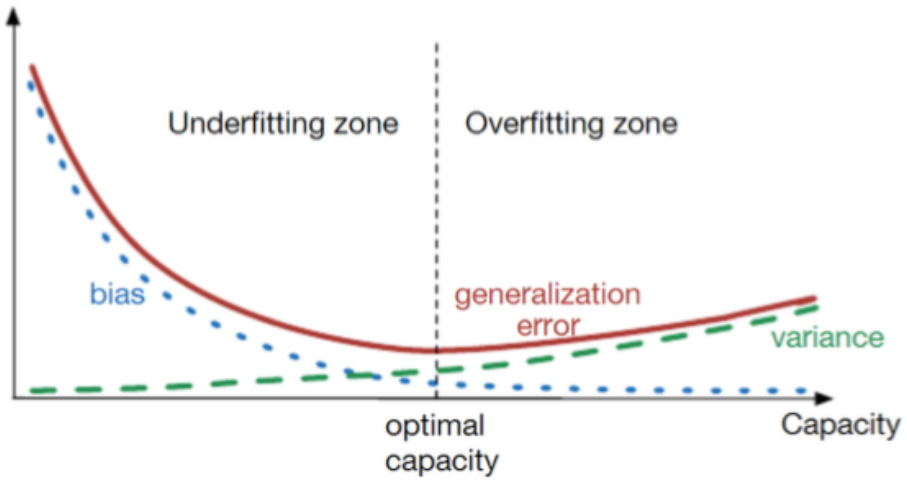
# Early-Stop

- 在梯度下降训练的过程中，由于过拟合的原因，在训练样本上收敛的参数，并不一定在测试集上最优。因此，我们使用一个**验证集**（Validation Dataset）（也叫**开发集**（Development Dataset））来测试每一次迭代的参数在验证集上是否最优。如果在验证集上的错误率不再下降，就停止迭代。这种策略叫 Early-Stop。如果没有验证集，可以在训练集上进行**交叉验证**

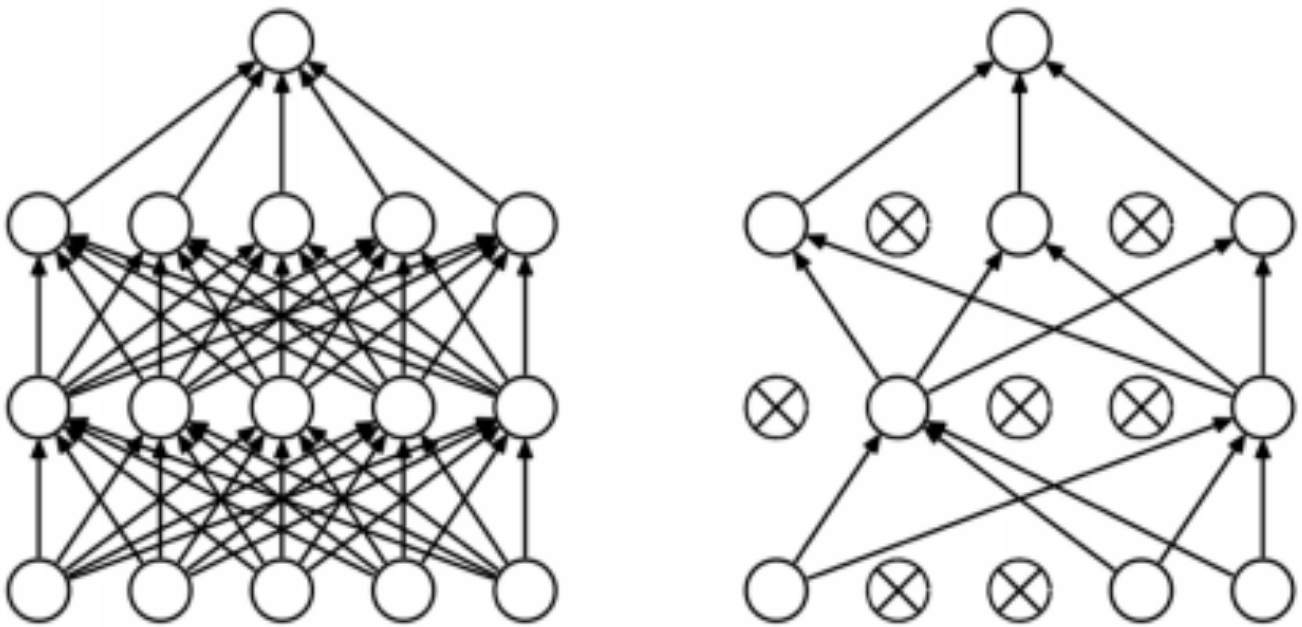


# Early Stop

- 利用开发集
  - learn model on training data
  - valuate_score_of_current_model = evaluate on validation
  - if valuate_score_of_current_model > best_validation_scor
    - best_validation_score = valuate_score_of_current_model
    - save current model
    - patient = 0
  - else: patient += 1
  - if patient > max_patient: – break
- 第二轮训练
  - 找到最佳训练轮数
  - 在训练集和开发集上再训练，当在开发集上达到一个较小值则停止

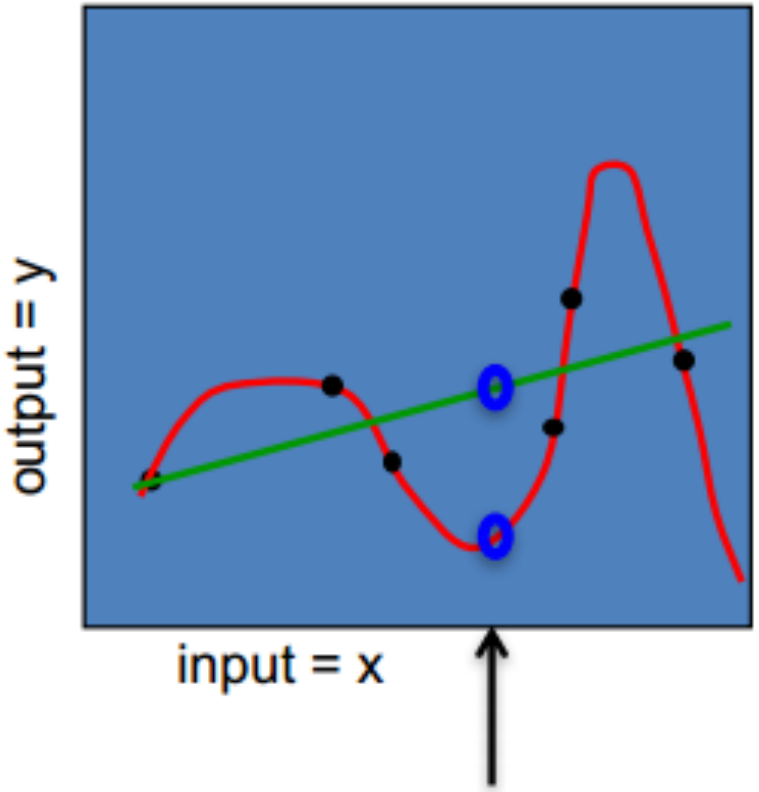


# Dropout



(a) Standard Neural Net (b) After applying dropout.

# 正则化 Regularization




Which output value should
you predict for this test input?

# 例子

- L1正则项
  - $L = L + \alpha|w|$

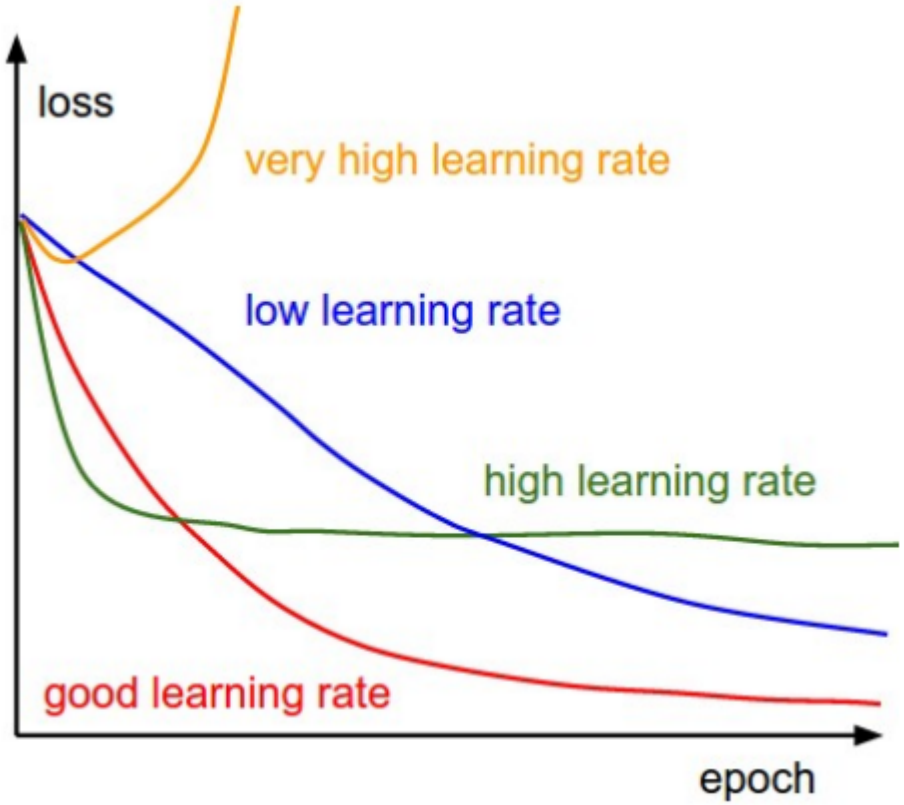
- L2正则项
  - 目标函数：$L = L + \alpha\frac{1}{2}||w||^2$
  - 梯度：$\nabla L = \nabla L + \alpha\, w$
  - 权重更新：$w = w - \lambda(\nabla L + \alpha\, w) = (1 - \lambda\alpha)w - \lambda\nabla L$

# 数据增强

- 不过拟合的另一个途径是数据增强，即自动生成更多的训练数据
  - 增加正、负样本
  - 增加伪噪声数据

# 学习率

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial E}{\partial w_{ij}}$$



# 加速

- 自适应变长加速
  - AdaGard
  - RMSprop
  - Adam
  - AdaDelta

$$\theta_t = \theta_{t-1} - \frac{\rho}{\sqrt{\sum_{\tau=1}^{t} g_\tau^2}} g_t$$

其中，$\rho$ 是初始的学习率，$g_\tau$ 是第 $\tau$ 次迭代时的梯度。随着迭代次数的增加，梯度逐渐缩小

# 步骤

- 设计网络结构
- 设定每层节点数
- 算出梯度
- 写代码
- 利用数据调参

# 小结

- 用ReLU作为激活函数
- 分类时用交叉熵作为损失函数
- SGD+mini-batch
- 数据预处理（归一化）
- 动态学习率（动态调整步长）
- 用L1或者L2正则化（跳过前几轮）
- Dropout
- 层数深（分层训练 ）
- 数据增强

# 第3章 知识表示方法

王 泉

中国科学院信息工程研究所
中国科学院大学网络空间安全学院

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

# 知识定义

- Feigenbaum
  知识是经过削减、塑造、解释和转换的信息。简单地说，知识是经过加工的信息。
- Bernstein
  知识是由特定领域的描述、关系和过程组成的。
- Hayes-Roth
  知识是事实、信念和启发式规则。从知识库的观点看，知识是某领域中所涉及的各有关方面的一种符号表示。

# 知识分类

- 陈述性知识 (declarative knowledge)：用于描述领域内有关概念、事实、事物的属性和状态等。
  - 太阳从东方升起
  - 一年有春夏秋冬四个季节
- 过程性知识 (procedural knowledge)：用于指出如何处理与领域相关的信息，以求得问题的解。
  - 菜谱中的炒菜步骤
  - 如果信道畅通，请发绿色信号
- 元知识 (meta knowledge)：关于知识的知识，包括怎样使用规则、解释规则、校验规则、解释程序结构等知识。

# 知识表示

- 知识表示可看成是一组事物的约定，以把人类知识表示成机器能处理的数据结构。对知识进行表示的过程就是把知识编码成某种数据结构的过程。
- 知识表示方法分为：
  - 陈述性知识表示：将知识表示与知识的运用分开处理，在知识表示时，并不涉及如何运用知识的问题，是一种静态的描述方法。
  - 过程性知识表示：将知识表示与知识的运用相结合，知识寓于程序中，是一种动态的描述方法。

# 知识表示准则

- 表示知识的范围是否广泛
- 是否适于推理
- 是否适于加入启发信息
- 是否适于计算机处理
- 是否有高效的求解算法
- 陈述性表示还是过程性表示
- 能否表示不精确知识
- 能否在同一层次上和不同层次上模块化
- 知识和元知识能否用统一的形式表示
- 表示方法是否自然

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

一阶谓词逻辑是最早用在人工智能领域的知识表示方法之一。它以数理逻辑为基础，是到目前为止能够表达人类思维和推理的一种最精确的形式语言。其表现方式和人类自然语言也非常接近，容易为计算机理解和操作，并支持精确推理。

# 基本概念：命题与联结词

- 命题 (proposition)：具有真假意义的陈述句。
  - 太阳从东方升起
  - 一年有春夏秋冬四个季节
- 逻辑联结词 (logical connective)：用于将多个原子命题组合成复合命题。
  - $\neg$：否定联结词，$\neg P$表示对原命题的否定。
  - $\vee$：析取联结词，$P \vee Q$表示两个命题存在“或”的关系。
  - $\wedge$：合取联结词，$P \wedge Q$表示两个命题存在“与”的关系。
  - $\rightarrow$：蕴含联结词，$P \rightarrow Q$表示“如果$P$，则$Q$”。
  - $\leftrightarrow$：等价联结词，$P \leftrightarrow Q$表示“$P$当且仅当$Q$”。

# 基本概念：命题与联结词

- 复合命题与原子命题的真值关系表

| $P$ | $Q$ | $\neg P$ | $P \vee Q$ | $P \wedge Q$ | $P \rightarrow Q$ | $P \leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| 真 | 真 | 假 | 真 | 真 | 真 | 真 |
| 真 | 假 | 假 | 真 | 假 | 假 | 假 |
| 假 | 真 | 真 | 真 | 假 | 真 | 假 |
| 假 | 假 | 真 | 假 | 假 | 真 | 真 |

# 基本概念：个体词、谓词与量词

- 个体词：领域内可以独立存在的具体或抽象的客体。
  - 太阳从东方升起：太阳
  - 小王与小张同岁：小王、小张
- 在谓词逻辑中，个体可以是常量也可以是变量（变元）。
  - 个体常量：表示具体的或特定的个体。
  - 个体变量：表示抽象的或泛指的个体。
  - 个体域（论域）：个体变量的取值范围，可以是有限集合，也可以是无穷集合。

# 基本概念：个体词、谓词与量词

- 谓词（predicate）：用来刻画个体性质以及个体之间相互关系的词。
  - 命题：$x$是有理数

    $x$是个体变量，“…是有理数”是谓词，记为Rational，命题符号化为Rational($x$)。
  - 命题：小王与小张同岁

    小王和小张是个体常量，“…与…同岁”是谓词，记为SameAge，命题符号化为SameAge(小王, 小张)。
- n元谓词：含有n个个体符号的谓词$\mathrm{P}(x_1, x_2, \cdots, x_n)$。
  - 一元谓词（$n = 1$）：表示$x_1$具有性质P。
  - 多元谓词（$n \geq 2$）：表示$x_1, x_2, \cdots, x_n$具有关系P。

# 基本概念：个体词、谓词与量词

- 函数：又称函词，是从若干个个体到某个个体的映射。
  - Father(小张)：小张的父亲
  - Sum(1,2)：1与2的加和
- 谓词与函数的区别：
  - 谓词实现的是从个体域中的个体到真或假的映射，而函数实现的是从个体域中的一个（或若干）个体到另一个个体的映射，无真值可言。
  - 在谓词逻辑中，函数本身不能单独使用，它必须嵌入到谓词中。

# 基本概念：个体词、谓词与量词

- 量词（quantifier）：是表示个体数量属性的词。
  - 全称量词：符号化为∀（All）
    - 日常生活和数学中所用的“一切的”、“所有的”、“每一个”、“任意的”、“凡”、“都”等词可统称为全称量词。
    - $x$表示个体域里的某个个体，$\forall x\ \mathrm{P}(x)$表示个体域里所有个体都具有性质P。
  - 存在量词：符号化为∃（Exist）
    - 日常生活和数学中所用的“存在”、“有一个”、“有的”、“至少有一个”等词可统称为存在量词。
    - $y$表示个体域里的某个个体，$\exists y\ \mathrm{Q}(y)$表示个体域里存在个体$y$具有性质Q。

# 谓词逻辑表示法

- 用谓词逻辑既可以表示事物的状态、属性、概念等事实性知识，也可以表示事物间具有因果关系的规则性知识。
- 用谓词逻辑表示知识的一般步骤：
  - 定义谓词及个体，确定每个谓词及个体的确切含义。
  - 根据所要表达的事物或概念，为每个谓词中的变量赋以特定的值。
  - 根据所要表达的知识的语义，用适当的逻辑联结词将各个谓词连接起来形成谓词公式。

# 谓词逻辑表示法示例

- 用谓词逻辑表示下列知识：
  - 北京是一个美丽的城市，但她不是一个沿海城市。
  - 北京是中国的首都。
  - 每一个国家的首都都必定位于这个国家。

① 定义谓词和个体域如下：

- $\mathrm{BCity}(x)$：$x$是一个美丽的城市
- $\mathrm{CCity}(x)$：$x$是一个沿海城市
- $\mathrm{CapitalOf}(x, y)$： $x$是$y$的首都
- $\mathrm{LocatedIn}(x, y)$： $x$位于$y$
- $x \in \{城市\}, y \in \{国家\}$

# 谓词逻辑表示法示例

- 用谓词逻辑表示下列知识：
  - 北京是一个美丽的城市，但她不是一个沿海城市。
  - 北京是中国的首都。
  - 每一个国家的首都都必定位于这个国家。

② 将个体带入谓词中，得到：

- $\mathrm{BCity}(\mathrm{Beijing})$
- $\mathrm{CCity}(\mathrm{Beijing})$
- $\mathrm{CapitalOf}(\mathrm{Beijing}, \mathrm{China})$
- $\mathrm{CapitalOf}(x, y)$
- $\mathrm{LocatedIn}(x, y)$

# 谓词逻辑表示法示例

- 用谓词逻辑表示下列知识：
  - 北京是一个美丽的城市，但她不是一个沿海城市。
  - 北京是中国的首都。
  - 每一个国家的首都都必定位于这个国家。

③ 根据语义，用逻辑联结词连接：

- $\mathrm{BCity}(\mathrm{Beijing}) \wedge \neg \mathrm{CCity}(\mathrm{Beijing})$
- $\mathrm{CapitalOf}(\mathrm{Beijing}, \mathrm{China})$
- $\forall \mathrm{x}, \mathrm{y}\ \ \mathrm{CapitalOf}(x, y) \rightarrow \mathrm{LocatedIn}(x, y)$

# 谓词逻辑表示法特性

- 优点：
  - 精确性：可以较准确地表示知识并支持精确推理。
  - 通用性：拥有通用的逻辑演算方法和推理规则。
  - 自然性：是一种接近于人类自然语言的形式语言系统。
  - 模块化：各条知识相对独立，它们之间不直接发生联系，便于知识的添加、删除和修改。
- 缺点：
  - 表示能力差：只能表示确定性知识，不能表示非确定性知识、过程性知识和启发式知识。
  - 管理困难：缺乏知识的组织原则，知识库管理困难。
  - 效率低：把推理演算与知识含义截然分开，往往使推理过程冗长，降低了系统效率。

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

产生式系统是用规则序列的形式来描述问题的思维过程，形成求解问题的思维模式。系统中的每一条规则称为一个产生式。目前产生式规则表示法已成为专家系统首选的知识表示方式，也是人工智能中应用最多的一种知识表示方式。

# 基本概念：事实与规则

- 事实：断言一个语言变量的值或断言多个语言变量之间关系的陈述句。
  - 雪是白的

    语言变量：雪；语言变量的值：白的
  - 小王与小张同岁

    语言变量：小王、小张；语言变量之间的关系：同岁
- 确定性事实：一般用三元组的形式表示为
  - （对象，属性，值）或（关系，对象1，对象2）
- 不确定性事实：一般用四元组的形式表示为
  - （对象，属性，值，置信度）
  - （关系，对象1，对象2，置信度）

# 基本概念：事实与规则

- 规则：也称为产生式，通常用于表示事物之间的因果关系。
- 确定性规则：通常表示为

$$P \rightarrow Q \quad \text{或} \quad \text{IF } P \text{ THEN } Q$$

  - $P$是产生式的前提或条件；
  - $Q$是一组结论或操作，用于指出前提$P$所指示的条件被满足时，应该得出的结论或应该执行的操作。
- 不确定性规则：通常表示为

$$P \rightarrow Q\ (\text{置信度}) \quad \text{或} \quad \text{IF } P \text{ THEN } Q\ (\text{置信度})$$

  - $P$是产生式的前提或条件，$Q$是一组结论或操作。
  - 已知事实与前提条件不能精确匹配时，只要按照置信度的要求模糊匹配，再按特定算法将不确定性传递到结论。

# 产生式系统结构

- 产生式系统（production system）由数据库、规则库和推理机三部分组成。

  - 数据库：用来存放问题的初始状态、已知事实、推理的中间结果和最终结论等。
  - 规则库：用来存放与求解问题有关的所有规则。
  - 推理机：用来控制整个系统的运行，决定问题求解的线路，包括匹配、冲突消解、路径解释等。

# 正向推理的产生式系统

- 正向推理：从已知事实出发，通过规则求得结论，也称为数据驱动方式或自底向上的方式。
- 推理过程：
  - 规则库中的规则前件与数据库中的事实进行匹配，得到匹配的规则集合；
  - 使用冲突消解算法，从匹配规则集合中选择一条规则作为启用规则；
  - 执行启用规则的后件，并将该规则的后件送入数据库；

重复上述过程直至达到目标。

# 正向推理的产生式系统示例

动物识别产生式系统：目标 = A是猎豹？

- 已有知识（规则库）：

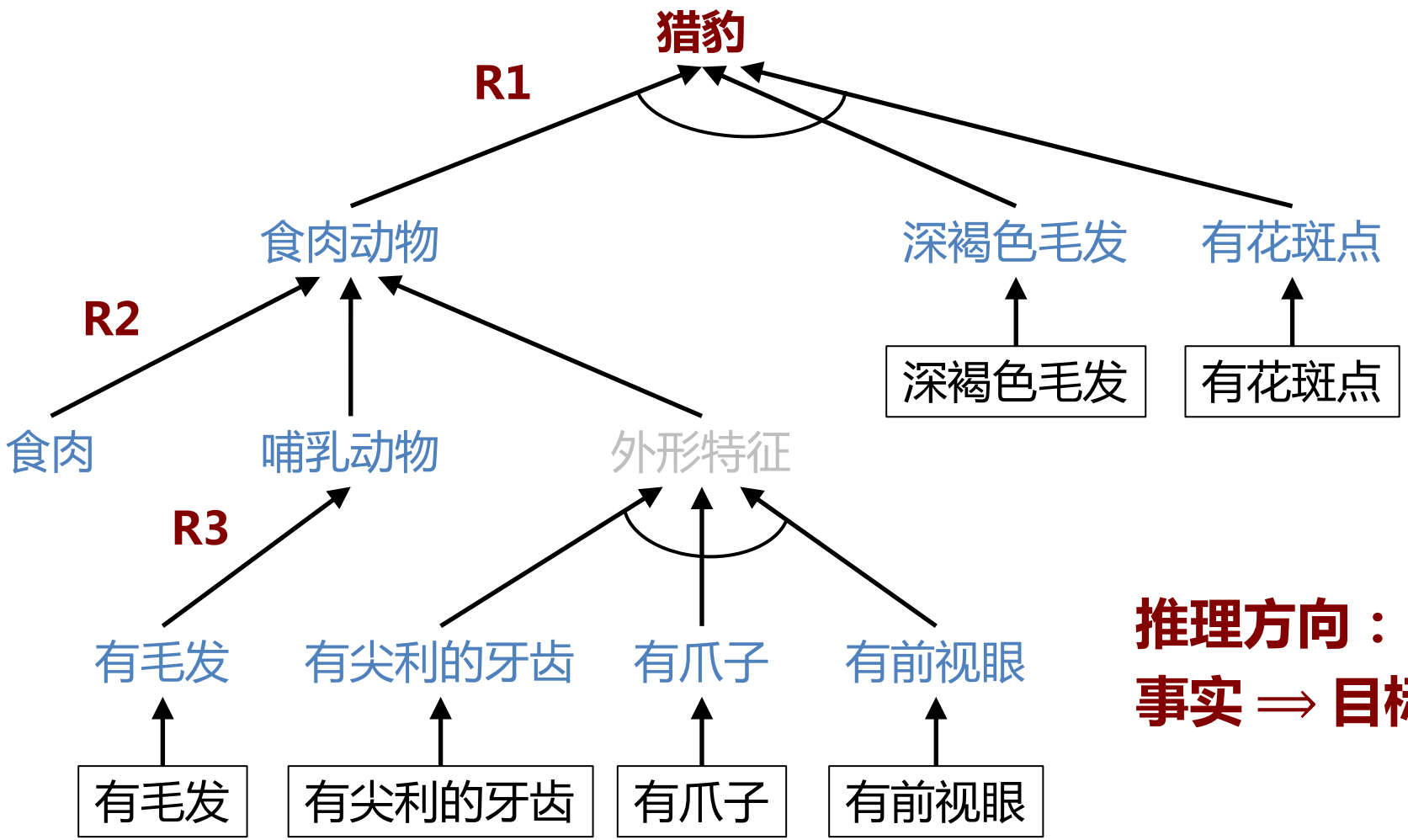
| | | |
|---|---|---|
| R1： | IF | X是食肉动物 ∧ X毛发是深褐色 ∧ X是有花斑点 |
| | THEN | X是猎豹 |
| R2： | IF | (X是哺乳动物) ∨ (X食肉) ∨ (X有尖利的牙齿 ∧ X有爪子 ∧ X有前视眼) |
| | THEN | X是食肉动物 |
| R3： | IF | X有毛发 |
| | THEN | X是哺乳动物 |

- 已知事实（事实库）

A有毛发；　A有尖利的牙齿；　A有爪子；　A有前视眼；

A毛发是深褐色；　A有花斑点

# 正向推理的产生式系统示例




**推理方向：**

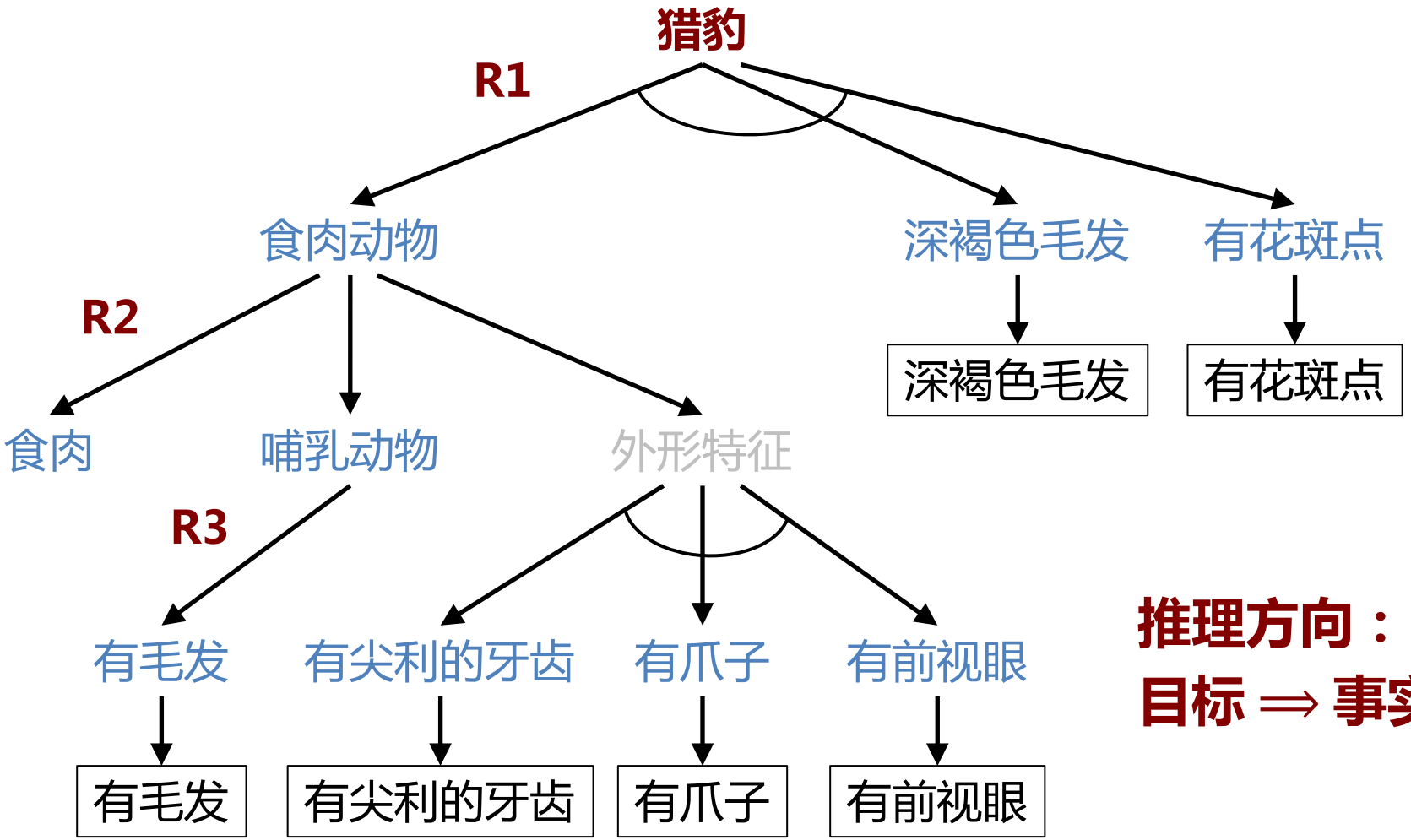
**事实 ⇒ 目标**

# 反向推理的产生式系统

- 反向推理：从目标出发，反向使用规则，求得已知事实，也称为目标驱动方式或自顶向下的方式。
- 推理过程：
  - 规则库中的规则后件与目标事实进行匹配，得到匹配的规则集合；
  - 使用冲突消解算法，从匹配规则集合中选择一条规则作为启用规则；
  - 将启用规则的前件作为子目标；

  重复上述过程直至各子目标均为已知事实。

# 反向推理的产生式系统示例




**推理方向：**

**目标 ⇒ 事实**

# 产生式规则表示法特性

- 优点：
  - 有效性：既可以表示确定性知识，又可以表示不确定性知识，有利于启发性和过程性知识的表达。
  - 自然性：用“如果…，则…”表示知识，直观、自然。
  - 一致性：所有规则具有相同的格式，并且数据库可被所有规则访问，便于统一处理。
  - 模块化：各条规则之间只能通过数据库发生联系，不能相互调用，便于知识的添加、删除和修改。
- 缺点：
  - 效率低：求解是反复进行的“匹配—冲突消解—执行”过程，执行效率低。
  - 表示的局限性：不能表示结构性或层次性知识。

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

框架表示法是以框架理论为基础发展起来的一种结构化知识表示方式，适用于表达多种类型的知识。框架理论认为人们对现实世界中各种事物的认识都是以一种类似于框架的结构存储在记忆当中的，当面临一个新事物时，就从记忆中找出一个适合的框架，并根据实际情况对其细节加以修改补充，从而形成对当前事物的认识。

# 框架

- 框架（frame）：是一种描述所论对象属性的数据结构。
  - 框架名：用来指代某一类或某一个对象。
  - 槽：用来表示对象的某个方面的属性。
  - 侧面：有时一个属性还要从不同侧面来描述。
  - 值：槽/侧面的取值，可以为原子型，也可以为集合型。

| | | | |
|---|---|---|---|
| <框架名> | | | |
| 槽名1 | 侧面名11 | 侧面值11 | … |
| | 侧面名12 | 侧面值12 | … |
| | … | | |
| 槽名2 | 侧面名21 | 侧面值21 | … |
| | 侧面名22 | 侧面值22 | … |
| | … | | |
| 槽名n | … | | |

# 框架

- 框架分为两种类型：
  - 类框架（class frame）用于描述一个概念或一类对象。
  - 实例框架（instance frame）用于描述一个具体的对象。
- 框架的层次结构：
  - 子类-subclass of->父类：类框架之间的包含关系。
  - 实例-instance of->类：实例框架和类框架的从属关系。

  下层框架可以从上层框架继承某些属性和值。后文对两者不做区分，统称为“类属”关系。

# 框架示例

**框架名：<灾难>**

时间：
地点：
伤亡：
死亡人数：
受伤人数：
失踪人数：
损失：
直接经济损失：
间接经济损失：

**框架名：<地震>**

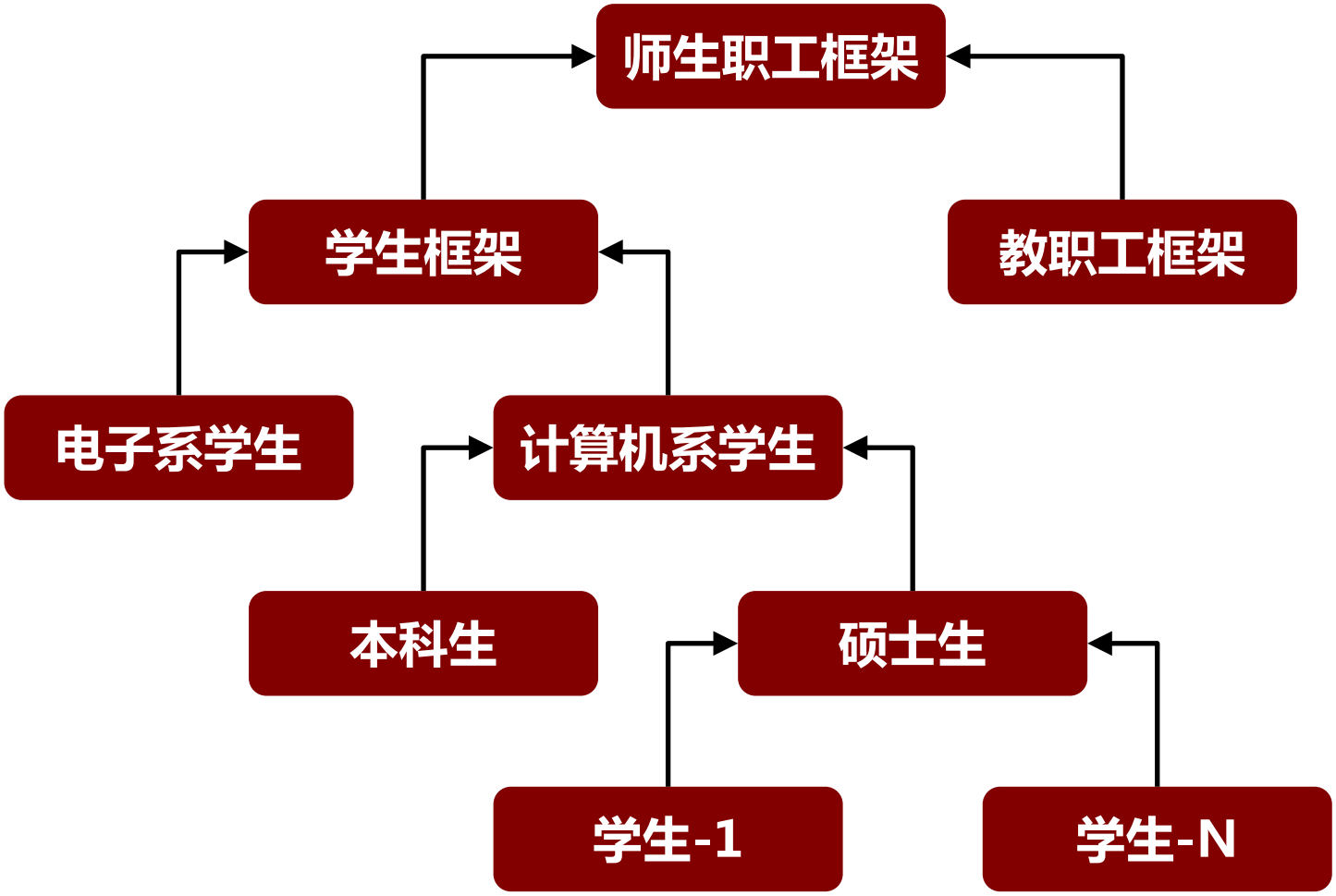
类　　属：<灾难>
震　　级：
震源深度：
断　　层：

**框架名：<汶川地震>**

类属：<地震>
时间：2008年5月12日14时28分04秒
地点：汶川
伤亡：
死亡人数：69227人
受伤人数：374643人
失踪人数：17923人
损失：
直接经济损失：8452亿元
震级：8.0级

汶川地震是中华人民共和国自建国以来影响最大的一次地震，发生于2008年5月12日14时28分04秒，里氏震级8.0级，直接严重受灾地区达10万平方公里。这次地震危害极大，共遇难69227人，受伤374643人，失踪17923人。其中四川省68712名同胞遇难，17921名同胞失踪，共有5335名学生遇难，1000多名失踪。直接经济损失8452亿元。

# 框架系统



# 框架表示法特性

- 优点：
  - 结构化：分层次嵌套式结构，既可以表示知识的内部结构，又可以表示知识之间的联系。
  - 继承性：下层框架可以从上层框架继承某些属性或值，也可以进行补充修改，减少冗余信息并节省存储空间。
  - 自然性：框架理论符合人类认知的思维过程。
  - 模块化：每个框架是相对独立的数据结构，便于知识的添加、删除和修改。
- 缺点：
  - 不能表示过程性知识。
  - 缺乏明确的推理机制。

# 代表性知识库：FrameNet[1]

- 针对词汇的概念进行框架形式的建模

## Apply_heat

Lexical Unit Index

**Definition:**

A Cook applies heat to Food, where the Temperature_setting of the heat and Duration of application may be specified. A Heating_instrument, generally indicated by a locative phrase, may also be expressed. Some cooking methods involve the use of a Medium (e.g. milk or water) by which heat is transferred to the Food. A less semantically prominent Food or Cook is marked Co-participant.

Sally FRIED an egg in butter.

Sally FRIED an egg in a teflon pan.

Ellen FRIED the eggs with chopped tomatoes and garlic.

This frame differs from Cooking_creation in focusing on the process of handling the ingredients, rather than the edible entity that results from the process.

**FEs:**

**Core:**

| | |
|---|---|
| Container [Container]<br>**Semantic Type:** Container | The Container holds the Food to which heat is applied.<br>BOIL the potatoes in a medium-sized pan.<br>Things that apply the heat directly are Heating_Instruments, e.g. crock-pot, electric skillet. |
| Cook [Cook]<br>**Semantic Type:** Sentient | The Cook applies heat to the Food.<br>Drew SAUTEED the garlic in butter. |
| Food [Food] | Food is the entity to which heat is applied by the Cook.<br>Suzy usually STEAMS the broccoli.<br>In instructional imperatives, this FE, which would be used for the (missing) object, is tagged CNI:<br>COOK on low heat for two hours.CNI |
| Heating_instrument [Heat_instr]<br>**Semantic Type:** Physical_entity | This FE identifes the entity that directly supplies heat to the Food.<br>Jim BROWNED the roast in the oven.<br>This FE will take precedence over Container when both are expressed in the same constituent. For example:<br>Kate COOKED the rice in a rice-cooker. |
| Temperature_setting [Temp]<br>**Semantic Type:** Temperature | This FE identifies the Temperature_setting of the Heating_instrument for the Food.<br>He BAKED the cookies at 350 degrees for 11 minutes.<br>She MICROWAVED the popcorn on high. |

[1]https://framenet.icsi.berkeley.edu/fndrupal/

# 代表性知识库：FrameNet[1]

- 针对词汇的概念进行框架形式的建模

**Apply_heat** 框架名

Lexical Unit Index

**Definition:**

A Cook applies heat to Food, where the Temperature_setting of the heat and Duration of application may be specified. A Heating_instrument, generally indicated by a locative phrase, may also be expressed. Some cooking methods involve the use of a Medium (e.g. milk or water) by which heat is transferred to the Food. A less semantically prominent Food or Cook is marked Co-participant.

Sally FRIED an egg in butter.

Sally FRIED an egg in a teflon pan.

Ellen FRIED the eggs with chopped tomatoes and garlic.

This frame differs from Cooking_creation in focusing on the process of handling the ingredients, rather than the edible entity that results from the process.

**FEs:** Frame elements：框架元素

**Core:**

| | |
|---|---|
| Container [Container]<br>**Semantic Type:** Container | The Container holds the Food to which heat is applied.<br>BOIL the potatoes in a medium-sized pan.<br>Things that apply the heat directly are Heating_Instruments, e.g. crock-pot, electric skillet. |
| Cook [Cook]<br>**Semantic Type:** Sentient | The Cook applies heat to the Food.<br>Drew SAUTEED the garlic in butter. |
| Food [Food] | Food is the entity to which heat is applied by the Cook.<br>Suzy usually STEAMS the broccoli.<br>In instructional imperatives, this FE, which would be used for the (missing) object, is tagged CNI:<br>COOK on low heat for two hours.CNI |
| Heating_instrument [Heat_instr]<br>**Semantic Type:** Physical_entity | This FE identifes the entity that directly supplies heat to the Food.<br>Jim BROWNED the roast in the oven.<br>This FE will take precedence over Container when both are expressed in the same constituent. For example:<br>Kate COOKED the rice in a rice-cooker. |
| Temperature_setting [Temp]<br>**Semantic Type:** Temperature | This FE identifies the Temperature_setting of the Heating_instrument for the Food.<br>He BAKED the cookies at 350 degrees for 11 minutes.<br>She MICROWAVED the popcorn on high. |

[1]https://framenet.icsi.berkeley.edu/fndrupal/

# 代表性知识库：FrameNet[1]

- 针对词汇的概念进行框架形式的建模

**Non-Core:**

| | |
|---|---|
| Beneficiary [] | THe person for whose benefit the food is cooked. |
| Co-participant [co-p] | Co-participant is the gramatically less prominent of two foods to which heat is applied by the Cook.<br>Kate BAKED the apples with cinnamon.<br>Joe ROASTED potatoes with the chicken. |
| Degree [Degr]<br>**Semantic Type:** Degree | This FE identifies the Degree to which heat application occurs. |
| Duration [Dur]<br>**Semantic Type:** Duration | Duration is the amount of Time heat is applied to the Food.<br>Jim BOILED the egg for three minutes. |
| Manner [Manr]<br>**Semantic Type:** Manner | Any description of the cooking event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). In addition, it may indicate salient characteristics of an Cook that also affect the action (presumptuously, coldly, deliberately, eagerly, carefully).<br>SAUTE the onions gently. |
| Means [Mns]<br>**Semantic Type:** State_of_affairs | This FE identifies the Means by which heat is applied to Food.<br>Kate BLANCHED the broccoli by throwing it into a pot of boiling water. |
| Medium [Med] | Medium is the substance through which heat is applied to the Food.<br>Bobby FRIED the potatoes in oil. |
| Place [Place]<br>**Semantic Type:** Locative_relation | This FE identifies the Place where the heat application occurs.<br>We BAKED the cake in the kitchen. |
| Purpose [Purp]<br>**Semantic Type:** State_of_affairs | This FE identifies the Purpose for which an intentional act is performed.<br>John BAKED cookies to show Patti that he was sorry. |
| Time [Time]<br>**Semantic Type:** Time | This FE identifies the Time at which the heat application occurs.<br>Jim ROASTED the chestnuts in the evening. |

[1]https://framenet.icsi.berkeley.edu/fndrupal/

# 代表性知识库：FrameNet[1]

- 针对词汇的概念进行框架形式的建模

**FE Core set(s):**

{Container, Heating_instrument}

**Frame-frame Relations:** 框架间关系

Inherits from: Activity, Intentionally_affect
Is Inherited by:
Perspective on:
Is Perspectivized in:
Uses:
Is Used by: Cooking_creation
Subframe of:
Has Subframe(s):
Precedes:
Is Preceded by:
Is Inchoative of:
Is Causative of: Absorb_heat
See also: Cooking_creation

**Lexical Units:** 词法单元，表示最能触发该框架的词

*bake.v, baking.n, barbecue.v, blanch.v, boil.v, boiling.n, braise.v, broil.v, broiling.n, brown.v, char.v, coddle.v, cook.v, cooking.n, deep fry.v, fry.v, frying.n, grill.v, grilling.n, melt.v, melting.n, microwave.v, parboil.v, plank.v, poach.v, roast.v, roasting.n, saute.v, scald.v, scorch.v, sear.v, simmer.v, simmering.n, singe.v, steam.v, steaming.n, steep.v, stew.v, stewing.n, toast.v, toasting.n*

[1]https://framenet.icsi.berkeley.edu/fndrupal/

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

脚本是一种与框架类似的知识表示方法，由一组槽组成，用来表示特定领域内一些事件的发生序列，类似于电影剧本。脚本表示的知识有明确的时间或因果顺序，必须是前一个动作完成后才会触发下一个动作。与框架相比，脚本用来描述一个过程而非静态知识。

# 脚本

- Winston
  一个脚本（script）是一个事件序列，包含了紧密相关的动作及改变状态的框架。
- Luger-Stubblfield
  一个脚本是一个描述特定上下文中的原型事件序列（stereotyped event sequence）的结构化表示。

# 脚本组成

- 进入条件：给出脚本中所描述事件的前提条件。
- 角色：用来描述事件中可能出现的人物。
- 道具：用来描述事件中可能出现的相关物体。
- 场景：用来描述事件发生的真实顺序。一个事件可以由多个场景组成，而每个场景又可以是其他事件的脚本。
- 结果：给出在脚本所描述事件发生以后所产生的结果。

# 脚本示例

## 例：用脚本表示去餐厅吃饭

**(1) 进入条件：**① 顾客饿了，需要进餐；② 顾客有足够的钱。

**(2) 角色**：顾客，服务员，厨师，老板。

**(3) 道具**：食品，桌子，菜单，账单，钱。

**(4) 场景**：

场景1：进入—— ① 顾客进入餐厅；② 寻找桌子；③ 在桌子旁坐下。

场景2：点菜—— ① 服务员给顾客菜单；② 顾客点菜；
③ 顾客把菜单还给服务员；④ 顾客等待服务员送菜。

场景3：等待—— ① 服务员告诉厨师顾客所点的菜；②厨师做菜，顾客等待。

场景4：吃饭—— ① 厨师把做好的送给服务员；② 服务员把菜送给顾客；
③ 顾客吃菜。

场景5：离开—— ① 服务员拿来账单；② 顾客付钱给服务员；
③ 顾客离开餐厅。

**(5) 结果：**① 顾客吃了饭，不饿了；② 顾客花了钱；③ 老板赚了钱；④ 餐厅食品少了。

# 脚本表示法特性

- 缺点：
  - 相较于框架表示，脚本表示表达能力更受约束，表示范围更窄，不具备对于对象基本属性的描述能力，也难以描述复杂事件发展的可能方向。
- 优点：
  - 在非常狭小的领域内，脚本表示却可以更细致地刻画步骤和时序关系，适合于表达预先构思好的特定知识或顺序性动作及事件，如故事情节理解、智能对话系统、机票酒店预订等。

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

语义网的概念来源于万维网（world wide web），是万维网的变革与延伸，是Web of documents向Web of data的转变，其目标是让机器或设备能够自动识别和理解万维网上的内容，使得高效的信息共享和机器智能协同成为可能。

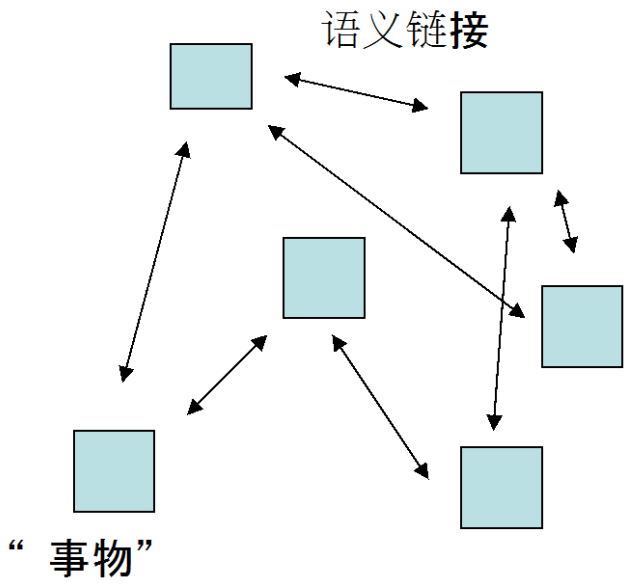
语义网（semantic web）≠ 语义网络（semantic network）

# 语义网

- 起源：传统万维网中HTML所表达的页面信息和组织方法，主要面向用户直接阅读，没有将信息的表现形式、内在结构和表达内容分离，没有提供机器可读的语义信息，因而非常不利于机器的理解和处理。
- 契机：XML的出现将数据的内容与布局区分开来，为语义更丰富、更自然的Web内容表达打开了新局面。
- 目标：为Web信息提供机器可以理解的语义，实现Web信息资源在语义层上的全方位互联，从而帮助机器对Web上异构、分布式信息进行有效检索和访问。

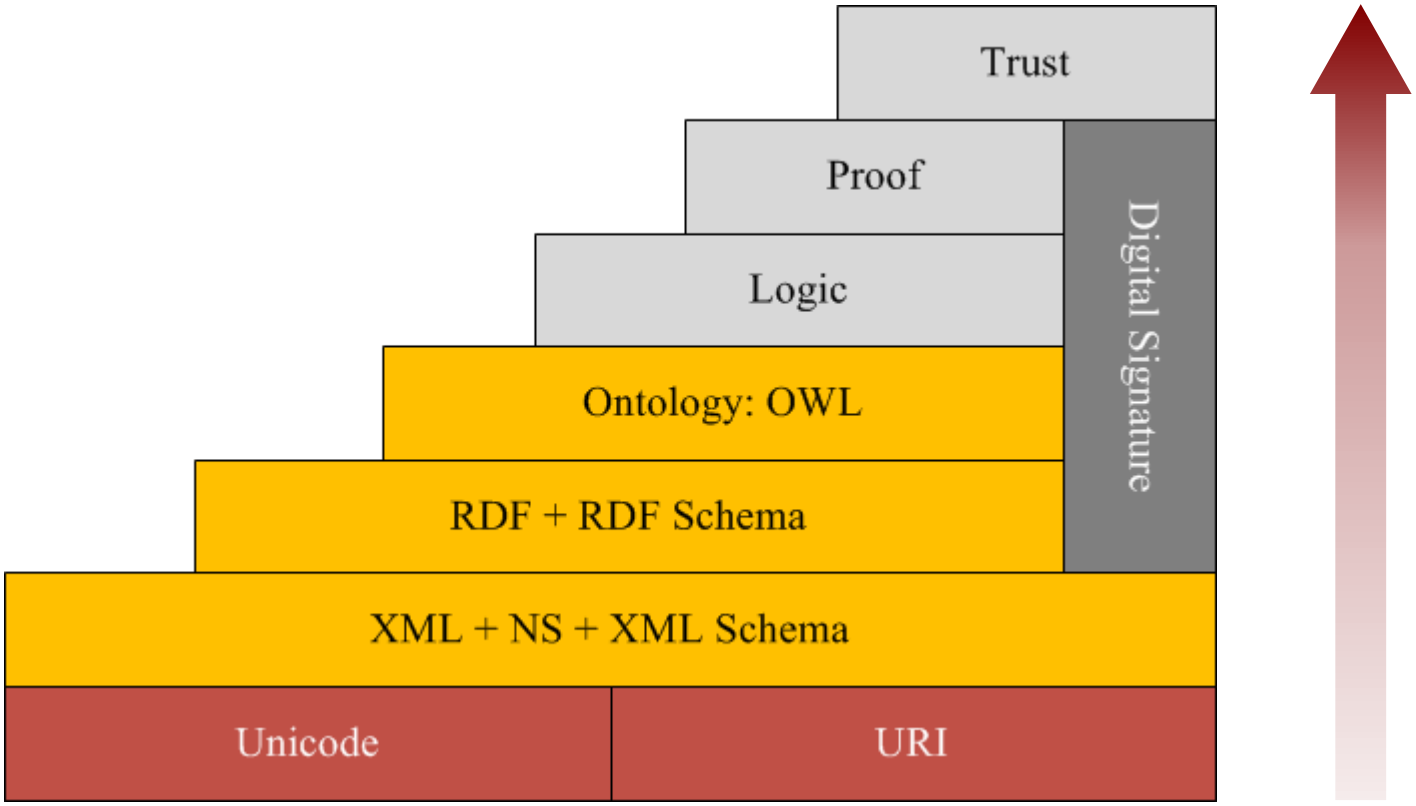
# 语义网

- 本质：以Web数据的内容（即语义）为核心，用机器能够理解和处理的方式链接起来的海量分布式数据库。
- 特征：
  - Web上的事物拥有唯一的URI
  - 事物之间由链接关联(如人物、地点、事件、建筑物)
  - 事物之间链接显式存在并拥有不同类型
  - Web上事物的结构显式存在



# 语义网体系结构

- 语义网提供了一套为描述数据而设计的表示语言和工具，用于形式化的描述一个知识领域内的概念、术语和关系。



# 语义网体系结构

- 第一层：Unicode和URI (uniform resource identifier，通用资源标识符)，是整个语义网的基础。Unicode处理资源的编码，实现网上信息的统一编码；URI 负责标识资源，支持网上对象和资源的精细标识。
- 第二层：XML+NS+XML Schema，用于表示数据的内容和结构。通过XML标记语言将网上资源信息的结构、内容和数据的表现形式进行分离。
- 第三层：RDF+RDF Schema，用于描述网上资源及其类型，为网上资源描述提供一种通用框架和实现数据集成的元数据解决方案。

# 语义网体系结构

- 第四层：Ontology Vocabulary，用于描述各种资源之间的联系，揭示资源本身及资源之间更为复杂和丰富的语义联系，明确定义描述属性或类的术语语义及术语间关系。
- 第五层：逻辑层，主要提供公理和推理规则，为智能推理提供基础。该层用来产生规则。
- 第六层：证明层，执行逻辑层产生的规则，并结合信任层的应用机制来评判是否能够信赖给定的证明。
- 第七层：信任层，注重于提供信任机制，以保证用户代理在网上进行个性化服务和彼此间交互合作时更安全可靠。

核心层为**XML**、**RDF**、**ONTOLOGY**，用于表示信息的语义。

# XML

- XML（eXtensible Markup Language，可扩展标记语言）是最早的语义网表示语言，它取消了HTML的显示样式和布局描述能力，突出了数据的语义和元素结构描述能力。

**HTML**

```
<html>
  <body>
    <h2>Explorer’s Guide to Semantic Web</h2>
    <i>by <b>Thomas B. Passin</b></i>
    <br>Springer 2004</br>
    <br>ISBN 1-932394-20-6</br>
  </body>
</html>
```

用于显示数据，侧重于如何表现信息

**XML**

```
<book>
  <title>
    Explorer’s Guide to Semantic Web
  </title>
  <author>Thomas B. Passin</author>
  <publisher>Springer</publisher>
  <year>2004</year>
  <ISBN>1-932394-20-6</ISBN>
</book>
```

用于存储和传输数据，侧重于如何结构化地描述信息

# XML：元素

- XML的元素代表XML文档所描述的“事件”，比如书籍、作者和出版商。
- 一个元素由起始标签、元素内容和结尾标签构成。

```
<author>Thomas B. Passin</author>
```

用户可以随意地选择标签名，只有很少的限制。

- 元素具有嵌套结构，并且没有约束嵌套的深度。

```
<author>
    <name>Thomas B. Passin</name>
    <gender>Male</gender>
    <phone>+61-7-3875 507</phone>
</author>
```

# XML：属性

- 与HTML类似，XML也可拥有属性，即元素名称-值对，可以表达与元素相同的语义。

```
<author  name="Thomas B. Passin"  gender="Male"
            phone="+61-7-3875 507"/>
```

- 属性也可以与元素混合使用，但是不能嵌套。

```
<author  name="Thomas B. Passin"  gender="Male" >
    <phone>+61-7-3875 507</phone>
</author>
```

# XML特性

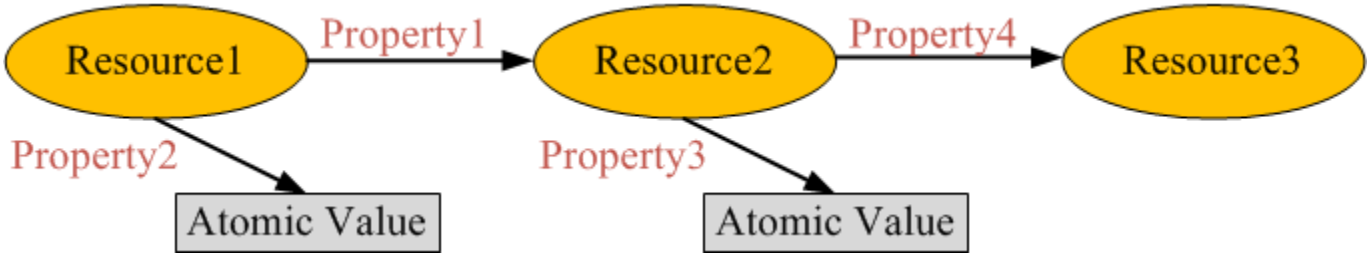
- 优点：
  - 结构化的数据表示方式，使得数据内容与其形式分离。
  - 良好的可扩展性，使用者可创建和使用自己的标记，可定义行业领域特殊的标记语言，进行数据共享和交换。
  - 包含文档类型声明，其数据可被任何XML解析器提取、分析、处理，可以轻松地跨平台应用。
- 缺点：
  - XML是一种元标记语言，任何组织或个人都可以用它来定义新的标记和标准，容易产生冲突和混乱。
  - XML文档作为数据集合使用时，相当于一个数据库，确不具备数据库管理系统那样完备的功能。
  - 数据是以树状结构存储的，插入和修改比较困难。

# RDF

- RDF（Resource Description Framework，资源描述框架）是一种资源描述语言，利用当前的多种元数据标准来描述各种网络资源，形成人机可读，并可由机器自动处理的文件。
- RDF的核心思想：
  利用Web标识符（URI）来标识事物，通过指定的属性和相应的值描述资源的性质或资源之间的关系。

# RDF数据模型

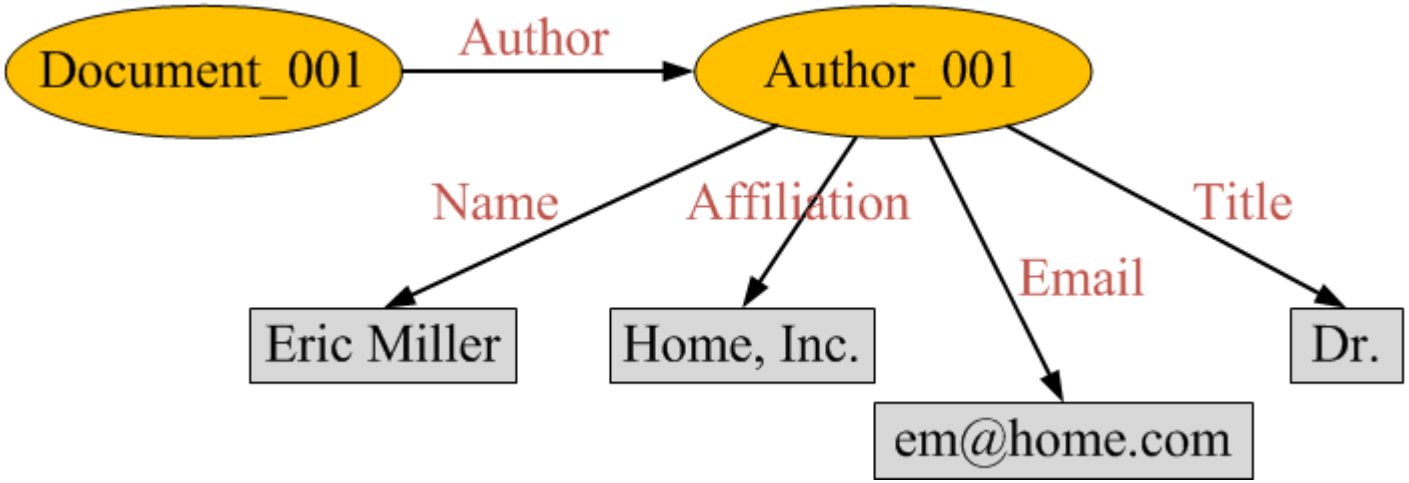
- RDF的基本数据模型包括资源、属性和陈述。
  - 资源（resource）：一切能够以RDF描述的对象都叫资源，用唯一的URI来表示。
  - 属性（property）：用来描述资源的特征或资源之间的关系，每一个属性都有特定的意义。
  - 陈述（statement）：特定的资源加上一个属性和相应的属性值就是一个陈述，其中资源是主体（subject），属性是谓词（predicate），属性值是客体（object）。



# RDF描述示例

考查以下陈述的RDF图表示

这篇文章（Document_001）的作者（Author_001）名为Eric Miller，他的工作单位是Home, Inc.，他的电子邮件地址是em@home.com，他的头衔是Dr.



# RDF Schema

- RDFS是RDF的扩展，它在RDF的基础上提供了一组建模原语，用来描述类、属性以及它们之间的关系。
  - Class, subClassOf：描述类别层次结构。
  - Property, subPropertyOf：描述属性层次结构。
  - domain, range：声明属性所应用的资源类和属性值类。
  - type：声明一个资源是一个类的实例。

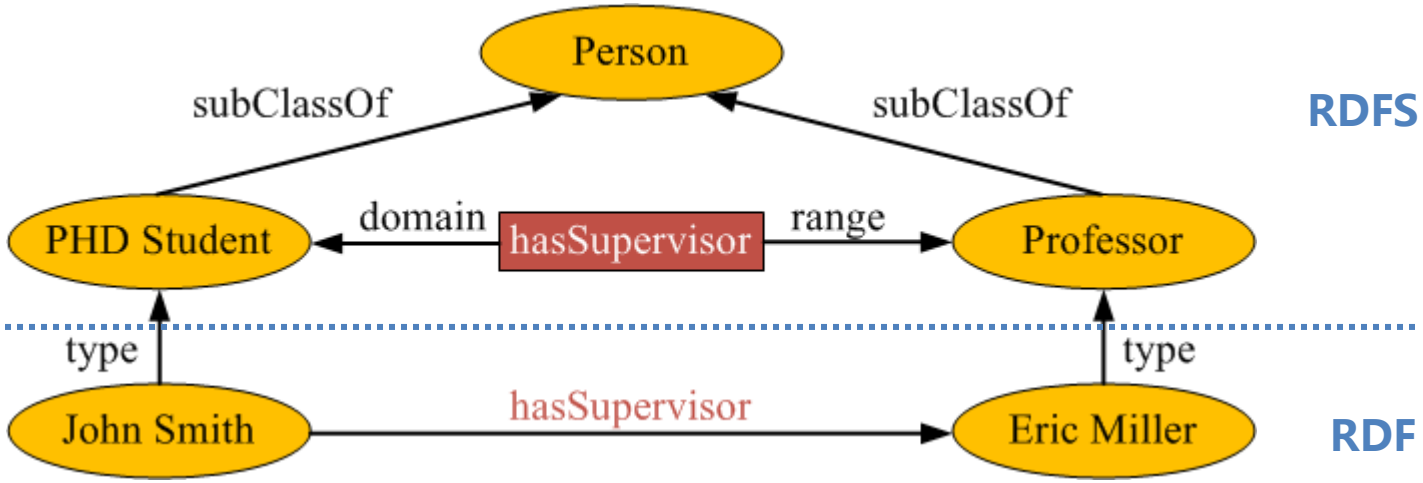
# RDF Schema

- RDFS是RDF的扩展，它在RDF的基础上提供了一组建模原语，用来描述类、属性以及它们之间的关系。
  - Class, subClassOf：描述类别层次结构。
  - Property, subPropertyOf：描述属性层次结构。
  - domain, range：声明属性所应用的资源类和属性值类。
  - type：声明一个资源是一个类的实例。

RDF

# RDF Schema

- RDFS是RDF的扩展，它在RDF的基础上提供了一组建模原语，用来描述类、属性以及它们之间的关系。
  - Class, subClassOf：描述类别层次结构。
  - Property, subPropertyOf：描述属性层次结构。
  - domain, range：声明属性所应用的资源类和属性值类。
  - type：声明一个资源是一个类的实例。



# RDF Schema示例

```
<rdf:Description ID="MotorVehicle">
   <rdf:type resource="http://www.w3.org/…#Class"/>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/…#Resource"/>
</rdf:Description>

<rdf:Description ID="Truck">
   <rdf:type resource="http://www.w3.org/…#Class"/>
   <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="registeredTo">
   <rdf:type resource="http://www.w3.org/…#Property"/>
   <rdfs:domain rdf:resource="#MotorVehicle"/>
   <rdfs:range rdf:resource="#Person"/>
</rdf:Description>

<rdf:Description ID="ownedBy">
   <rdf:type resource="http://www.w3.org/…#Property"/>
   <rdfs:subPropertyOf rdf:resource="#registeredTo"/>
</rdf:Description>
```

# RDFS词汇表

- RDFS允许定义自己的词汇表（vocabulary）：
  - 类别集合和属性集合
  - 与其它词汇表中词汇的关系
- 词汇表示例：
  - Dublin core terms: creator, date, …
  - FOAF terms: characterization of persons
  - Good relations: e-Commerce terms
  - Creative commons: copyright classes, license relations, …
  - schema.org terms: events, organizations, places, reviews, …

# RDF(S)特性

- 优点：
  - 简单：资源以三元组的形式描述，简单、易控制。
  - 易扩展：描述和词汇集分开，具备良好的可扩展性。
  - 包容性：允许定义自己的词汇集，并可以无缝使用多种词汇集来描述资源。
  - 易综合：RDF认为一切都是资源，这样很容易综合描述。
- 缺点：
  - 不能准确描述语义：同一个概念有多种词汇表示，同一个词汇有多种含义（概念）。
  - 没有推理模型，不具备推理能力。

# RDF(S)特性

- 优点：
  - 简单：资源以三元组的形式描述，简单、易控制。
  - 易扩展：描述和词汇集分开，具备良好的可扩展性。
  - 包容性：允许定义自己的词汇集，并可以无缝使用多种词汇集来描述资源。
  - 易综合：RDF认为一切都是资源，这样很容易综合描述。
- 缺点：
  - 不能准确描述语义：同一个概念有多种词汇表示，同一个词汇有多种含义（概念）。
  - 没有推理模型，不具备推理能力。

Web Ontology Language (OWL)

# 为什么需要Ontology？

- Ontology（本体）通过对概念的严格定义和概念与概念之间的关系来确定概念的精确含义，表示共同认可的、可共享的知识。
- 对于ontology来说，author，creator和writer是同一个概念，而doctor在大学和医院分别表示的是两个概念。因此在语义网中，ontology具有非常重要的地位，是解决语义层次上Web信息共享和交换的基础。

# 本体的定义

- 哲学界：对世界上客观存在物的系统地描述，即存在论。
- 知识工程界：
  - 1991/Neches：本体定义了主题领域的词汇的基本术语和关系，以及利用这些术语和关系构成的规定这些词汇外延的规则。
  - 1993/Gruber：本体是概念模型的明确的规范说明。
  - 1997/Borst：本体是共享概念模型的形式化规范说明。
  - 1998/Studer：本体是共享概念模型的明确的形式化规范说明。

# 本体的定义

Studer：本体是共享概念模型的明确的形式化规范说明。

An ontology is a **formal**, **explicit** specification of a **shared conceptualization**.

- 概念模型 (conceptualization)：本体是通过抽象客观世界的概念而得到的模型，其表示的含义独立于具体的环境状态。
- 明确性 (explicit)：本体所使用的概念及使用这些概念的约束都有明确的定义，没有二义性。
- 形式化 (formal)：本体是计算机可处理的，而非自然语言。
- 共享 (shared)：本体体现的是共同认可的知识，反映的是相关领域中公认的概念集合，它所针对的是团体而非个体。

# 本体的组成：$O=\{C,R,F,A,I\}$

- 概念 (concept) 或类 (class)：可以指任何事物，从语义上讲，它表示的是对象的集合。
  - 学生、教授、演员、歌手
- 关系 (relation)：描述概念之间的语义关联。
  - part-of、kind-of、instance-of、attribute-of
- 函数 (function)：一类特殊的关系，该关系中的第n个元素可由前n-1个元素唯一决定。
  - father-of(x, y)表示y是x的父亲
- 公理 (axiom)：代表永真断言。
  - 如果A是B的子女，B是C的子女，则A是C的子孙
- 实例 (instance)：某类概念所指的具体个体，即对象。
  - John Smith是概念学生的实例

# OWL

- OWL (Web Ontology Language，Web本体语言)是在语义网上表示本体的推荐语言，作为RDF(S)的扩展，其目的是提供更多原语以支持更加丰富的语义表达并支持推理。
- OWL的三个子语言：
  - OWL Lite：提供一个分类层次和简单属性约束。
  - OWL DL：提供推理系统，保证计算完备性和可判定性。
  - OWL Full：支持完全自由的RDF语法，但是不具备可计算性保证。

表达能力：OWL Lite < OWL DL < OWL Full

# OWL建模原语

- 类运算式 (class descriptions)：
  - 枚举
    owl:oneOf
  - 属性值约束
    owl:allValuesFrom, owl:someValuesFrom, owl:hasValue
  - 属性基数约束
    owl:maxCadinality, owl:minCadinality, owl:cadinality
  - 交集、并集、补集
    owl:intersectionOf, owl:unionOf, owl:complementOf
- 类公理 (class axioms)：
  - 子类、等价类、不相交类
    rdfs:subClassOf, owl:equivalentClass, owl:disjointWith

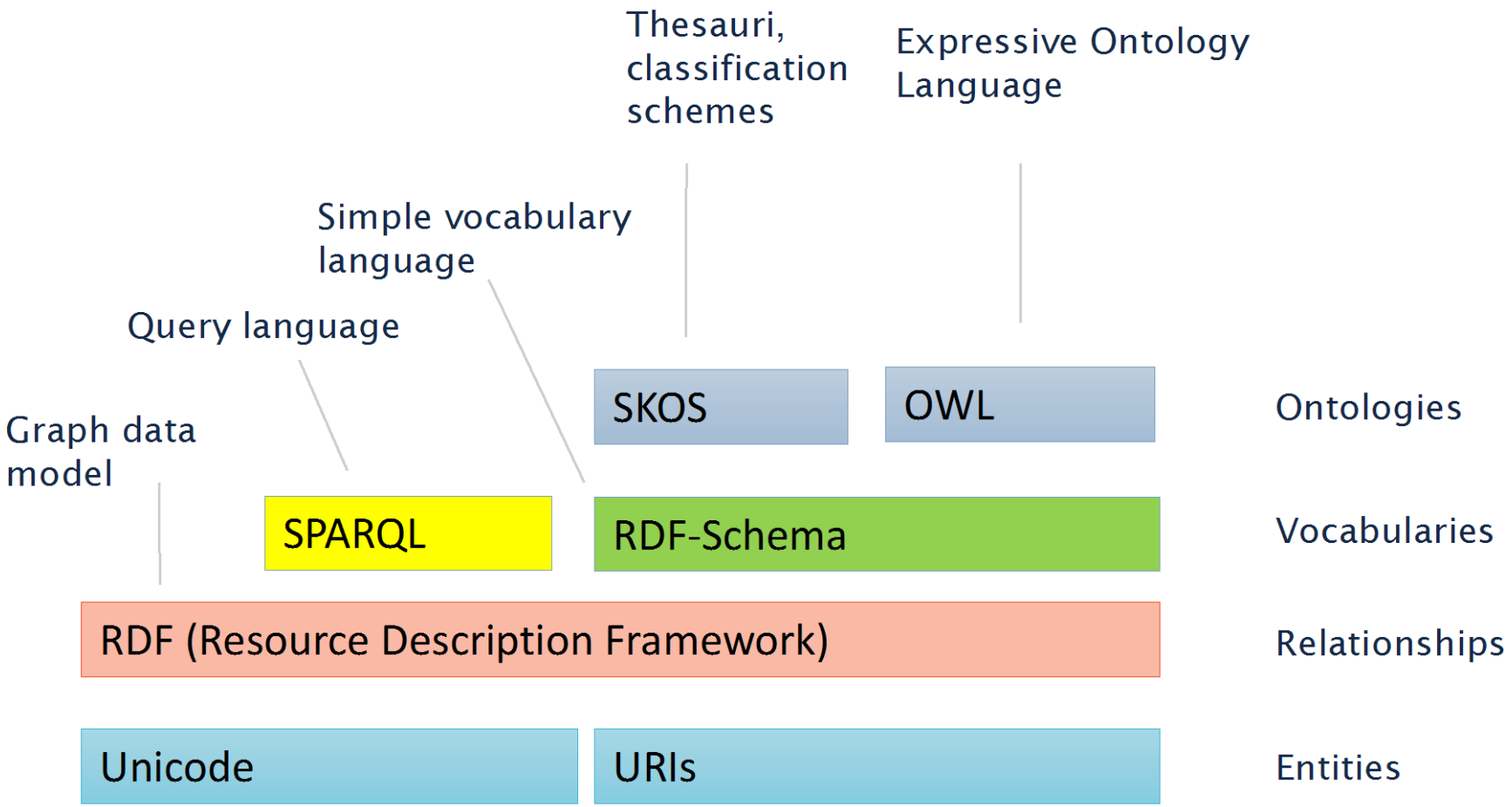
# OWL建模原语

- 属性公理：
  - RDFS属性结构
    rdfs:subPropertyOf, rdfs:domain, rdfs: range
  - 与其他属性的关系
    owl:equivalentProperty, owl:inverseOf
  - 关于属性的全局基数约束
    owl:FunctionalProperty, owl:InverseFunctionalProperty
  - 属性的逻辑特征
    owl:TransitiveProperty, owl:SymmetricProperty
- 个体公理：
  - 个体同一性
    owl:sameAs, owl:differentFrom, owl:allDifferent

# 语义网知识描述语言体系

- **XML** (http://www.w3.org/XML/) 提供了一种结构化文档的表层语法，但没有对文档含义施加任何语义约束 。
- **RDF** (http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/) 是一个关于对象（资源）和它们之间关系的数据模型，该模型具备简单语义，能够用XML语法表示。
- **RDF Schema** (http://www.w3.org/TR/2002/WD-rdf-schema-20021112/) 是一组描述RDF资源的类和属性的建模原语，提供了关于这些类和属性的层次结构的语义。
- **OWL** (http://www.w3.org/TR/2004/REC-owl-ref-20040210/) 添加了更多用于描述类和属性的建模原语，支持更加丰富的语义表达并支持推理。

# 语义网知识描述语言体系

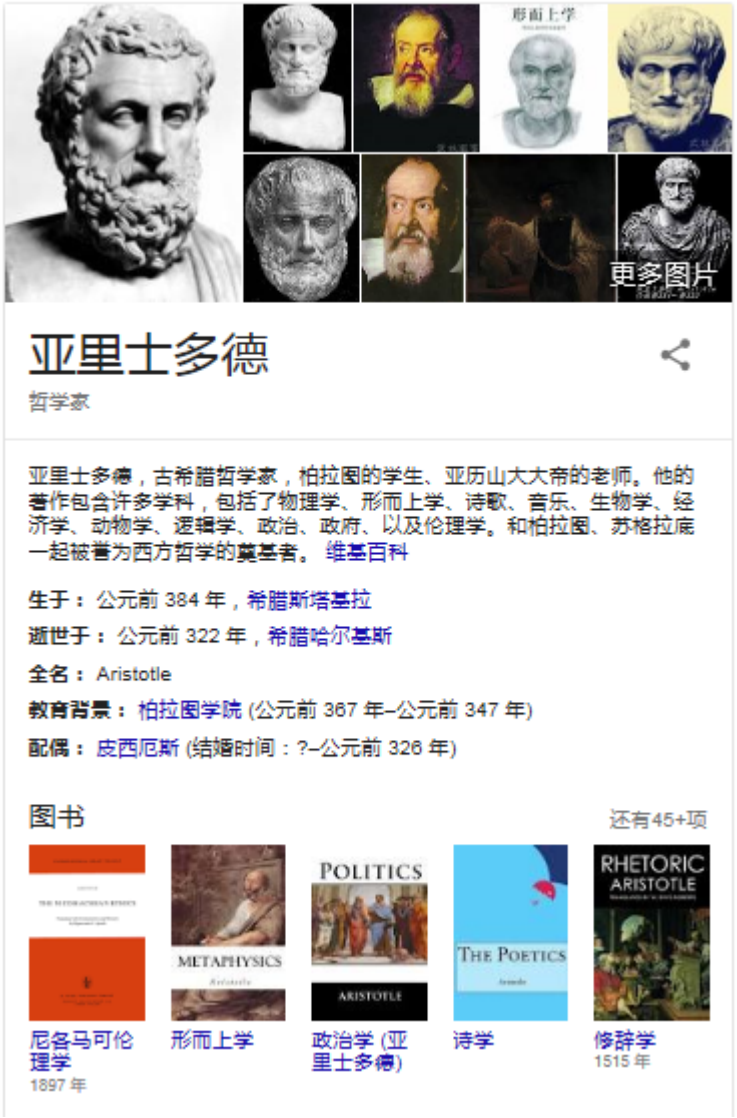


# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

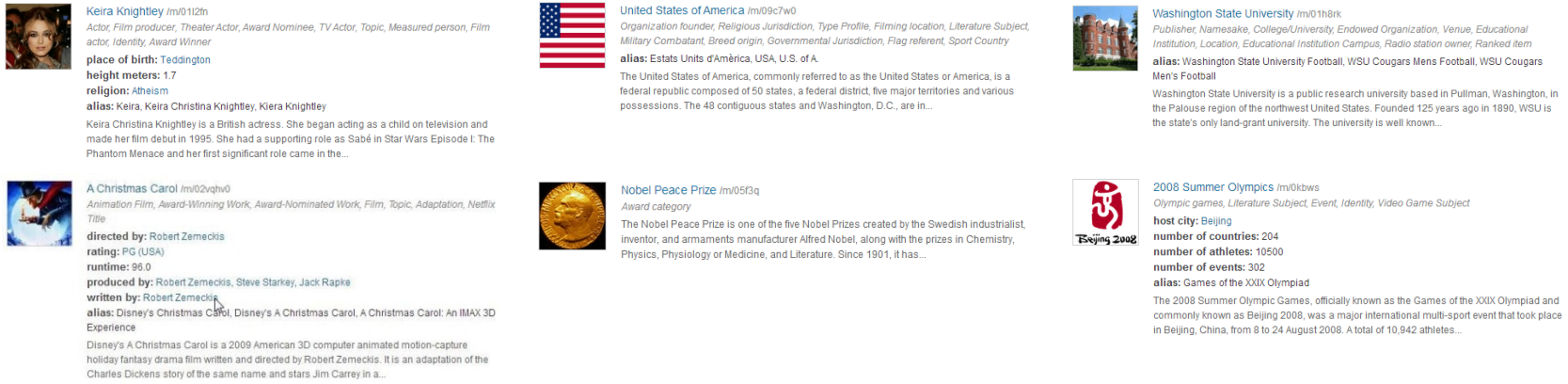
# 知识图谱概念起源

- 知识图谱的概念最早出现于Google公司的知识图谱项目，体现在使用Google搜索引擎时，出现于搜索结果右侧的相关知识展示。
- 截止到2016年底，Google知识图谱中的知识数量已经达到了600亿条，关于1500个类别的5.7亿个实体，以及它们之间的3.5万种关系。



# 实体、关系和事实

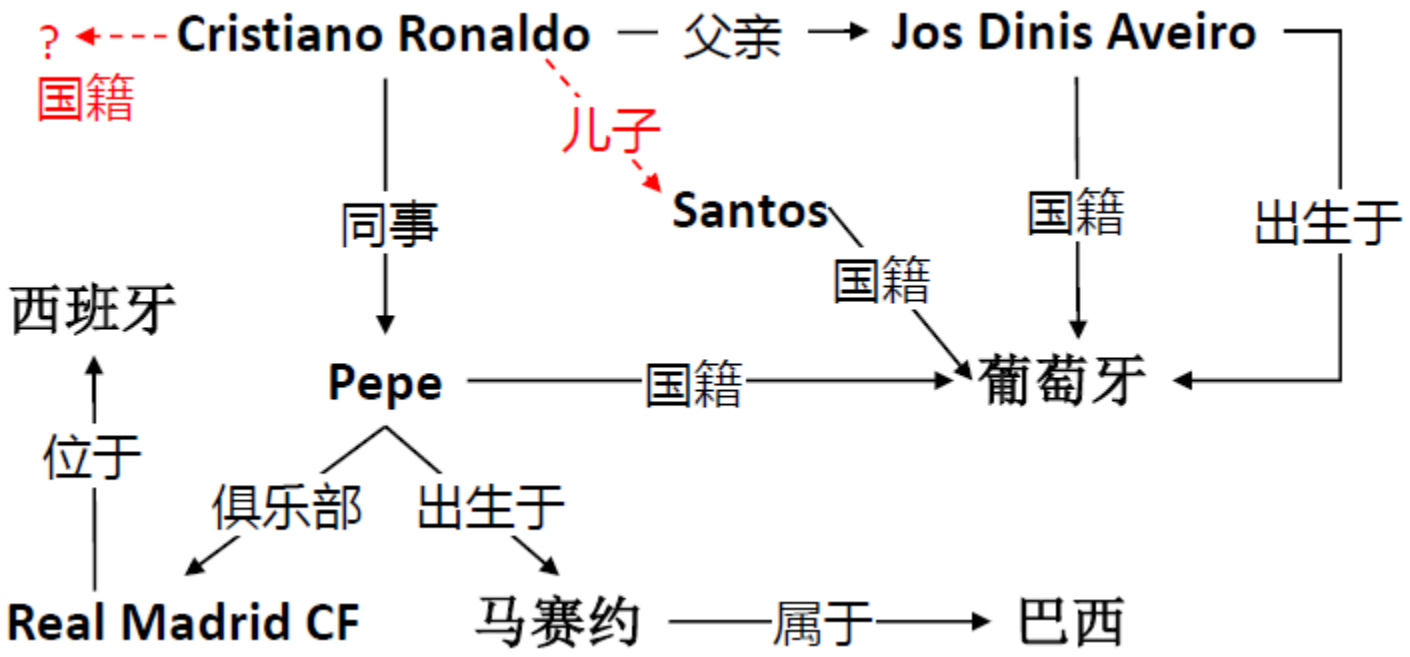
- 实体 (entity)：现实世界中可区分、可识别的事物或概念。
  - 客观对象：人物、地点、机构
  - 抽象事件：电影、奖项、赛事




- 关系 (relation)：实体和实体之间的语义关联。
  - BornInCity, IsParentOf, AthletePlaysForTeam
- 事实 (fact)：陈述两个实体之间关系的断言，通常表示为 (head entity, relation, tail entity) 三元组形式。

# 狭义知识图谱

- 狭义知识图谱：具有图结构的三元组知识库。
  - 知识库中的实体作为知识图谱中的节点。
  - 知识库中的事实作为知识图谱中的边，边的方向由头实体指向尾实体，边的类型就是两实体间关系类型。



# 知识图谱特性

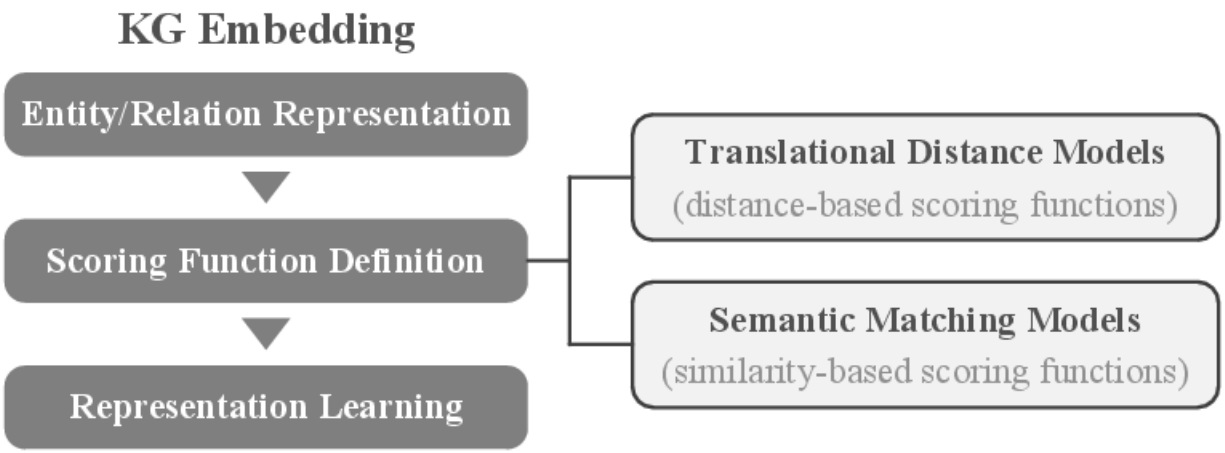
- 知识图谱不太专注于对知识框架的定义，而专注于如何以工程的方式，从文本中自动抽取或依靠众包的方式获取并组建广泛的、具有平铺结构的知识实例，最后再要求使用它的方式具有容错、模糊匹配等机制。
- 知识图谱的真正魅力在于其图结构，可以在知识图谱上运行搜索、随机游走、网络流等大规模图算法，使知识图谱与图论、概率图等碰撞出火花。

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

# 分布式知识表示[1]

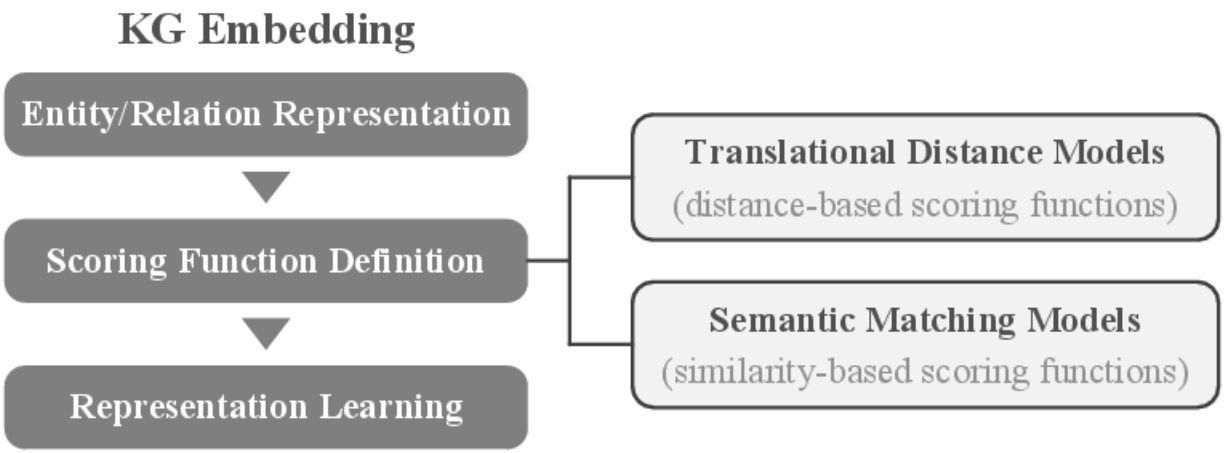
- 核心思想：将符号化的实体和关系在低维连续向量空间进行表示，在简化计算的同时最大程度保留原始的图结构。
  - 将实体和关系在向量空间进行表示（向量/矩阵/张量）。
  - 定义打分函数，衡量每个三元组成立的可能性。
  - 构造优化问题，学习实体和关系的低维连续向量表示。




[1]Wang et al. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE TKDE, to appear, 2017.
http://ieeexplore.ieee.org/document/8047276/

# 分布式知识表示[1]

- 方法类型：
  - 位移距离模型 (translational distance models)：采用基于距离的打分函数来衡量三元组成立的可能性。
  - 语义匹配模型 (semantic matching models)：采用基于相似度的打分函数来衡量三元组成立的可能性。




[1]Wang et al. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE TKDE, to appear, 2017.
http://ieeexplore.ieee.org/document/8047276/

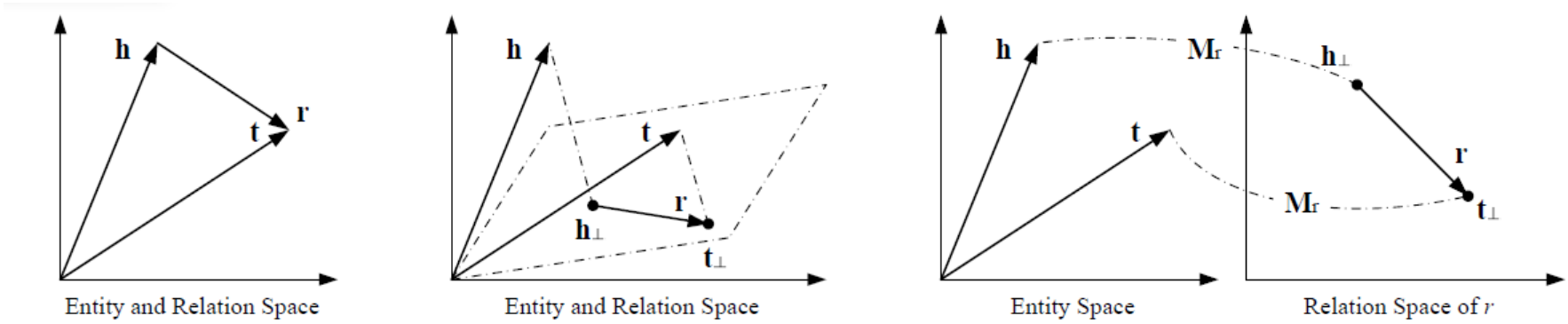
# 位移距离模型

- 代表性方法：TransE及其变种
  - head entity + relation = tail entity

    China – Beijing = France – Paris = capital-of

    Beijing + capital-of = China

    Paris + capital-of = France




(a) TransE. (b) TransH. (c) TransR.

$$f_r(h,t) = -\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}.$$

$$\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r, \quad \mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r.$$

$$f_r(h,t) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2,$$

$$\mathbf{h}_\perp = \mathbf{M}_r^1\mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r^2\mathbf{t}.$$

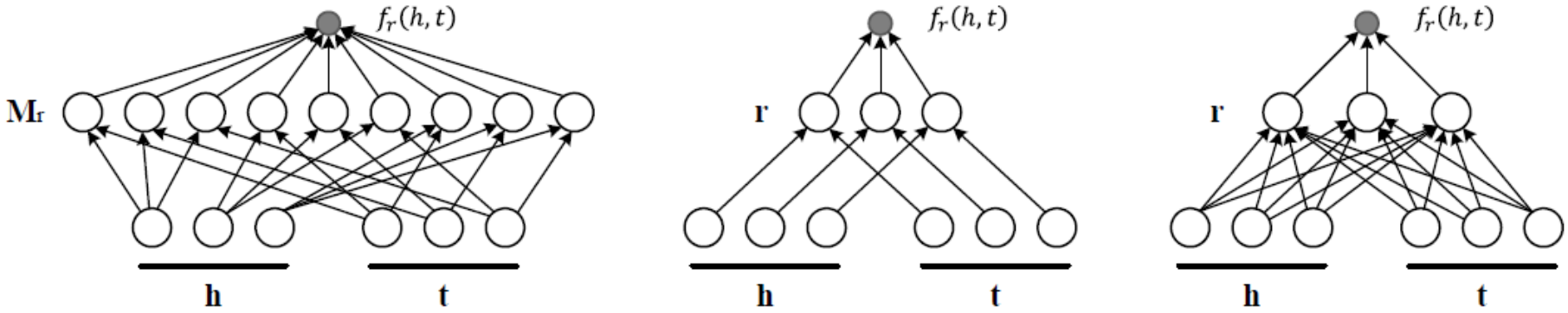
$$f_r(h,t) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2,$$

# 位移距离模型

| Method | Ent. embedding | Rel. embedding | Scoring function $f_r(h,t)$ | Constraints/Regularization |
|---|---|---|---|---|
| TransE [14] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$ | $\lVert\mathbf{h}\rVert_2=1,\lVert\mathbf{t}\rVert_2=1$ |
| TransH [15] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$ | $-\lVert(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1$<br>$\lvert\mathbf{w}_r^\top\mathbf{r}\rvert/\lVert\mathbf{r}\rVert_2\le\epsilon,\lVert\mathbf{w}_r\rVert_2=1$ |
| TransR [16] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k,\mathbf{M}_r\in\mathbb{R}^{k\times d}$ | $-\lVert\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r\mathbf{h}\rVert_2\le 1,\lVert\mathbf{M}_r\mathbf{t}\rVert_2\le 1$ |
| TransD [50] | $\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$<br>$\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$ | $-\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\rVert_2\le 1$<br>$\lVert(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2\le 1$ |
| TranSparse [51] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k,\mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$<br>$\mathbf{M}_r^1(\theta_r^1),\mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$ | $-\lVert\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_{1/2}^2$<br>$-\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_{1/2}^2$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r(\theta_r)\mathbf{h}\rVert_2\le 1,\lVert\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\rVert_2\le 1,\lVert\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_2\le 1$ |
| TransM [52] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-\theta_r\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$ | $\lVert\mathbf{h}\rVert_2=1,\lVert\mathbf{t}\rVert_2=1$ |
| ManifoldE [53] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-(\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_2^2-\theta_r^2)^2$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$ |
| TransF [54] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$ |
| TransA [55] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d,\mathbf{M}_r\in\mathbb{R}^{d\times d}$ | $-(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)^\top\mathbf{M}_r(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)$ | $\lVert\mathbf{h}\rVert_2\le 1,\lVert\mathbf{t}\rVert_2\le 1,\lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r\rVert_F\le 1,[\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge 0$ |
| KG2E [45] | $\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$<br>$\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$<br>$\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$<br>$\boldsymbol{\Sigma}_h,\boldsymbol{\Sigma}_t\in\mathbb{R}^{d\times d}$ | $\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$<br>$\boldsymbol{\mu}_r\in\mathbb{R}^d,\boldsymbol{\Sigma}_r\in\mathbb{R}^{d\times d}$ | $-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$<br>$-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$<br>$\boldsymbol{\mu}=\boldsymbol{\mu}_h+\boldsymbol{\mu}_r-\boldsymbol{\mu}_t$<br>$\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_t$ | $\lVert\boldsymbol{\mu}_h\rVert_2\le 1,\lVert\boldsymbol{\mu}_t\rVert_2\le 1,\lVert\boldsymbol{\mu}_r\rVert_2\le 1$<br>$c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$<br>$c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_t\le c_{max}\mathbf{I}$<br>$c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_r\le c_{max}\mathbf{I}$ |
| TransG [46] | $\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\sigma_h^2\mathbf{I})$<br>$\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma_t^2\mathbf{I})$<br>$\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ | $\boldsymbol{\mu}_r^i\sim\mathcal{N}(\boldsymbol{\mu}_t-\boldsymbol{\mu}_h,(\sigma_h^2+\sigma_t^2)\mathbf{I})$<br>$\mathbf{r}=\sum_i\pi_r^i\boldsymbol{\mu}_r^i\in\mathbb{R}^d$ | $\sum_i\pi_r^i\exp\left(-\frac{\lVert\boldsymbol{\mu}_h+\boldsymbol{\mu}_r^i-\boldsymbol{\mu}_t\rVert_2^2}{\sigma_h^2+\sigma_t^2}\right)$ | $\lVert\boldsymbol{\mu}_h\rVert_2\le 1,\lVert\boldsymbol{\mu}_t\rVert_2\le 1,\lVert\boldsymbol{\mu}_r^i\rVert_2\le 1$ |
| UM [56] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | —— | $-\lVert\mathbf{h}-\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2=1,\lVert\mathbf{t}\rVert_2=1$ |
| SE [57] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$ | $-\lVert\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\rVert_1$ | $\lVert\mathbf{h}\rVert_2=1,\lVert\mathbf{t}\rVert_2=1$ |

# 语义匹配模型

- 代表性方法：RESCAL及其变种
  - matching(relation, composition(head, tail))




(a) RESCAL. (b) DistMult. (c) HolE.

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} = \sum_{i=0}^{d-1}\sum_{j=0}^{d-1} [\mathbf{M}_r]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j,$$

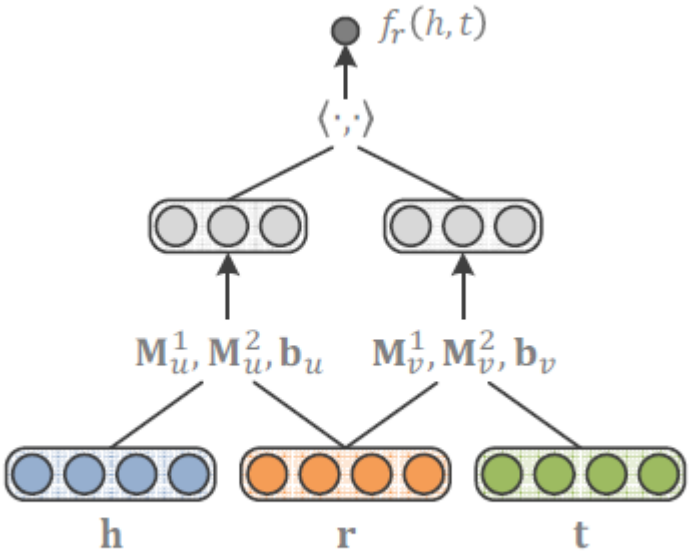
$$f_r(h,t) = \mathbf{h}^\top \mathrm{diag}(\mathbf{r})\mathbf{t} = \sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_i.$$

$$[\mathbf{h} \star \mathbf{t}]_i = \sum_{k=0}^{d-1} [\mathbf{h}]_k \cdot [\mathbf{t}]_{(k+i) \bmod d}.$$

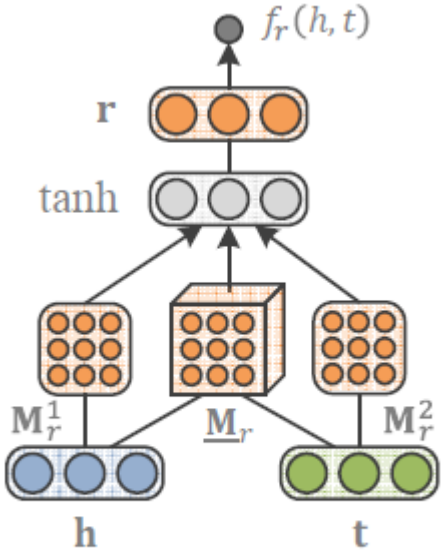
$$f_r(h,t) = \mathbf{r}^\top (\mathbf{h} \star \mathbf{t})$$

# 语义匹配模型

- 代表性方法：神经网络
  - matching via neural network architectures




(a) SME.



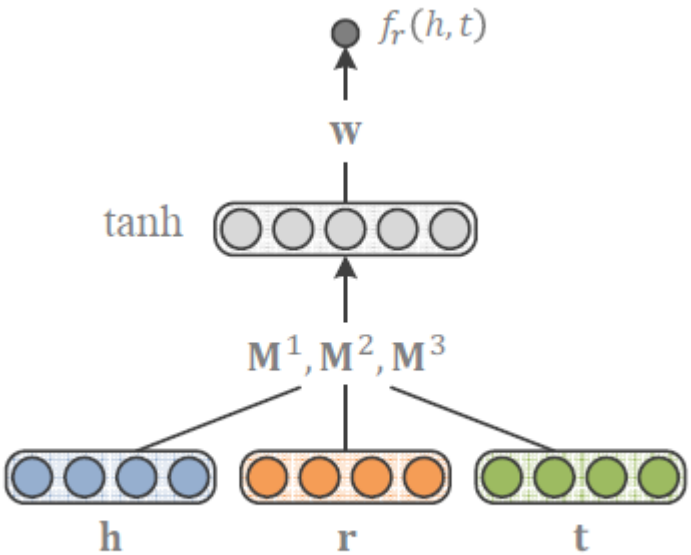

(b) NTN.

$$f_r(h,t) = g_u(\mathbf{h},\mathbf{r})^\top g_v(\mathbf{t},\mathbf{r}).$$

$$f_r(h,t) = \mathbf{r}^\top \tanh(\mathbf{h}^\top \underline{\mathbf{M}}_r \mathbf{t} + \mathbf{M}_r^1 \mathbf{h} + \mathbf{M}_r^2 \mathbf{t} + \mathbf{b}_r),$$

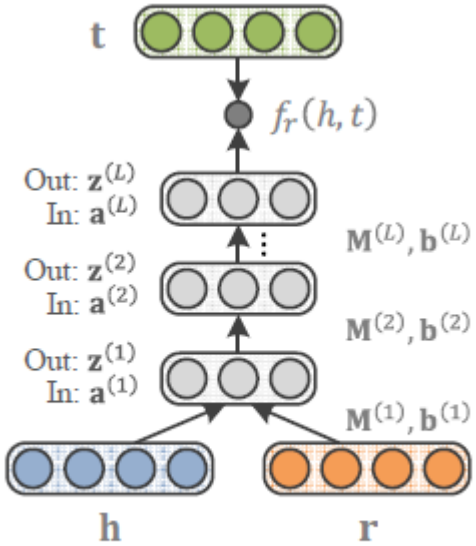
# 语义匹配模型

- 代表性方法：神经网络
  - matching via neural network architectures




(c) MLP.

$$f_r(h,t) = \mathbf{w}^\top \tanh(\mathbf{M}^1\mathbf{h} + \mathbf{M}^2\mathbf{r} + \mathbf{M}^3\mathbf{t}).$$




(d) NAM.

$$\mathbf{z}^{(0)} = [\mathbf{h}; \mathbf{r}] \in \mathbb{R}^{2d}.$$
$$\mathbf{z}^{(\ell)} = \mathrm{ReLU}(\mathbf{a}^{(\ell)}), \quad \mathbf{a}^{(\ell)} = \mathbf{M}^{(\ell)}\mathbf{z}^{(\ell-1)} + \mathbf{b}^{(\ell)}$$
$$f_r(h,t) = \mathbf{t}^\top \mathbf{z}^{(L)}.$$

# 语义匹配模型

| Method | Ent. embedding | Rel. embedding | Scoring function $f_r(h,t)$ | Constraints/Regularization |
|---|---|---|---|---|
| RESCAL [13] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{M}_r\in\mathbb{R}^{d\times d}$ | $\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{M}_r\rVert_F\le 1$<br>$\mathbf{M}_r=\sum_i \pi_r^i\mathbf{u}_i\mathbf{v}_i^\top$ (required in [17]) |
| TATEC [64] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$ | $\mathbf{h}^\top\mathbf{M}_r\mathbf{t}+\mathbf{h}^\top\mathbf{r}+\mathbf{t}^\top\mathbf{r}+\mathbf{h}^\top\mathbf{D}\mathbf{t}$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r\rVert_F\le 1$ |
| DistMult [65] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\mathbf{t}$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1, \lVert\mathbf{r}\rVert_2\le 1$ |
| HolE [62] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $\mathbf{r}^\top(\mathbf{h}\star\mathbf{t})$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ |
| ComplEx [66] | $\mathbf{h},\mathbf{t}\in\mathbb{C}^d$ | $\mathbf{r}\in\mathbb{C}^d$ | $\mathrm{Re}(\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\bar{\mathbf{t}})$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ |
| ANALOGY [68] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{M}_r\in\mathbb{R}^{d\times d}$ | $\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{M}_r\rVert_F\le 1$<br>$\mathbf{M}_r\mathbf{M}_r^\top=\mathbf{M}_r^\top\mathbf{M}_r$<br>$\mathbf{M}_r\mathbf{M}_{r'}=\mathbf{M}_{r'}\mathbf{M}_r$ |
| SME [18] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $(\mathbf{M}_u^1\mathbf{h}+\mathbf{M}_u^2\mathbf{r}+\mathbf{b}_u)^\top(\mathbf{M}_v^1\mathbf{t}+\mathbf{M}_v^2\mathbf{r}+\mathbf{b}_v)$<br>$((\mathbf{M}_u^1\mathbf{h})\circ(\mathbf{M}_u^2\mathbf{r})+\mathbf{b}_u)^\top((\mathbf{M}_v^1\mathbf{t})\circ(\mathbf{M}_v^2\mathbf{r})+\mathbf{b}_v)$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$ |
| NTN [19] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{b}_r\in\mathbb{R}^k, \underline{\mathbf{M}}_r\in\mathbb{R}^{d\times d\times k}$<br>$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$ | $\mathbf{r}^\top\tanh(\mathbf{h}^\top\underline{\mathbf{M}}_r\mathbf{t}+\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t}+\mathbf{b}_r)$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{b}_r\rVert_2\le 1, \lVert\underline{\mathbf{M}}_r^{[:,:,i]}\rVert_F\le 1$<br>$\lVert\mathbf{M}_r^1\rVert_F\le 1, \lVert\mathbf{M}_r^2\rVert_F\le 1$ |
| SLM [19] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$ | $\mathbf{r}^\top\tanh(\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t})$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$<br>$\lVert\mathbf{M}_r^1\rVert_F\le 1, \lVert\mathbf{M}_r^2\rVert_F\le 1$ |
| MLP [69] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $\mathbf{w}^\top\tanh(\mathbf{M}^1\mathbf{h}+\mathbf{M}^2\mathbf{r}+\mathbf{M}^3\mathbf{t})$ | $\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ |
| NAM [63] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $f_r(h,t)=\mathbf{t}^\top\mathbf{z}^{(L)}$<br>$\mathbf{z}^{(\ell)}=\mathrm{ReLU}(\mathbf{a}^{(\ell)}),\ \mathbf{a}^{(\ell)}=\mathbf{M}^{(\ell)}\mathbf{z}^{(\ell-1)}+\mathbf{b}^{(\ell)}$<br>$\mathbf{z}^{(0)}=[\mathbf{h};\mathbf{r}]$ | —— |

# 模型训练

- 开放世界假设 (Open World Assumption, OWA)：知识图谱仅包括正确的事实，那些不在其中出现的事实要么是错误的，要么是缺失的。
  - Logistic loss

$$\min_{\Theta} \sum_{\tau \in \mathbb{D}^+ \cup \mathbb{D}^-} \log\left(1 + \exp(-y_{hrt} \cdot f_r(h,t))\right),$$

  - Pairwise ranking loss

$$\min_{\Theta} \sum_{\tau^+ \in \mathbb{D}^+} \sum_{\tau^- \in \mathbb{D}^-} \max\left(0, \gamma - f_r(h,t) + f_{r'}(h',t')\right)$$

关键：以何种策略生成负样本。

# 模型训练

- 开放世界假设 (Open World Assumption, OWA)：知识图谱仅包括正确的事实，那些不在其中出现的事实要么是错误的，要么是缺失的。
  - Logistic loss

$$\min_{\Theta} \sum_{\tau \in \mathbb{D}^+ \cup \mathbb{D}^-} \log\left(1 + \exp(-y_{hrt} \cdot f_r(h,t))\right),$$

  - Pairwise ranking loss

$$\min_{\Theta} \sum_{\tau^+ \in \mathbb{D}^+} \sum_{\tau^- \in \mathbb{D}^-} \max\left(0, \gamma - f_r(h,t) + f_{r'}(h',t')\right)$$

关键：以何种策略生成负样本。

$$\begin{aligned}\mathbb{D}^- = &\{(h',r,t) | h' \in \mathbb{E} \wedge h' \neq h \wedge (h,r,t) \in \mathbb{D}^+\} \\ &\cup \{(h,r,t') | t' \in \mathbb{E} \wedge t' \neq t \wedge (h,r,t) \in \mathbb{D}^+\}.\end{aligned}$$

# 模型训练

- 封闭世界假设 (Closed World Assumption, CWA)：但凡未在知识图谱中出现的事实都是错误的。
  - Squared loss

$$\min_{\Theta} \sum_{h,t\in\mathbb{E},r\in\mathbb{R}} \left(y_{hrt} - f_r(h,t)\right)^2,$$

等价于分解由知识图谱表示成的三阶张量。

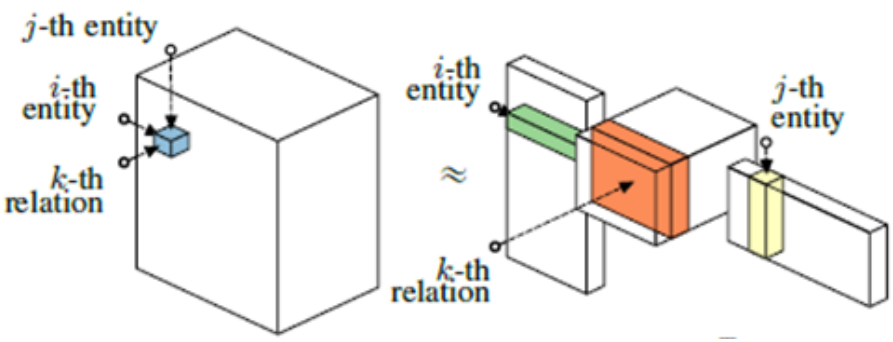
# 模型训练

- 封闭世界假设 (Closed World Assumption, CWA)：但凡未在知识图谱中出现的事实都是错误的。
  - Squared loss

$$\min_{\Theta} \sum_{h,t\in\mathbb{E},r\in\mathbb{R}} \left(y_{hrt} - f_r(h,t)\right)^2,$$

等价于分解由知识图谱表示成的三阶张量。

RESCAL




$$\min_{\mathbf{E},\{\mathbf{M}_r\}} \sum_{r\in\mathbb{R}} \|\underline{\mathbf{Y}}^{[:,:,r]} - \mathbf{E}\mathbf{M}_r\mathbf{E}^\top\|_F^2 + \lambda_1\|\mathbf{E}\|_F^2 + \lambda_2 \sum_{r\in\mathbb{R}} \|\mathbf{M}_r\|_F^2.$$

# 提纲

- 概述
- 一阶谓词逻辑表示法
- 产生式规则表示法
- 框架表示法
- 脚本表示法
- 语义网表示法
- 知识图谱中的知识表示
- 分布式知识表示
- 本章小结

# 一阶谓词逻辑表示法

- 通过个体、谓词、量词、逻辑联结词来表示事物的状态、属性等事实性知识，以及事物间因果关系的规则性知识。
  - 定义谓词及个体，确定每个谓词及个体的确切含义。
  - 依据所要表达的事物，为谓词中的变量赋以特定的值。
  - 用逻辑联结词将各个谓词连接起来形成谓词公式。
- 优点：精确性、通用性、自然性、模块化。
- 缺点：表示能力差、管理困难、效率低。

# 产生式规则表示法

- 产生式系统结构：数据库 + 规则库 + 推理机

  - 数据库：存放已知事实、推理的中间结果和最终结论。
  - 规则库：存放与求解问题有关的所有规则。
  - 推理机：控制整个系统的运行，决定问题求解的线路。
- 优点：有效性、自然性、一致性、模块化。
- 缺点：效率低、不能表示结构性或层次性知识。

# 框架表示法

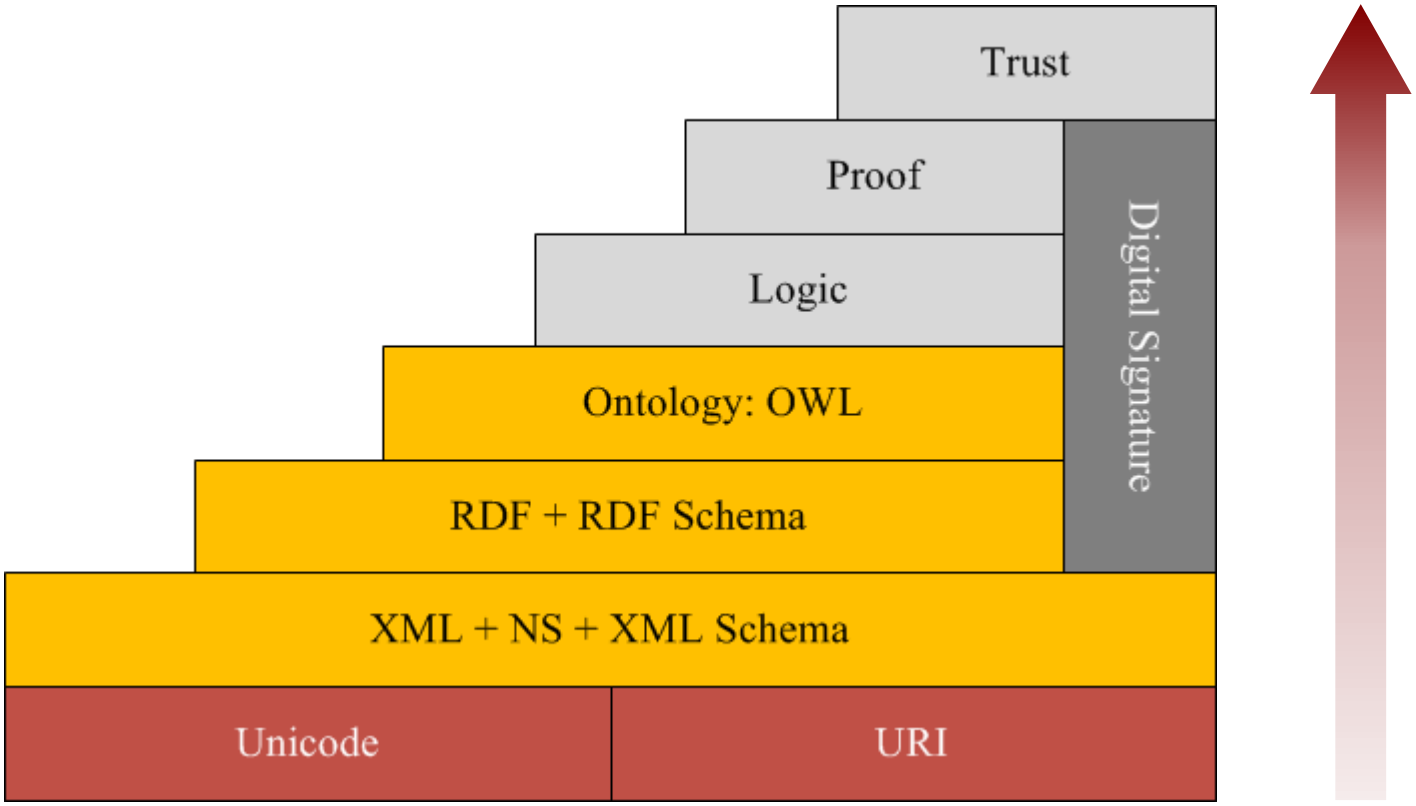
- 框架是一种描述所论对象属性的数据结构，由框架名、槽、侧面和值四部分组成。
- 框架类型：类框架 + 实例框架
- 层次结构：子类-subclass of->父类、实例-instance of->类
- 优点：结构化、继承性、自然性、模块化。
- 缺点：不能表示过程性知识、缺乏明确的推理机制。

# 脚本表示法

- 脚本是描述特定上下文中的原型事件序列的结构化表示，由进入条件、角色、道具、场景和结果五部分组成。
- 缺点：表达能力非常有限，表示范围窄，不具备对于对象基本属性的描述能力，也难以描述复杂事件发展的可能方向。
- 优点：在非常狭小的领域内，脚本表示却可以更细致地刻画步骤和时序关系，适合于表达预先构思好的特定知识或顺序性动作及事件。

# 语义网表示法

- 语义网提供了一套为描述数据而设计的表示语言和工具，用于形式化的描述一个知识领域内的概念、术语和关系。

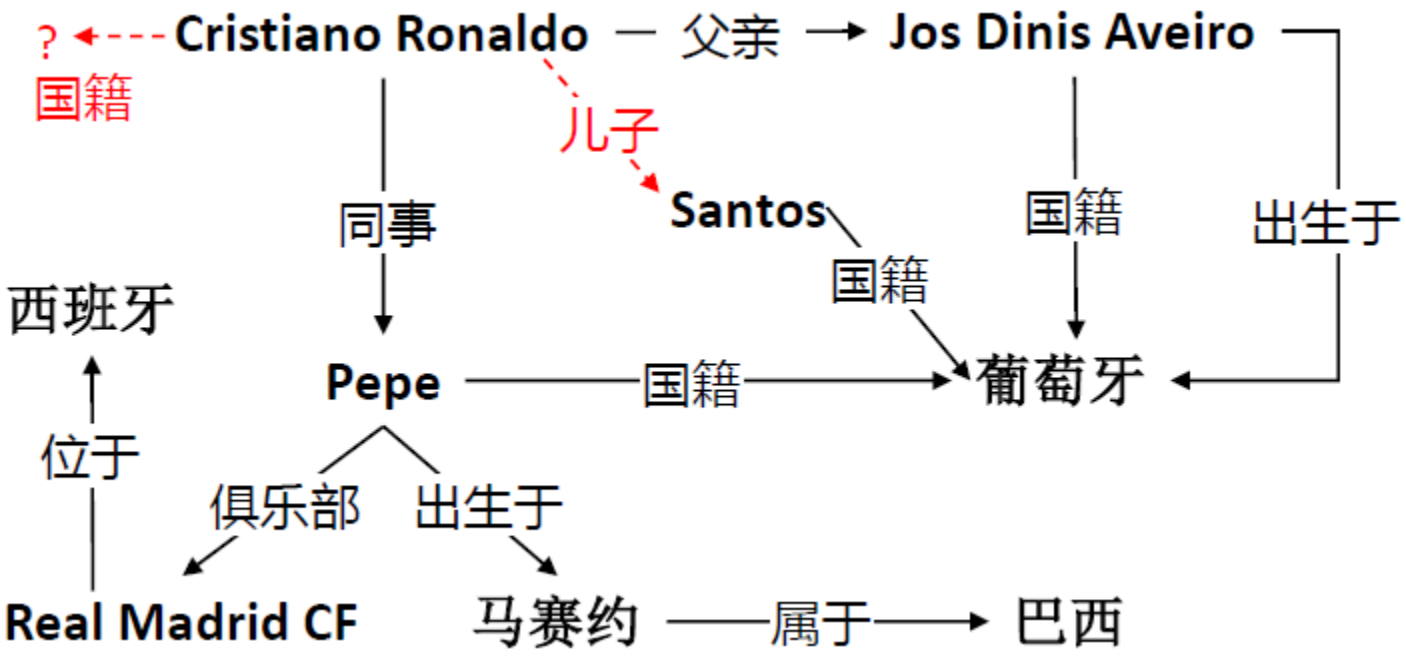


# 语义网表示法

- **XML** (http://www.w3.org/XML/) 提供了一种结构化文档的表层语法，但没有对文档含义施加任何语义约束 。
- **RDF** (http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/) 是一个关于对象（资源）和它们之间关系的数据模型，该模型具备简单语义，能够用XML语法表示。
- **RDF Schema** (http://www.w3.org/TR/2002/WD-rdf-schema-20021112/) 是一组描述RDF资源的类和属性的建模原语，提供了关于这些类和属性的层次结构的语义。
- **OWL** (http://www.w3.org/TR/2004/REC-owl-ref-20040210/) 添加了更多用于描述类和属性的建模原语，支持更加丰富的语义表达并支持推理。

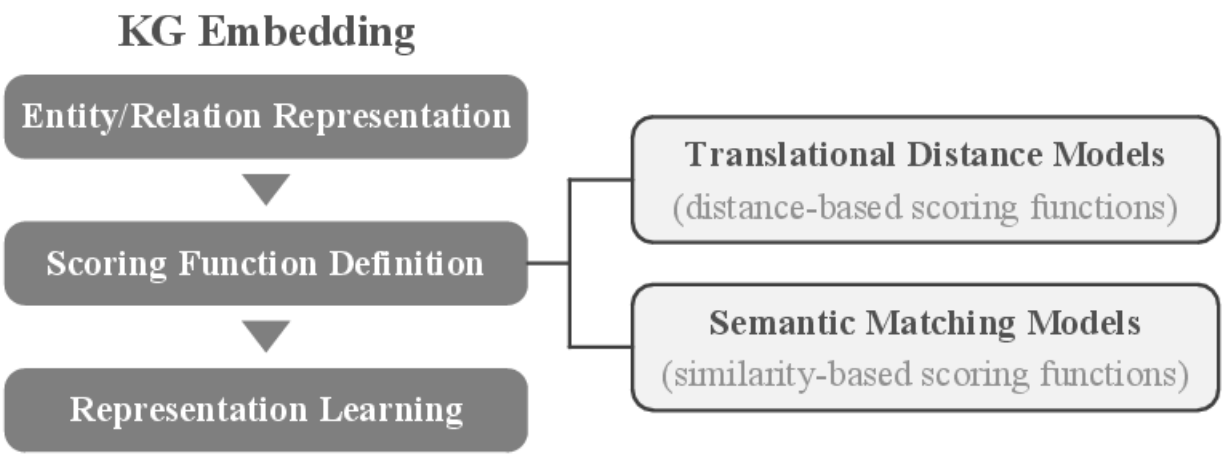
# 知识图谱中的知识表示

- 狭义知识图谱：具有图结构的三元组知识库。
  - 知识库中的实体作为知识图谱中的节点。
  - 知识库中的事实作为知识图谱中的边，边的方向由头实体指向尾实体，边的类型就是两实体间关系类型。



# 分布式知识表示

- 核心思想：将符号化的实体和关系在低维连续向量空间进行表示，在简化计算的同时最大程度保留原始的图结构。
  - 实体关系表示（向量/矩阵/张量）
  - 打分函数定义（距离函数/相似度函数）
  - 表示学习（开放世界假设/封闭世界假设）



谢谢！

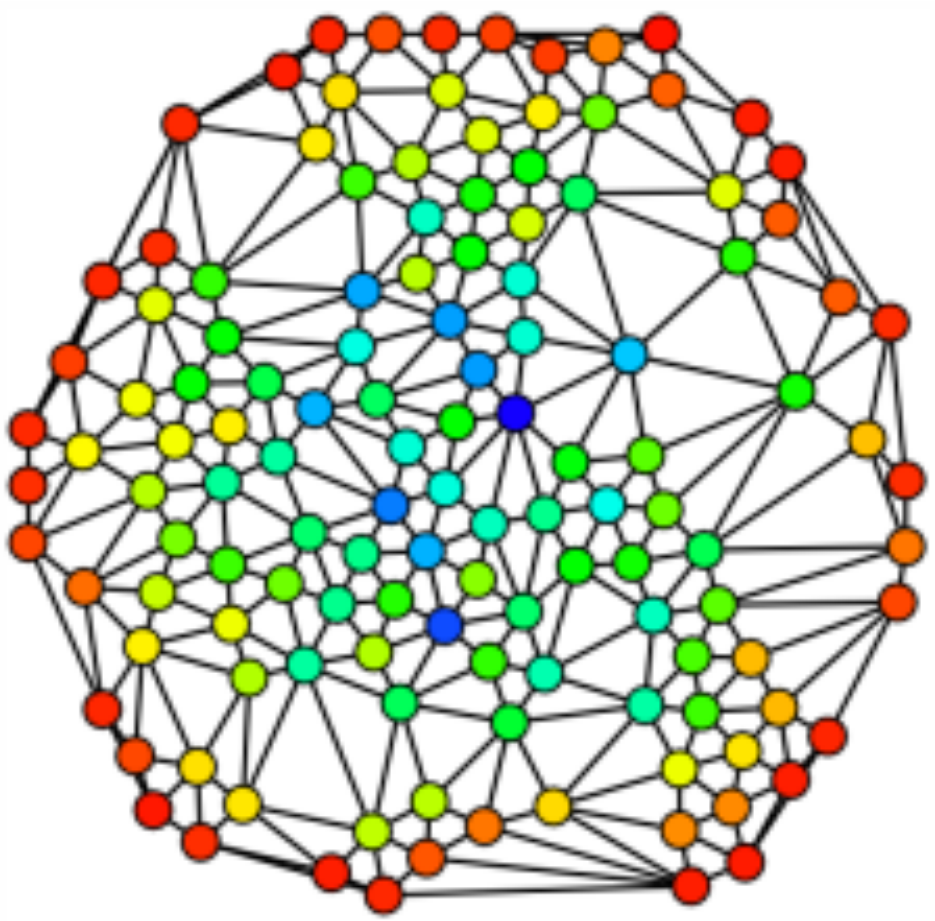
wangquan@iie.ac.cn

# 知识框架学习与融合

刘康
中国科学院自动化研究所

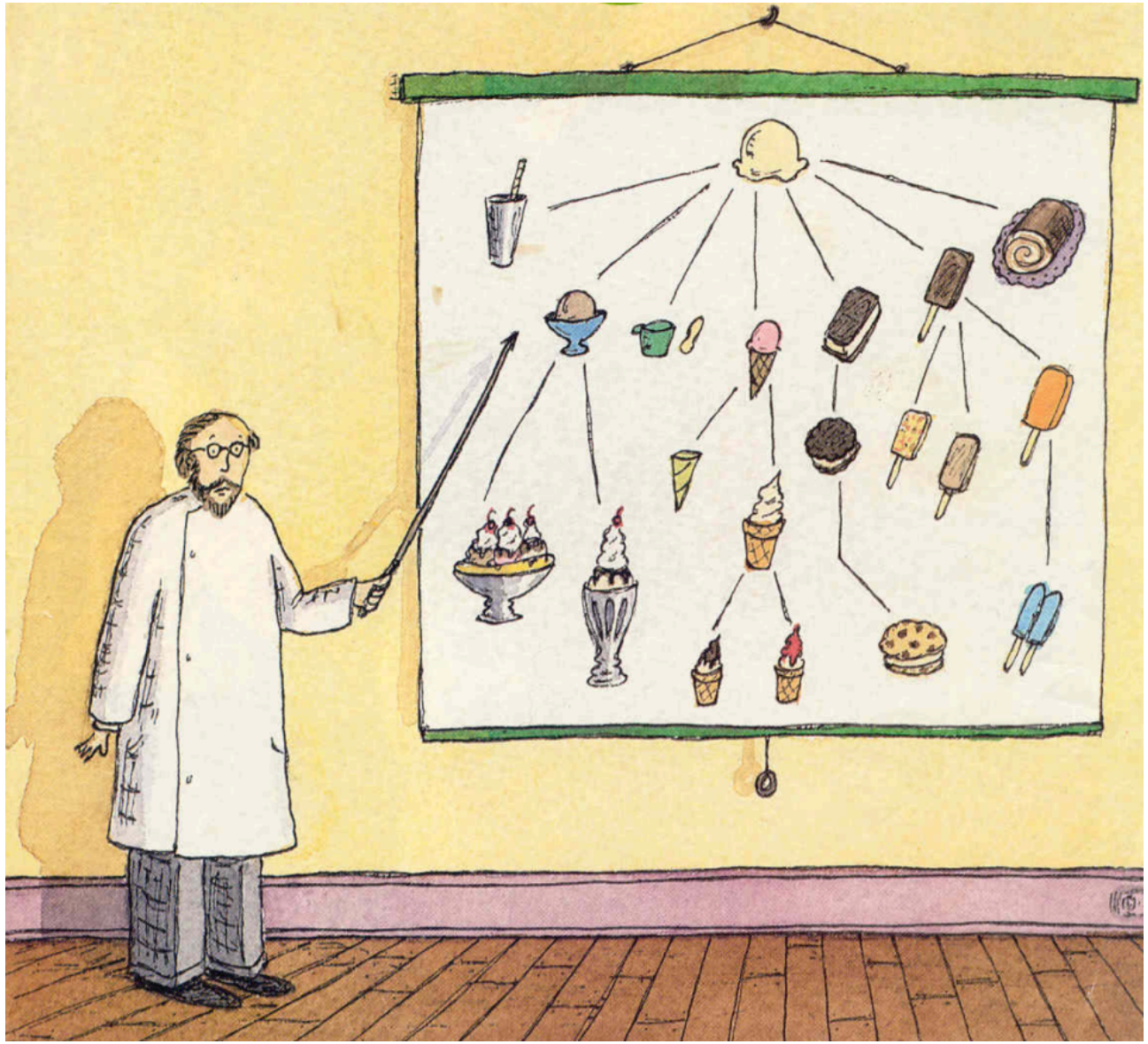
# 目录

- 什么是Ontology
- 如何自动构建Ontology
  - Ontology Learning
- 如何融合多个异构Ontology
  - Ontology Matching



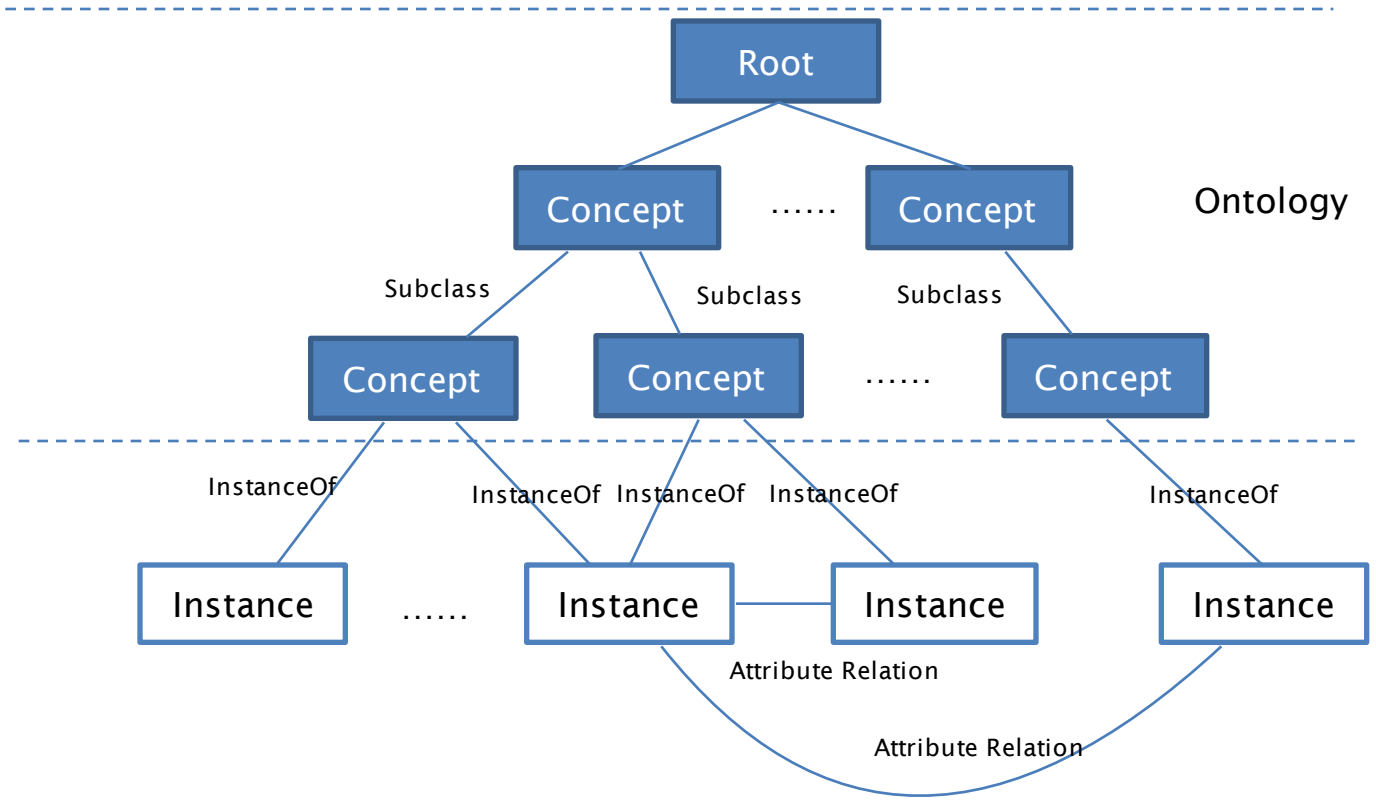
# Ontology

- Ontology is the study of the nature of being
- Ontologies 是（特定领域）信息组织的一种形式



# Ontology是什么

- 本体(Ontology)通过对于概念(Concept)、术语(Terminology)及其相互关系(Relation, Property)的规范化(Conceptualization)描述，勾画出某一领域的基本知识体系和描述语言



# Ontology是什么

**Definition 2.2 (Ontology).** *An ontology is a tuple* $o = \langle C, I, R, T, V, \leq, \perp, \in, = \rangle$, *such that:*

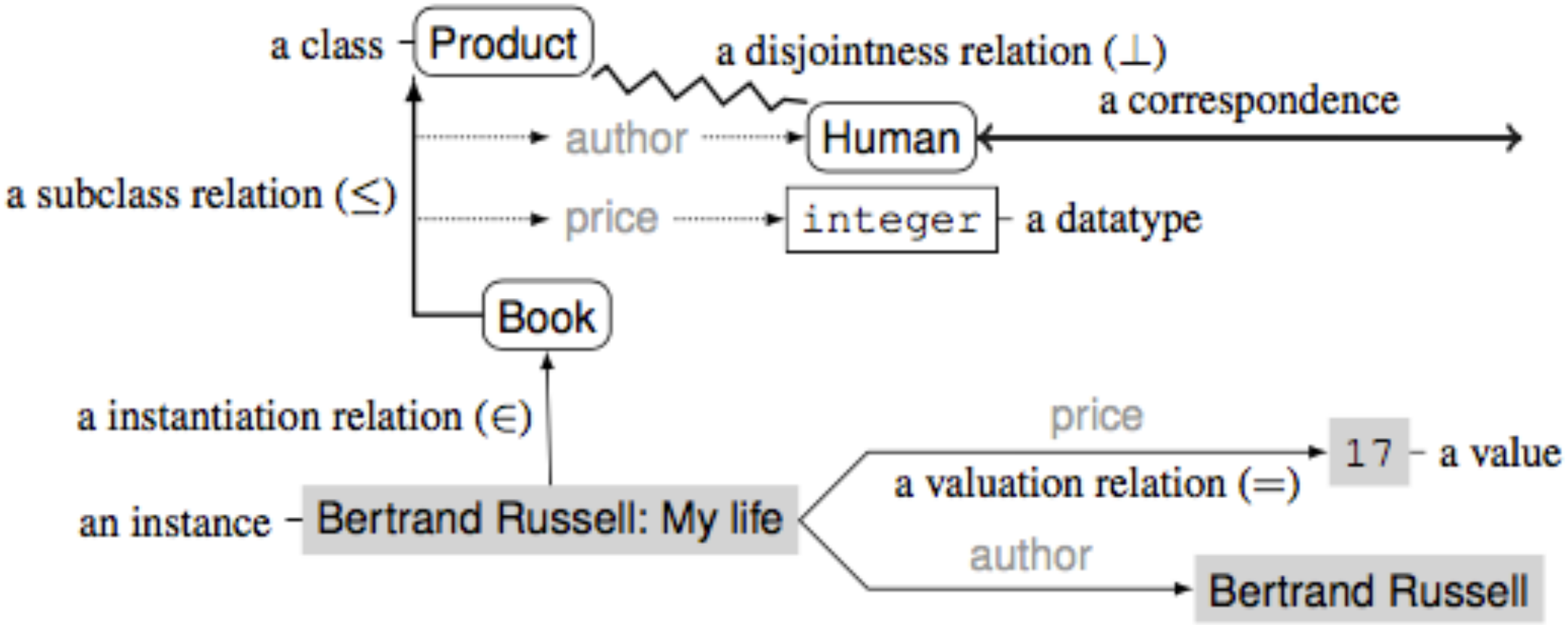
$C$ *is the set of classes;*
$I$ *is the set of individuals;*
$R$ *is the set of relations;*
$T$ *is the set of datatypes;*
$V$ *is the set of values* ($C$, $I$, $R$, $T$, $V$ *being pairwise disjoint);*
$\leq$ *is a relation on* $(C \times C) \cup (R \times R) \cup (T \times T)$ *called specialisation;*
$\perp$ *is a relation on* $(C \times C) \cup (R \times R) \cup (T \times T)$ *called exclusion;*
$\in$ *is a relation over* $(I \times C) \cup (V \times T)$ *called instantiation;*
$=$ *is a relation over* $I \times P \times (I \cup V)$ *called assignment.*

Ontology Matching *Jérôme Euzenat, Pavel Shvaiko*

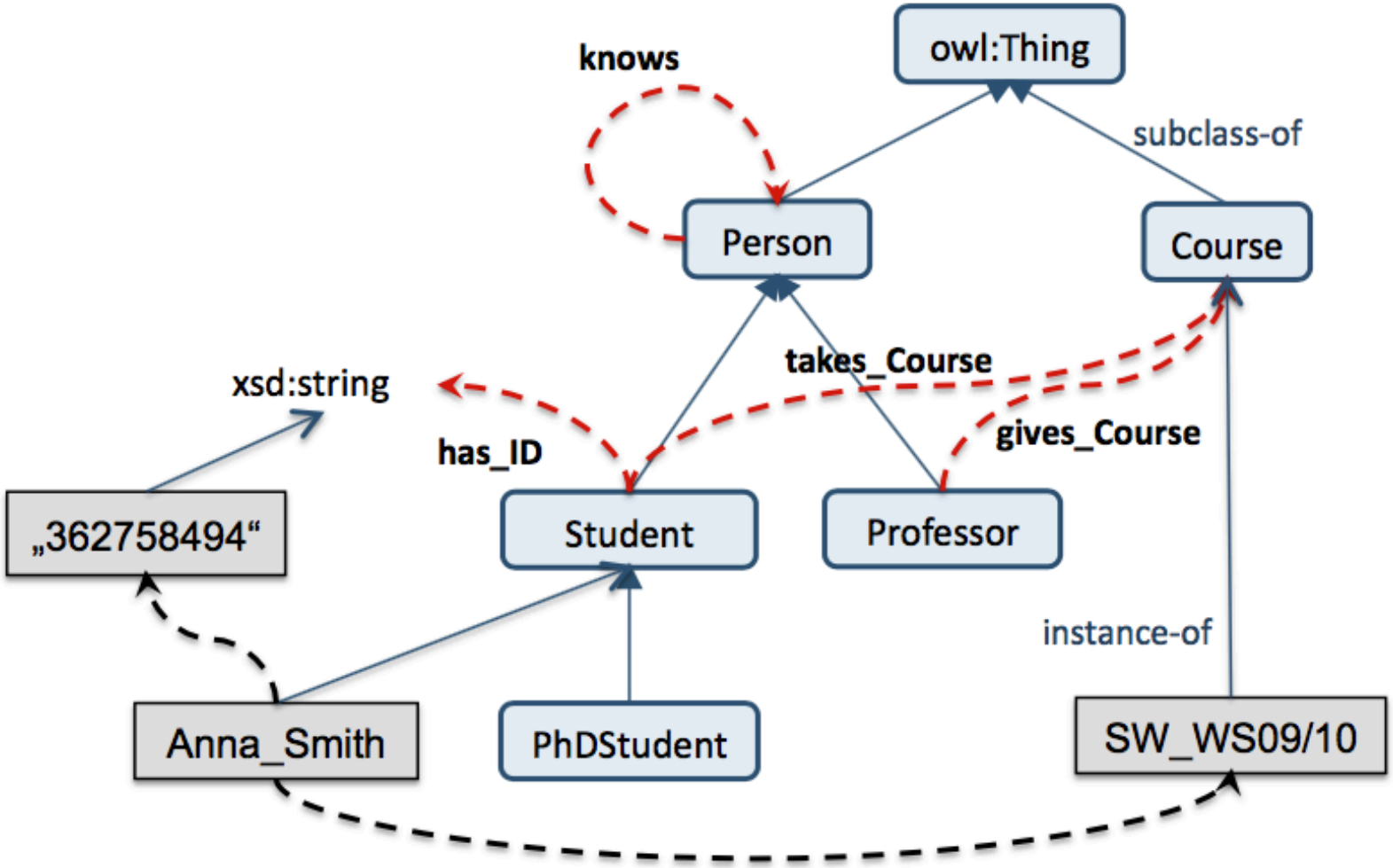
# Ontology是什么

- Conceptualization
  - Abstract model (concepts)
  - 指某一概念系统所蕴涵的语义结构，它是对某一事实结构的一组非正式的约束规则。它可以理解和/或表达为一组概念（如实体、属性、过程）及其定义和相互关系
- Explicit
  - The concepts are explicitly defined
- Formal
  - Machine readable
- Shared
  - Accepted by a group and not private to some individual

# Ontology 例子


a class
Product
a disjointness relation (⊥)
a correspondence
author
Human
a subclass relation (≤)
price
integer
a datatype
Book
a instantiation relation (∈)
price
17
a value
a valuation relation (=)
an instance
Bertrand Russell: My life
author
Bertrand Russell

# Ontology 例子


knows
owl:Thing
subclass-of
Person
Course
takes_Course
xsd:string
gives_Course
has_ID
„362758494“
Student
Professor
instance-of
Anna_Smith
PhDStudent
SW_WS09/10

# Various Forms of Ontologies

'Ordinary' glossaries | Data dictionaries | Structured glossaries | Principled, informal hierarchies | XML schemas | Formal taxonomies | Description logics

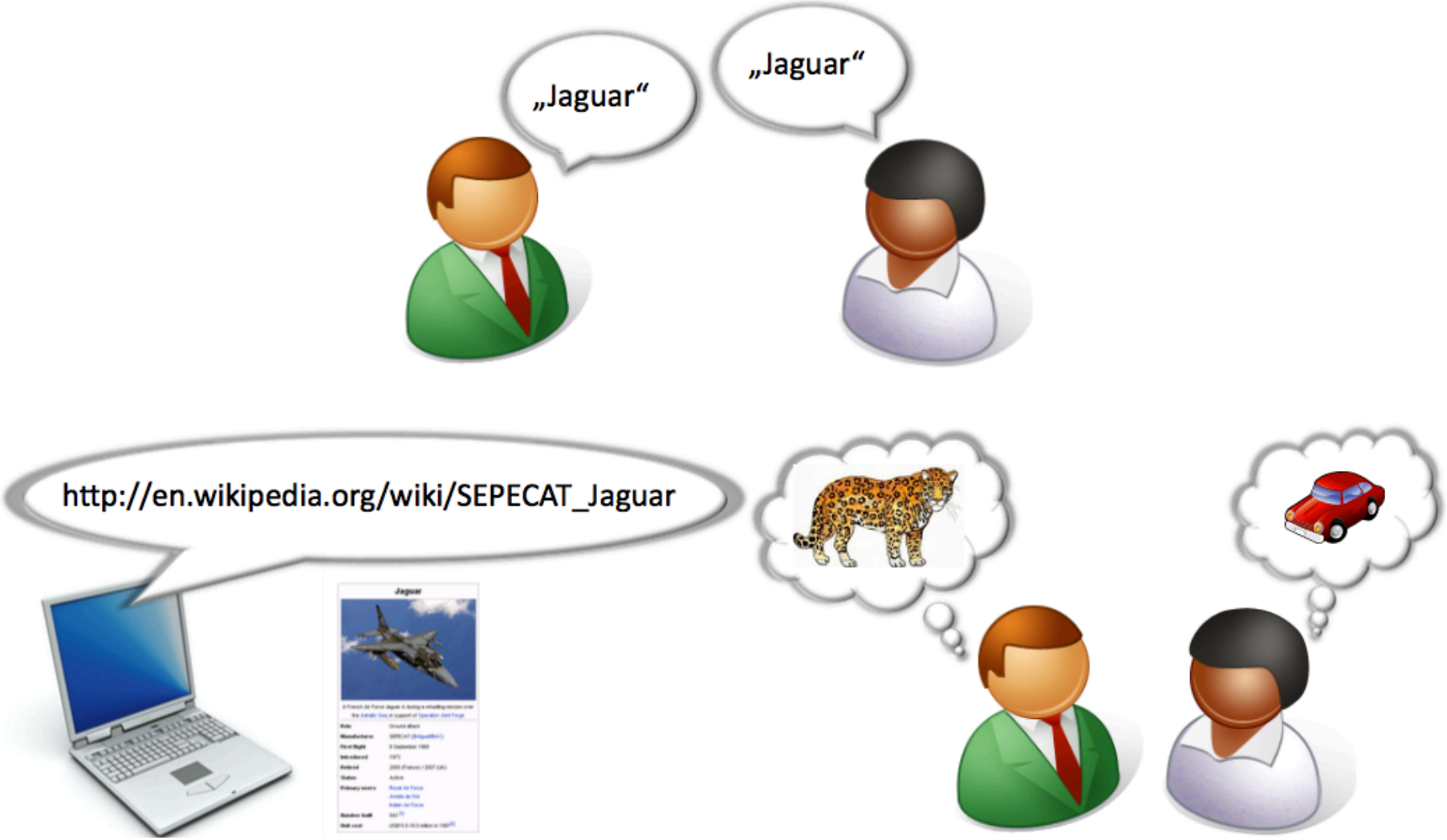
→ *expressivity*

Terms | Ad hoc hierarchies | Thesauri | XML DTDs | Database schemas | Entity-relationship models | Frames | Logics

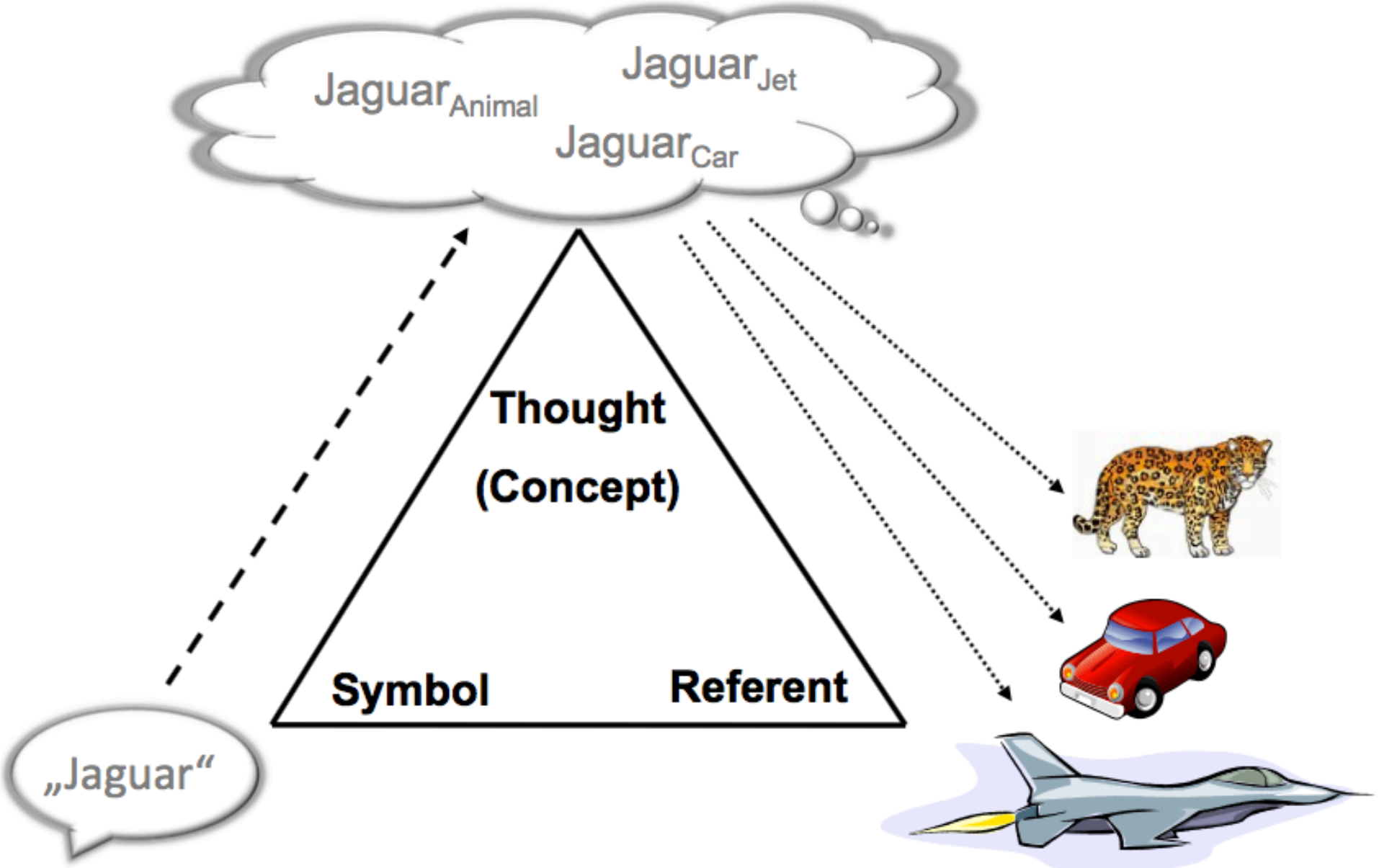
**Glossaries and data dictionaries** | **Thesauri and taxonomies** | **Metadata and data models** | **Formal ontologies**

# Ontology应用

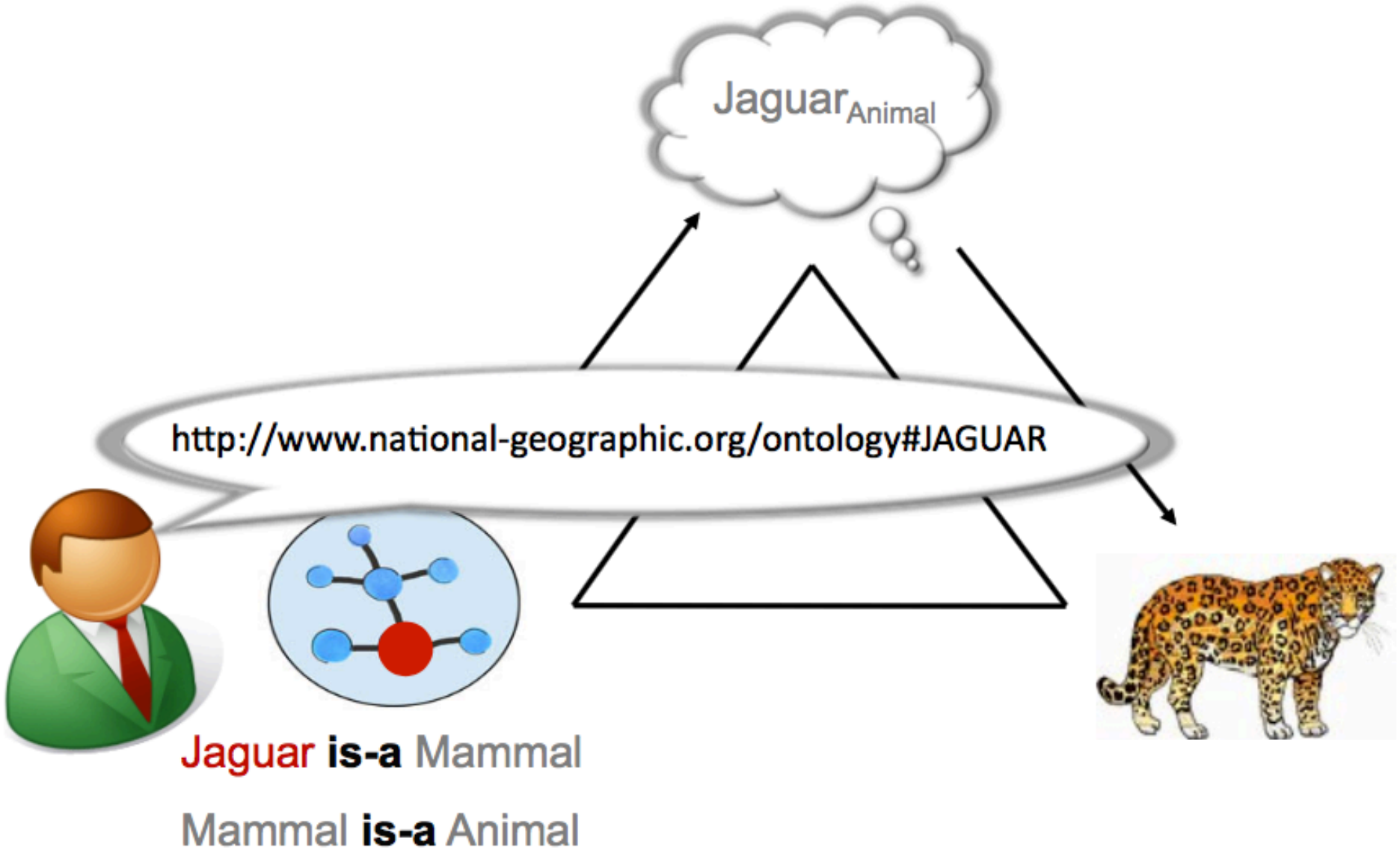
- 消歧

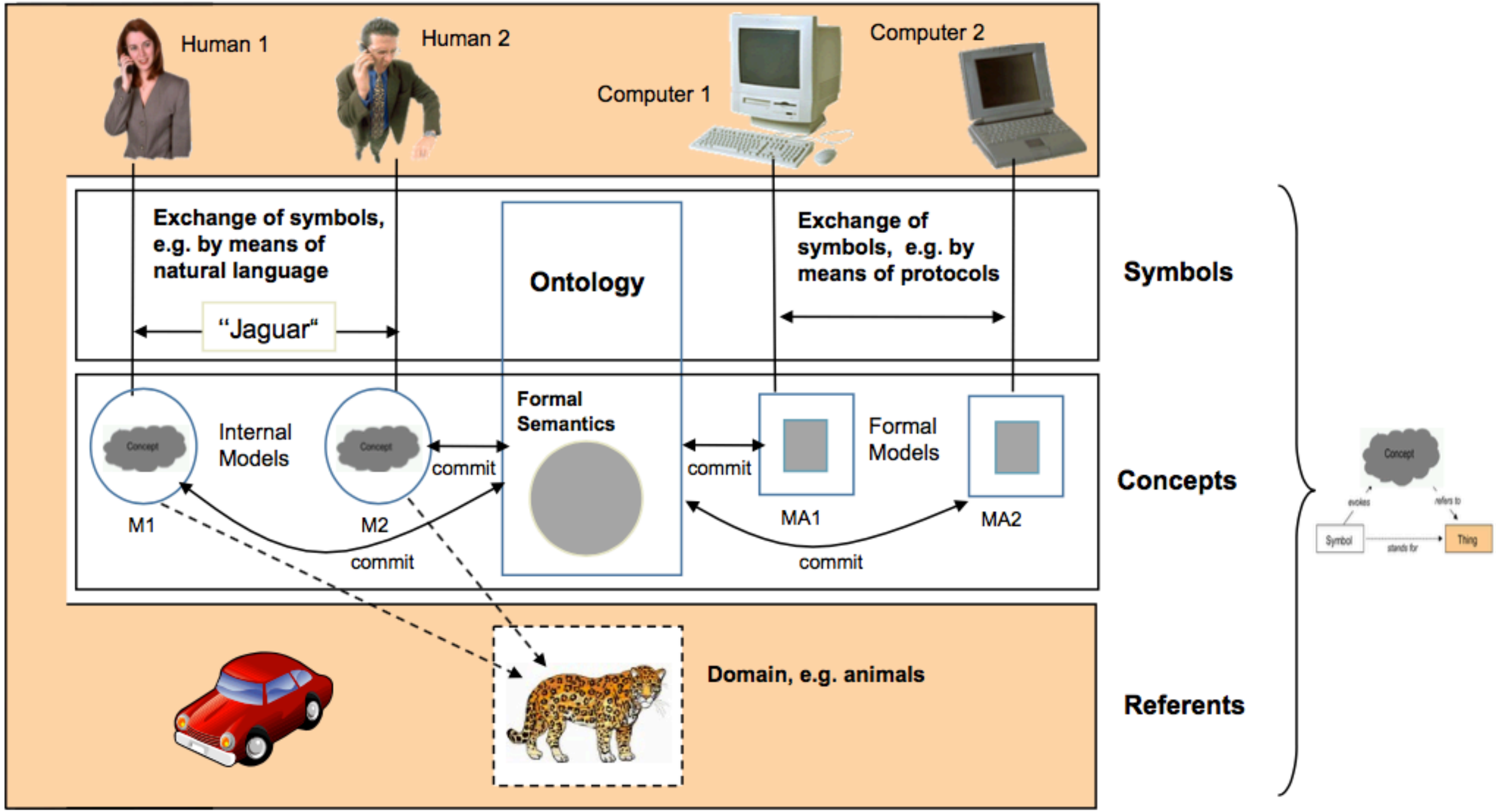


# Ontology应用


Jaguar Animal
Jaguar Jet
Jaguar Car
Thought
(Concept)
Symbol
Referent
„Jaguar“

# Ontology应用


Jaguar Animal
http://www.national-geographic.org/ontology#JAGUAR
Jaguar is-a Mammal
Mammal is-a Animal


Human 1
Human 2
Computer 1
Computer 2
Exchange of symbols, e.g. by means of natural language
"Jaguar"
Ontology
Exchange of symbols, e.g. by means of protocols
Symbols
Concept
Internal Models
Concept
commit
Formal Semantics
commit
Formal Models
M1
M2
commit
MA1
MA2
commit
Concepts
Domain, e.g. animals
Referents
Concept
evokes
refers to
Symbol
stands for
Thing

# Ontology vs. Knowledge Base

- Ontology：共享概念化的规范，涉及概念、关系和公理三个要素
- Knowledge：服从于ontology 控制的知识单元的载体

Ontology

Knowledge Base

# 知识的其他组织形式

Folksonomy

# 知识的其他组织形式

## 百科分类树 知识地图

搜分类 显示树型结构

- 页面总分类 收起
  - + 自然 展开
  - + 文化 展开
  - + 人物 展开
  - + 历史 展开
  - + 生活 展开
  - + 社会 展开
  - + 艺术 展开
  - + 经济 展开
  - + 科学 展开
  - – 体育 收起
    - + 奥林匹克运动会 展开
    - + 体育组织 展开
    - + 体育赛事 展开
    - + 体育奖项 展开
    - + 体育理论 展开
    - + 体育传媒 展开
    - + 体育运动 展开
    - + 体育设施 展开
    - + 体育院校 展开
  - + 技术 展开
  - + 地理 展开
  - + HOT 展开
  - + 企业专题 展开

Taxonomy

# 知识的其他组织形式

- Ontology
  - 树状结构，不同层节点之间是严格的IsA关系
  - 优点：可以适用于知识的推理
  - 缺点：无法表示概念的二义性（运动员：体育？人物？）
- Taxonomy
  - 树状结构，上下位节点之间非严格的IsA关系
  - 优点：可以表示概念的二义性（体育—>运动员）
  - 缺点：不适用于推理，无法避免概念冗余（餐厅：美食？机构？地点？）
- Folksonomy
  - 类别标签，更加开放
  - 优点：能够涵盖更多的概念
  - 缺点：如何进行标签管理？

# 知识的其他组织形式

- 目前的知识资源多是采用Folksonomy与Taxonomy相结合的组织形式
- 但是能够覆盖的类别还很少

| 全部 | 含有开放分类(Folksonomy)的页面数比例 |
| --- | --- |
| 互动百科 | 70.19% |
| 百度百科 | 64.38% |

- 这些开放式类别标签存在冗余、不规范的问题，标签之间也缺乏关联
  - 体育、人物
  - 1980年、购房、房产、房地产……

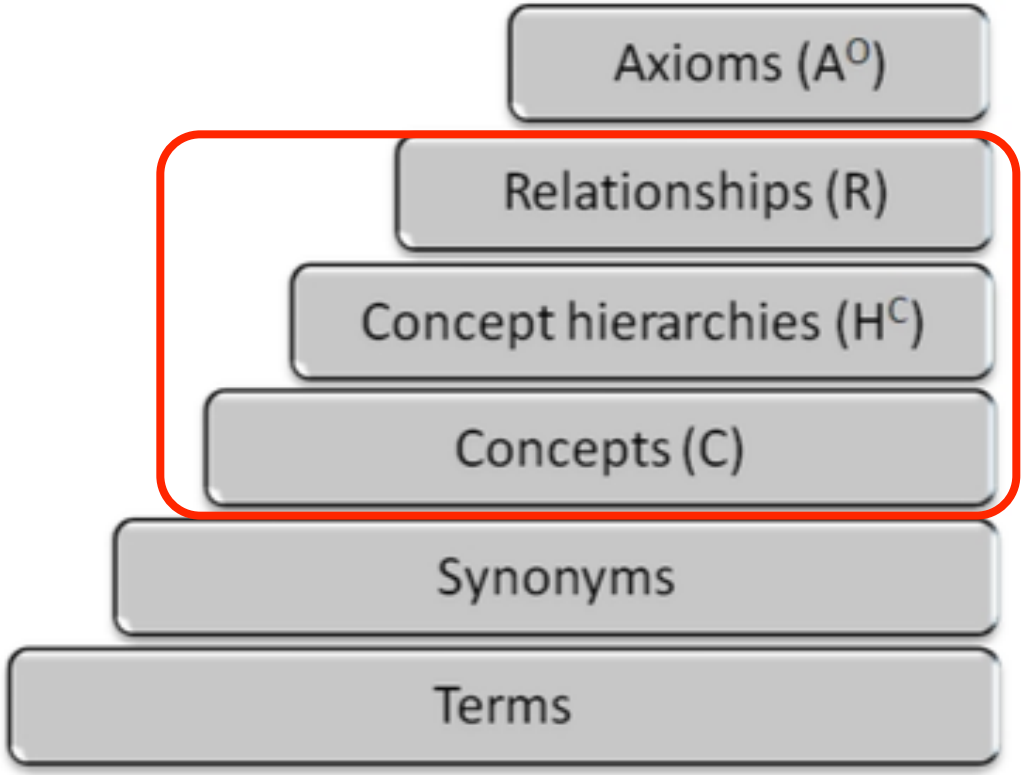
# 几个概念

- Semantic Annotation
  - Semantic annotation is the process of annotating a document with ontology elements
- Ontology Population
  - Ontology population adds instances to a given ontology
- Ontology Learning
  - Constructing Ontology

# Ontology Learning

- 手工构建Domain Ontology费时费力
- 自动构建
  - 目前还不能直接使用
  - 但是可以节省人力

- Concept Extraction（节点）
- Taxonomical Relation Extraction（关系）
- No-taxonomical Relation Extraction （属性）




Ontology Learning on Different Level

# Ontology Extraction from (Semi-)Structured Data

# Ontology Extraction from (Semi-)Structured Data

## 刘德华

开放分类 | 人物 | 歌手 | 演员

### 基本信息

| | | | |
|---|---|---|---|
| 中文名: | 刘德华 | 英文名: | Andy Lau,Lau Tak Wah |
| 其它外文名: | Lau TAK Wah | 别名: | 华仔，老大，华哥，华Dee，刘天王，华神，刘主席 |
| 性别: | 男 | 出生日期: | 1961年9月27日 |
| 国籍: | 中国 | 籍贯: | 广东新会 |
| 出生地: | 香港新界大埔镇泰亨村 | 民族: | 汉族 |
| 毕业院校: | 可立中学，第十期无线艺员培训班 | 身高: | 174cm |
| 宗教信仰: | 佛教 | 体重: | 63kg |
| 职业: | 演员、歌手、填词人、电影人 | 音乐代表作品: | 忘情水、冰雨、练习、中国人、笨小孩 |
| 音乐类型: | 流行音乐 | 影视代表作品: | 暗战、无间道、盲探、天若有情等 |
| 出道地区: | 香港 | 活跃年代: | 上世纪80年代至今 |
| 经纪公司: | 东亚唱片、映艺娱乐 | 星座: | 天秤座 |
| 血型: | AB型 | 爱好: | 保龄球、羽毛球、台球、驾驶 |
| 特长: | 唱歌、跳舞、拍戏 | 其他信息: | 第一部书:《浓情爱不完》；自传体散文集《我是这样长大的》 |

同花顺 WWW.10JQKA.COM.CN | 个股

输入股票/港股代码、简称或关键字 搜索

首页概览 | 资金流向 | 公司资料 | 新闻公告 | 财务分析 | 经营分析 | 股东股本 | 主力持仓 | 公司大事 | 分红融资 | 价值分析 | 行业分析

详细情况 | 高管介绍 | 发行相关 | 参股控股公司

华通热力 002893

加入自选

26.44 ↑ 2.40 9.98%

涨停：26.44 跌停：21.64

2017年10月4日 15:00:00

| | | |
|---|---|---|
| 今开：26.44 | 成交量：16.50万 | 振幅：0.00% |
| 最高：26.44 | 成交额：436.19万 | 换手：0.55% |
| 最低：26.44 | 总市值：31.73亿 | 市净率：5.13 |
| 昨收：24.04 | 流通市值：7.93亿 | 市盈率(动)：20.56 |

## 详细情况

华通热力

| | | |
|---|---|---|
| 公司名称：北京华远意通热力科技股份有限公司 | | 所属地域：北京市 |
| 英文名称：Beijing Huayuan Yitong Thermal Technology Co., Ltd. | | 所属行业：公用事业 — 电力 |
| 曾 用 名：- | | 公司网址：www.huatongreli.com |

主营业务：热力供应、节能技术服务

产品名称：热力供应 + 、节能技术服务 +

控股股东：赵一波 (持有北京华远意通热力科技股份有限公司股份比例：28.24%)

实际控制人：赵一波 (持有北京华远意通热力科技股份有限公司股份比例：28.24%)

最终控制人：赵一波 (持有北京华远意通热力科技股份有限公司股份比例：28.24%)

| | | |
|---|---|---|
| 董事长：赵一波 | 董 秘：石秀杰 | 法人代表：赵一波 |
| 总 经 理：赵一波 | 注册资金：1.2亿元 | 员工人数：1013 |
| 电 话：86-10-52917878 | 传 真：86-10-52917676 | 邮 编：100070 |

办公地址：北京市丰台区南四环西路188号三区8号楼8层

公司简介：

北京华远意通热力科技股份有限公司主营业务为热力供应、节能技术服务。 公司是国家发改委首批备案合同能源管那推掉单位，是北京市认定的首批.综合类核心供热节能技术产品公司".曾多次被北京市市政市容管理委员会评为"北京市供热先进单位"、"北京市供热优秀单位二井曾荣获"全国节能突出贡献示范单位一、"质址安全信并AAA级企业'等多项荣咨称号。此外.公司先后通过ISO9001质最管理体系、1SO14001环境管理体系认证和职业健康安全体系认证。

# Ontology Extraction from (Semi-)Structured Data

## 发行相关

| | | |
|---|---|---|
| 成立日期：2002-12-12 | 发行数量：3000.00万股 | 发行价格：8.56元 |
| 上市日期：2017-09-15 | 发行市盈率：22.9700倍 | 预计募资：2.18亿元 |
| 首日开盘价：10.27元 | 发行中签率：0.02% | 实际募资：2.57亿元 |

主承销商：招商证券股份有限公司　　上市保荐人：招商证券股份有限公司

历史沿革：

北京华远意通热力科技股份有限公司曾用名北京华远意通热力科技股份有限公司。

1,2002年12月，华通有限设立。

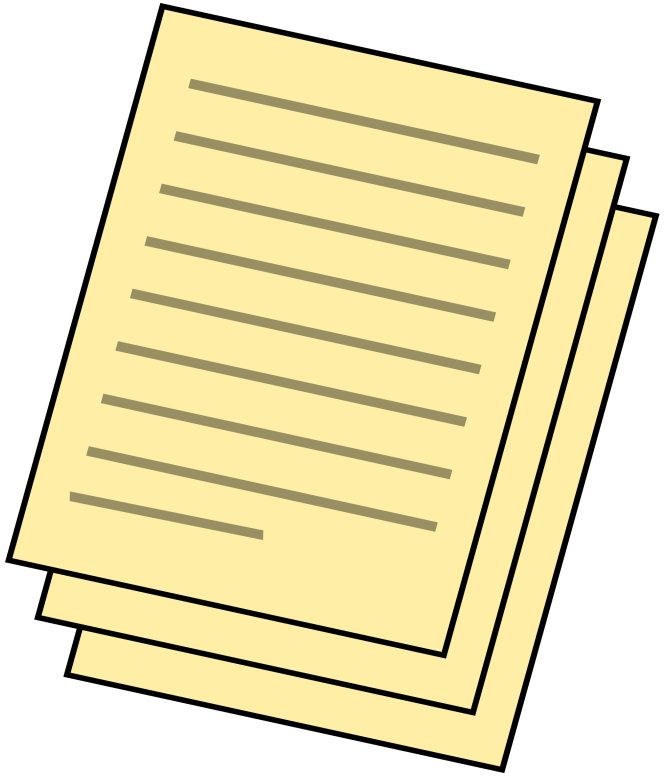
本公司前身华通有限系赵一波、张莹、李萍、孙福童四名自然人于2002年10月以货币出资方式共同设立，注册资本为50万元，实收资本为50万元，法定代表人为赵一波。

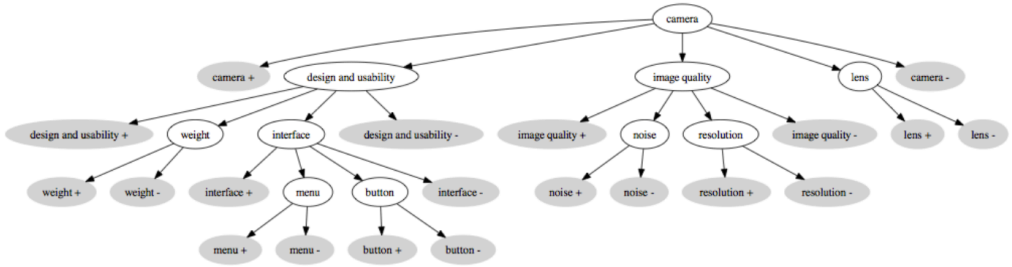
2002年12月10日，北京瑞文成联合会计师事务所出具京瑞联验字[2002]第09-B-3236号《验资报告》。2002年12月12日华通有限取得北京市工商行政管理局颁发的注册号为1101012518525的《企业法人营业执照》。

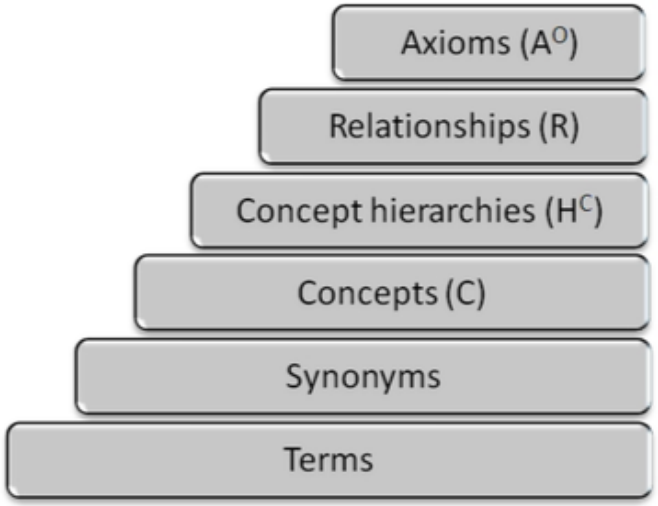
2,2003年11月，第一次股权转让。

查看全部▼

# Ontology Extraction from Texts



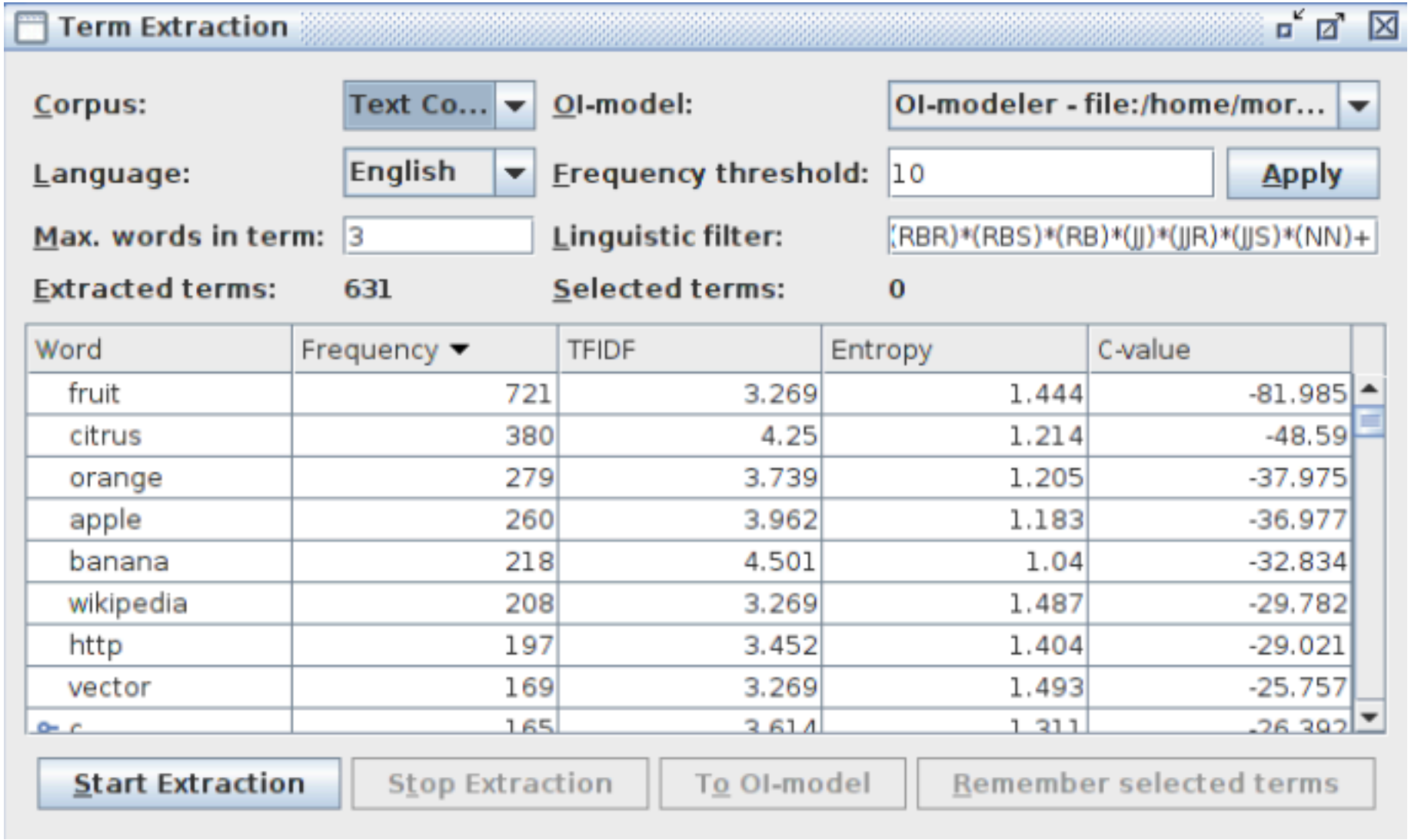


# Terminology

- Terms Extraction
  - Single words or multi-words expression
  - Linguistic methods
    - POS + Patterns: N-N, Adj-N
  - Statistical methods
    - N-grams with high frequencies
    - TFIDF
    - High frequencies in in-domain and low frequencies in out-domains

# Terminology

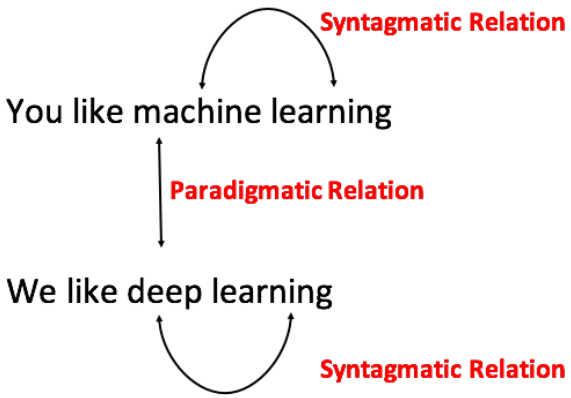



| Word | Frequency | TFIDF | Entropy | C-value |
|---|---|---|---|---|
| fruit | 721 | 3.269 | 1.444 | -81.985 |
| citrus | 380 | 4.25 | 1.214 | -48.59 |
| orange | 279 | 3.739 | 1.205 | -37.975 |
| apple | 260 | 3.962 | 1.183 | -36.977 |
| banana | 218 | 4.501 | 1.04 | -32.834 |
| wikipedia | 208 | 3.269 | 1.487 | -29.782 |
| http | 197 | 3.452 | 1.404 | -29.021 |
| vector | 169 | 3.269 | 1.493 | -25.757 |
| c | 165 | 3.614 | 1.311 | -26.392 |

Text To Onto Term Extraction

# Synonyms

- Clustering：Semantic Terms Similarity
  - Character based Similarity: 编辑距离
  - 利用外部资源：WordNet、HowNet、同义词词林
  - 基于统计的方法
    - 基本假设（distributed representation）：语义相同的词（Terms）具有相同的上下文



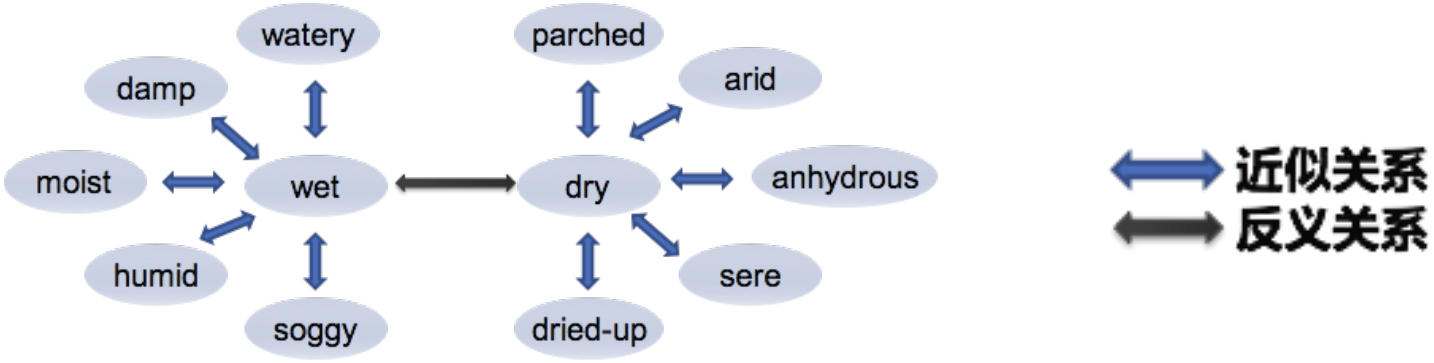
# WordNet

- Synset： WordNet 将英语的名词、动词、形容词、和副词组织为Synsets，每一个Synset表示一个基本的词汇概念

- 概念关系
- 同义关系
- 反义关系
- 上位关系
- 下位关系
- 整体关系（名词）
- 部分关系（名词）
- 蕴含关系（动词）
- 因果关系（动词）
- 近似关系（形容词）

**newspaper词义的上位synsets**

newspaper, paper
=> press, public press
=> print media
=> medium
=> instrumentality
=> artifact, artefact
=> whole, unit
=> object, physical object
=> physical entity
=> entity



# Concept and Concept Hierarchies

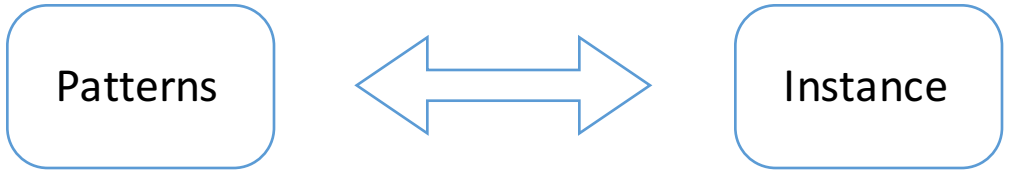
- Heuristic Patterns
  - such NP as {NP,}*{or|and} NP
    - such games as baseball and cricket
  - NP {,NP}* {,} {and¦or} other NP
    - rabbits and other animals (rabbits and other pets)
  - NP {,} including {NP,}* {or¦and} NP
    - fruits including apples and pears
  - NP {,} especially {NP,}* {or¦and} NP
    - Europeans, especially Italians (US presidents, especially democrats)
  - NP is a NP
    - A kangaroo is an animal living in Australia
  - a NP {named¦called} NP
    - Japanese people like to play a game called Go

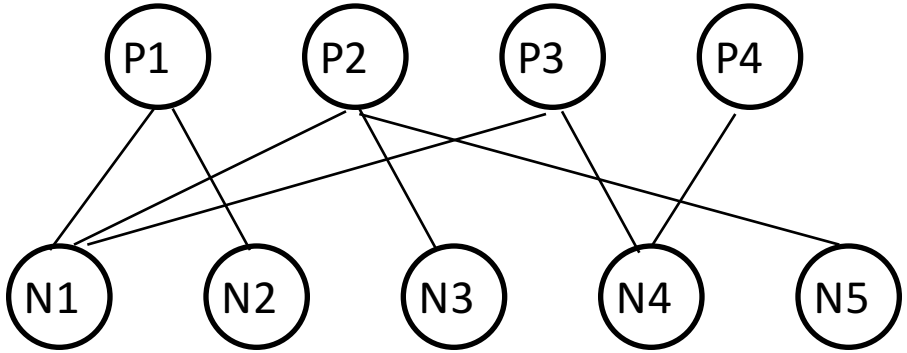
# Concept and Concept Hierarchies

- NP. The NP
  - John loves his Ferrari. The car …
- NP and other NP
  - universities and other institutions
- NP such as NP
  - sports such as tennis
- Among NP, NP
  - Among all musical instruments, violins are …
- NP {except for¦other than} NP
  - Employees except for managers suffer from …

# Concept and Concept Hierarchies

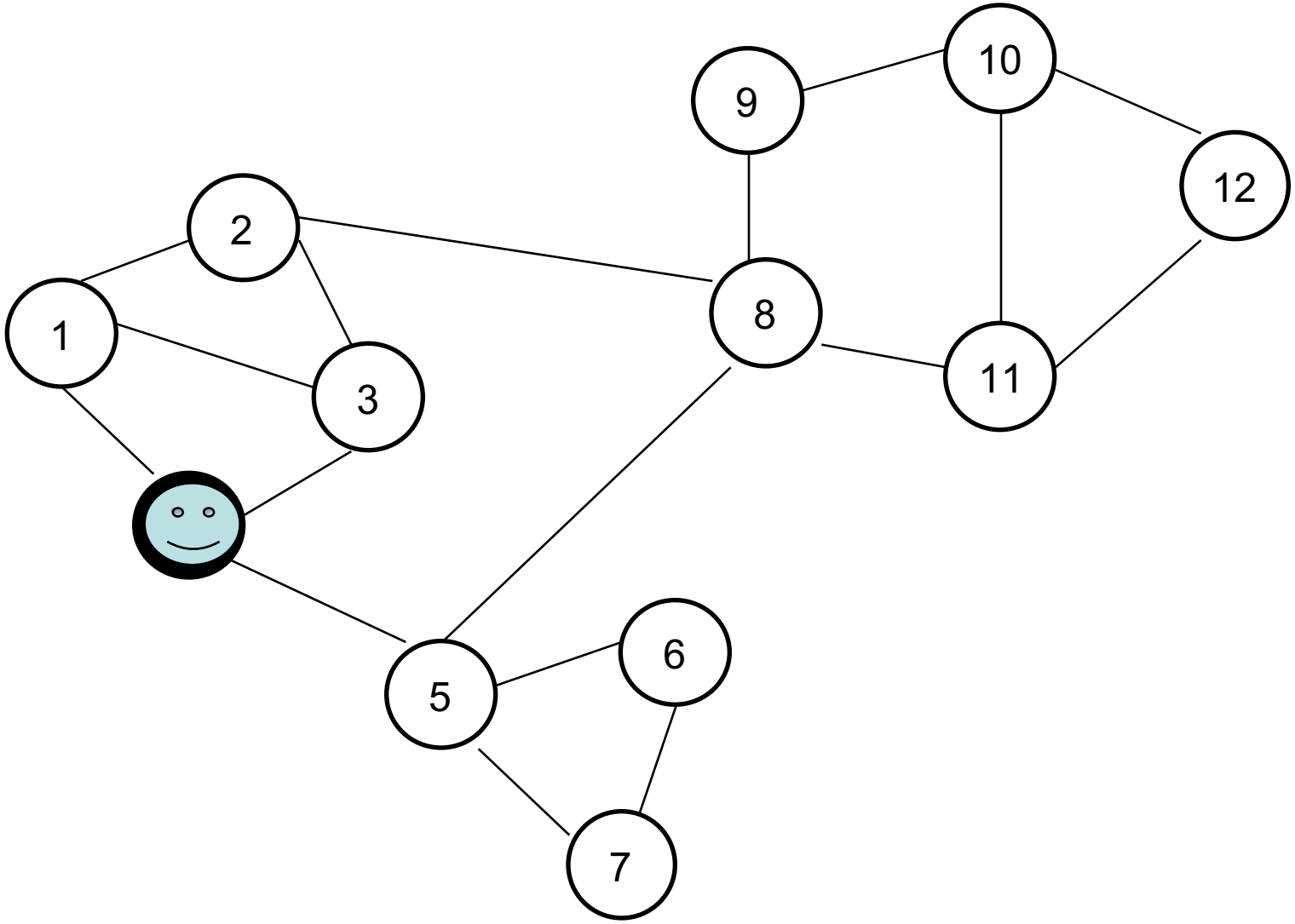
- Learned Patterns
  - Pattern Confidence



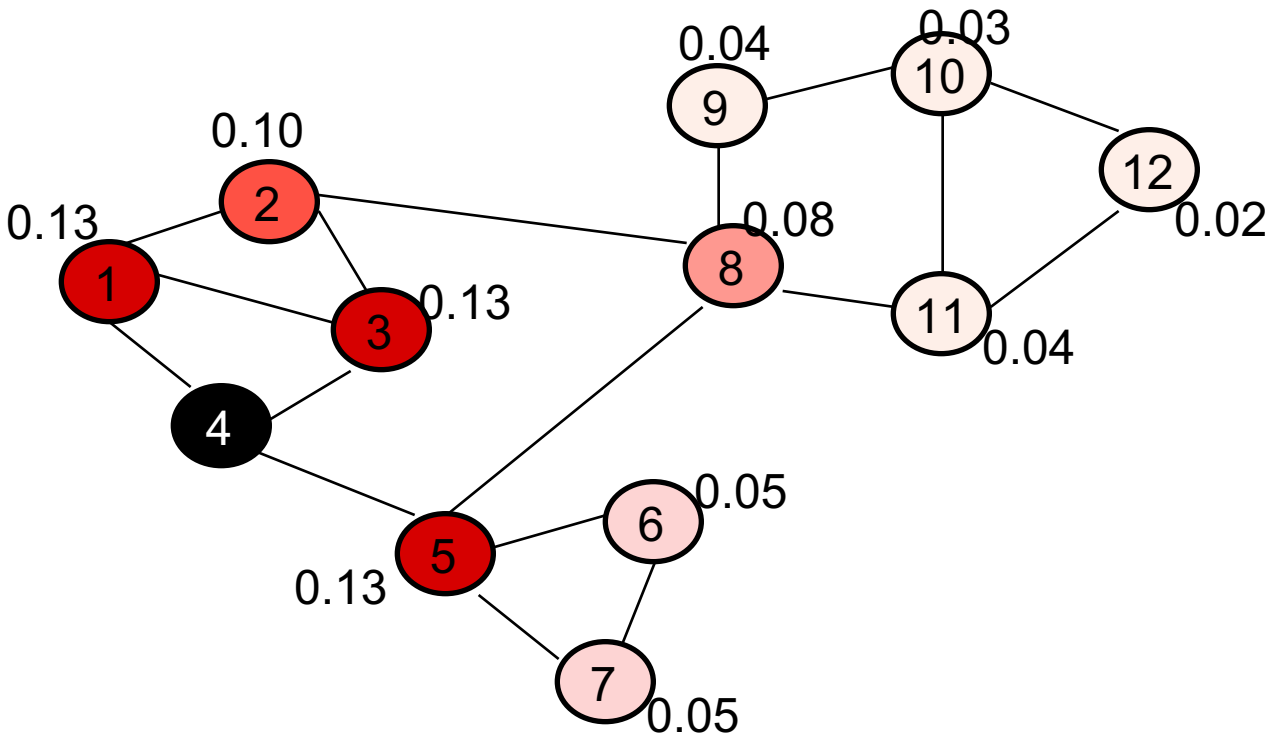




Radom Walking

# Radom Walking



# Radom Walking




| | Node 4 |
|---|---|
| Node 1 | 0.13 |
| Node 2 | 0.10 |
| Node 3 | 0.13 |
| Node 4 | 0.22 |
| Node 5 | 0.13 |
| Node 6 | 0.05 |
| Node 7 | 0.05 |
| Node 8 | 0.08 |
| Node 9 | 0.04 |
| Node 10 | 0.03 |
| Node 11 | 0.04 |
| Node 12 | 0.02 |

Ranking vector

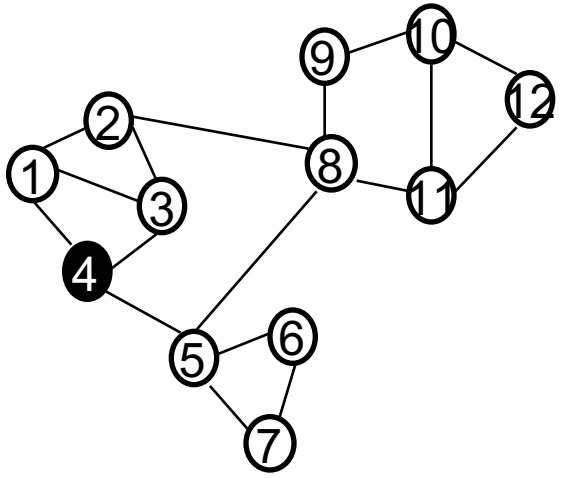
# Radom Walking

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$$

Ranking vector Adjacent matrix starting vector

$$\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} = 0.9 \times \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \end{pmatrix} \begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} + 0.1 \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

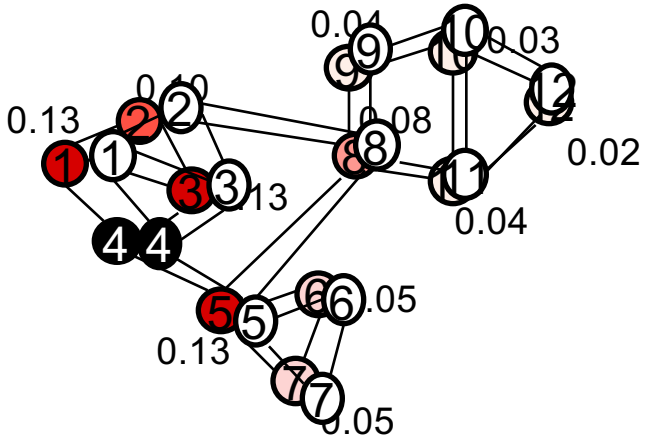
n x 1 n x n n x 1



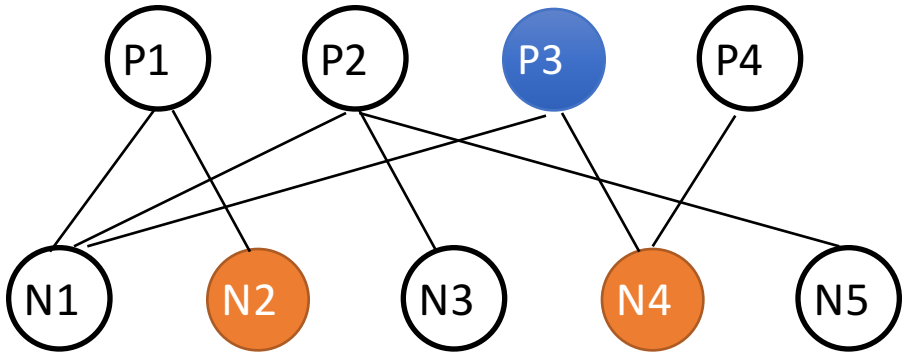
# Radom Walking

$$\vec{r} \leftarrow c\tilde{W}\vec{r} + (1-c)\vec{e}$$

$$\underset{\vec{r}}{\begin{pmatrix} 0.018 \\ 0.018 \\ 0.018 \\ 0.025 \\ 0.018 \\ 0.005 \\ 0.005 \\ 0.006 \\ 0.004 \\ 0.003 \\ 0.004 \\ 0.002 \end{pmatrix}} \leftarrow 0.9 \times \underset{\tilde{W}}{\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \end{pmatrix}} \underset{\vec{r}}{\begin{pmatrix} 0.013 \\ 0.010 \\ 0.013 \\ 0.022 \\ 0.013 \\ 0.005 \\ 0.005 \\ 0.008 \\ 0.004 \\ 0.003 \\ 0.004 \\ 0.002 \end{pmatrix}} + 0.1 \times \underset{\vec{e}}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}$$

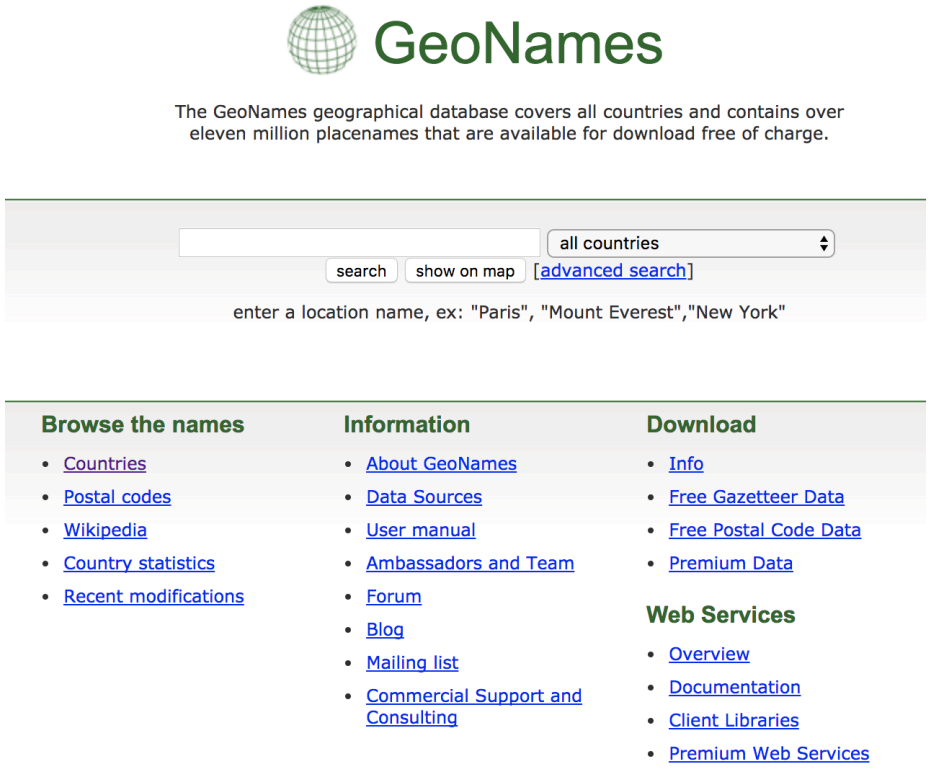


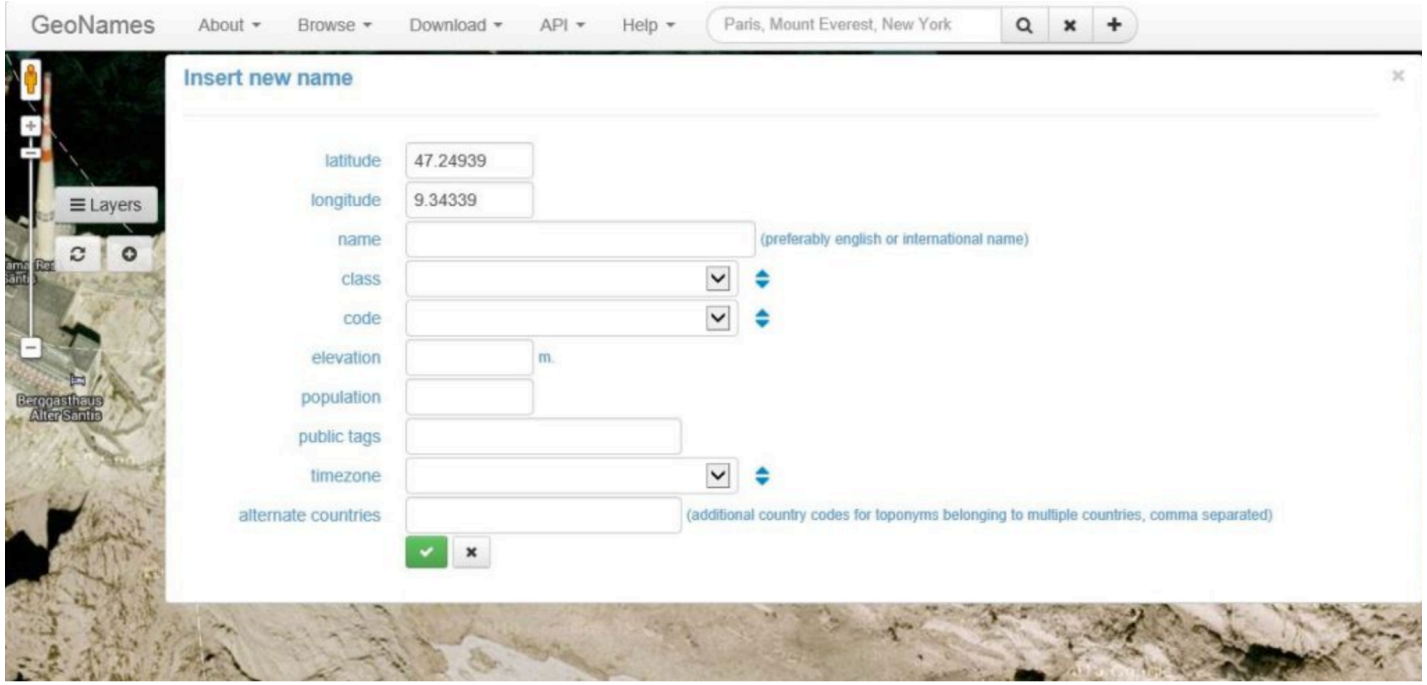
# Radom Walking



# 已有的Ontology

- GeoNames



# 已有的Ontology

- DBpedia Ontology

## Ontology Classes

- owl:Thing
  - Activity (edit)
    - Game (edit)
      - BoardGame (edit)
      - CardGame (edit)
    - Sales (edit)
    - Sport (edit)
      - Athletics (edit)
      - Boxing (edit)
        - BoxingCategory (edit)
        - BoxingStyle (edit)
      - HorseRiding (edit)
      - TeamSport (edit)
        - Soccer (edit)
  - Agent (edit)
    - Deity (edit)
    - Employer (edit)
    - Family (edit)
      - NobleFamily (edit)
    - FictionalCharacter (edit)
      - ComicsCharacter (edit)
        - AnimangaCharacter (edit)
      - DisneyCharacter (edit)
      - MythologicalFigure (edit)
      - NarutoCharacter (edit)
      - SoapCharacter (edit)
    - Person (edit)
      - Ambassador (edit)
      - Archeologist (edit)
      - Architect (edit)
      - Aristocrat (edit)
      - Artist (edit)
        - Actor (edit)
          - AdultActor (edit)
          - VoiceActor (edit)
        - Comedian (edit)
        - ComicsCreator (edit)
        - Dancer (edit)
        - FashionDesigner (edit)
        - Humorist (edit)
        - MusicalArtist (edit)
          - BackScene (edit)
          - ClassicalMusicArtist
          - Instrumentalist (edit)
            - Guitarist (edit)
          - MusicDirector (edit)
          - Singer (edit)
        - Painter (edit)
        - Photographer (edit)
        - Sculptor (edit)
      - Astronaut (edit)
      - Athlete (edit)
        - ArcherPlayer (edit)
  - Place (edit)
    - ArchitecturalStructure (edit)
      - AmusementParkAttraction (edit)
        - RollerCoaster (edit)
        - WaterRide (edit)
      - Arena (edit)
      - Building (edit)
        - Casino (edit)
        - Castle (edit)
        - Factory (edit)
        - HistoricBuilding (edit)
        - Hospital (edit)
        - Hotel (edit)
        - Museum (edit)
        - Prison (edit)
        - ReligiousBuilding (edit)
          - Abbey (edit)
          - Church (edit)
          - Monastery (edit)
          - Mosque (edit)
          - Shrine (edit)
          - Synagogue (edit)
          - Temple (edit)
        - Restaurant (edit)
        - ShoppingMall (edit)
        - Skyscraper (edit)
      - Gate (edit)
      - Infrastructure (edit)
        - Airport (edit)

**Instances per class**

| Class | Instances |
|---|---|
| Resource (overall) | 4,233,000 |
| Place | 735,000 |
| Person | 1,450,000 |
| Work | 411,000 |
| Species | 251,000 |
| Organisation | 241,000 |

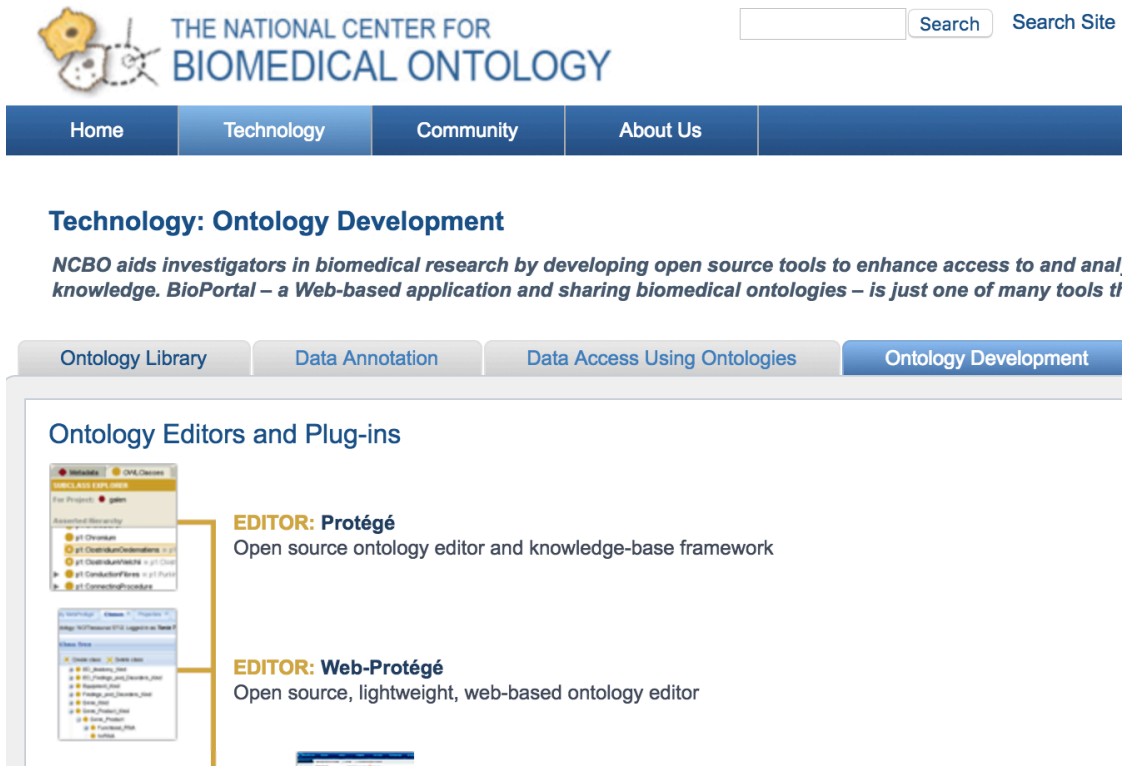
# 已有的Ontology

- Taxon Concept Ontology
- Knowledge Organization System (KOS)
- …

Schema **v101.xsd**

| | |
|---|---|
| schema location: | v101.xsd |
| attribute form default: | **unqualified** |
| element form default: | **qualified** |
| targetNamespace: | **http://www.tdwg.org/schemas/tcs/1.01** |

| Elements | Complex types | Simple types |
|---|---|---|
| DataSet | AccordingToType | NomenclaturalCodesEnum |
| | AgentNames | NomenclaturalTypeStatusOfUnitsEnum |
| | CanonicalAuthorship | TaxonomicRankAboveSuperfamilyEnum |
| | CanonicalName | TaxonomicRankBelowSubspeciesEnum |
| | NameCitation | TaxonomicRankCultivatedPlants |
| | NomenclaturalNoteType | TaxonomicRankEnum |
| | PlaceholderType | TaxonomicRankFamilyGroupEnum |
| | ReferenceType | TaxonomicRankFamilySubdivisionEnum |
| | RelationshipType | TaxonomicRankGenusGroupEnum |
| | ScientificName | TaxonomicRankGenusSubdivisionEnum |
| | TaxonomicRank | TaxonomicRankSpeciesGroupEnum |



# Ontology定义不统一

| | | | |
|---|---|---|---|
| 中文名: | 李娜 | 籍贯: | 武汉市 |
| 性别: | 女 | 民族: | 汉族 |
| 国籍: | 中国 | 出生年月: | 1982年2月26日 |
| 星座: | 双鱼座 | 职业: | 运动员 女子网球选手 |
| 毕业院校: | 华中科技大学 | 身高: | 172厘米 |

互动百科

属性定义不统一

| | | | |
|---|---|---|---|
| 中文名 | 李娜 | 主要奖项 | |
| 外文名 | Li Na | 重要事件 | |
| 别　名 | 娜姐 | | |
| 国　籍 | 中国 | | |
| 民　族 | 汉 | 启蒙教练 | |
| 出生地 | 湖北省武汉市江岸区 | 训练地 | |
| 出生日期 | 1982年2月26日 | 教　练 | |
| 毕业院校 | 华中科技大学（新闻学专业） | 丈　夫 | |

百度百科

**人物**
政治人物　话题人物
历史人物
文化人物
虚拟人物
经济人物

**自然**
动物
植物
自然灾害
自然资源
自然现象

**文化**
美术　书画
戏剧　建筑
舞蹈　语言
摄影
曲艺

**历史**
各国历史
历史事件
历史著作
文物考古

**地理**
行政区划
地形地貌

**科技**
科研机构
互联网
航空航天
医学
电子产品

分类体系不统一

– 人物 收起
+ 政治人物 展开
+ 中国摇滚歌手 展开
+ 热点人物 展开
+ 娱乐人物 展开
+ 中国手球运动员 展开
+ 人物传记 展开
+ 自然科学人物 展开
+ 社会科学人物 展开
+ 古代美女 展开
+ 网络美女 展开
+ 女优 展开
+ 财经人物 展开
+ 性感美女 展开
+ 经济人物 展开
+ 虚拟人物 展开
+ 公益人物 展开
+ 人物励志图书 展开
+ 体育人物 展开
+ 话题人物 展开
+ 文化人物 展开
+ 香港TVB女演员 展开
+ IT风云人物 展开
+ 中国政治犯 展开
+ 海聚工程 展开
+ 人物野史 展开
+ 互动百科智愿者拾贝 展开

# 统一标准：Schema.org

schema.org

## Organization of Schemas

The schemas are a set of 'types', each associated with a set of properties. The types are arranged in a hierarchy. The core vocabulary currently consists of 597 Types, 867 Properties, and 114 Enumeration values.

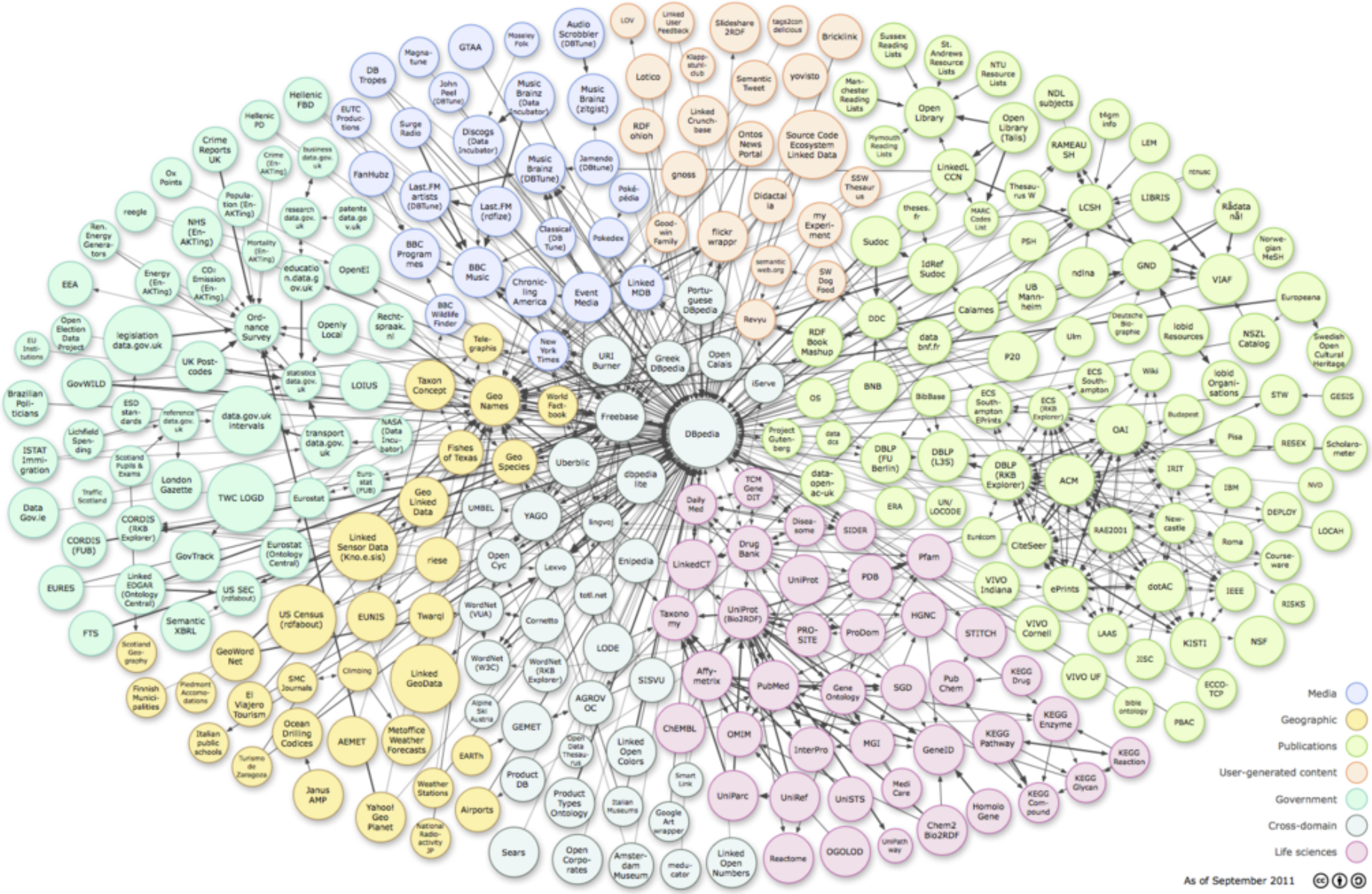
Browse the full hierarchy:

- One page per type
- Full list of types, shown on one page

Or you can jump directly to a commonly used type:

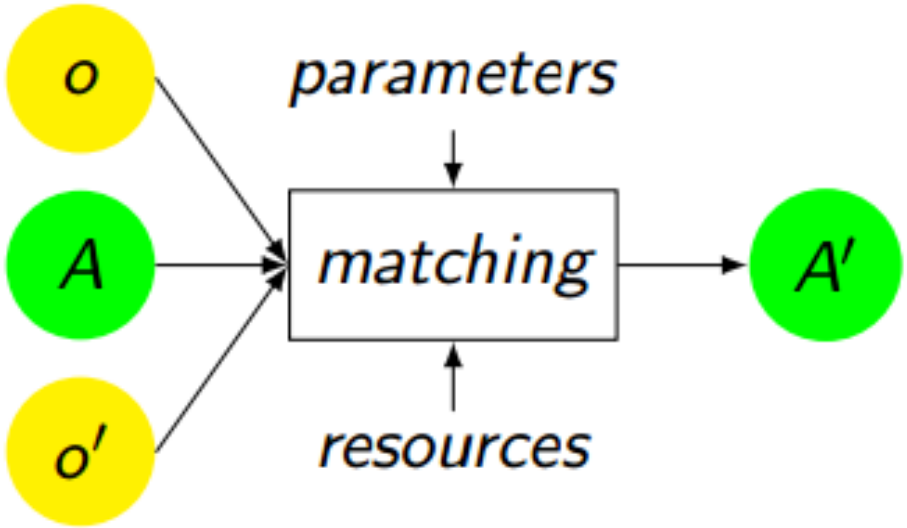
- Creative works: CreativeWork, Book, Movie, MusicRecording, Recipe, TVSeries ...
- Embedded non-text objects: AudioObject, ImageObject, VideoObject
- Event
- Health and medical types: notes on the health and medical types under MedicalEntity.
- Organization
- Person
- Place, LocalBusiness, Restaurant ...
- Product, Offer, AggregateOffer
- Review, AggregateRating
- Action

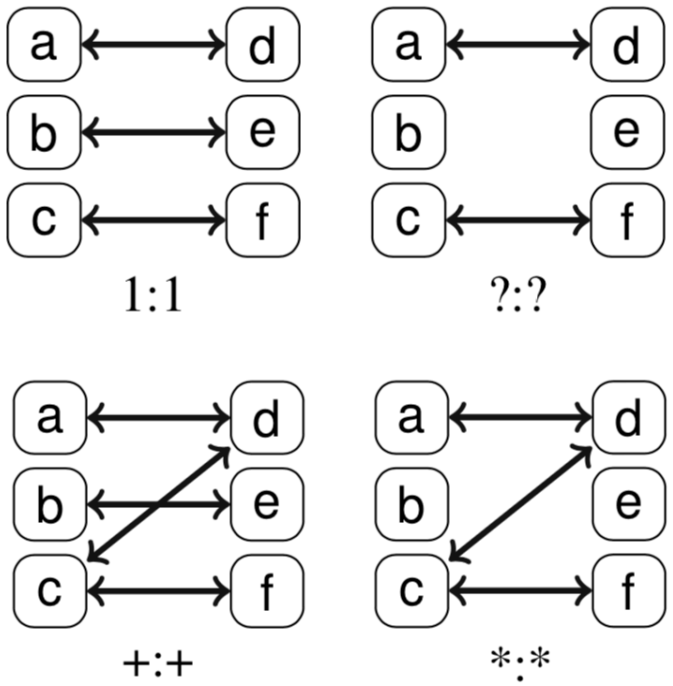
# Ontology Matching



# Ontology Matching

- 任务定义
  - 给定两个本体 $o$ 和 $o'$，$e$ 和 $e'$ 分别是两个本体中元素（实体，关系），本体对齐的目标是建立三元组 $< e, e', r >$ 为两个本体中元素之间的关系，例如： $=$ $\subseteq$ $\supseteq$ $\perp$

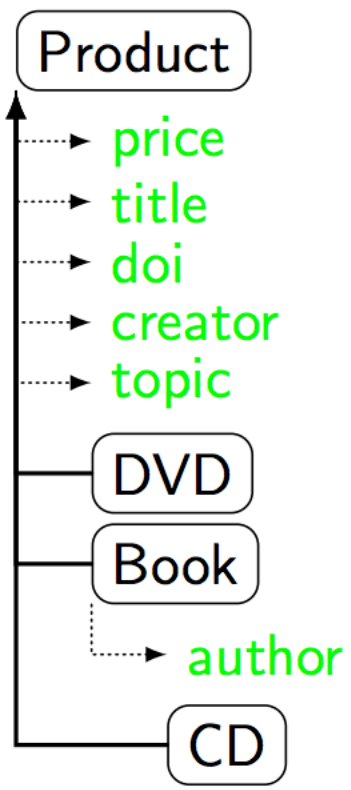


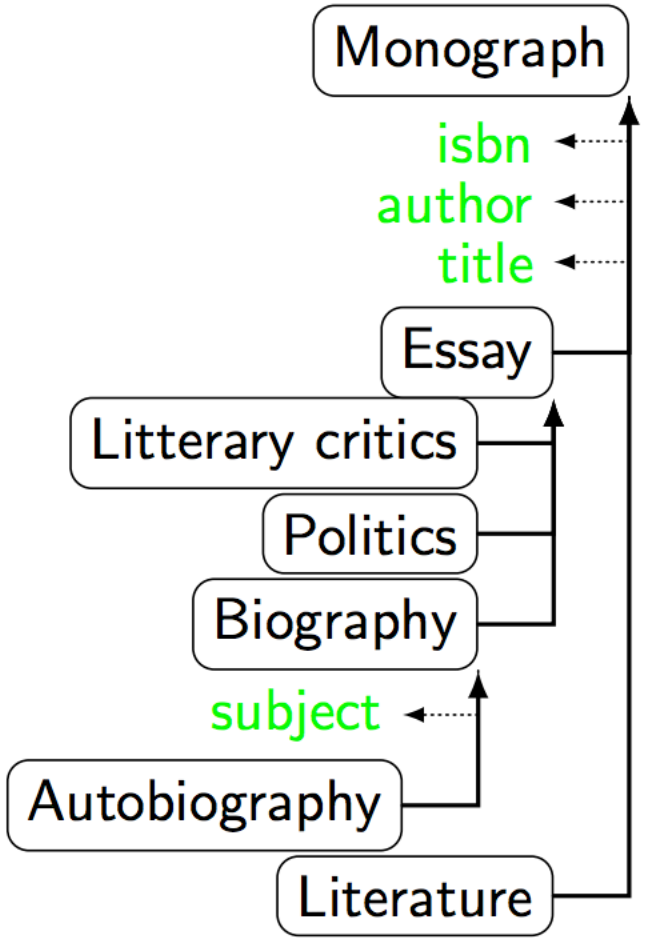



# 例子

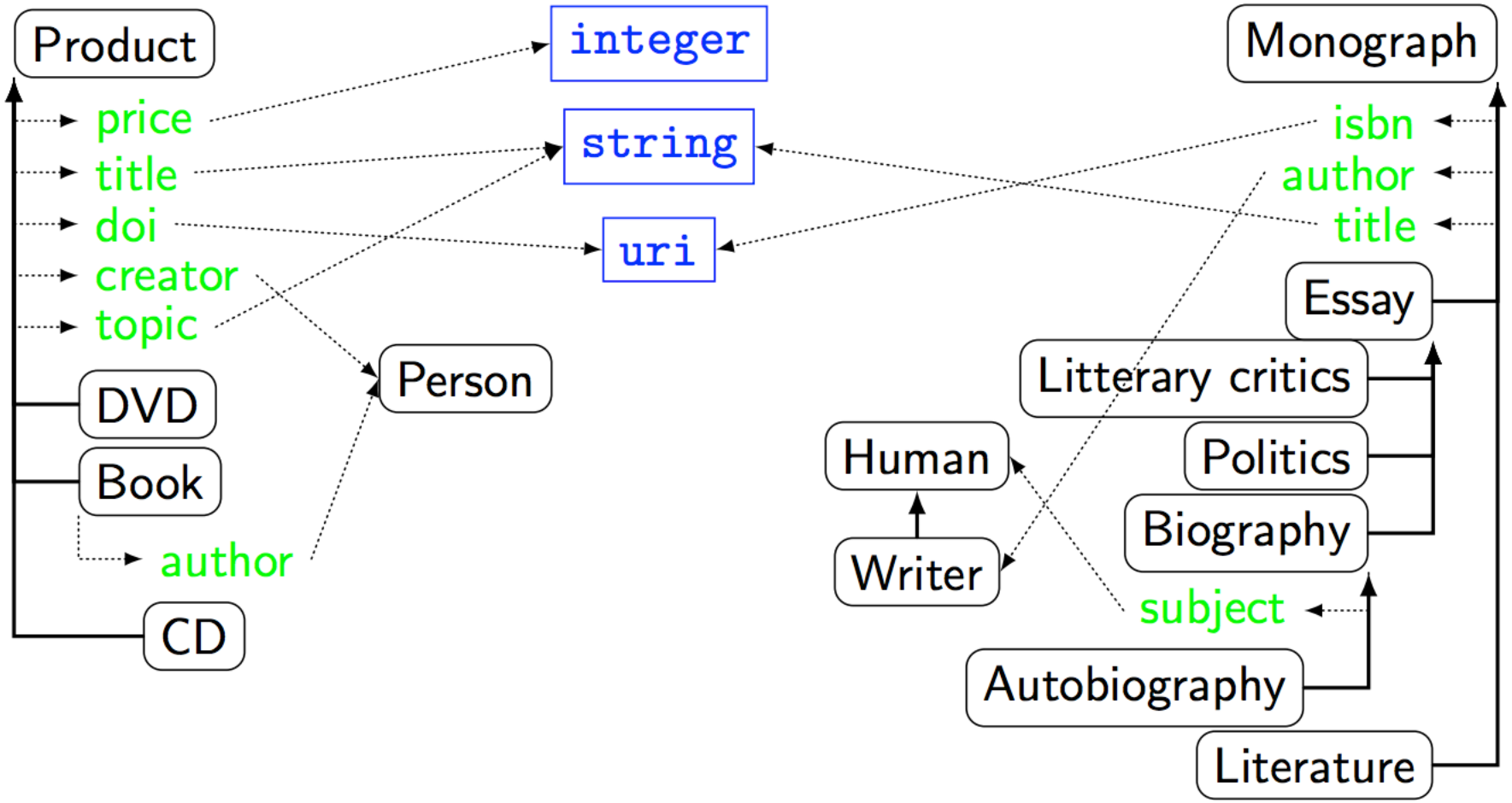
Product
DVD
Book
CD
Monograph
Essay
Litterary critics
Politics
Biography
Autobiography
Literature

# 例子

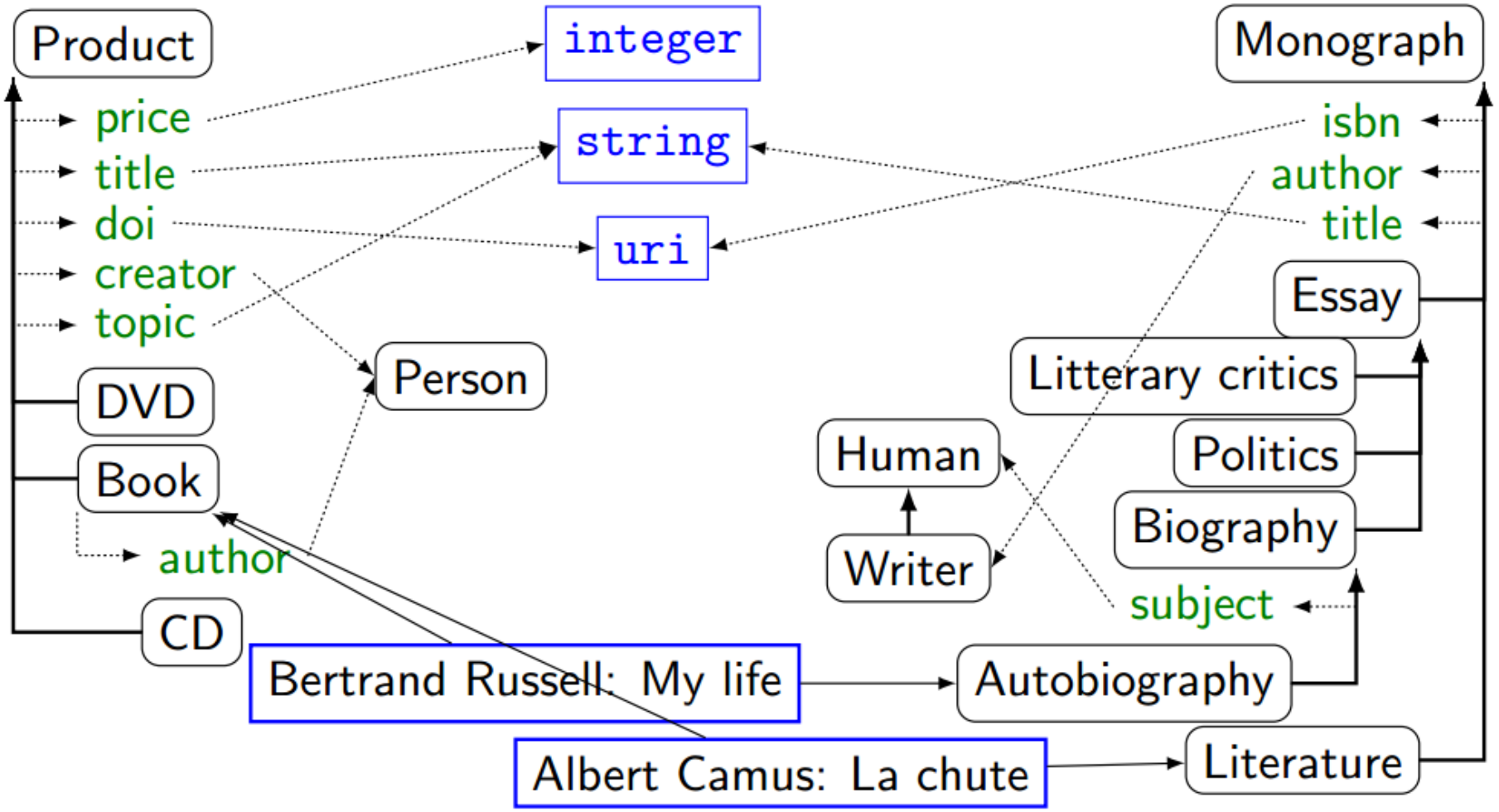

Product
price
title
doi
creator
topic
DVD
Book
author
CD


Monograph
isbn
author
title
Essay
Litterary critics
Politics
Biography
subject
Autobiography
Literature

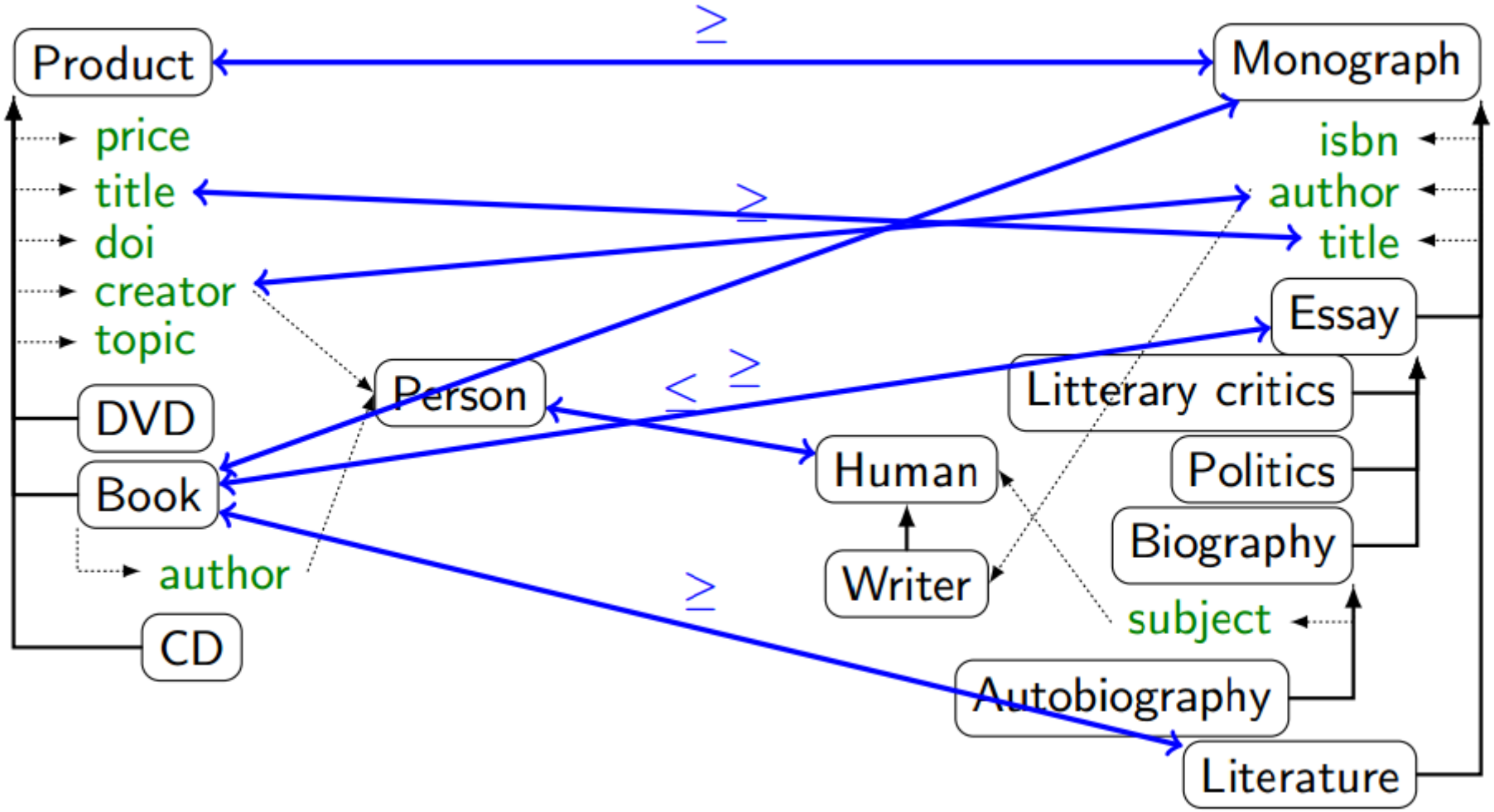
# 例子


Product
price
title
doi
creator
topic
DVD
Book
author
CD
integer
string
uri
Person
Monograph
isbn
author
title
Essay
Litterary critics
Politics
Biography
Human
Writer
subject
Autobiography
Literature

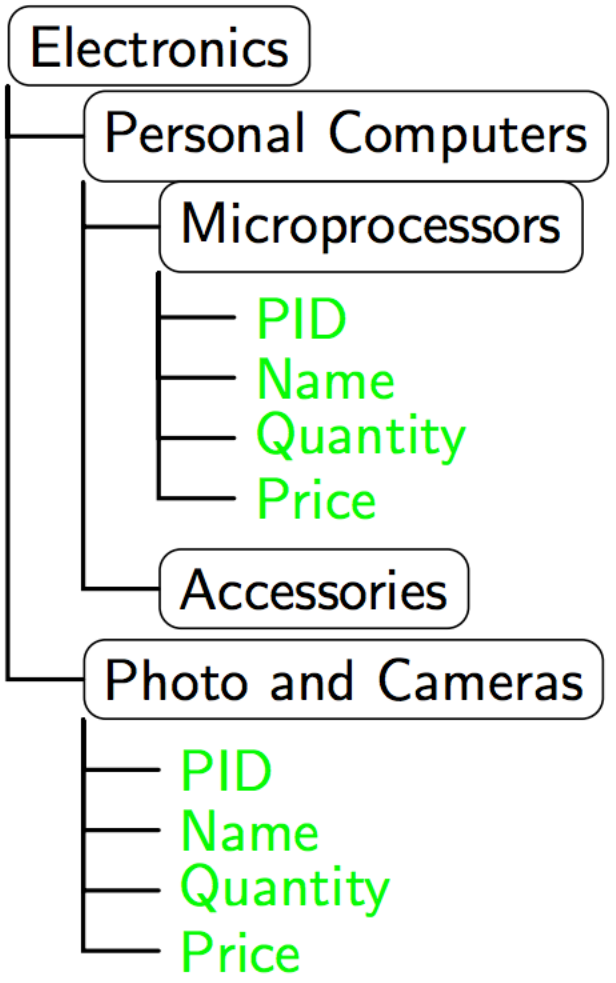
# 例子

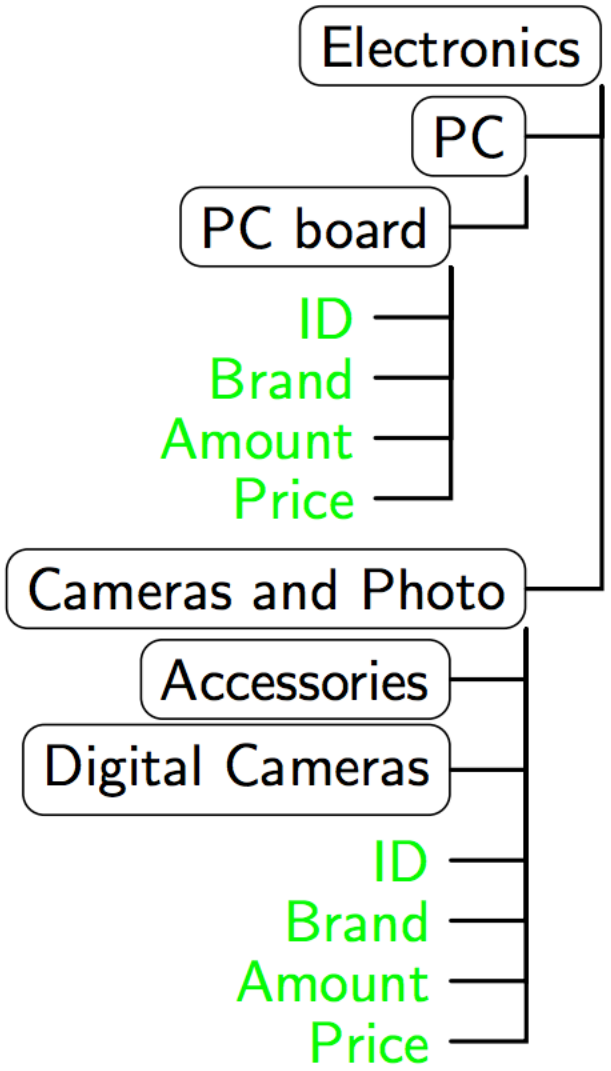

Product
price
title
doi
creator
topic
DVD
Book
author
CD
integer
string
uri
Person
Bertrand Russell: My life
Albert Camus: La chute
Monograph
isbn
author
title
Essay
Litterary critics
Politics
Biography
Human
Writer
subject
Autobiography
Literature

# 例子

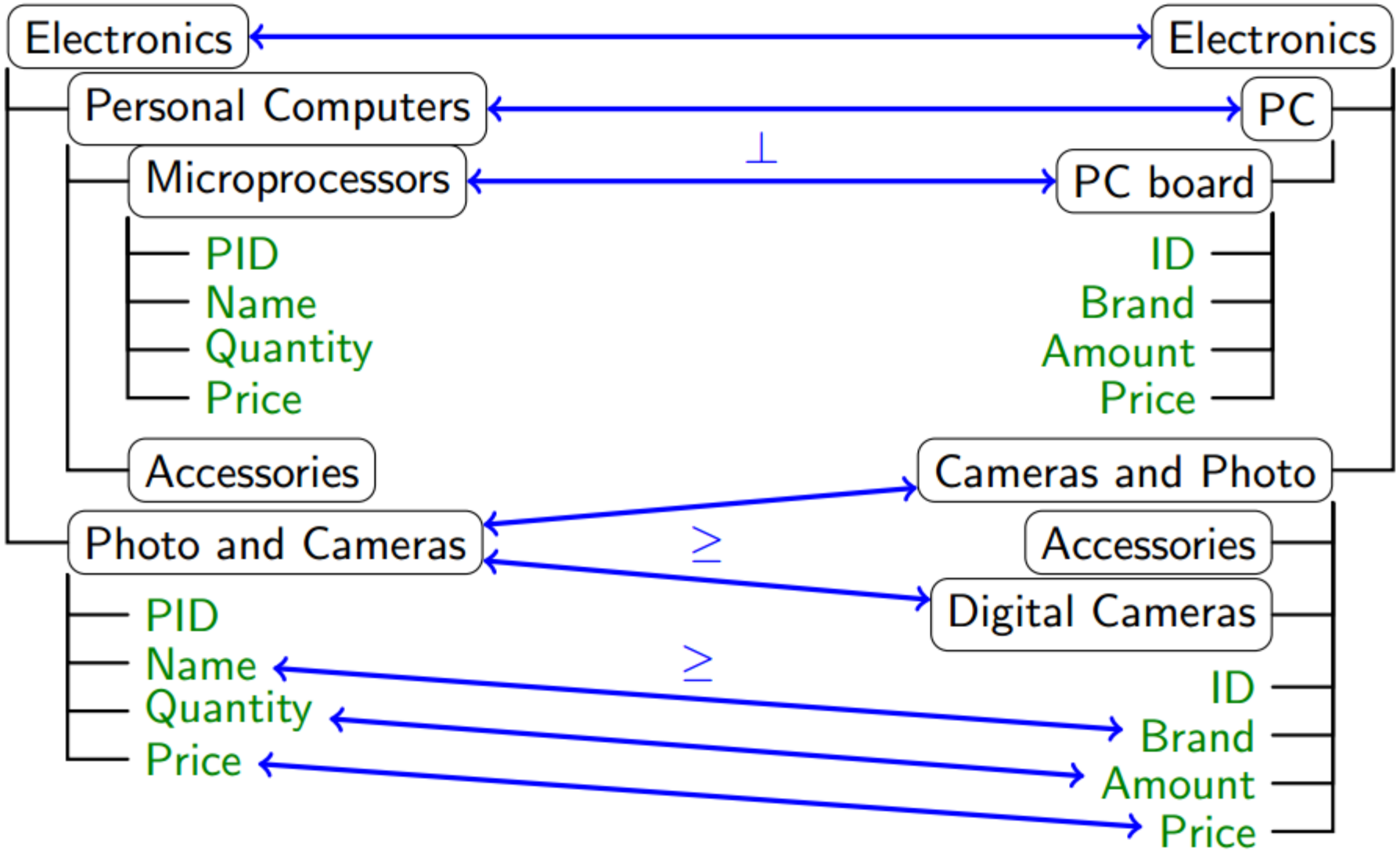

Product
Monograph
price
title
doi
creator
topic
isbn
author
title
Essay
Person
Litterary critics
DVD
Book
Human
Politics
Biography
author
Writer
subject
CD
Autobiography
Literature

# 例子：Two Schema

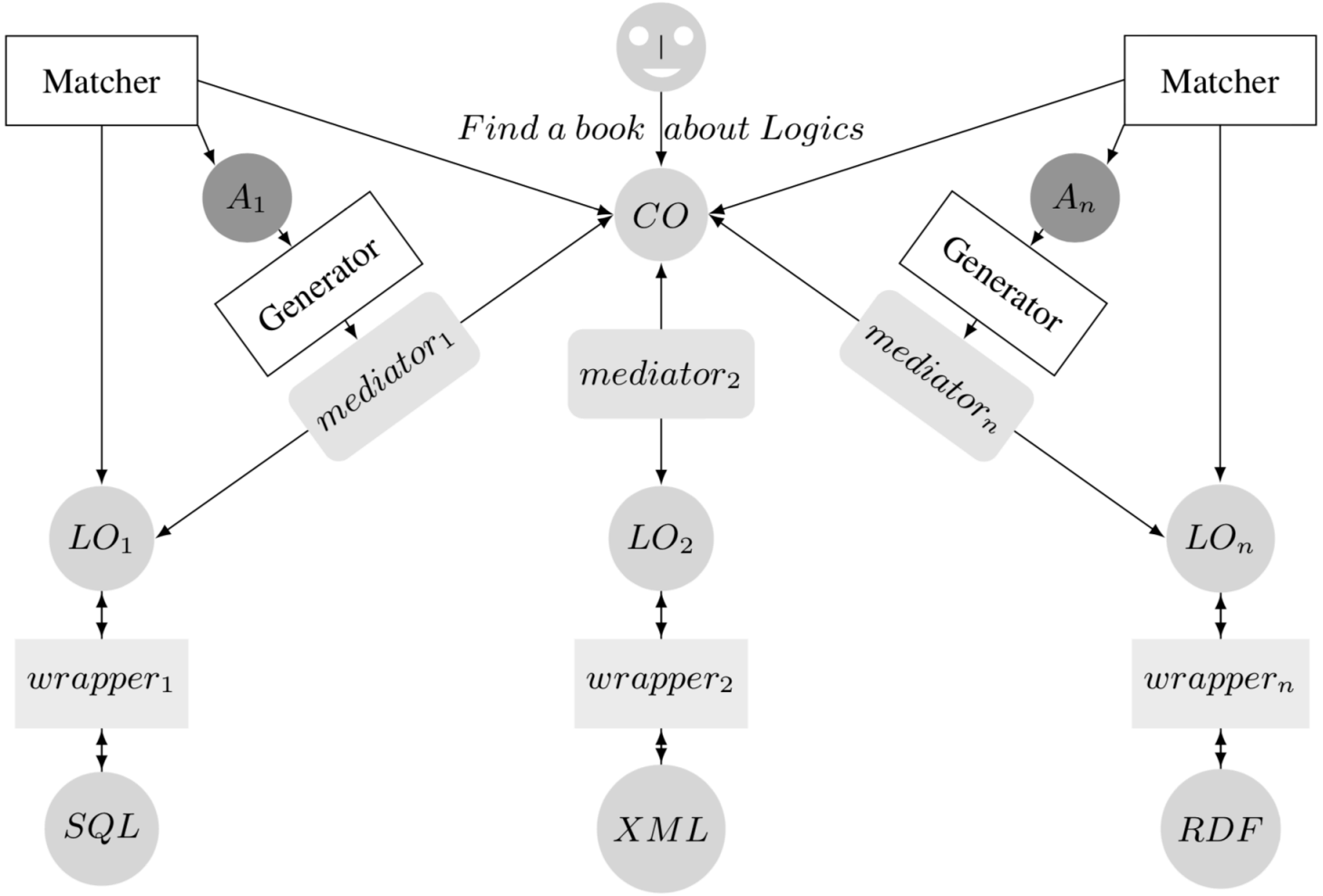


# 例子：Two Schema



# 应用：信息融合




| | | |
|---|---|---|
| 中文名 | 李娜 | 主要奖项 |
| 外文名 | Li Na | 重要事件 |
| 别　名 | 娜姐 | |
| 国　籍 | 中国 | |
| 民　族 | 汉 | 启蒙教练 |
| 出生地 | 湖北省武汉市江岸区 | 训练地 |
| 出生日期 | 1982年2月26日 | 教　练 |
| 毕业院校 | 华中科技大学（新闻学专业） | 丈　夫 |

百度百科

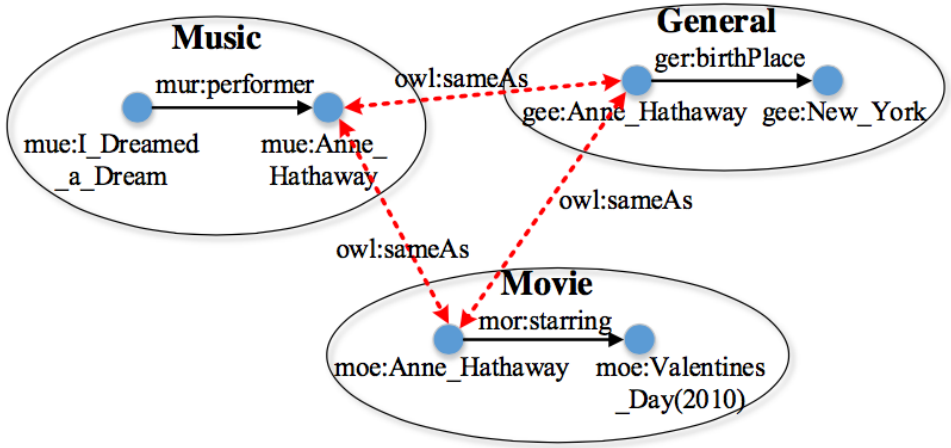
| | | | |
|---|---|---|---|
| 中文名: | 李娜 | 籍贯: | 武汉市 |
| 性别: | 女 | 民族: | 汉族 |
| 国籍: | 中国 | 出生年月: | 1982年2月26日 |
| 星座: | 双鱼座 | 职业: | 运动员 女子网球选手 |
| 毕业院校: | 华中科技大学 | 身高: | 172厘米 |

互动百科

# 应用：多知识库问答

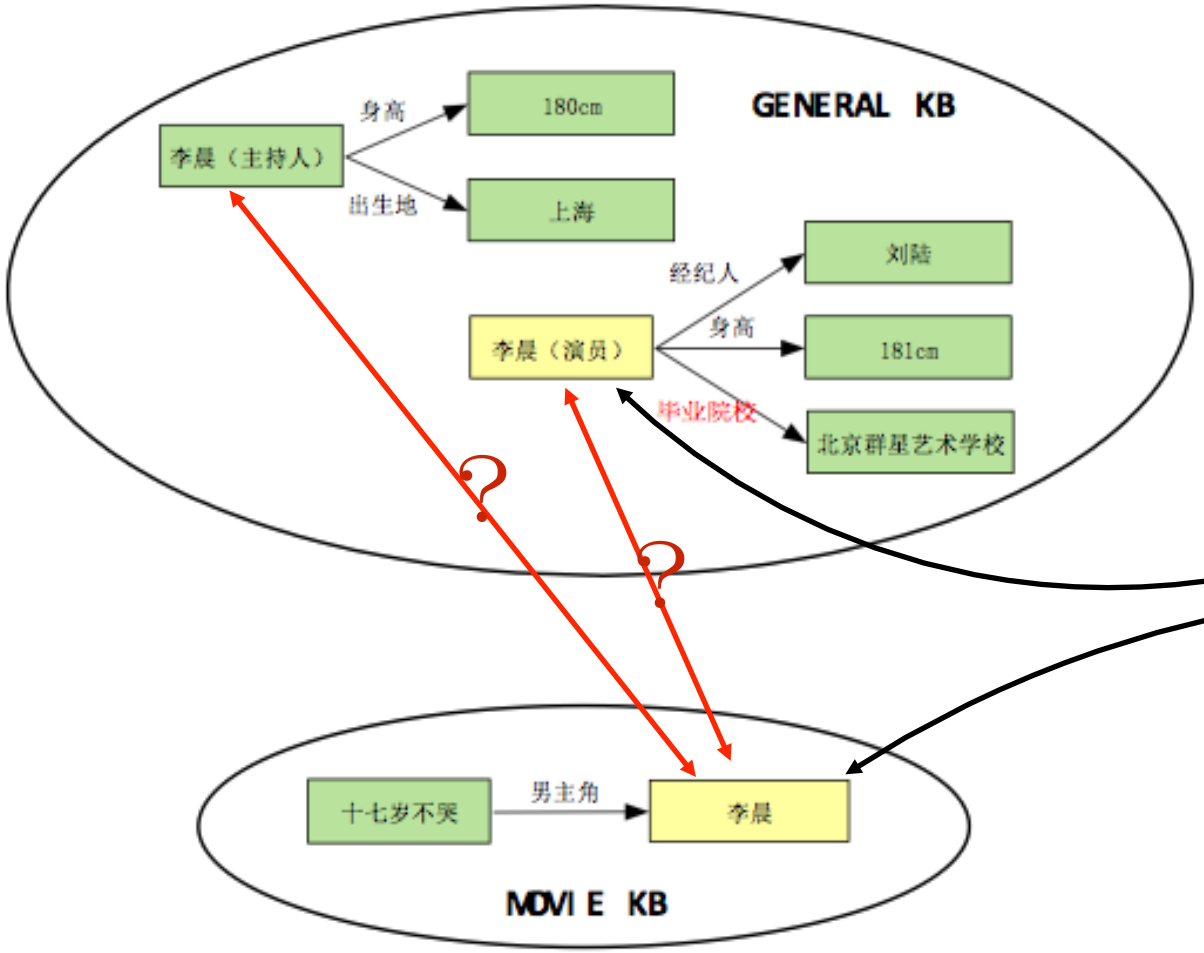
Which song are performed by person who was born in New York and played a role in Valentine's Day

music KB　　people KB　　movie KB




SELECT ?v1 WHERE {
⟨?v1, mur:perfomer[1], ?v2⟩
⟨?v2, owl:sameAs, ?v3⟩
⟨?v3, mor:starring, moe:Valentines_Day(2010)⟩
⟨?v3, owl:sameAs, ?v4⟩
⟨?v4, ger:birthPlace, gee:New_York⟩ }

# 应用：多知识库问答




北京群星艺术学校毕业并且演过十七岁不哭的男演员身高有多少？

<x, dbr:毕业院校, dbo:北京群星艺术学校>

<x, dbr:主演, dbo:十七岁不哭>

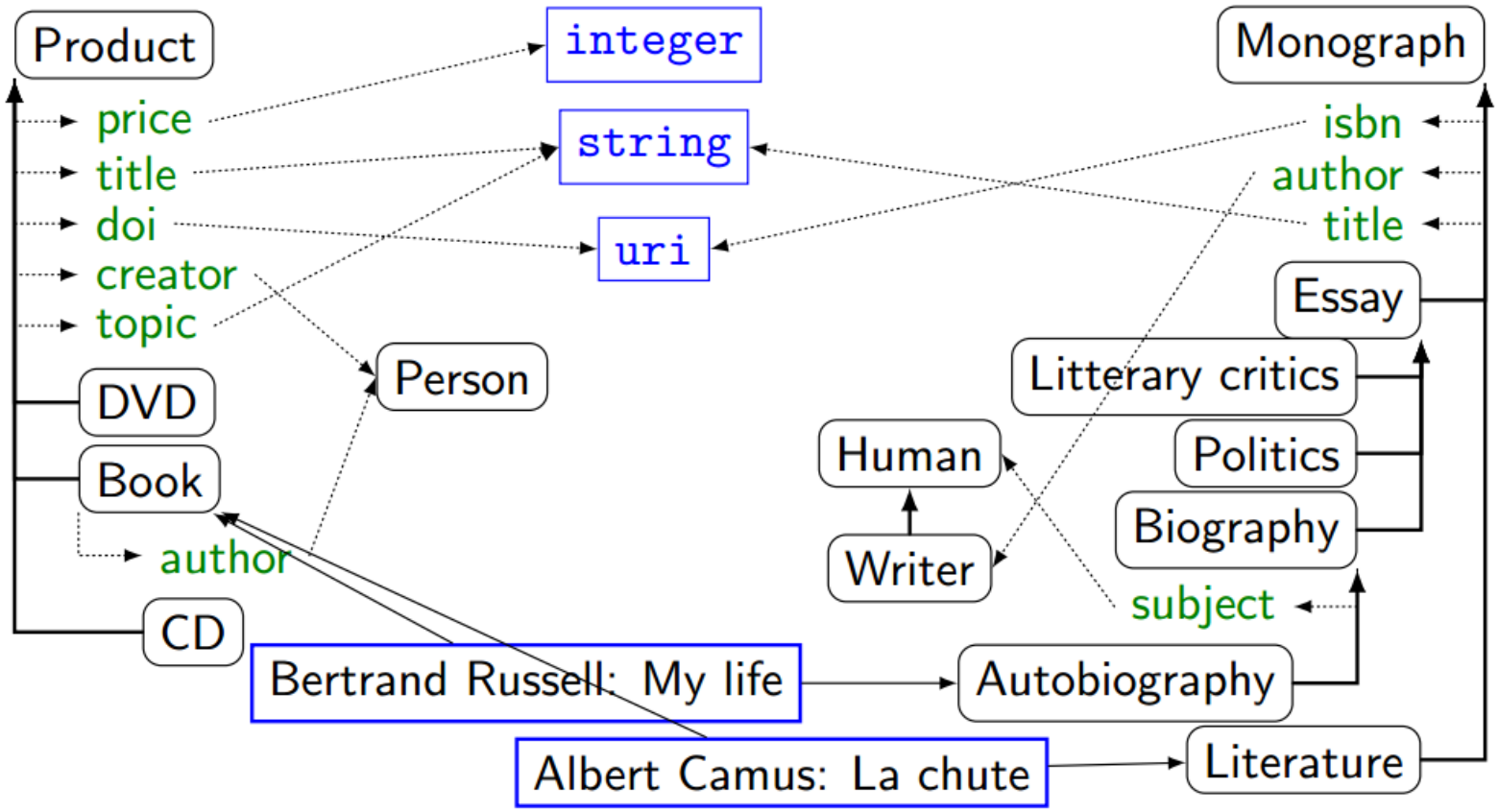
<x, dbr:身高, y？>

# 评测：OAEI (Ontology Alignment Evaluation Initiative)

2017 OAEI–2017 at the ISWC ontology matching workshop, Wien
2016 OAEI–2016 at the ISWC ontology matching workshop, Kobe
2015 OAEI–2015 at the ISWC ontology matching workshop, Bethlehem
2014 OAEI–2014 at the ISWC ontology matching workshop, Riva del Garda
2013 OAEI–2013 at the ISWC ontology matching workshop, Sydney
2012 OAEI–2012 at the ISWC ontology matching workshop, Boston
2012 OAEI–2011.5 at ESWC 2012 iWEST workshop, Hersounissos
2011 OAEI–2011 at the ISWC ontology matching workshop, Bonn
2010 OAEI–2010 at the ISWC ontology matching workshop and IWEST workshop, Shanghai
2009 OAEI–2009 at the ISWC ontology matching workshop, Fairfax
2008 OAEI–2008 at the ISWC ontology matching workshop, Karlsruhe
2007 OAEI–2007 at the ISWC ontology matching workshop, Busan
2006 OAEI–2006 at the ISWC ontology matching workshop, Athens
2005 OAEI–2005 at the K–Cap workshop on integrating ontologies, Banff
2004 EON 2004 Ontology alignment contest, Hiroshima
2004 ParMIS 2004 I3CON, Gaithersburg

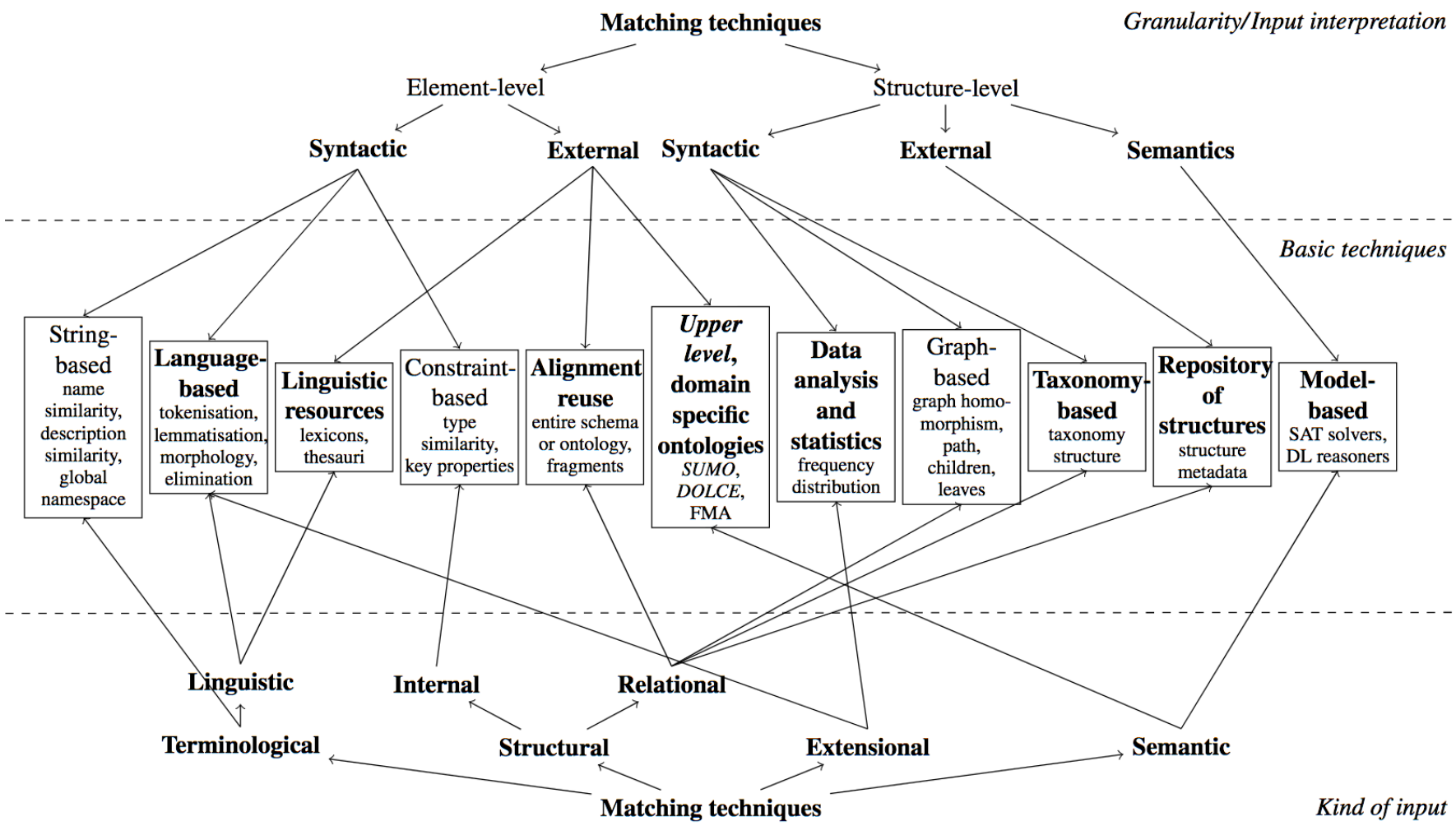
| Dataset | Formalism | Relations | Confidence | Modalities | Language |
|---|---|---|---|---|---|
| benchmarks | OWL | = | [0 1] | open | EN |
| anatomy | OWL | = | [0 1] | blind | EN |
| conference | OWL-DL | =, <= | [0 1] | blind+open | EN |
| directory | OWL | =, <, >, ⊥ | 1 | blind+open | EN |
| library | SKOS +OWL | exact-,narrow-, broadMatch | 1 | blind | EN+NL+FR |
| benchmarksubs | OWL | =,<,> | [0 1] | open | EN |
| ars | RDF | = | [0 1] | open | EN |
| tap | RDF | = | [0 1] | open | EN |
| iimb | RDF | = | [0 1] | open | EN |
| vlcr | SKOS +OWL | exact-, closeMatch | [0 1] | blind expert | NL+EN |

# 方法




核心问题：语义匹配

- 字符
- 结构
- 语义


Matching techniques
Granularity/Input interpretation
Element-level
Structure-level
Syntactic
External
Syntactic
External
Semantics
Basic techniques
String-based
name similarity, description similarity, global namespace
Language-based
tokenisation, lemmatisation, morphology, elimination
Linguistic resources
lexicons, thesauri
Constraint-based
type similarity, key properties
Alignment reuse
entire schema or ontology, fragments
Upper level, domain specific ontologies
SUMO, DOLCE, FMA
Data analysis and statistics
frequency distribution
Graph-based
graph homo-morphism, path, children, leaves
Taxonomy-based
taxonomy structure
Repository of structures
structure metadata
Model-based
SAT solvers, DL reasoners
Linguistic
Internal
Relational
Terminological
Structural
Extensional
Semantic
Matching techniques
Kind of input

# String-based Methods

- String equality
  - Hamming Distance
- Substring test
  - Substring Similarity
    - Prefix: net=network; hot=hotel
    - Suffix: ID = PID; word=sword
  - N-gram Similarity
- Edit Distance
- Token based Distance

$$\delta(s,t) = \frac{\left(\sum_{i=1}^{\min(|s|,|t|)} s[i] \neq t[i]\right) + ||s| - |t||}{\max(|s|,|t|)}$$

$$\sigma(x,y) = \frac{2|t|}{|x| + |y|}$$

$$\overline{\sigma}(s,t) = \frac{|ngram(s,n) \cap ngram(t,n)|}{\min(|s|,|t|) - n + 1}$$

$$\delta(s,t) = \min_{(op_i)_I; op_n(\ldots op_1(s))=t} \left(\sum_{i \in I} w_{op_i}\right)$$

$$\sigma_V(s,t) = \frac{\sum_{i \in |V|} \overrightarrow{s}_i \times \overrightarrow{t}_i}{\sqrt{\sum_{i \in |V|} \overrightarrow{s}_i^2 \times \sum_{i \in |V|} \overrightarrow{t}_i^2}}$$

# Language based Methods

- Linguistic Normalization
  - Tokenization
    - Hands-Free_Kits ⇨ <hands, free, kits>
  - Lemmatization
    - Kits ⇨ Kit
  - Elimination
    - Stop words list: a, the, by, type of, their, from

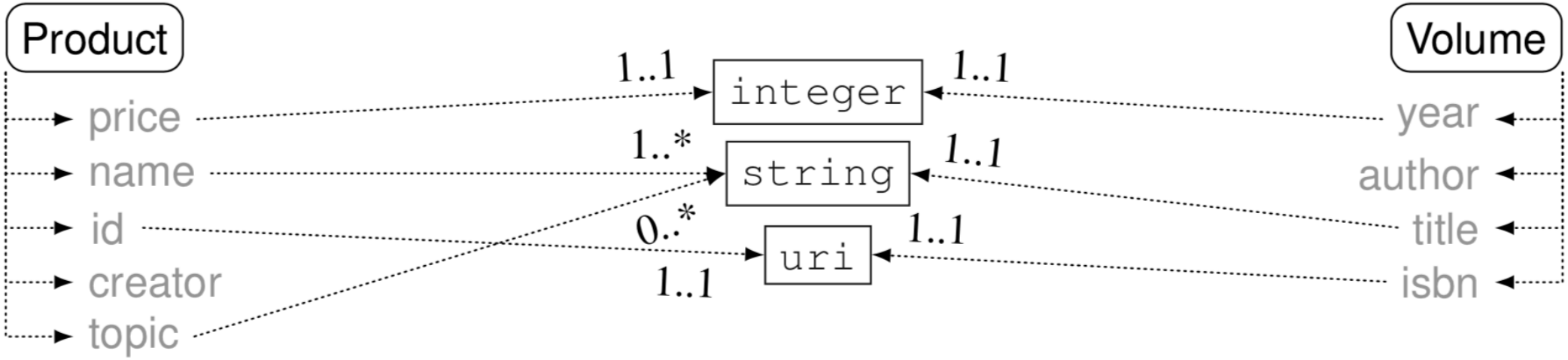
# Linguistic resources

- Sense-based: WordNet
  - A ⊑ B if A is a hyponym or meronym of B
    - Brand ⊑ Name
  - A ⊒ B if A is a hypernym or holonym of B
    - Europe ⊒ Greece
  - A = B if they are synonyms
    - Quantity = Amount
  - A ⊥ B if they are antonyms or the siblings in the part of hierarchy
    - Microprocessors ⊥ PC Board

$$\sigma(s,t) = \frac{|\Sigma(s) \cap \Sigma(t)|}{|\Sigma(s) \cup \Sigma(t)|}$$

# Constraint-based methods

- 利用internal structures (property 信息 )
  - property
  - datatype
  - domain
  - …

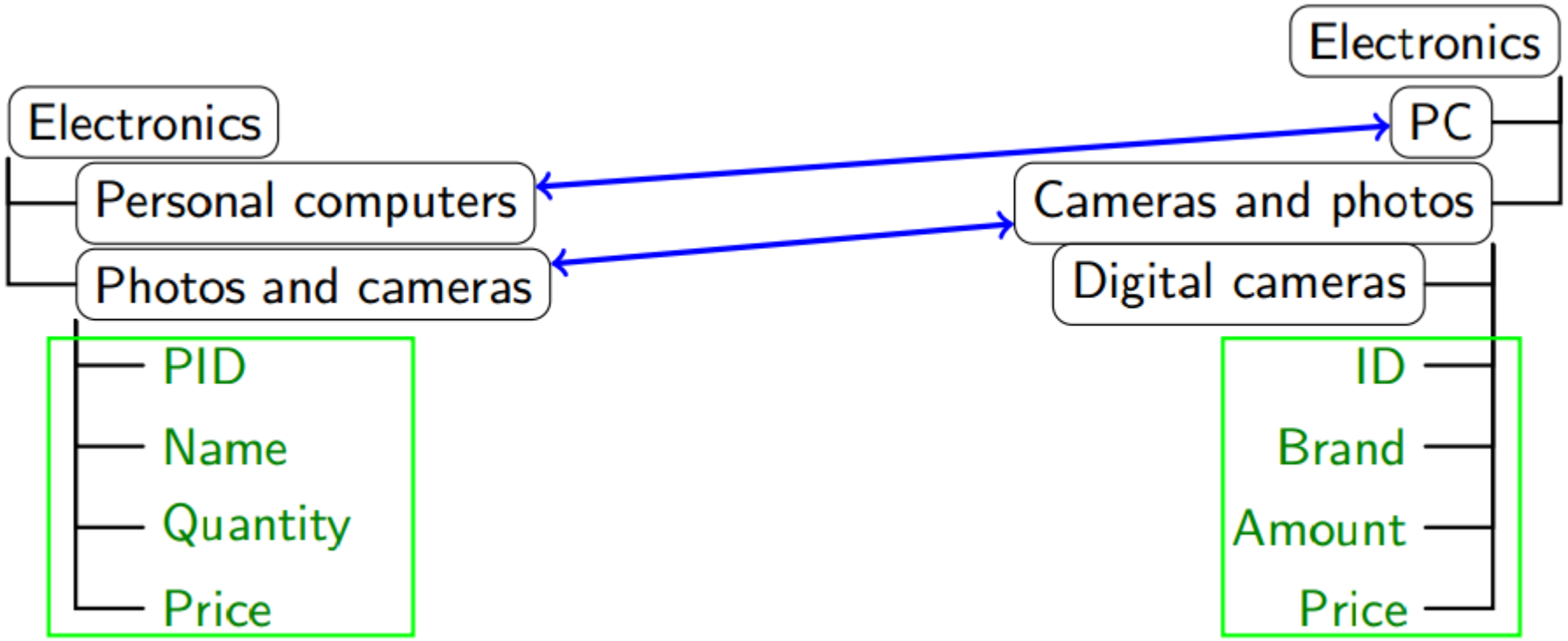


# Graph-based techniques

- 基本假设
  - If the neighbors of two nodes of the two ontologies are similar, they will be more similar.

# Taxonomy-based techniques

- 如果两个非叶节点的孩子节点是可以对齐的，那么这两个非叶节点也应该对齐
- 如果两个非叶节点的孩子节点(叶节点)是可以对齐的，即使他们的孩子节点（非叶结点）不相似，那么这两个非叶节点也应该对齐

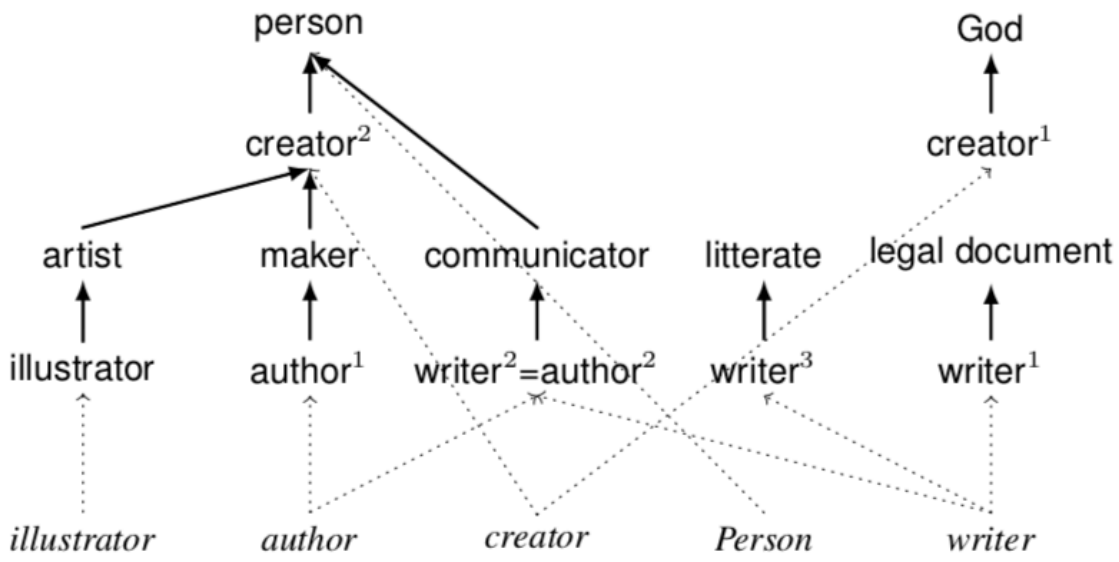


# 相似度计算方法

- Structural topological dissimilarity

$$\forall e, e' \in o, \delta(e, e') = \min_{c \in o}[\delta(e, c) + \delta(e', c)]$$

$\delta(e, c)$ is the number of intermediate edges between e and



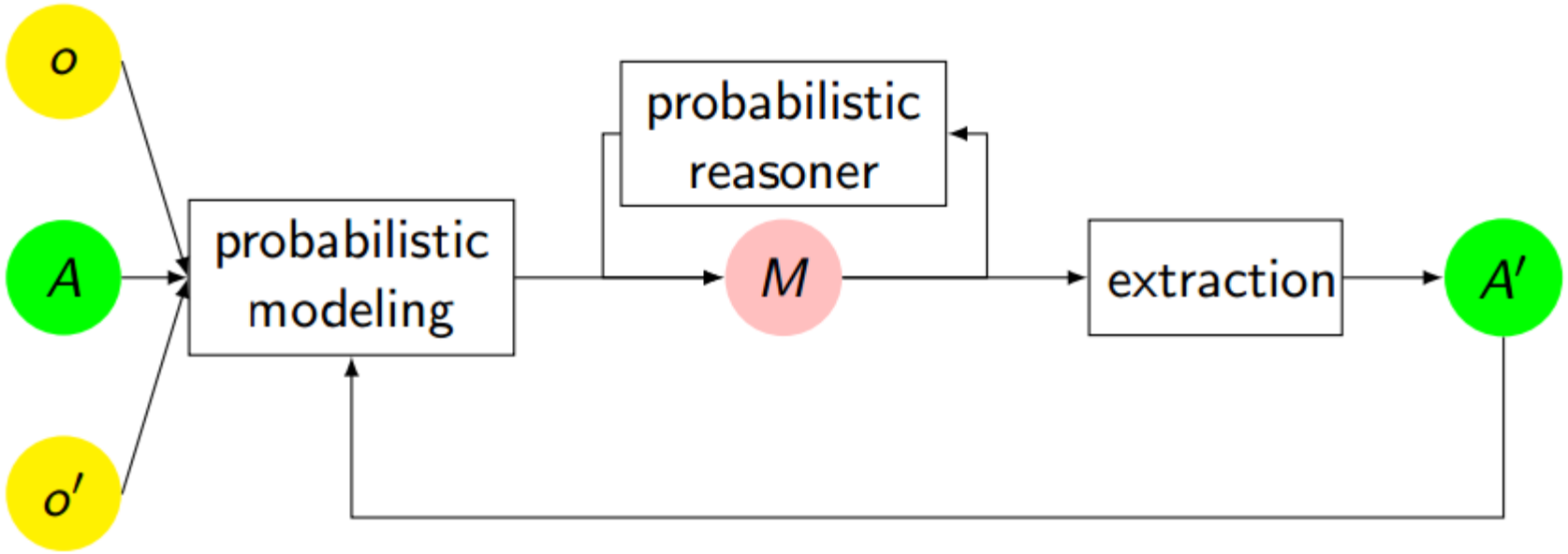

- Upward co-topic similarity

The set of superclasses of c

$$\sigma(c, c') = \frac{|UC(c, H) \cap UC(c', H)|}{|UC(c, H) \cup UC(c', H)|}$$

# Model-based techniques

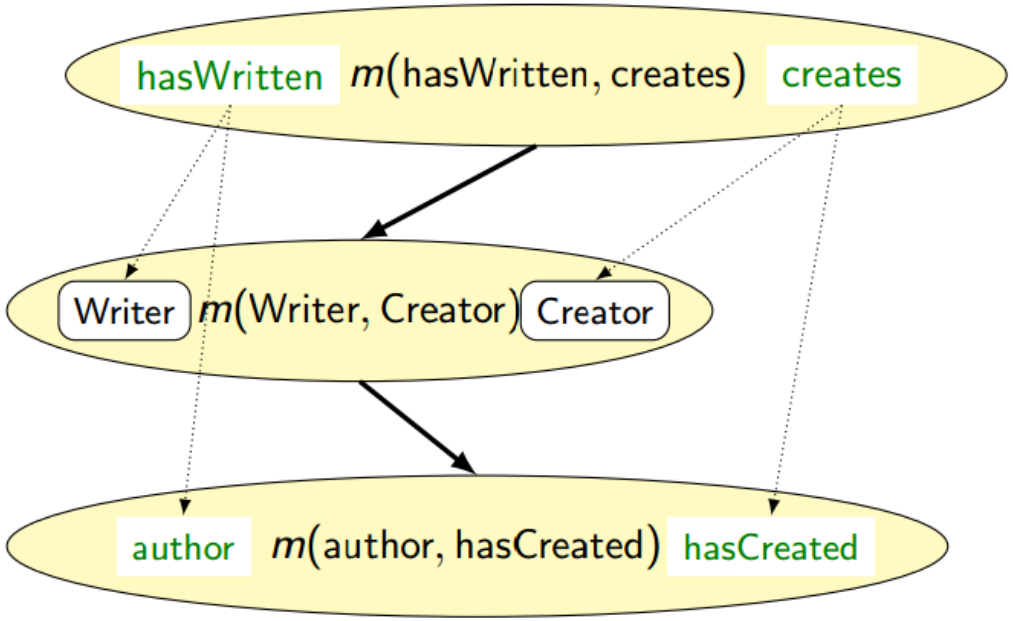
- Probabilistic Matching



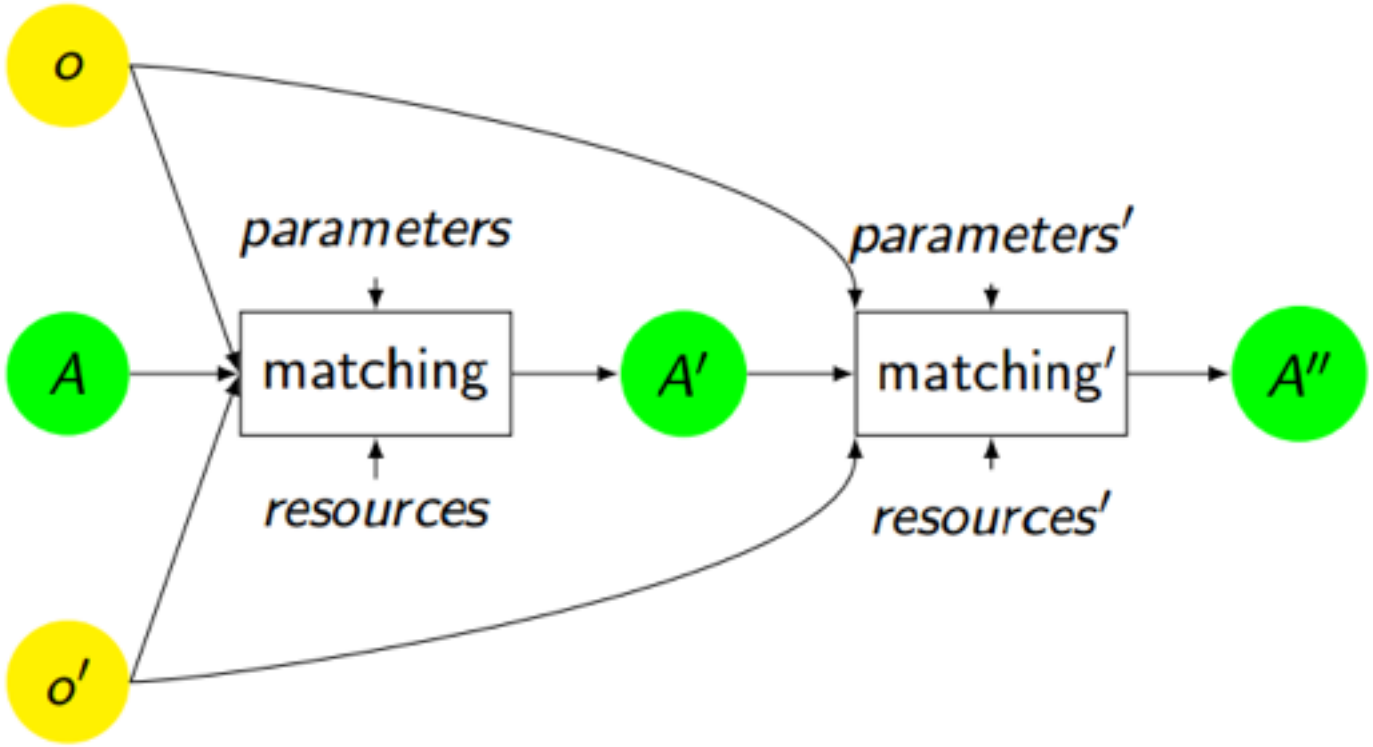
# Global Methods: Bayesian networks

- 全局寻优、联合推断：贝叶斯网

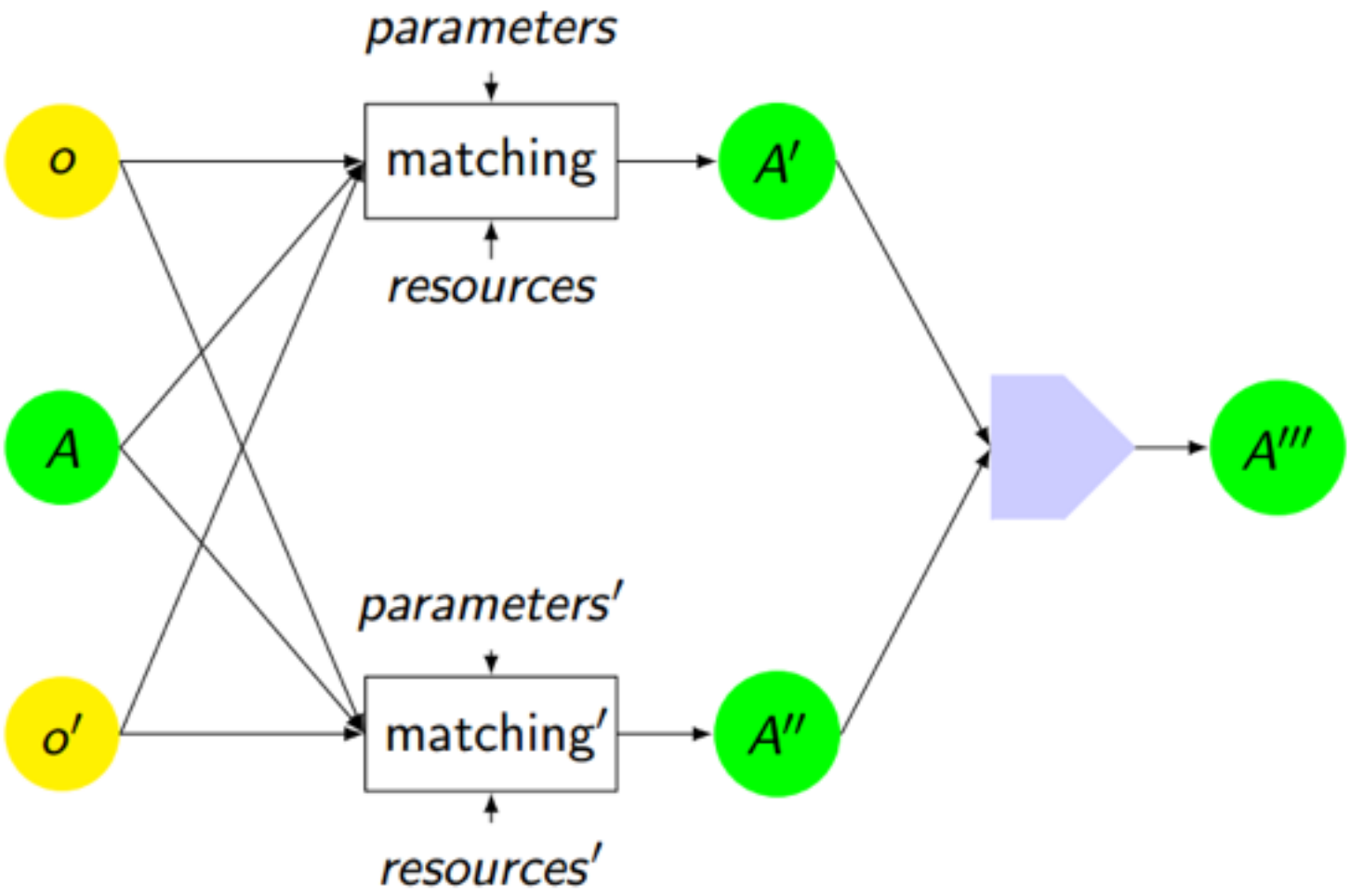
$$P(X_1, \ldots, X_n) = \prod_i P(X_i | parents(X_i)), i = 1, \ldots, n$$



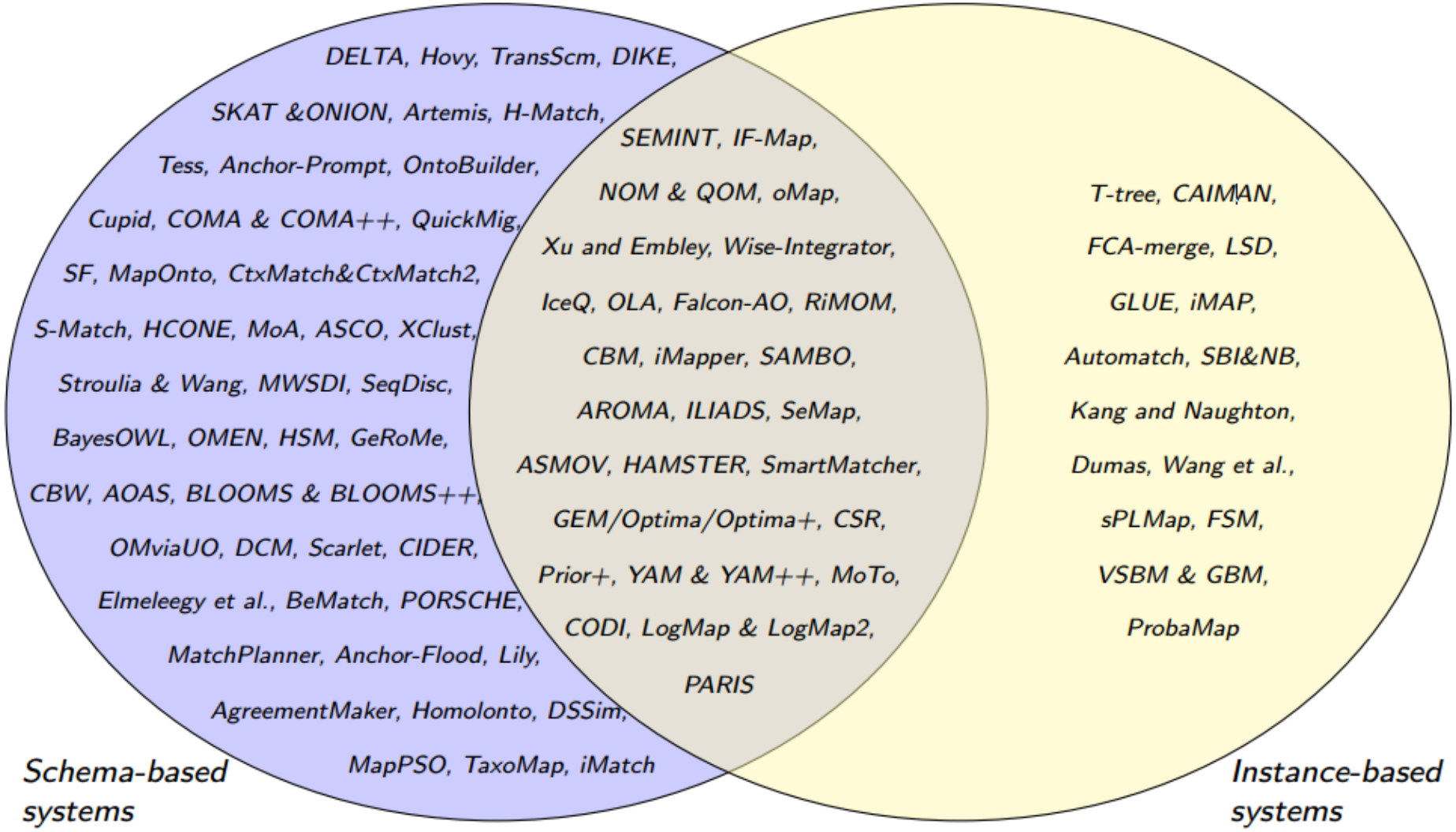
# Sequential composition



# Parallel composition



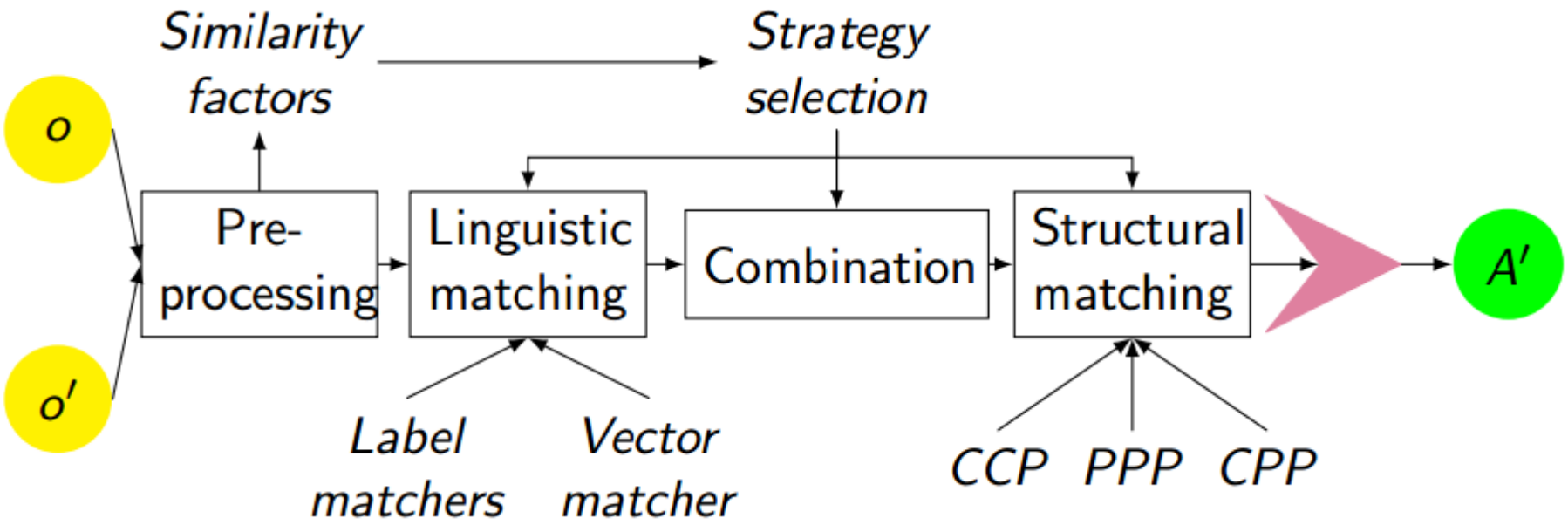
# 已有的系统


DELTA, Hovy, TransScm, DIKE,
SKAT &ONION, Artemis, H-Match,
Tess, Anchor-Prompt, OntoBuilder,
Cupid, COMA & COMA++, QuickMig,
SF, MapOnto, CtxMatch&CtxMatch2,
S-Match, HCONE, MoA, ASCO, XClust,
Stroulia & Wang, MWSDI, SeqDisc,
BayesOWL, OMEN, HSM, GeRoMe,
CBW, AOAS, BLOOMS & BLOOMS++,
OMviaUO, DCM, Scarlet, CIDER,
Elmeleegy et al., BeMatch, PORSCHE,
MatchPlanner, Anchor-Flood, Lily,
AgreementMaker, Homolonto, DSSim,
MapPSO, TaxoMap, iMatch
SEMINT, IF-Map,
NOM & QOM, oMap,
Xu and Embley, Wise-Integrator,
IceQ, OLA, Falcon-AO, RiMOM,
CBM, iMapper, SAMBO,
AROMA, ILIADS, SeMap,
ASMOV, HAMSTER, SmartMatcher,
GEM/Optima/Optima+, CSR,
Prior+, YAM & YAM++, MoTo,
CODI, LogMap & LogMap2,
PARIS
T-tree, CAIMAN,
FCA-merge, LSD,
GLUE, iMAP,
Automatch, SBI&NB,
Kang and Naughton,
Dumas, Wang et al.,
sPLMap, FSM,
VSBM & GBM,
ProbaMap
Schema-based systems
Instance-based systems

# 已有的系统

- Cupid (U. of Washington, Microsoft Corporation and U. of Leipzig)
- S-Match (U. of Trento)
- OLA (INRIA and U. de Montreal)
- Falco-AO (China Southwest U.)
- RiMoM (Tsinghua U.)
- ASMOV (INFOTECH Soft, Inc., U. of Miami)
- LogMap (U. of Oxford)
- eTuner (U. of lllinois and The MITRE Corporation)
- …

# RiMOM

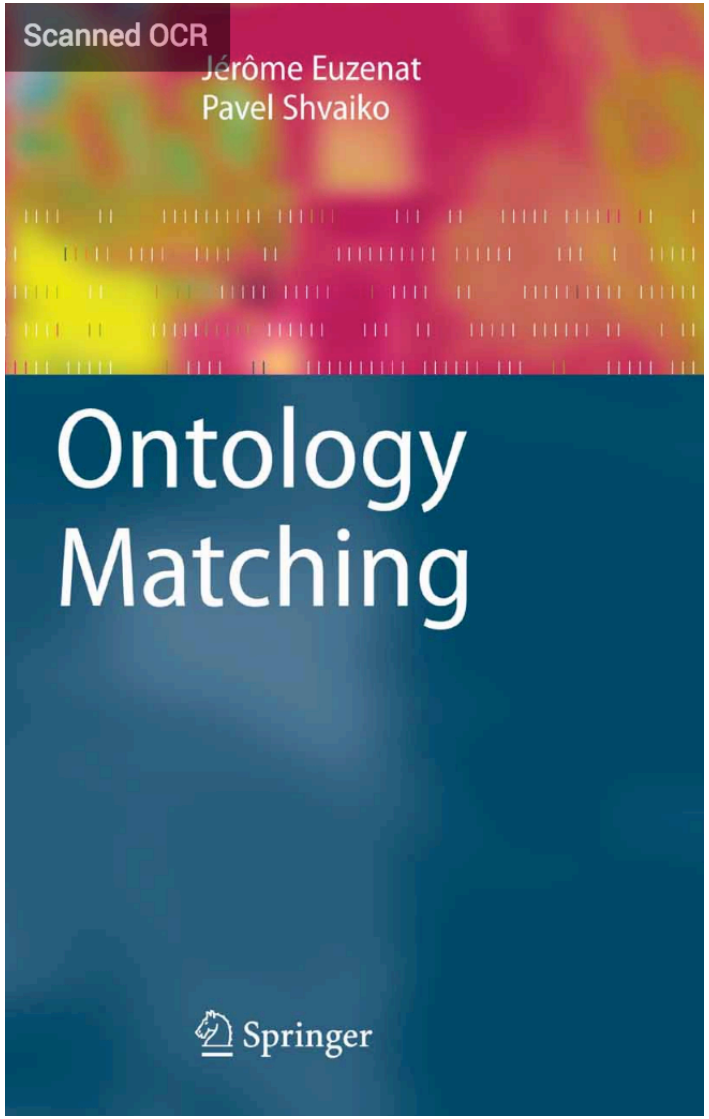



concept2concept
concept2property
property2property

# 小结

- 本节课介绍了
  - Ontology
  - Ontology Learning
  - Ontology Mapping
- 挑战
  - NLP技术面临巨大的挑战
  - Large Scale
  - Efficiency
  - 背景知识
  - 多系统融合
  - 融入人工（小样本学习）
  - 结果的可解释性
  - Crowdsourcing

# Ontology Matching




*Jérôme Euzenat, Pavel Shvaiko*

1. Applications
2. The matching problem
3. Methodology
4. Classification
5. Basic similarity measures
6. Global matching methods
7. Strategies
8. Systems
9. Evaluation
10. Representation
11. User involvement
12. Processing

http://book.ontologymatching.org

# 谢谢

# 实 体 识 别

赵 军（jzhao@nlpr.ia.ac.cn）

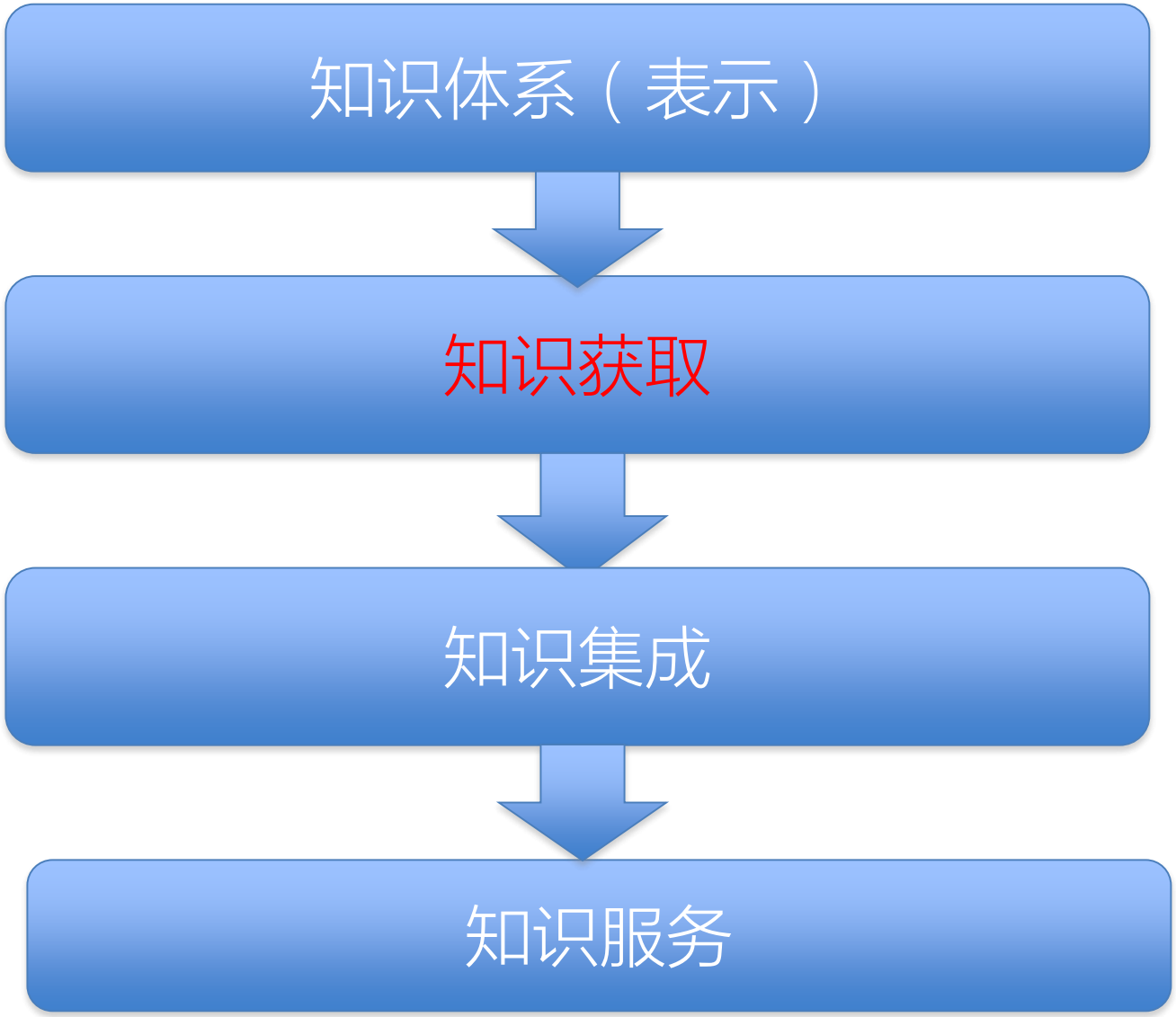
中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

# 提纲

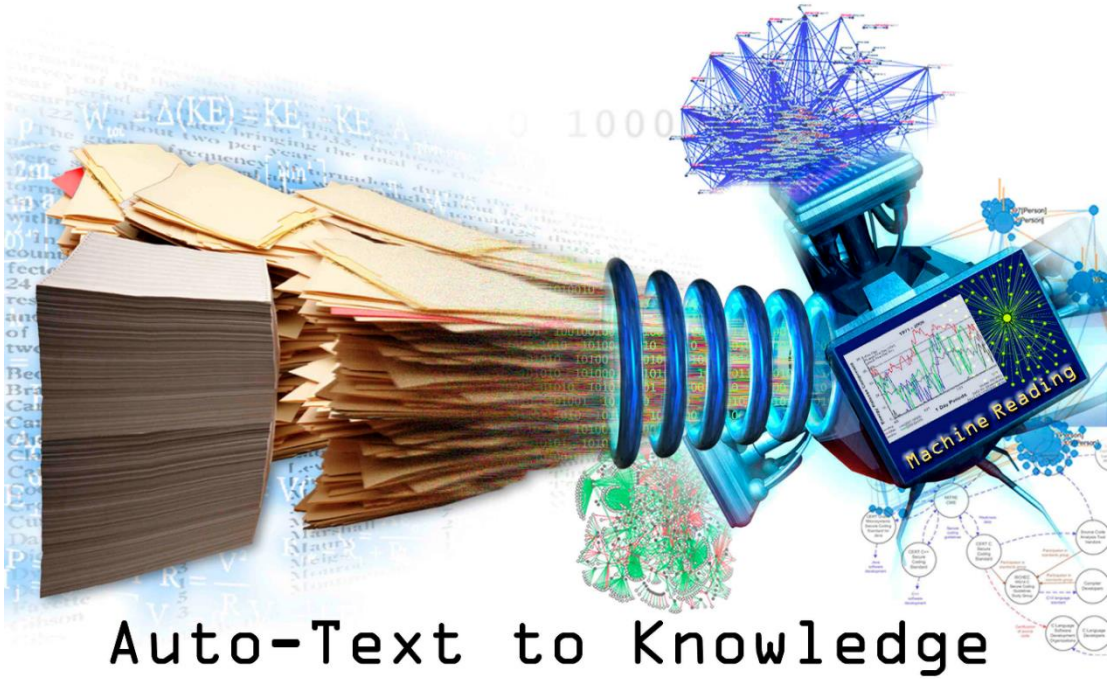
- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 知识图谱中的四个关键问题



# 生命周期—知识获取

- 输入：
  - 领域知识本体
  - 海量数据：文本、垂直站点、百科
- 输出：领域知识
  - 实体集合
  - 实体关系/属性
- 主要技术：
  - 信息抽取
  - 文本挖掘




信息抽取是自然语言处理中的关键技术

# 自然语言处理发展历程（1）

- 50-60年代：自然语言处理研究领域是作为人工智能的应用发展起来的
  - 最早的自然语言理解方面的研究工作是机器翻译，20世纪60年代，国外对机器翻译曾有大规模的研究
  - 普遍采用基于规则的方法，或者基于知识库的方法，在限定领域取得成功
  - 但人们低估了自然语言的复杂性，在开放领域遇到很大的困难
  - 1966年，美国科学院语言自动处理咨询委员会公布了《语言与机器》，否定了机译的可行性

# 自然语言处理发展历程（2）

- 90年代开始：随着大规模词典和真实语料库的研制，给自然语言处理领域的研究带来了巨大变化
  - 基于语料库的统计自然语言学习成为一种重要的方法
  - 自然语言处理系统面向大规模真实文本的处理，研制的系统开始面向实用
    - 机器翻译：统计机器翻译
    - 信息抽取：系统并不要求能对自然语言文本进行深层理解，而是从中抽取一些有用信息，作为自然语言部分理解的一种形式
  - 统计自然语言学习方法受限于训练集的规模，过拟合问题严重，推广能力不足

# 自然语言处理发展历程（3）

- 近几年
  - 深度学习方法基于分布学习词语的语义，在自然语言处理的很多任务中取得了很好的效果。
    - 基于深度学习的自然语言处理方法
    - 神经机器翻译
  - 随着Web2.0的普及，网络上积累了规模巨大的User Generated Content，为建立大规模知识库奠定基础
    - 基于知识的方法在开放域自然语言处理处理任务中的应用成为可能
  - 基于知识的方法和基于统计的方法的融合受到关注

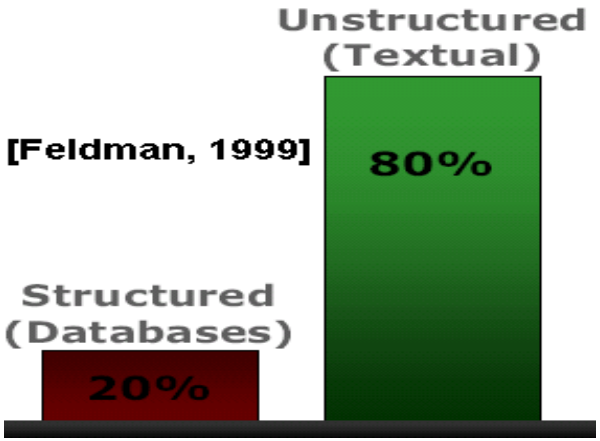
# 提纲

- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 背景

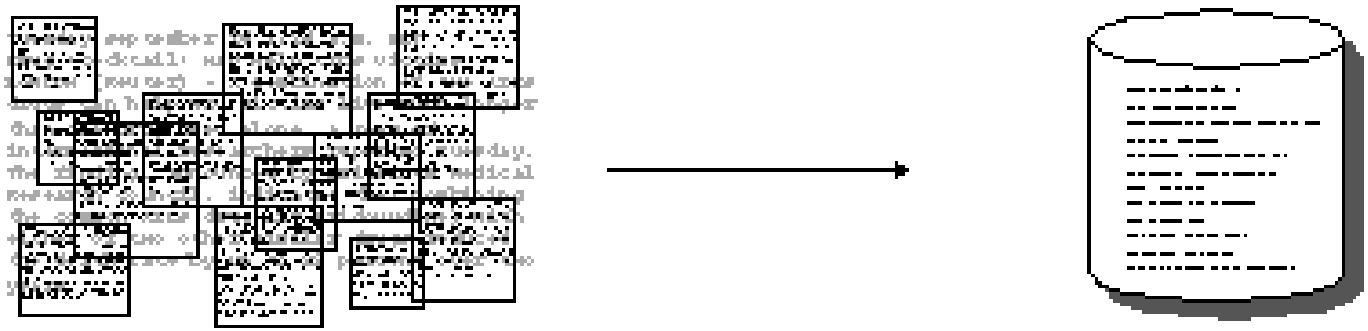
- 互联网的迅速普及和发展
  - 信息资源极大丰富
  - 但“信息过载”问题日趋严重

- 迫切需要快速、准确获取信息的技术手段
  - 信息抽取技术应运而生
    - 文本信息抽取
    - 自然语言文本信息抽取



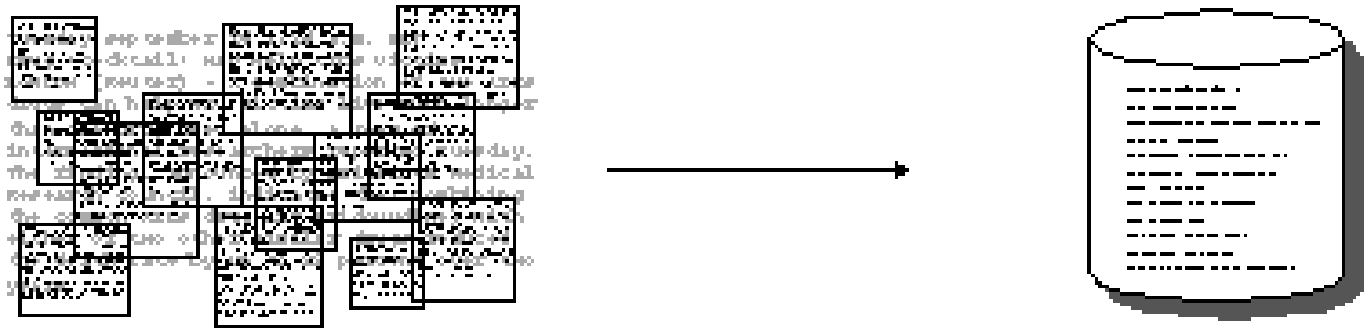
# 信息抽取定义

- 信息抽取定义(Grishman, 1997)
  - 从自然语言文本中抽取指定类型的实体、关系、事件等事实信息，并形成结构化数据输出的文本处理技术
  - Making information more machine-readable (Wu, 2010)



# 信息抽取定义

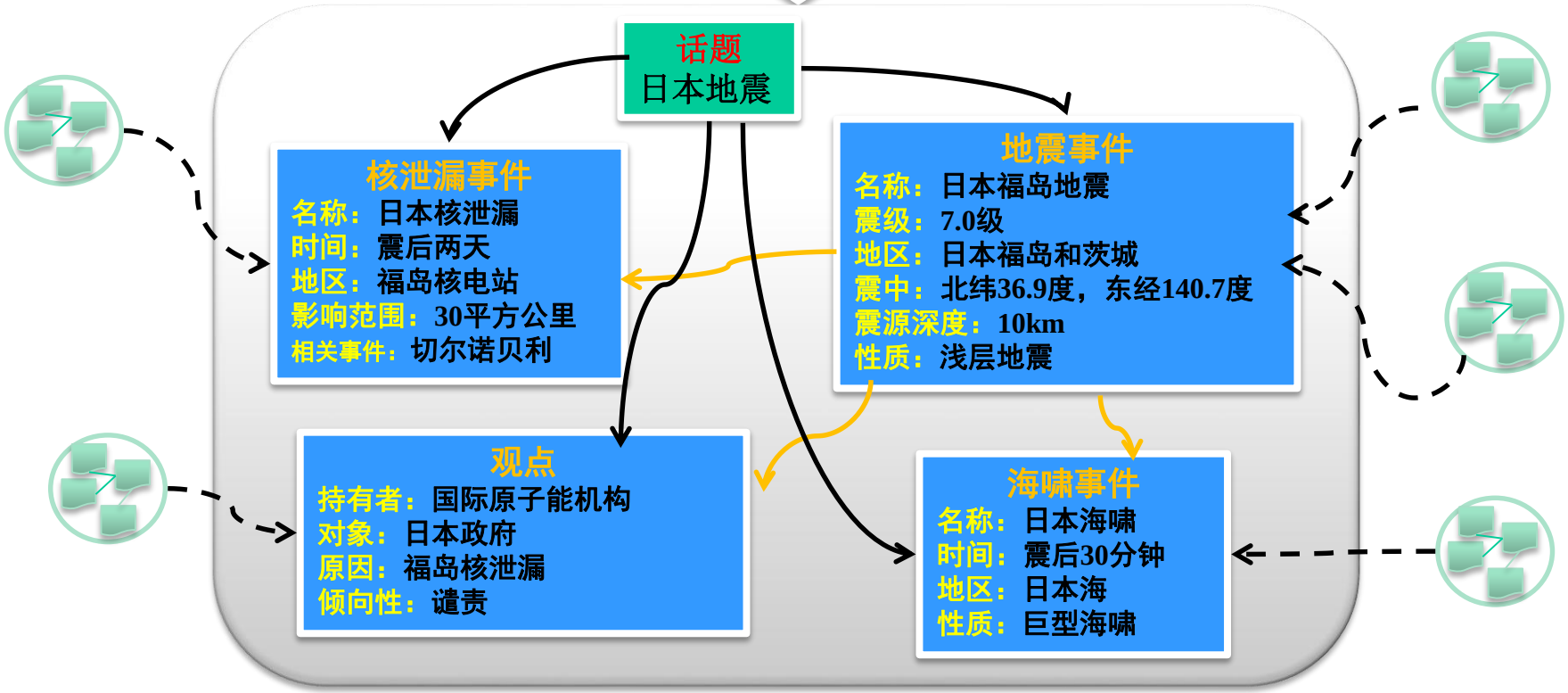
- 信息抽取定义(Grishman, 1997)
  - 从自然语言文本中抽取指定类型的实体、关系、事件等事实信息，并形成结构化数据输出的文本处理技术
  - Making information more machine-readable (Wu, 2010)



# 信息抽取的主要任务

# 信息抽取的目标示例

2011年4月11日17点16分，日本东北部的福岛和茨城地区发生里氏7.0级强烈地震（震中北纬36.9度、东经140.7度，即福岛西南30公里左右的地方，震源深度10公里，属于浅层地震）当局已经发布海啸预警震后约30分钟后在日本海地区发生巨型海啸，同时造成福岛核电站出现核泄漏震后第十天，国际原子能机构对于日本政府反应迟钝进行了谴责



# 信息抽取的历史(1/2)

- 信息抽取的研究最早于上世纪70年代末期
- 最早的信息抽取系统于上世纪80年代中期面世，由路透社研制的JASPER，用作向金融从业者提供结构化的金融新闻

# 信息抽取的历史 (2/2)

- MUC（Message Understanding Conferences, 1987-1997）
  - 由美国国防高级研究计划委员会DARPA资助
  - 主要是英文，后两届扩展到中文
  - 任务命名实体识别，共指消解，模板关系抽取等等
- ACE（Automatic Content Extraction, 1999-2008 ）
  - 由美国国家标准与技术研究所NIST主办
  - 2009起，ACE变成了TAC（Text Analysis Conference）的一项子任务
  - 英文、中文、阿拉伯文等
  - 任务命名实体识别，关系抽取，事件抽取等等
- TAC-KBP(Knowledge Base Population)子任务 (2009-今)
  - 实体链接、属性抽取
  - 事件、事件关系抽取

# 小结

| 时间 | 评测阶段 | 任务 | 语料 |
| --- | --- | --- | --- |
| 1987-1993 | MUC1-MUC5 | 命名实体识别，共指消解，模板关系抽取等等 | 限定领域文本<br>(海军军事情报、恐怖袭击) |
| 1995-1997 | MUC6-MUC7 | 模板填充、命名实体识别、共指关系确定等 | 限定领域文本<br>(人事职位变动、飞机失事) |
| 1999 | ACE-Pilot----ACE-1 | 命名实体识别 | 新闻语料 |
| 2002 | ACE2 | 命名实体识别、关系识别与描述等 | 新闻语料 |
| 2003-2007 | ACE2003-ACE2007 | 命名实体识别、关系识别与描述、时间表达式识别、事件抽取等 | 新闻语料、对话语料 |
| 2009-2017 | TAC1-TAC7 | 实体链接、属性抽取、关系抽取<br>事件抽取、事件关系抽取 | 新闻语料、Web页面 |

- 封闭语料→开放语料：限定领域、新闻语料→Web页面
- 限定类别→开放类别：有限类别的实体、关系、事件→维基百科条目
- 文本内信息抽取→与真实世界关联

# 提纲

- 引言
- 信息抽取概述
- 信息抽取的基础：分词和词性标注
- 命名实体识别
- 开放域实体识别

# 什么是中文分词

- 中文以字为基本书写单位，词语之间没有明显的区分标记
- 通俗的说，中文分词就是要由机器在中文文本中词与词之间加上标记。
  - 输入：我是学生。
  - 输出：我/是/学生/。

  - 输入：知识图谱和深度学习是人工智能的两个核心技术。
  - 输出：知识图谱/和/深度学习/是/人工智能/的/两个/核心/技术。
- 和中文相比，英语切分问题较为容易。

# 什么是词性标注

- Part-of-speech（POS）tagging（词性标注）：消除词性兼类歧义，即确定当前上下文中每个词是名词、动词、形容词或其他词性的过程。

  这个/rz 人/n 什么/ry 都/d 好/a，/wd 就/d 是/vshi 好/V 赌博/vn 。/wj

- 例子：
  - 名词和动词兼类：爱好，把握，报道
  - 形容词和名词兼类：本分，标准，典型
  - 形容词和动词兼类：抽象
  - 名词和量词：行
  - 形容词和副词：真

# 分词的重要性

- 分词是汉语句子分析的基础
- 中文的语义与字词的搭配相关

唐朝时，一个穷人到他朋友家去玩，这个穷人非常的聪明。有一天，下起了雨，他朋友想让他快点回去，不好意思说，于是，写了一封信，但没加标点。原话是：下雨天留客天留人不留。他朋友的意思是：下雨天留客，天留人不留。没想到，这位聪明的穷人在信中加了标点，意思全变了：下雨天，留客天，留人不？留！其它分词方法：

1、下雨天留客，天留，我不留。
2、下雨天，留客，天留，我不留。
3、下雨天留客，天留我不？留！
4、下雨天，留客天，留我不留？
5、下雨天，留客天，留我？不留！
6、下雨天留客，天！留我不？留！

# 分词的重要性

- 词语的分析具有广泛的应用，互联网绝大多数应用都需要分词，典型应用实例
  - 汉字处理：拼音输入法、手写识别、简繁转换 …
  - 信息检索：Google 、Baidu …
  - 内容分析：机器翻译、广告推荐、内容监控 …
  - 语音处理：语音识别、语音合成 …
  - …

# 中文分词的难点

■汉语分词规范

中华人民共和国国家标准

信息处理用现代汉语分词规范　　GB/T 13715—92

Contemporary Chinese language word segmentation specification for information processing

3.2　词

最小的能独立运用的语言单位。

3.3　词组

由两个或两个以上的词，按一定的语法规则组成，表达一定意义的语言单位。

3.4　分词单位

汉语信息处理使用的、具有确定的语义或语法功能的基本单位。它包括本规范的规则限定的词和词组。

3.5　汉语分词

从信息处理需要出发，按照特定的规范，对汉语按分词单位进行划分的过程。

# 中文分词的难点

- 汉语分词规范
  - 汉语中，字、词、词素和词组的界限模糊
    - 象牙 兔牙
    - 吃饭 吃鱼 吃羊肉 吃羊肉串
    - 新华社25日讯
    - 花草、房顶、小鸟、担水、一层

# 中文分词的难点

- 歧义切分字段处理
  - 交集型歧义：对于汉字串ABC，AB、BC同时成词
    - 例如：研究生物、从小学起、中国产品质量、部分居民生活水平、大学生、为人民工作等都是交集型歧义
    - 内塔内亚胡/说/的/确实/在理
    - 内塔内亚/胡说/的确/实在/理
    - 中国人/为了/实现/自己/的梦想
    - 中国/人为/了/实现/自己/的/梦想

# 中文分词的难点

- 歧义切分字段处理
  - 组合型歧义：对于汉字串AB，A、B、AB同时成词
    - 门/ 把/ 手/ 弄/ 坏/ 了/ 。
    - 门/ 把手/ 弄/ 坏/ 了/ 。
    - 这个人/手上有痣
    - 我们公司/人手/充足
    - 例如：才能、现在、学生会、将来等都是组合型歧义

# 中文分词的难点

- 歧义切分字段处理
  - 真歧义：歧义字段在不同的语境中确实有多种切分形式
    - 乒乓球拍/卖/完了
    - 乒乓球/拍卖/完了

# 中文分词的难点

- 未登录词识别
  - 实体名词和专有名词
    - 中国人名：李素丽、老张、李四、王二麻子、李爱国、张建国、高升、高山
    - 中国地名：定福庄、白沟、三义庙、韩村、河马甸
    - 翻译人名：乔治·布什、叶利钦、包法利夫人、酒井法子
    - 翻译地名：阿尔卑斯山、新奥尔良、约克郡
    - 机构名　：方正公司、联想集团、国际卫生组织、外贸部
    - 商标字号：百度、乐视、小米 、苹果、非常可乐 、乐凯、波导、杉杉、同仁堂

# 中文分词的难点

- 未登录词识别
  - 专业术语和新词语
    - 专业术语：万维网、主机板、模态、逻辑、贝叶斯算法、卷积神经网络、深度学习、知识图谱、人工智能、循环神经网络
    - 缩略语：一带一路、发改委、三个代表、五讲四美 、打假、扫黄打非、计生办
    - 新词语：老铁、扎心了、 666 、 555 、 333 、微博、微信、王者荣耀、撸啊撸、给力、点赞、广场舞、尬舞、尬聊、直播、网红脸、喊麦、美刀

# 中文分词结果的评价

- 两种测试
  - 封闭测试和开放测试
    - 开放测试指的是测试样本不属于训练样本集合，否则称为封闭测试；
    - 封闭测试相当于考试试题都出自于学习过的书本，实际上，通过机械记忆小样本的封闭测试取得100%的精度不存在任何问题。
  - 专项测试和总体测试
    - 专项测试是对特定领域或者特定类型的样本进行测试，反之成为总体测试
    - 总体测试能反映分词系统的综合效果，专项测试可以反映分词系统针对某个特定领域或者特定类型文本的效果

# 中文分词结果的评价

- 评价指标
  - 正确率(Correct ratio/Precision, P )：测试结果中正确切分或标注的个数占系统所有输出结果的比例。假设系统输出N个，其中，正确的结果为n个，那么，

$$P = \frac{n}{N} \times 100\%$$

# 中文分词结果的评价

- 评价指标
  - 召回率 (Recall ratio, R ): 测试结果中正确结果的个数占标准答案总数的比例。假设系统输出N 个结果，其中，正确的结果为n个，而标准答案的个数为M 个，那么，

$$R = \frac{n}{M} \times 100\%$$

# 中文分词结果的评价

- 评价指标
  - F-值(F-Measure): 正确率与召回率的综合值。计算公式为：

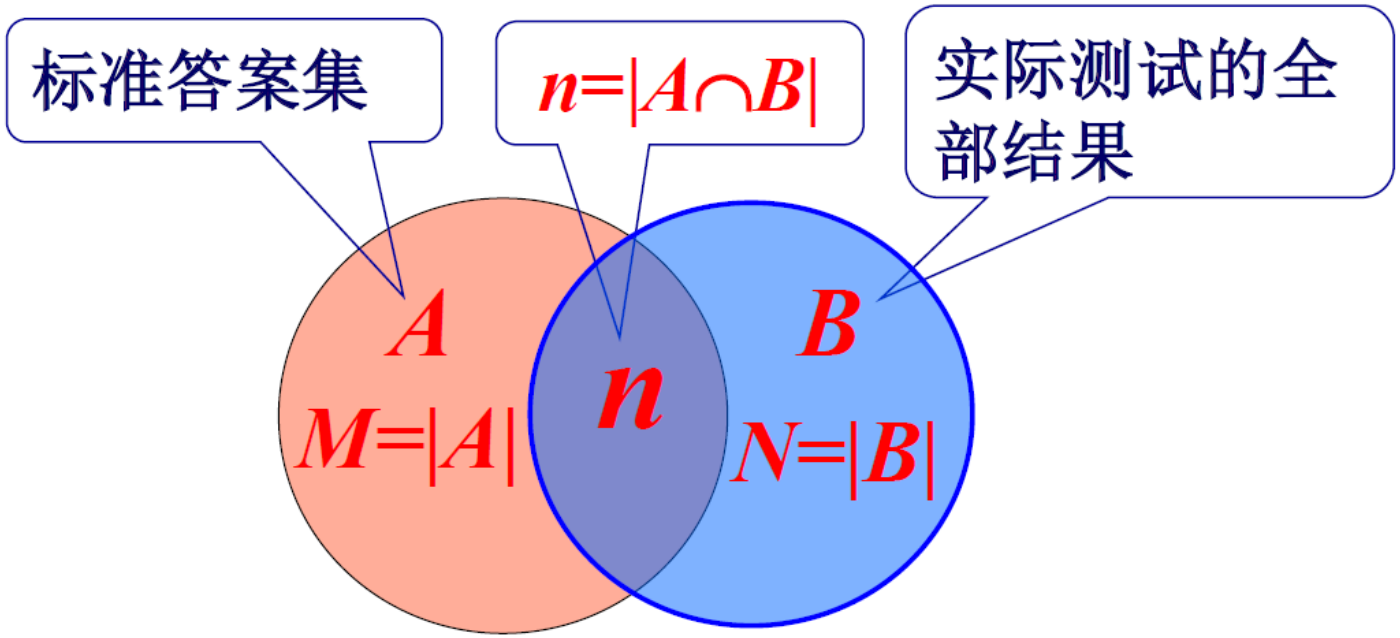
$$F-measure=\frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R}\times 100\%$$

  - 一般地，取 $\beta$ =1，即

$$F1=\frac{2\times P\times R}{P+R}\times 100\%$$

# 中文分词结果的评价

■评价指标




$$P = \frac{n}{N} \times 100\%$$

$$R = \frac{n}{M} \times 100\%$$

# 中文分词结果的评价

- 例子：假设某个汉语分词系统在一测试集上输出**5000**个分词结果，而标准答案是**4000** 个词语，根据这个答案，系统切分出来的结果中有**3000**个是正确的。那么：

$$\mathrm{P}=\frac{3000}{5000}\times 100\% = 60\%$$

$$\mathrm{R}=\frac{3000}{4000}\times 100\% = 75\%$$

$$\mathrm{F}=\frac{2PR}{P+R}\times 100\% = 66.67\%$$

# 中文分词的方法

- 方法分类
  - 有词典切分/无词典切分
    - 在分词的过程中使用词典的方法是有词典切分，反之是无词典切分
    - 有字典切分的方法一般是基于规则的，无词典切分的方法一般是基于统计的
  - 基于规则的方法/基于统计的方法
    - 基于规则的方法不需要标注训练语料，能直接根据词典和规则进行分词
    - 基于统计的方法需要标注训练语料训练模型
    - 基于统计的方法可以分为生成式统计分词和判别式统计分词

# 基于字典的分词方法

- 方法概述：按照一定的策略将待分析的汉字串与一个充分大的词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功
- 典型方法
  - 正向最大匹配法
  - 反向最大匹配法
  - 最短路径法（最少分词法）
- 举例

  句子：中医治白癜风　　词典：中、医、治、中医、医治、白癜风
  - 正向最大匹配法：中医 / 治 / 白癜风
  - 反向最大匹配法：中 / 医治 / 白癜风
  - 最短路径法：

    独立自主 / 和平 / 等 / 互利 / 的 / 原则

    独立自主 / 和 / 平等互利 / 的 / 原则

# 基于字典的分词方法

- 正向最大匹配（Forward Maximum Matching，FMM）

(1) 令i=0，当前指针$p_i$指向输入字串的初始位置，执行下面的操作：

(2) 计算当前指针$p_i$ 到字串末端的字数（即未被切分字串的长度）n，如果n=1，转(4)，结束算法。否则，令m=词典中最长单词的字数，如果n<m, 令m=n；

(3)从当前$p_i$ 起取m个汉字作为词$w_i$，判断：

(a) 如果$w_i$确实是词典中的词，则在$w_i$后添加一个切分标志，转(c)；

(b) 如果$w_i$不是词典中的词且$w_i$的长度大于1，将$w_i$从右端去掉一个字，转(a)步；否则（$w_i$的长度等于1），则在$w_i$后添加一个切分标志，将$w_i$作为单字词添加到词典中，执行(c)步；

(c) 根据$w_i$的长度修改指针$p_i$的位置，如果$p_i$指向字串末端，转(4)，否则，i=i+1，返回(2)；

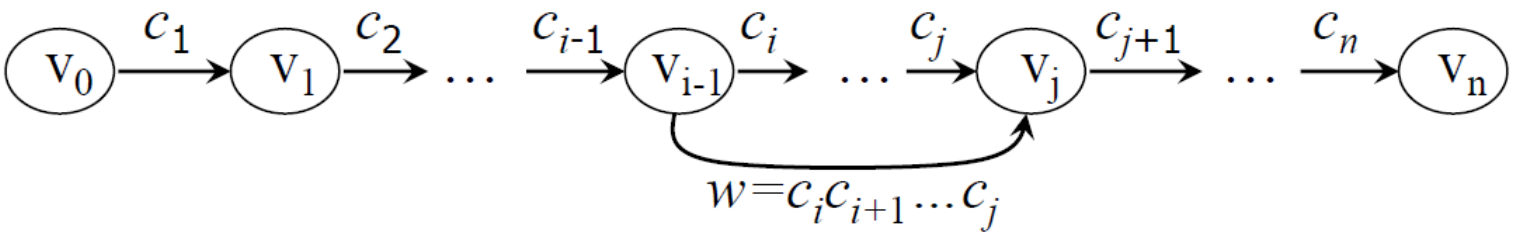
(4) 输出切分结果，结束分词程序。

# 基于字典的分词方法

- 最短路径法
  - 算法概述：

(1) 相邻节点$v_{k-1}$, $v_k$之间建立有向边<$v_{k-1}$, $v_k$>，边对应的词默认为$c_k$( $k$ =1, 2, …, $n$)。

(2) 如果$w$= $c_i c_{i+1} \ldots c_j$ (0<$i$<$j$<=$n$) 是一个词，则节点$v_{i-1}$, $v_j$之间建立有向边<$v_{i-1}$, $v_j$>，边对应的词为$w$。




(3) 重复步骤(2)，直到没有新路径(词序列)产生。

(4) 从产生的所有路径中，选择路径最短的(词数最少的)作为最终分词结果。

# 基于字典的分词方法

- 优点
  - 程序简单易行，开发周期短；
  - 仅需要很少的语言资源（词表），不需要任何词法、句法、语义资源；
- 缺点
  - 歧义消解的能力差：基于规则的，很容易错误；
  - 切分正确率不高。

# 基于统计的分词方法：生成式方法

- 原理：首先建立学习样本的生成模型，再利用模型对预测结果进行间接推理
- 典型算法：

  HMM, PCFG等
- 马尔可夫(**Andrei Andreyevich Markov**)

  前苏联数学家。切比雪夫(1821年5月16日～1894年12月8日)的学生。在概率论、数论、函数逼近论和微分方程等方面卓有成就。<u>他提出了用数学分析方法研究自然过程的一般图式—马尔可夫链，并开创了随机过程(马尔可夫过程)的研究。</u>

# 马尔可夫模型

存在一类重要的随机过程（马尔可夫过程）：如果一个系统有 $N$ 个状态 $S_1, S_2, \ldots, S_N$, 随着时间的推移，该系统从某一状态转移到另一状态。如果用 $q_t$ 表示系统在时间 $t$ 的状态变量，那么，$t$ 时刻的状态取值为 $S_j$ $(1 \le j \le N)$ 的概率取决于前 $t$-1 个时刻 (1, 2, ..., $t$-1) 的状态，该状态的概率为：

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots)$$

# 马尔可夫模型：假设

■假设1：一阶马尔可夫假设

如果在特定情况下，系统在时间t 的状态只与其在时间t-1的状态相关，则该系统构成一个离散的一阶马尔可夫链：

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots) = P(q_t = S_j \mid q_{t-1} = S_i)$$

■假设2：不动性假设

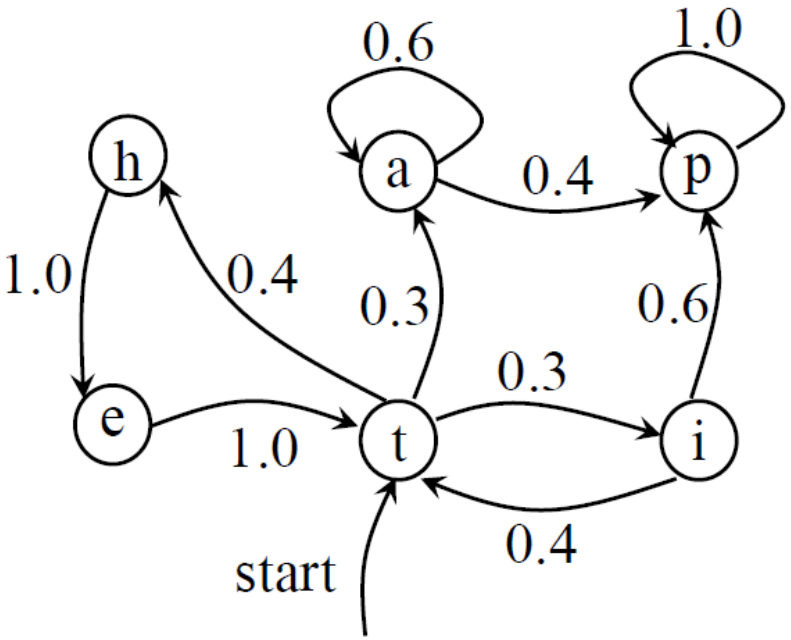
如果只考虑上述公式独立于时间$t$的随机过程， 状态与时间无关，那么：

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}, \quad 1 \le i, j \le N$$

$$a_{ij} \ge 0 \qquad \sum_{j=1}^{N} a_{ij} = 1$$

# 马尔可夫模型：状态表示

- 马尔可夫链可以表示成状态图（转移弧上有概率的非确定的有限状态自动机）
  - 零概率的转移弧省略
  - 每个节点上所有发出弧的概率之和等于1



# 马尔可夫模型：状态序列的概率

■状态序列 $S_1, \cdots, S_T$的概率：

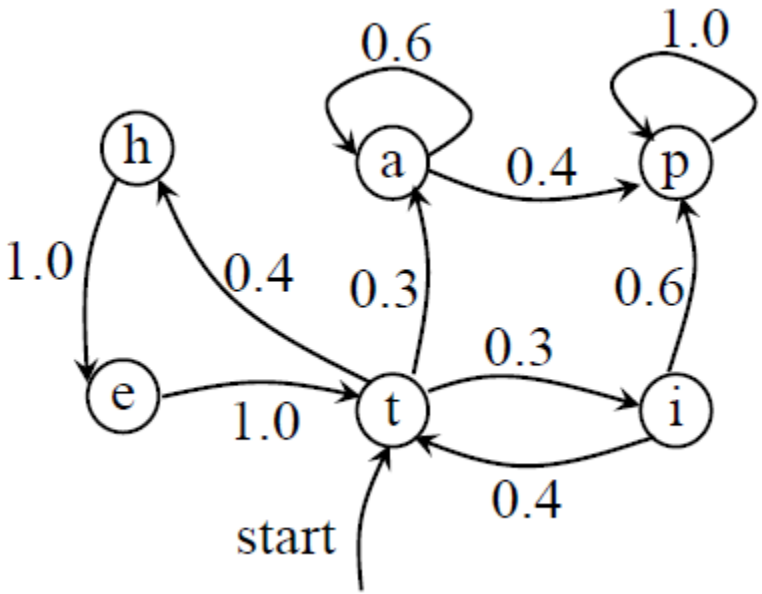
$$P(S_1, \cdots, S_T) = P(S_1)P(S_2 \mid S_1)P(S_3 \mid S_1, S_2) \cdots P(S_T \mid S_1, \cdots, S_{T-1})$$

**一阶马尔可夫假设** $= P(S_1)P(S_2 \mid S_1)P(S_3 \mid S_2) \cdots P(S_T \mid S_{T-1})$

**不动性假设** $= \pi_{S_1} \prod_{t=1}^{T-1} a_{S_t S_{t+1}}$

其中，$\pi = p(q_1 = S_i)$ 为初始状态的概率

# 马尔可夫模型：例子




$$P(t,i,p) = P(S_1 = t) \times P(S_2 = i \mid S_1 = t) \times P(S_3 = p \mid S_2 = i)$$

$$= 1.0 \times 0.3 \times 0.6$$

$$= 0.18$$

# 隐马尔可夫模型

- 隐马尔可夫模型(Hidden Markov Model, HMM)

创建于**20**世纪**70**年代，是美国数学家鲍姆**(Leonard E. Baum)**等人提出来的。隐马尔可夫模型是关于时序的概率模型，是一个双重随机过程。

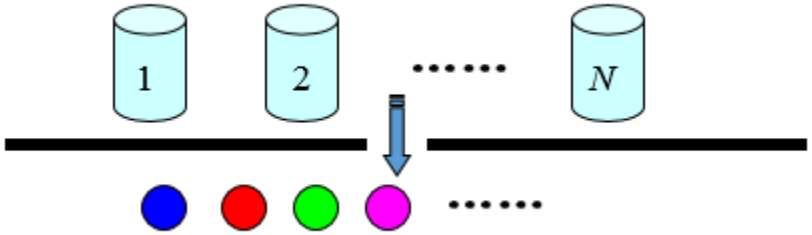
描述由一个隐藏的马尔可夫链随机生成不可观察的状态随机序列(state sequence)，再由各个状态生成一个观察，从而产生随机观察序列（observation sequence )的过程，序列的每一个位置又可以看作是一个时刻。

# 隐马尔可夫模型：例子

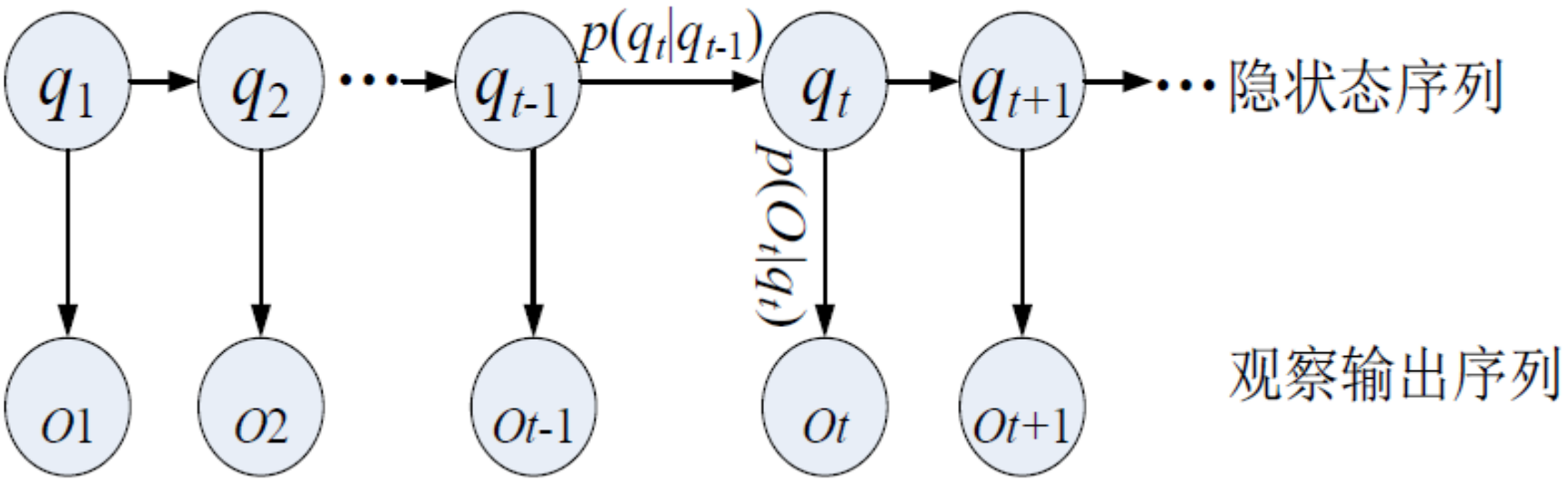
房间里有*N*个盒子，每个盒子中有*M*种不同颜色的球。一实验员进入房间根据某一概率分布选择一个盒子，然后根据盒子中不同颜色球的概率分布随机取出一个球，并向房间外的人报告该球的颜色。

对房间外的人：可观察的过程是不同颜色球的序列，而盒子的序列是不可观察的。

每只盒子对应HMM中的一个状态；球的颜色对应于HMM中状态的输出。



# 隐马尔可夫模型：图解



# 隐马尔可夫模型：HMM的组成

- 状态集合：模型中的状态数为N（盒子的数量）
- 观察集合：从每一个状态可能输出的不同的符号数M(不同颜色球的数目)
- 状态转移概率矩阵：$A = a_{ij}$ ($a_{ij}$ 为实验员从一个盒子(状态$S_i$)转向另一个盒子(状态$S_j$) 取球的概率)。其中,

$$A = [a_{ij}]_{N \times N}$$

$$a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i) \qquad i = 1,2,\ldots N; j = 1,2,\ldots, N$$

$$a_{ij} \geq 0 \qquad \sum_{j=1}^{N} a_{ij} = 1$$

# 隐马尔可夫模型：HMM的组成

- 观察概率矩阵B，处于状态 $S_i$ 的条件下生成观察 $v_k$ 的概率

$$B = \left[b_j(k)\right]_{N\times M}$$

$$b_j(k) = P\left(o_t = v_k \middle| q_t = S_j\right), \mathrm{k} = 1,2,\cdots,M;\ j = 1,2,\cdots,N$$

$$\mathrm{b_j}(k) \geq 0 \qquad \sum_{k=1}^{M} b_j(K) = 1$$

# 隐马尔可夫模型：HMM的组成

- 初始状态的概率分布π：

$$\pi_i = P(q_1 = S_i), \qquad i = 1,2,\cdots,N$$

$$\pi_{\mathrm{i}} \geq 0 \qquad \sum_{i=1}^{N} \pi_i = 1$$

# 隐马尔可夫模型：假设

- 参数集合
  - 为了方便，一般将HMM记为：$\mu = (A, B, \pi)$

    或者 $\mu = (S, O, A, B, \pi)$
- 基本假设
  - 一阶马尔可夫性假设： 隐马尔可分链t的状态只和t-1状态有关

$$P(S_t \mid S_{t-1}, o_{t-1}, \ldots, S_1, o_1) = P(S_t \mid S_{t-1}), \qquad t = 1,2,\cdots,T$$

  - 观察独立性假设：观察值和当前时刻状态有关

$$P(o_t \mid S_t, o_{t-1}, \ldots, S_1, o_1) = P(o_t \mid S_t)$$

# 隐马尔可夫模型：生成观察序列

■给定模型 $\mu = (A, B, \pi)$，生成观察序列$O = O_1 O_2 \ldots O_T$

(1) 令t=1

(2) 按照初始状态分布$\pi$产生状态 $q_1 = S_i$

(3) 按照状态$S_t$的观察概率分布 $b_i(k)$ 生成 $o_t$

(4) 按照状态$S_t$的状态转移概率分布 $\{a_{i,j}\}$ 产生状态$q_{t+1} = S_j$，

(5) 令t=t+1；如果t<T，转步(3)；否则，终止

# 隐马尔可夫模型：三个问题

- 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，怎样快速计算概率 $P(O|\mu)$ ？（**概率计算问题**）
- 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，如何选择在一定意义下“最优” 的状态序列 $Q=q_1q_2\cdots q_T$，使得该状态序列 “最好地解释”观察序列 $O=O_1O_2\ldots O_T$？（**预测问题**）
- 给定一个观察序列 $O=O_1O_2\ldots O_T$，如何根据最大似然估计求模型的参数值？即如何调节模型 $\mu=(A,B,\pi)$ 的参数，使得 $P(O|\mu)$ 最大？（**学习问题**）

# 隐马尔可夫模型：概率计算问题

- 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，怎样快速计算概率 $P(O|\mu)$ ？

**遍历所有路径**

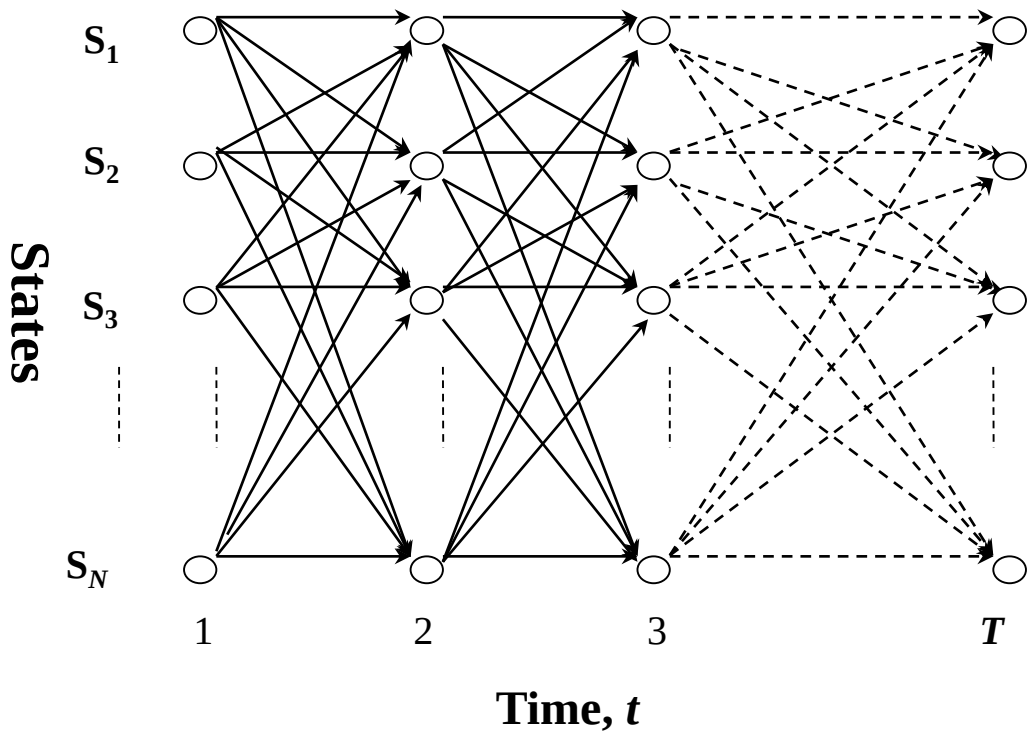
- 对于给定的状态序列 $Q=(q_1,q_2,\cdots,q_T)$ ， $P(O|\mu)$ ？

$$p(O\mid\mu)=\sum_{Q}p(O,Q\mid\mu)=\sum_{Q}p(Q\mid\mu)\times p(O\mid Q,\mu)$$

**一阶马尔可夫假设** $$p(Q|\mu)=\pi_{q_1}\times a_{q_1q_2}\times a_{q_2q_3}\times\ldots\times a_{q_{T-1}q_T}$$

**观察独立假设** $$p(O\mid Q,\mu)=b_{q_1}(O_1)\times b_{q_2}(O_2)\times\ldots\times b_{q_T}(O_T)$$

# 隐马尔可夫模型：概率计算问题




如果模型$\mu=(A, B, \pi)$ 有 N 个不同的状态，时间长度为 **T**，那么有 **$N^T$** 个可能的状态序列，搜索路径成指数级组合爆炸。

**时间复杂度：$O(TN^T)$**

**如何降低时间复杂度：动态规划，利用递推算法提高计算效率**

# 隐马尔可夫模型：概率计算问题

- 前向算法：
  - 前向变量定义：给定隐马尔可夫模型 $\mu$ ，定义到时刻t的部分观察序列为：$o_1, o_2, \cdots, o_t$ ，且**状态为** $S_i$ 的概率为前向概率，记作：

$$\alpha_t(i) = p(O_1O_2...O_t, q_t = S_i \mid \mu)$$

  - 如果可以高效地计算 $\alpha_t(i)$，就可以高效地求得 $P(O|\mu)$

$$p(O|\mu) = \sum_{S_i} p(O_1O_2\cdots O_T, q_T = S_i \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$

**T时刻输出观察序列$O_1O_2\cdots O_T$且状态为$S_i$的概率，再求和**

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] \times b_j(O_{t+1})$$

递推公式

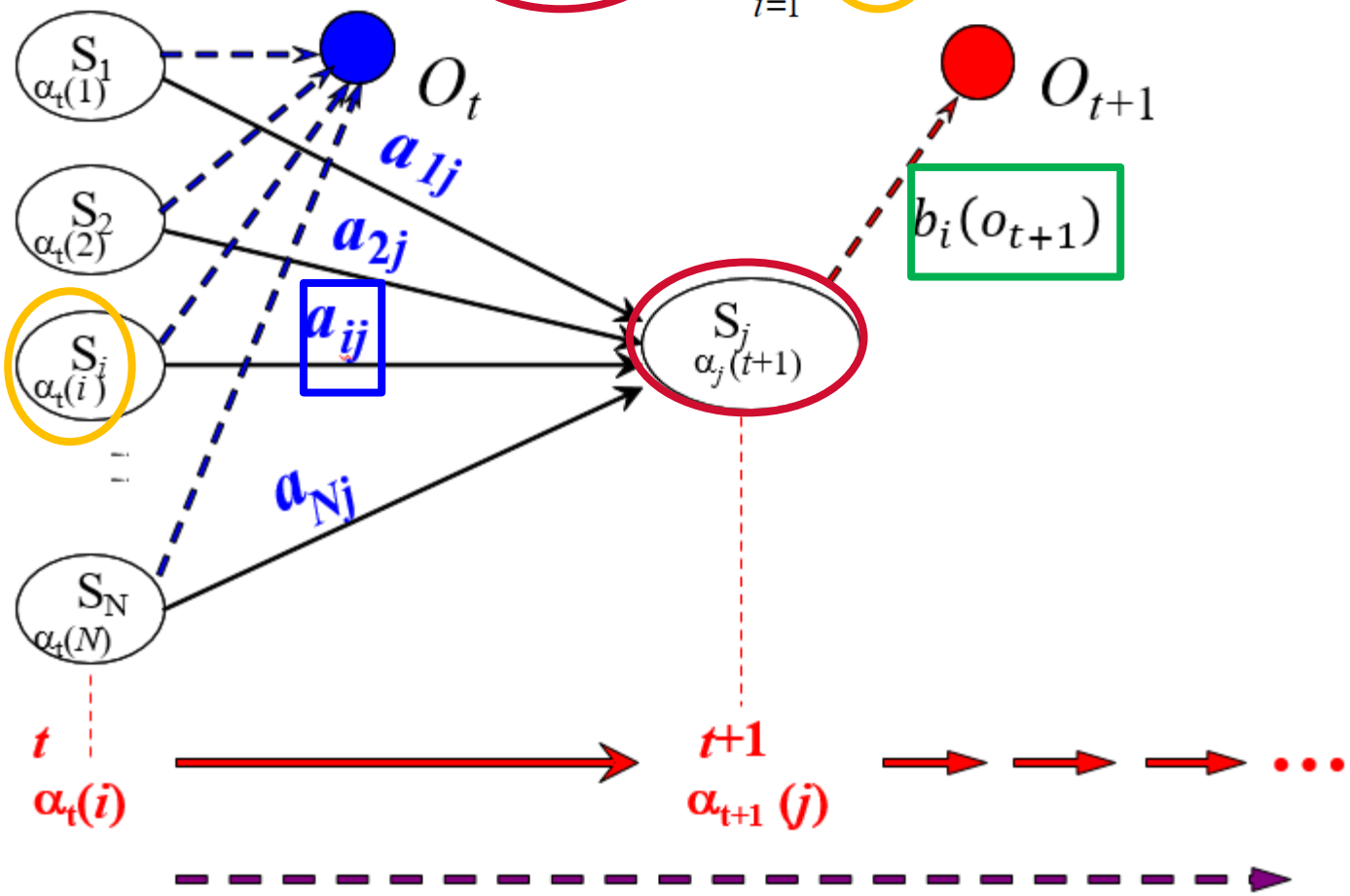
  - 递推顺序：

$$\alpha_1(x), \alpha_2(x), \ldots, \alpha_{T-1}(x), \alpha_T(x)$$

# 隐马尔可夫模型：概率计算问题

■前向算法示意图：

$$\alpha_{t+1}(j)=[\sum_{i=1}^{N}\alpha_t(i)a_{ij}]\times b_j(O_{t+1})$$



# 隐马尔可夫模型：概率计算问题

- 前向算法
  - 初始化：

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$$

  - 循环计算：

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] \times b_j(O_{t+1}), \quad 1 \le t \le T-1$$

  - 结束，输出：

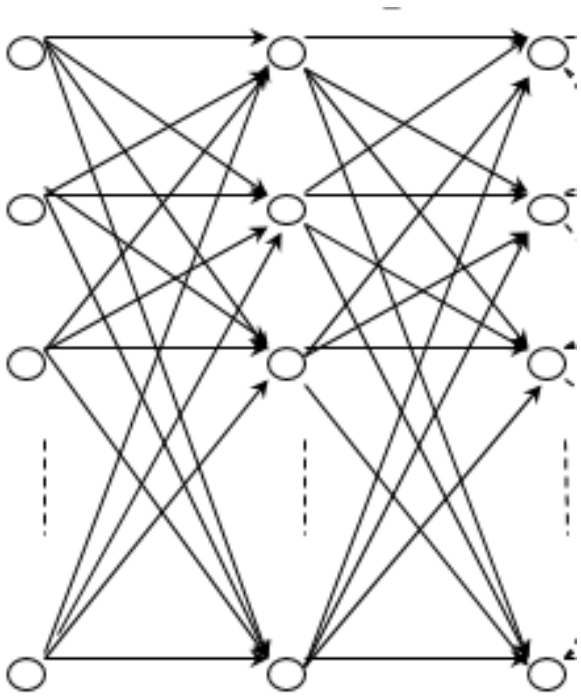
$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$

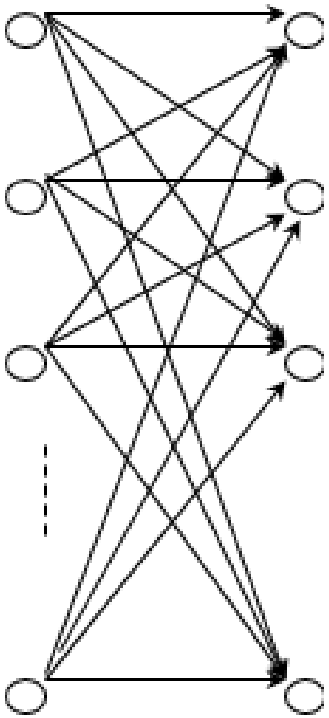
# 隐马尔可夫模型：概率计算问题

- 算法的时间复杂度
  - 每计算一个 $\alpha_t(i)$，必须考虑从 $t$-1 时的所有 $N$ 个状态转移到状态 $S_i$ 的可能性，时间复杂性为 $O(N)$
  - 对应每个时刻 $t$，要计算 $N$ 个前向变量 $\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(N)$
  - 所以时间复杂性为：$O(N) \times N = O(N^2)$
  - 又因 $t$= 1, 2, ..., $T$，所以前向算法总的**复杂性为** $O(N^2T)$

# 隐马尔可夫模型：概率计算问题

- 减少计算量的原因：递推计算中，每一次计算直接引用前一个时刻的计算结果，避免重复计算。





# 隐马尔可夫模型：概率计算问题

■例子

$$A=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}, B=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}, \pi=(0.2,0.4,0.4)^T$$

设时间长度为T=3（取了3次球）

观察序列O=（红，白，红）

试用前向算法计算P

(1)计算前向向量的初值

$$\alpha_1(1)=\pi_1 b_1(o_1)=0.10$$
$$\alpha_1(2)=\pi_2 b_2(o_1)=0.16$$
$$\alpha_1(3)=\pi_3 b_3(o_1)=0.28$$

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \pi = (0.2, 0.4, 0.4)^T$$

## (2) 递推计算

$$\alpha_2(1) = \left[\sum_{i=1}^{3} \alpha_1(i)\alpha_{i1}\right] b_1(o_2) = 0.154 \times 0.5 = 0.077$$

$$\alpha_2(2) = \left[\sum_{i=1}^{3} \alpha_1(i)\alpha_{i2}\right] b_2(o_2) = 0.184 \times 0.6 = 0.1104$$

$$\alpha_2(3) = \left[\sum_{i=1}^{3} \alpha_1(i)\alpha_{i3}\right] b_3(o_2) = 0.202 \times 0.3 = 0.0606$$

$$\alpha_3(1) = \left[\sum_{i=1}^{3} \alpha_2(i)\alpha_{i1}\right] b_1(o_3) = 0.04187$$

$$\alpha_3(2) = \left[\sum_{i=1}^{3} \alpha_2(i)\alpha_{i2}\right] b_2(o_3) = 0.03551$$

$$\alpha_3(3) = \left[\sum_{i=1}^{3} \alpha_2(i)\alpha_{i3}\right] b_3(o_3) = 0.05284$$

## (3) 终止

$$P = \sum_{i=1}^{3} \alpha_3(i) = 0.13022$$

# 隐马尔可夫模型：概率计算问题

- 后向算法
  - 后向变量：定义后向变量 $\beta_t(i)$ 是在给定模型 $\mu=(A,B,\pi)$ 和假定在时间 $t$ 状态为 $S_i$ 的条件下，模型输出观察序列 $O_{t+1}O_{t+2}\cdots O_T$ 的概率：

$$\beta_t(i)=p(O_{t+1}O_{t+2}\cdots O_T \mid q_t=S_i,\mu)$$

  - 与前向变量一样，运用动态规划计算后向量：
    - 从时刻t到t+1，模型由状态$S_i$转移到状态$S_j$，并从$S_j$输出$O_{t+1}$
    - 在时间t+1，状态为$S_j$的条件下，模型输出观察序列 $O_{t+2}O_{t+3}\cdots O_T$ （递推定义）

# 隐马尔可夫模型：概率计算问题

- 后向算法
  - 第一步的概率：$a_{ij} \times b_j(O_{t+1})$
  - 第二步的概率按后向变量的定义为：$\beta_{t+1}(j)$

    于是，有递推关系：

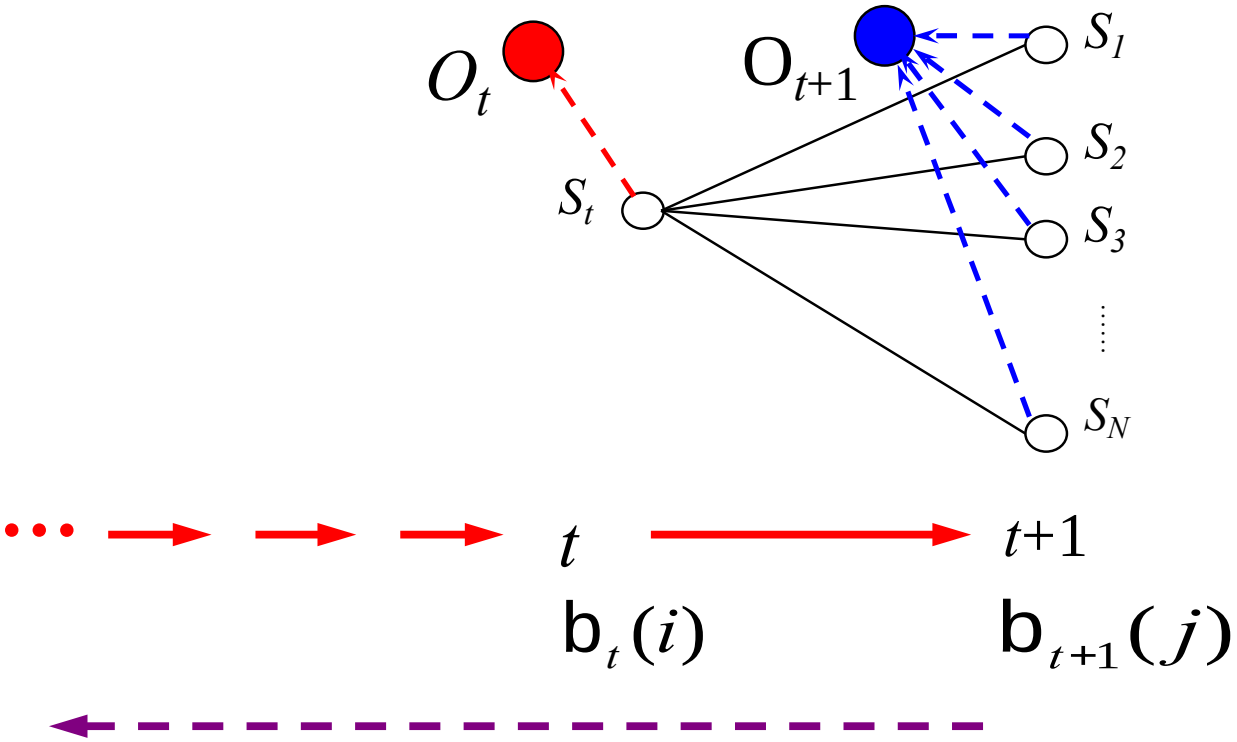
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j)$$

  - 归纳顺序：$\beta_1(x), \beta_2(x), \ldots, \beta_{T-1}(x), \beta_T(x)$

$$\longleftarrow$$

# 隐马尔可夫模型：概率计算问题

■后向算法：



# 隐马尔可夫模型：概率计算问题

- 后向算法-算法描述
  - 初始化：

$$\beta_T(i)=1,\quad 1\le i\le N$$

  - 循环计算：

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j),\quad T-1\ge t\ge 1,\quad 1\le i\le N$$

  - 结束，输出：

$$p(O\mid\mu)=\sum_{i=1}^{N}\beta_1(i)\times\pi_i\times b_i(O_1)$$

  - 时间复杂度：

$$O(N^2T)$$

# 隐马尔可夫模型：预测问题

- 在给定模型$\mu=(A,B,\pi)$和观察序列$O=O_1O_2\ldots O_T$的情况下，如何选择在一定意义下“最优” 的状态序列$q_1q_2\cdots q_T$，使得该状态序列 “最好地解释”观察序列$O=O_1O_2\ldots O_T$？（**预测问题**）

$$\hat{Q}=\arg\max_{Q} P(Q\mid O,\mu)$$

- 解释不是唯一的，关键在于如何理解 “最优” 的状态序列？
  - 一种解释是：状态序列中的每个状态都单独地具有概率，针对每个时刻t都找到具有最大概率的状态，顺序连接每个时刻具有最大概率的状态，进而得到“最优”状态序列
  - 另一种解释是：不单独考虑每个时刻的状态，而是考虑到达t时刻的状态序列的概率，从不同的状态序列中找到具有最大概率的状态序列，进而得到“最优”状态序列

# 隐马尔可夫模型：预测问题的解释之一

- 状态序列中的每个状态都单独地具有概率，针对每个时刻t都找到具有最大概率的状态，顺序连接每个时刻具有最大概率的状态，进而得到“最优”状态序列。
- 即对于每个时刻 $t$（$1 \le t \le T$），寻找 $q_t$ 使 $\gamma_t(i) = P(q_t = S_i \mid O, \mu)$ 最大

$$\gamma_t(i) = P(q_t = S_i \mid O, \mu) = \frac{P(q_t = S_i, O \mid \mu)}{P(O \mid \mu)}$$

分子

分母

**HMM 的输出序列 $O$，并且在时间 $t$ 到达状态 $i$ 的概率。**

- 分子部分：把这个事件进行拆分，按状态i分成前后两部分，可以得到前后两部分

# 隐马尔可夫模型：预测问题

- 前面的部分：模型在时间t 到达状态i, 并且输出 $O = O_1 O_2 \ldots O_t$ 。根据前向变量的定义，实现这一步的概率为 $\alpha_t(i)$
- 后面的部分：从时间t，状态Si 出发，模型输出 $O = O_{t+1} O_{t+2} \ldots O_T$ 。根据后向变量定义，实现这一步的概率为 $\beta_t(i)$
- 所以

$$p(q_t = S_i, O \mid \mu) = \alpha_t(i) \times \beta_t(i)$$

**分子部分**

- 而 $p(O|\mu)$ 与时间 $t$ 的状态无关，因此：

$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_t(i) \times \beta_t(i)$$

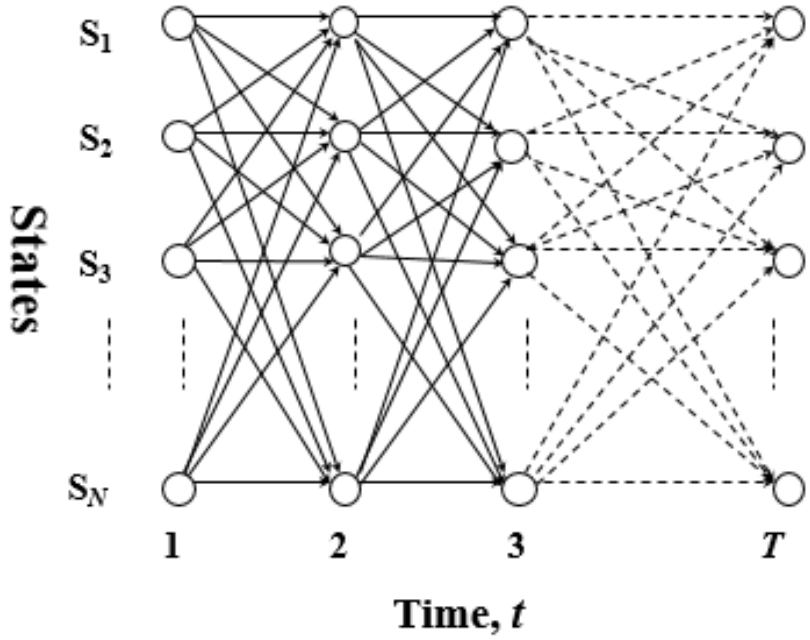
**分母部分**

- 所以

$$\gamma_t(i) = \frac{\alpha_t(i) \times \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \times \beta_t(i)} \qquad \hat{q}_t = \arg\max_{1 \le i \le N} (\gamma_t(i))$$

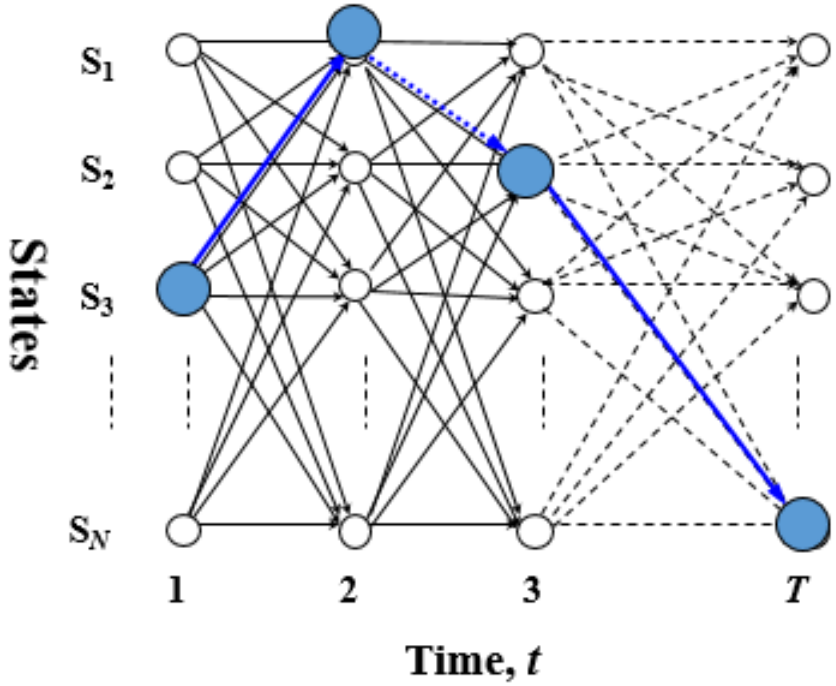
# 隐马尔可夫模型：预测问题

- 这种理解会遇到的问题：每一个状态单独最优不一定使整体的状态序列最优，可能两个最优的状态 $\hat{q}_t$ 和 $\hat{q}_{t+1}$ 之间的转移概率为0，即 $a_{\hat{q}_t \hat{q}_{t+1}} = 0$



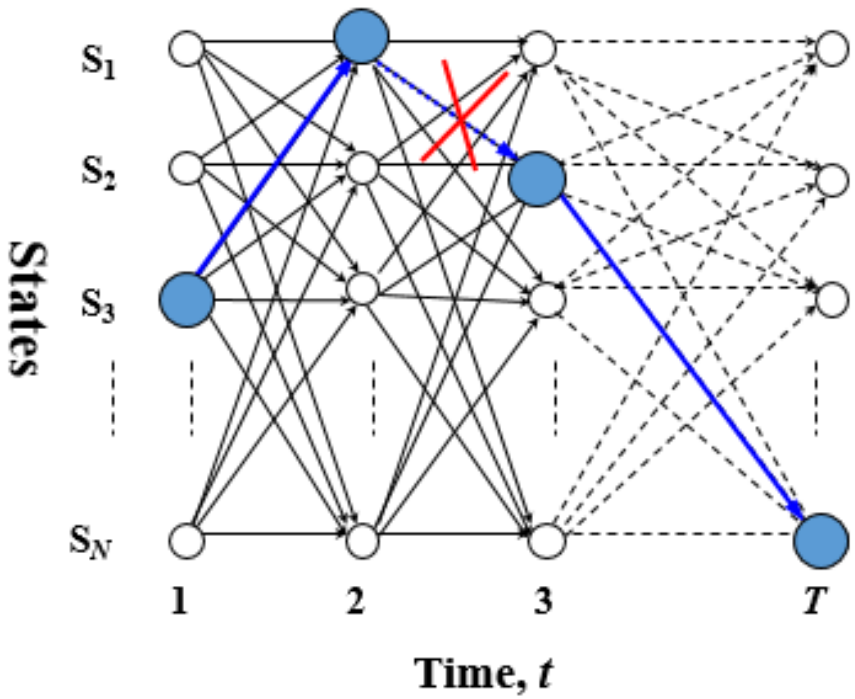
# 隐马尔可夫模型：预测问题

- 这种理解会遇到的问题：每一个状态单独最优不一定使整体的状态序列最优，可能两个最优的状态 $\hat{q}_t$ 和 $\hat{q}_{t+1}$ 之间的转移概率为0，即 $a_{\hat{q}_t \hat{q}_{t+1}} = 0$



# 隐马尔可夫模型：预测问题

- 这种理解会遇到的问题：每一个状态单独最优不一定使整体的状态序列最优，可能两个最优的状态 $\hat{q}_t$ 和 $\hat{q}_{t+1}$ 之间的转移概率为0，即 $a_{\hat{q}_t \hat{q}_{t+1}} = 0$



# 隐马尔可夫模型：预测问题的解释之二

- 给定模型$\mu$和观察序列$O$的条件下求**概率最大的状态序列**

$$\hat{Q} = \arg\max_{Q} P(Q \mid O, \mu)$$

**对比α和δ：**
**前向概率：输出序列$O_1O_2...O_t$，并且在时间 $t$ 到达状态 $S_i$ 的概率**

- Viterbi algorithm: 动态搜索最优状态序列
  - Viterbi 变量 $\delta_t(i)$的定义：时间 $t$ 时，HMM **沿着某一条路径**到达 $S_i$，并输出观察序列 $O = O_1O_2...O_t$ 的**最大概率**：

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1},} P(q_1, q_2, \ldots, q_{t-1}, q_t = S_i, O_1O_2...O_t \mid \mu)$$

  - 递推计算：

$$\delta_{t+1}(i) = \max_j [\delta_t(j) \cdot a_{ji}] \cdot b_i(O_{t+1})$$

$$\hat{Q}_T = \arg\max_{1 \le i \le N} [\delta_T(i)]$$

# 隐马尔可夫模型：预测问题

■Viterbi 算法

1时刻先找一个输出为$O_1$的最大概率的状态

□初始化：$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$

概率最大的路径变量：$\psi_1(i) = 0$

记录这个最佳状态

□递推计算：

t时刻HMM沿着某一条路径到达$S_j$，并生成$O_1O_2...O_t$的最大概率，等于t-1时刻HMM沿着某一条路径到达$S_i$，并生成$O_1O_2...O_{t-1}$的最大概率

计算概率：$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(O_t), \quad 2 \le t \le T, \quad 1 \le j \le N$

求解最优状态：$\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(O_t), \; 2 \le t \le T, \; 1 \le i \le N$

# 隐马尔可夫模型：预测问题

- Viterbi 算法
  - 结束：

结束时刻T，HMM沿着某一条路径到达$S_i$，并生成$O_1O_2...O_T$的最大概率

$$\hat{Q}_T = \arg\max_{1\le i\le N}[\delta_T(i)], \qquad \hat{P}(\hat{Q}_T) = \max_{1\le i\le N}\delta_T(i)$$

  - 通过回溯得到路径：

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad t = T-1, T-2, \cdots, 1$$

  - 时间复杂度：

$$O(N^2T)$$

# 隐马尔可夫模型：预测问题

## ■对比

$$\alpha_t(i) = P(O_1O_2...O_t, q_t = S_i \mid \mu)$$

参数为 $\mu$ ,输出序列 $O_1O_2...O_t$ ，并且在时间 $t$ 到达状态 $S_i$ 的概率

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T \mid q_t = S_i, \mu)$$

参数为 $\mu$ ,在时间 t 从状态 $S_i$ 出发， 输出序列 $O_{t+1}O_{t+2}...O_T$ 的概率

$$\gamma_t(i) = P(q_t = S_i \mid O, \mu)$$

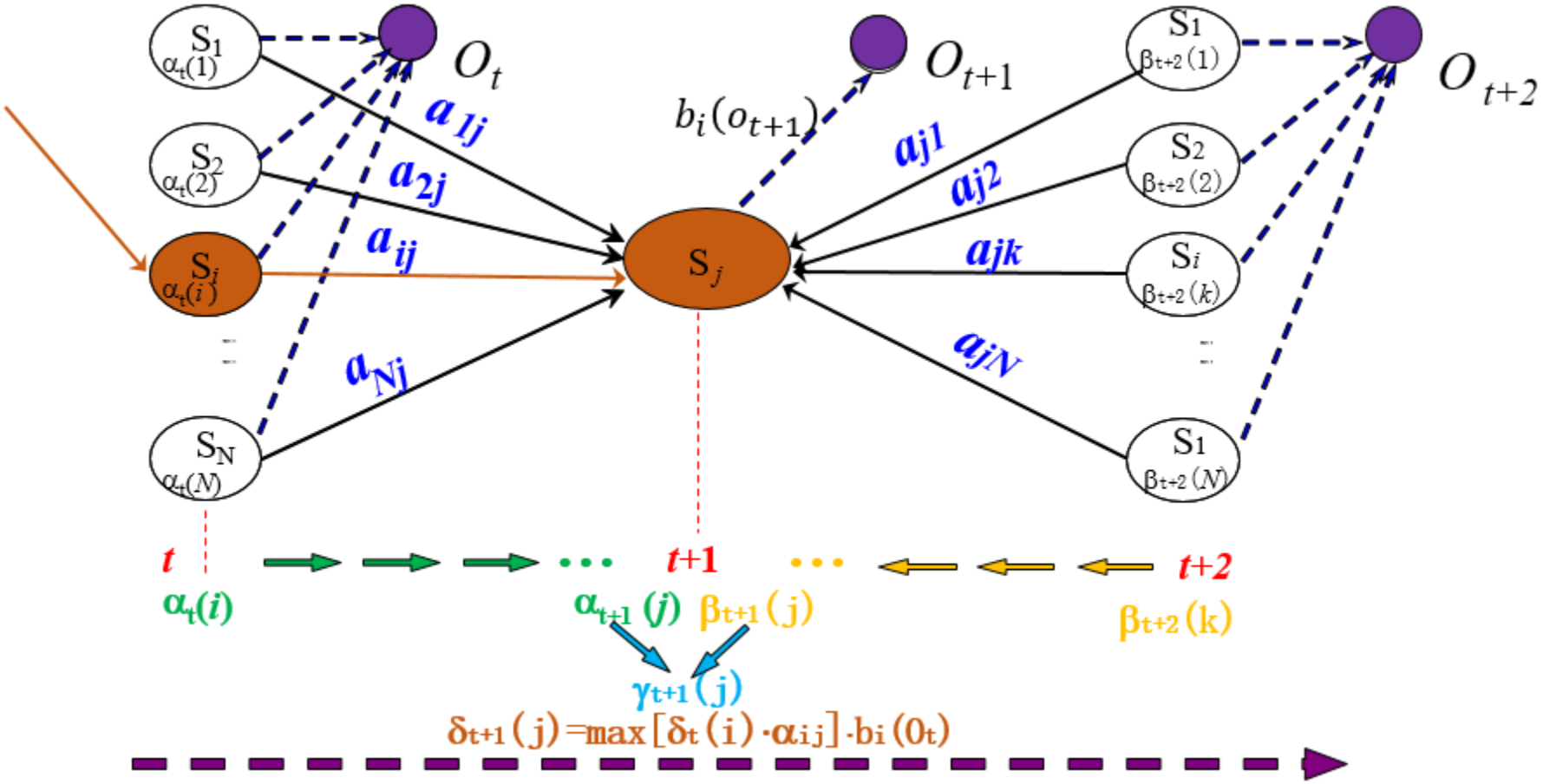
参数为 $\mu$ ,输出序列 $O_1O_2...O_T$ ， 并且在时间 t 到达状态 $S_i$ 的概率

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1},} P(q_1, q_2, \ldots, q_{t-1}, q_t = S_i, O_1O_2...O_t \mid \mu)$$

参数为 $\mu$ ，在时间 t 时，HMM **沿着某一条路径**到达状态 $S_i$，并输出观察序列 $O_1O_2...O_t$ 的最大概率

# 隐马尔可夫模型：预测问题

■对比



# 隐马尔可夫模型：预测问题

■Viterbi 搜索算法-例子

$$A=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}, B=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}, \pi=(0.2,0.4,0.4)^T$$

观察序列O=（红，白，红），试求最优状态序列，即最优路径 $S^*=(S_1^*,S_2^*,S_3^*)$

初始化： 在t=1时， 对每一个状态$S_i$， i=1,2,3， 求状态$S_i$观测$Q_1$为红的概率， 记为：$\delta_1(i)$

$$\delta_1(i)=\pi_i b_i(o_1)=\pi_i b_i(\text{红})\text{，}i=1,2,3$$
$$\delta_1(1)=0.10,\delta_1(2)=0.16,\delta_1(3)=0.28$$
$$\psi_1(i)=0,\qquad i=1,2,3$$

# 隐马尔可夫模型：预测问题

- Viterbi 搜索算法-例子
  - 在t=2时， 对每一个状态$S_i$， i=1,2,3， 求在t=1时状态为j观测$Q_1$为红并在t=2时状态为$S_i$观测$Q_2$ 为白的路径的最大概率， 记为 $\delta_2(i)$

$$\delta_2(i) = \max_{1\le j\le 3}\left[\delta_1(i)a_{ji}\right] b_i(o_2)$$

  - 同时， 对每个状态$S_i$，记录概率最大路径的前一个状态j

$$\psi_2(i) = arg \max_{1\le j\le 3}[\delta_1(i)a_{ji}], \quad i = 1,2,3$$

# 隐马尔可夫模型：预测问题

- Viterbi 搜索算法-例子
  - 计算：

$$\delta_2(1) = \max_{1\le j\le 3}\left[\delta_1(i)a_{j1}\right] b_1(o_2)$$
$$= \max_j\{0.10\times 0.5, 0.16\times 0.3, 0.28\times 0.2\}\times 0.5$$
$$= 0.028$$
$$\psi_2(1) = 3$$
$$\delta_2(2) = 0.0504, \psi_2(2) = 3$$
$$\delta_2(3) = 0.042, \psi_2(3) = 3$$

  - 同样，在t=3时 $\delta_3(3) = \max_{1\le j\le 3}\left[\delta_2(i)a_{ji}\right] b_i(o_3)$

$$\psi_3(i) = arg\max_{1\le j\le 3}[\delta_2(j)a_{ji}]$$
$$\delta_3(1) = 0.00756, \quad \psi_3(1) = 2$$
$$\delta_3(2) = 0.01008, \quad \psi_3(2) = 2$$
$$\delta_3(3) = 0.0147, \quad \psi_3(3) = 3$$

# 隐马尔可夫模型：预测问题

- Viterbi 搜索算法-例子
  - 以P*表示最优路径的概率：

$$P^* = \max_{1 \le i \le 3} \delta_3(i) = 0.0147$$

  - 最优路径的终点是：

$$S_3^* = arg \max_i [\delta_3(i)] = 3$$

  - 由最优路径的终点$S_3^*$，逆向找到$S_2^*$，$S_1^*$

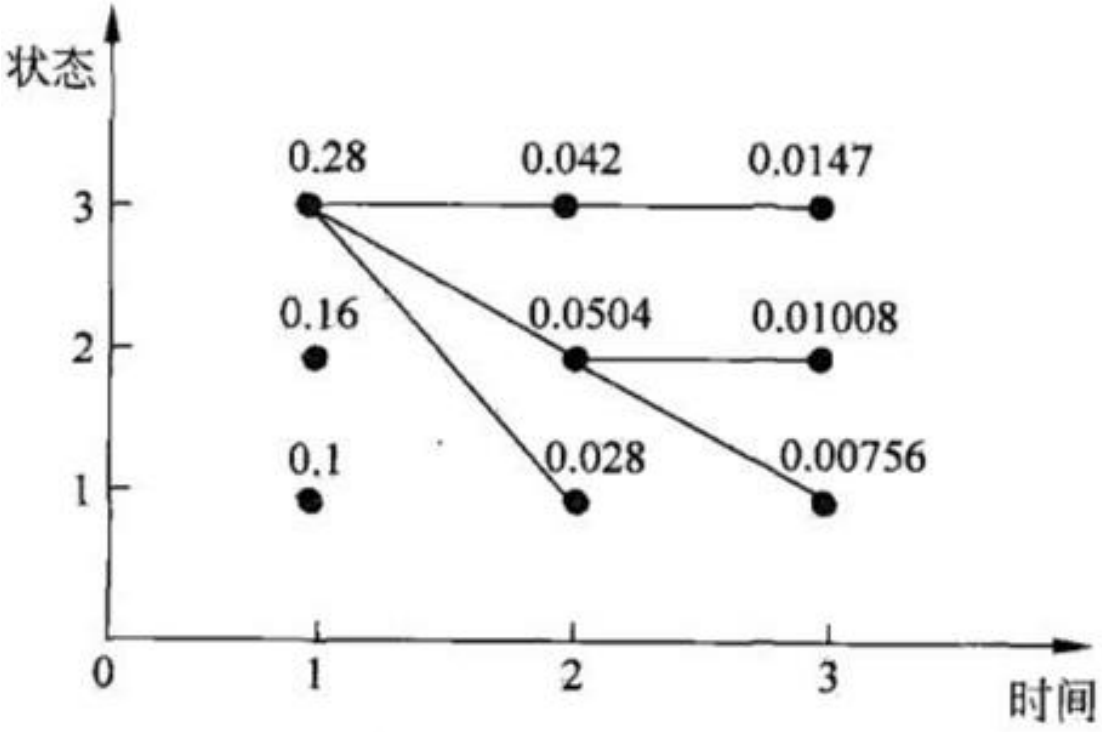
    在t=2时， $S_2^* = \psi_3(S_3^*) = \psi_3(3) = 3$

    在t=1时， $S_1^* = \psi_2(S_2^*) = \psi_2(3) = 3$
  - 于是求得最优路径，即最优状态序列：

$$S^* = (S_1^*, S_2^*, S_3^*) = (3,3,3)$$

# 隐马尔可夫模型：预测问题

- Viterbi 搜索算法-例子



# 隐马尔可夫模型：学习问题

- 给定一个观察序列$O = O_1 O_2 \ldots O_T$，如何根据最大似然估计求模型的参数值？即如何调节模型 $\mu = (A, B, \pi)$ 参数，使得 $P(O \mid \mu)$ 最大？（学习问题）

- 有监督学习的方法：假设训练数据是包括观测序列O和对应的状态序列 $Q = q_1 q_2 \ldots q_T$，则可以利用最大似然估计来计算模型的参数

$$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的次数}}{Q\text{中所有从状态}q_i\text{转移到另一状态(包括}q_j\text{自身)的总数}}$$

$$= \frac{\sum_{t=1}^{T-1} \delta(q_t, S_i) \times \delta(q_{t+1}, S_j)}{\sum_{t=1}^{T-1} \delta(q_t, S_i)}$$

# 隐马尔可夫模型：学习问题

- 有监督学习的方法
  - 其中，$\delta(x, y)$为克罗奈克(Kronecker) 函数，当$x= y$时，$\delta(x, y)=1$，否则$\delta(x, y) = 0$。
  - 类似地，

$$\bar{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的次数}}{Q\text{到达}q_j\text{的总次数}}$$

$$= \frac{\sum_{t=1}^{T} \delta(q_t, S_j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \delta(q_t, S_j)}$$

其中，$v_k$是模型输出符号集中的第$k$个符号。

  - 初始状态概率$\pi_i$的估计$\hat{\pi}_i$为S个样本中初始状态为$q_i$的频率

# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - 假设训练数据中只包括观测序列O，没有对应的状态序列 $Q = q_1 q_2 \cdots q_T$，此时只能利用期望值最大化算法（Expectation-Maximization, EM）
  - 基本思想：初始化时随机地给模型的参数赋值(遵循限制规则，如：从某一状态出发的转移概率总和为1，得到模型 $\mu_0$，然后可以从 $\mu_0$ 得到从某一状态转移到另一状态的期望次数，然后以期望次数代替公式中的次数，得到模型参数的新估计，由此得到新的模型 $\mu_1$，从 $\mu_1$ 又可得到模型中隐变量的期望值，由此重新估计模型参数。循环这一过程，参数收敛于最大似然估计值。

# 隐马尔可夫模型：学习问题

## ■算法流程

随机初始化参数
$\mu_0 = (A_0, B_0, \pi_0)$

E步：计算期望值
$\xi_t(i,j), \gamma_t(i)$

M步：估计参数
$\mu_k = (A_k, B_k, \pi_k)$

算法终止：
$|\log P(O|\mu_{k+1}) - \log P(O|\mu_k)| < \varepsilon$

观察序列
$O = O_1 O_2 \ldots O_T$

$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \le i \le N \\ \sum_{k=1}^{M} b_i(k) = 1 & 1 \le i \le N \end{cases}$$

# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - 定义：给定模型 $\mu$ 和观察序列 $O=O_1O_2\ldots O_T$，那么，在时间 $t$ 位于状态 $S_i$，时间 $t+1$ 位于状态 $S_j$ 的概率：

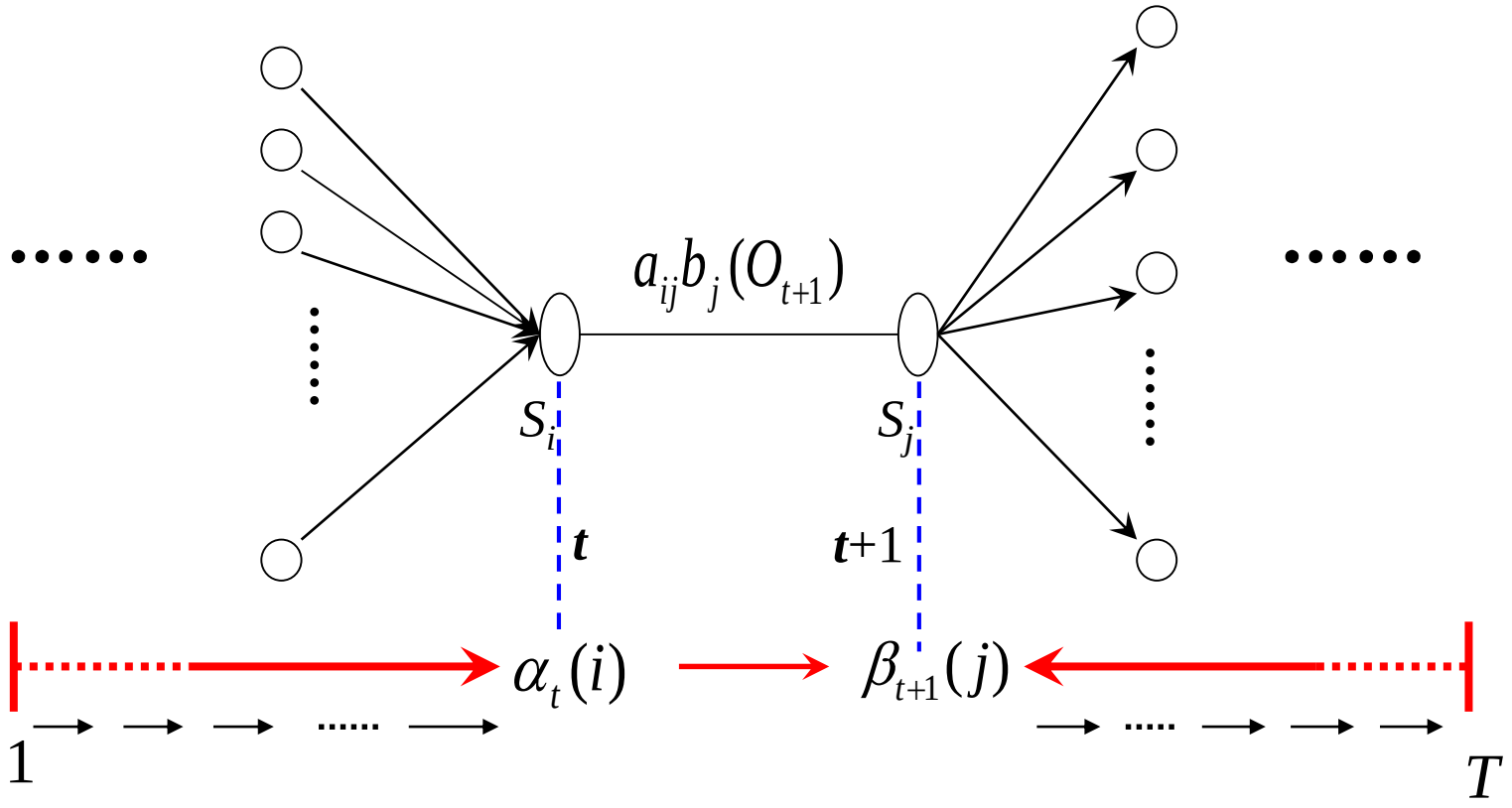
$$\xi_t(i,j)=p(q_t=S_i,q_{t+1}=S_j\mid O,\mu)=\frac{p(q_t=S_i,q_{t+1}=S_j,O\mid\mu)}{p(O\mid\mu)}$$

$$=\frac{\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}{p(O\mid\mu)}$$

$$=\frac{\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}$$

# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - 图搜索的过程



# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - 那么，给定模型 $\mu$ 和观察序列 $O = O_1 O_2 \cdots O_T$，在时间 t 位于状态 $S_i$ 的概率为：

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j)$$

下一页中的分母

  - 由此，模型$\mu$ 的参数可由下面的公式重新估计：

$q_1$ 为$S_i$的概率：

$$\pi_i = \gamma_1(i)$$

# 隐马尔可夫模型：学习问题

■无监督学习方法

$$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的期望次数}}{Q\text{中所有从状态}q_i\text{转移到下一状态(包括}q_j\text{自身)的期望次数}}$$

$$= \frac{\sum_{i=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的期望次数}}{Q\text{到达}q_j\text{的期望次数}}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \gamma_t(j)}$$

# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - Baum-Welch 算法(前向后向算法)描述:
    - 初始化：随机地给 $\pi_i$， $a_{ij}$， $b_j(k)$ 赋值

$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \le i \le N \\ \sum_{k=1}^{M} b_i(k) = 1 & 1 \le i \le N \end{cases}$$

由此得到模型$\mu_0$，令 i = 0。

# 隐马尔可夫模型：学习问题

- 无监督学习方法
  - Baum-Welch 算法(前向后向算法)描述:
    - 执行EM算法：

E-步：计算期望值$\xi_t(i, j)$、$\gamma_t(i)$ 。

M-步：用E-步中所得到的期望值，重新估计$\pi_i$ $a_{ij}$ $b_j(k)$ 得到模型$\mu_{i+1}$。

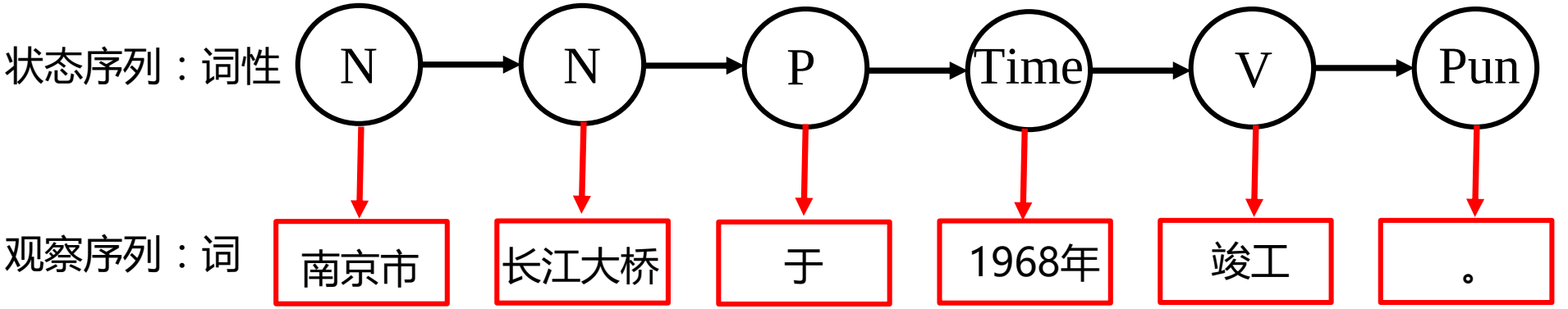
循环：i = i+1，重复执行 E-步和M-步，直至$\pi_i$ $a_{ij}$ $b_j(k)$ 的值收敛：

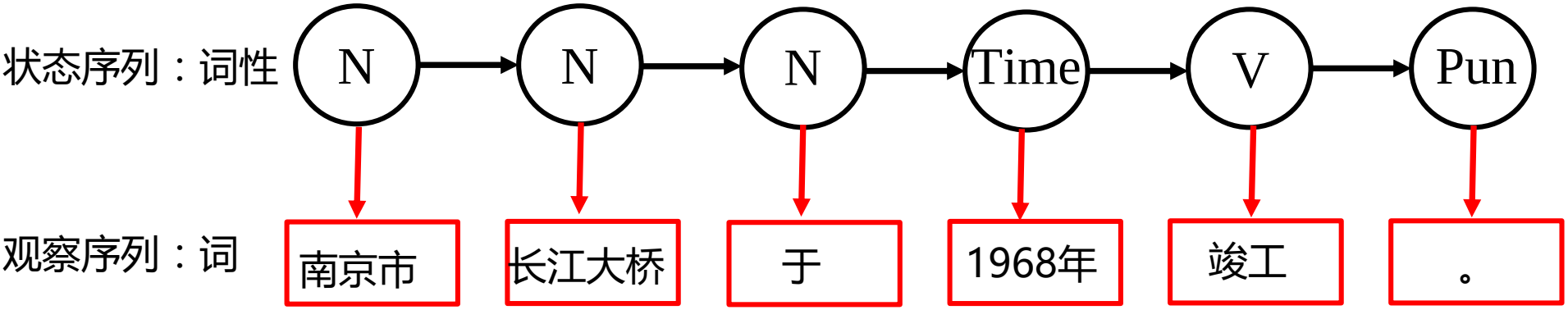
$$|\log P(O|\mu_{i+1}) - \log P(O|\mu_i)| < \varepsilon$$

    - 结束算法，获得相应的参数。

# 隐马尔可夫模型：分词+词性标注

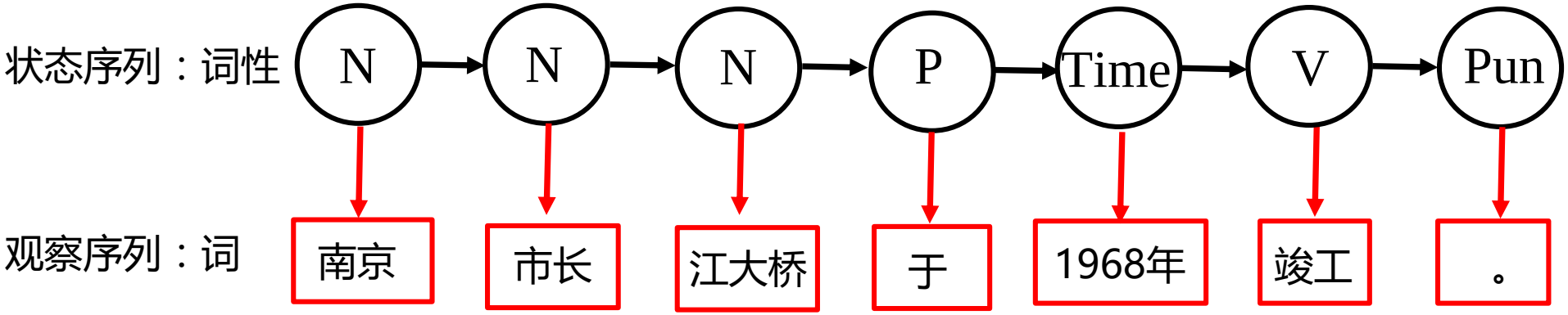
- 举例：
  - 待处理句子：南京市长江大桥于1968年竣工。

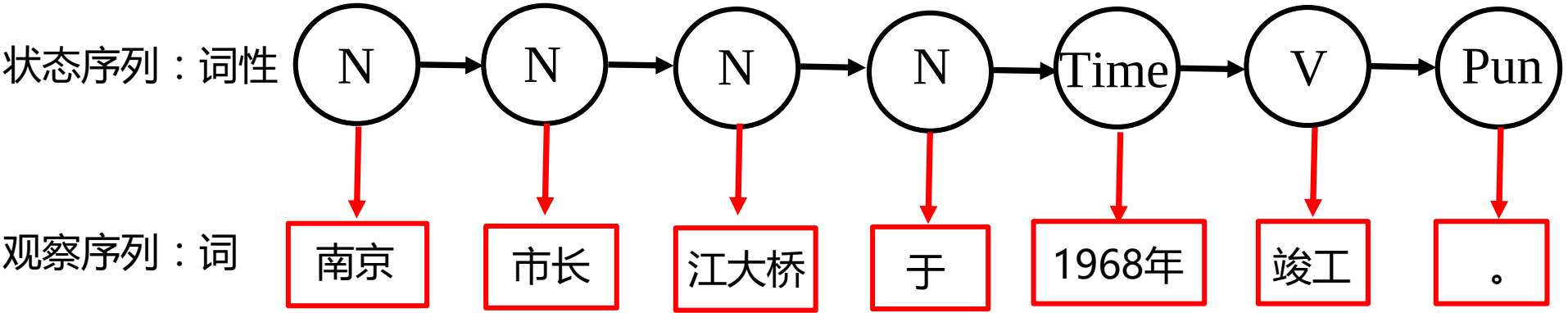


# 隐马尔可夫模型：分词+词性标注

- 举例：
  - 待处理句子：南京市长江大桥于1968年竣工。



# 隐马尔可夫模型：分词+词性标注

- 举例：南京市长江大桥于1968年竣工。
  - 模型对应
    - 观察序列：单词序列
    - 状态序列：词性标注序列
    - 状态数目N：为词类标记符号的个数，如UpennLDC汉语树库中有33个词类，北大语料库词类标记符号106个等
    - 输出符号数M：每个状态可输出的不同词汇个数，如汉语介词P 约有50个，连词C约有120个，即P 和C分别对应的输出符号数为50、120。
  - 求解目标
    - 分词结果： $\hat{O} = \arg\max_{O} p(O \mid \mu)$
    - 词性标注结果： $\hat{Q} = \arg\max_{Q} p(Q \mid O, \mu)$
  - 参数学习
    - 有监督训练：直接根据数据进行统计得到 $\mu = (A, B, \pi)$
    - 无监督训练：E-M算法求得 $\mu = (A, B, \pi)$

# 基于统计的分词方法：生成式方法

- 优点
  - 在训练语料规模足够大和覆盖领域足够多的情况下，可以获得较高的切分正确率
- 缺点
  - 训练语料的规模和覆盖领域不好把握。
  - 模型实现复杂、计算量较大。

# 基于统计的分词方法：判别式方法

- 原理：
  - 在有限样本条件下建立对于预测结果的判别函数，直接对预测结果进行判别
  - 由字构词的分词理念，将分词问题转化为判别式分类问题

- 典型算法：

  Maxent SVM CRF Perceptron

# 基于统计的分词方法

- 判别式分词
  - 流程：
    - 把分词问题转化为确定句中每个字在词中位置问题
    - 每个字在词中可能的位置可以分为以下三种
      - 词首B（日本 占领 了 东三省）
      - 词中M（游泳 比赛 菲尔普斯 独占鳌头）
      - 词尾E（中国队 抢占 了 风头）
      - 独字S （男生 占 全班 人数 的 百分之八十）
  - 分词结果展示：
    - 分词结果：毛/B新/M年/E2/B0/M0/M0/M年/E毕/B业/E/于/S 东/B北/M大/M学/E

# 最大熵模型

- 什么是熵？
  - 什么是熵？ 没有什么问题在科学史的进程中曾被更为频繁地讨论过

    （比利时物理化学家，诺贝尔奖，普里高津）

  - 熵定律是自然界一切定律中的最高定律

    （里夫金&霍华德，《熵:一种新的世界观》）

# 最大熵模型：熵的提出

- 热力学第二定律：物理过程总是自发地从有序走向无序，最后达“热寂”。
- 热寂是猜想宇宙终极命运的一种假说。作为一个“孤立”的系统，宇宙的熵会随着时间的流逝而增加，由有序向无序，当宇宙的熵达到最大值时，宇宙中的其他有效能量已经全数转化为热能，所有物质温度达到热平衡。这样的宇宙中再也没有任何可以维持运动或是生命的能量存在。

# 最大熵模型：熵的提出

- 德国物理学家克劳修斯（Rudolph J.E clausius）从热力学第二定律出发，于1865提出熵的概念用来描述一个系统的无序度。 因此热力学第二定律又被称为“增熵原理”，即系统的演进总是指向熵增加的方向。

- 克劳修斯的熵概念这是在热力学角度提出的，后被Boltzmann通过统计物理学的角度重新诠释。

# 最大熵模型：熵与信息

- 熵表示了一个事件的不确定性
- 信息可以理解为事件不确定性的减少
  - 原来不确定的事件现在确定下来，就是获得了信息
  - 原来不确定性越大的事情发生了，获得的信息越多

    例如：

    狗咬人不是新闻，人咬狗才是新闻

# 最大熵模型：信息熵

- 熵：1948年电气工程师香农( Shannon)创立了信息论，将信息量与熵联系起来。
- 他用非常简洁的数学公式定义了信息时代的基本概念：信息熵（简称熵），定量地描述事件的不确定性
- 设随机变量ξ ，它有A1，A2，…，An共n个可能的结局，每个结局出现的概率分别为$p_1$,$p_2$，… ，$p_n$，则ξ 的不确定程度，即信息熵为:

$$H(\xi)=-\sum_{i=1}^{n} p_i \log p_i$$

- 熵越大，越不确定；熵等于0，事件是确定的；通常对数底取2，熵的单位为比特（bit）

# 最大熵模型：例子

- 抛硬币
  - X={正面，反面}
  - p(正面)=p(反面)=0.5
  - H(X)=-(0.5log0.5+0.5log0.5)=1
- 掷骰子
  - X={1，2，3，4，5，6}
  - p(1)=p(2)=p(3)=p(4)=p(5)=p(6)=1/6
  - H(X)=log(6)=2.585
- 不对称的硬币
  - p(正面)=0.3, p(反面)=0.7
  - H(X)=-(0.3log0.3+0.7log0.7)=0.881

# 最大熵模型：通信中的信息量和熵

- 表示“是” 和 “否”
  - 1 = 是，0 =否
- 表示“是” 、 “否”和“可能是”
  - 11 =是，00 = 否 ，10(01) = 可能是
- 一条消息的熵就是编码这条消息所需二进制位即比特的个数
- 通信中，熵越大，需要的编码越长

# 最大熵模型：最大熵理论

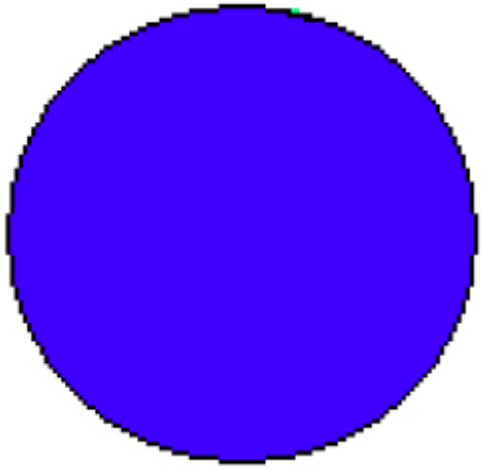
- 熵增原理
- 在无外力作用下，事物总是朝着最混乱的方向发展
- 事物是约束和自由的统一体
- 事物总是在约束下争取最大的自由权，这其实也是自然界的根本原则。
- 在已知条件下，熵最大的事物，最可能接近它的真实状态

# 最大熵模型

- 基于最大熵原理的模型选择
  - 任务：研究某个随机事件，根据已知信息，预测其未来行为。
  - 方法：当无法获得随机事件的真实分布时，构造统计模型对随机事件进行模拟。
  - 难点：满足已知信息要求的模型可能有很多个，用哪个模型来预测最合适呢？
    - 选择熵最大的模型
    - Jaynes证明：对随机事件的所有相容的预测中，熵最大的预测出现的概率占绝对优势
    - Tribus证明：正态分布、伽玛分布、指数分布等，都是最大熵原理的特殊情况

# 最大熵模型

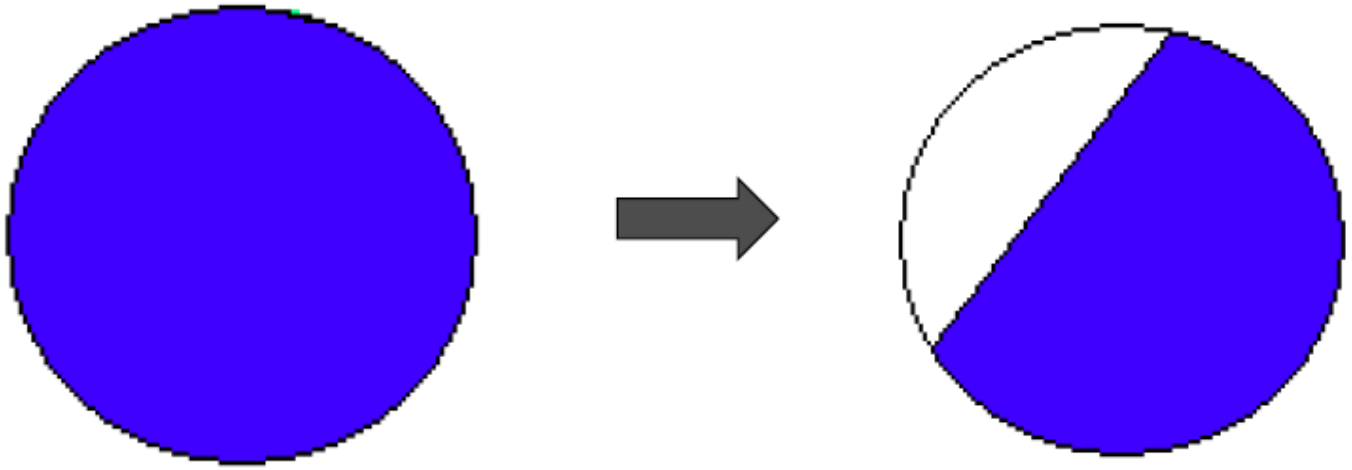
- 最大熵原则下点的分布



  - 对一个随机过程，如果没有任何观察量，既没有任何约束，则解为均匀分布

# 最大熵模型

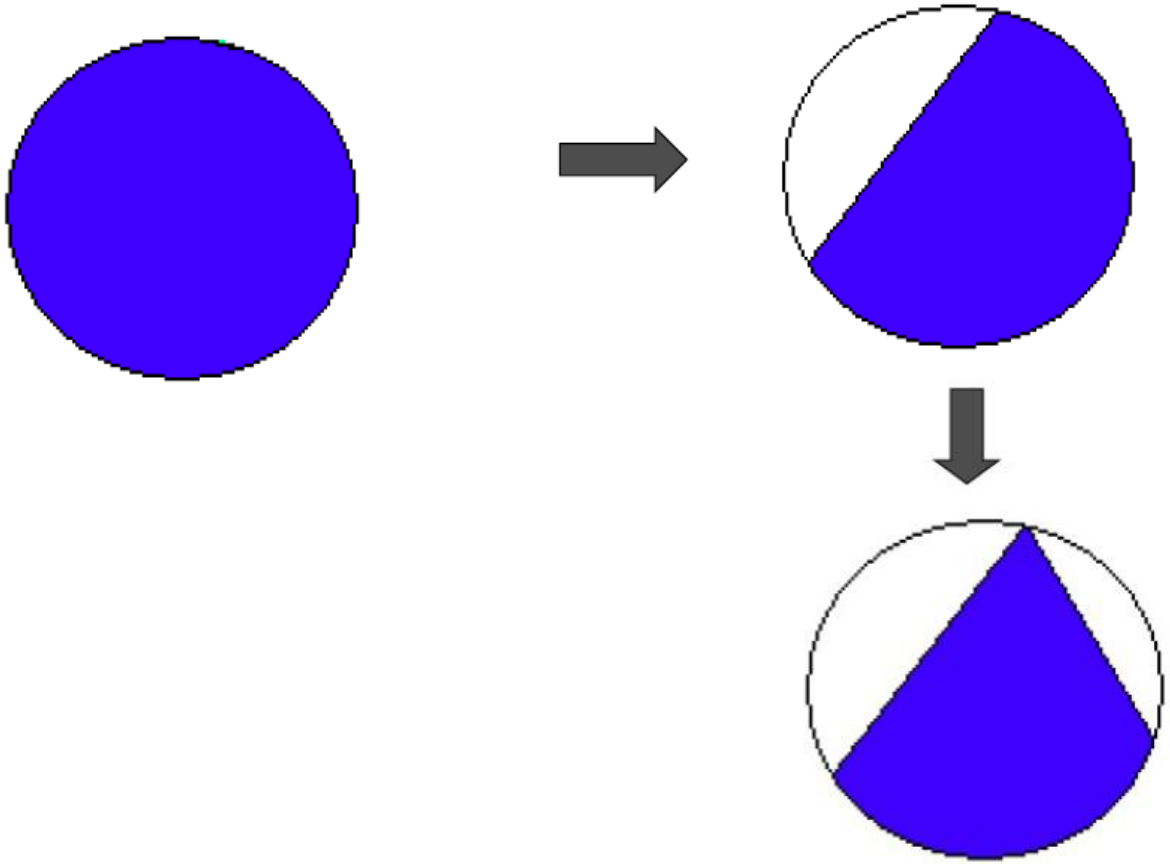
- 最大熵原则下点的分布



  - 加上一个约束条件，则在满足约束的范围内均匀分布

# 最大熵模型

- 最大熵原则下点的分布



- 加上更多约束条件，则在满足约束的范围内均匀分布

# 最大熵模型：天气预报

- 假设要用今天的天气预测明天的天气
  - 天气∈{晴、阴、雨}
  - 风向∈{无、南、北}
  - 已知今天的天气和风向，要预测明天的天气

- 已知历史天气数据（见下页表）
- 已知今天天气为阴，风向北，明天天气最可能是什么？

# 最大熵模型：天气预报

■样本数据：

| 昨天风向 | 昨天天气 | 今天天气 |
|---|---|---|
| 无 | 雨 | 晴 |
| 南 | 阴 | 阴 |
| 北 | 晴 | 晴 |
| 北 | 晴 | 晴 |
| 南 | 阴 | 雨 |
| 无 | 晴 | 晴 |
| 北 | 晴 | 阴 |
| 无 | 雨 | 晴 |
| 南 | 阴 | 雨 |
| 北 | 晴 | 阴 |

# 最大熵模型：天气预报

- 问题定义
  - 用A表示条件集合，B表示结论集合。
    随机事件 $X=(\alpha,\beta)\in\xi=A\times B,\ \alpha\in A,\ \beta\in B$
  - 已经有一些X的样本：

$$X_1=\{\alpha_1,\ \beta_1\},\ X_2=\{\alpha_2,\ \beta_2\},\ \ldots\ ,X_n=\{\alpha_n,\ \beta_n\}$$

  - 假设已知 $\alpha_{n+1}$，如何预测 $\beta_{n+1}$ ？
  - 求解一个X的概率分布，使得：
    - X的分布与已知的样本分布一致
    - X的熵最大

# 最大熵模型：天气预报

- 建模
  - 建立天气预报模型：给出所有可能的条件下结论的概率 $p(\beta|\alpha)$，$\alpha\in A$ ，$\beta\in B$
  - 为简化问题起见，通常用联合概率模型取代上述的条件概率模型$p(\alpha,\beta)$
  - 二者关系： $p(\beta|\alpha)= p(\alpha,\beta)/p(\alpha)$
  - 由于条件和结论都是离散量，理论上，所有的可能性是可以穷举的，因此只要给出所有可能性的概率即可

# 最大熵模型：天气预报

■建模目标：模型能预测所有的可能，并给出概率

所有的可能性

| 昨天风向 | 昨天天气 | 今天天气 | 概率 |
|---|---|---|---|
| 无 | 晴 | 晴 | 0.011 |
| 无 | 晴 | 阴 | 0.003 |
| 无 | 晴 | 雨 | 0.002 |
| 无 | 阴 | 晴 | 0.008 |
| 无 | 阴 | 阴 | 0.013 |
| 无 | 阴 | 雨 | 0.005 |
| …… | …… | …… | |
| 北 | 雨 | 晴 | 0.009 |
| 北 | 雨 | 阴 | 0.001 |
| 北 | 雨 | 雨 | 0.003 |

概率之和为一

# 最大熵模型：天气预报

- 建模
  - 上述模型需要满足以下两个条件：
    - 模型的概率分布应尽可能与样本一致
      - 模型中，“晴、阴、雨”天的概率分布应与样本一致
      - 模型中，昨天为“晴”、“阴”、“雨”时第二天的天气概率分布应与样本一致
      - 模型中，昨天为“无风”、“南风”、“北风”时第二天的天气概率分布应与样本一致
      - 模型中，昨天天气与风向组合给定的情况下，第二天的天气概率分布应与样本一致
      - ……
    - 模型的熵最大！

# 最大熵模型：天气预报

- 优化目标
  - X熵最大：可以表示为：

$$\hat{p} = \underset{p}{\operatorname{argmax}}\, H(p)$$

  - X的分布与已知的样本分布一致：如何表示？
  - 实际上，由于A（预测的条件）的可能性数量可能极其巨大，穷举所有可能性是不现实的，如何对任意的条件和结论给出其概率？

# 最大熵模型：天气预报

- 引入特征
  - 实际问题中，由于条件α和结果β取值多样化，为模型表示方便起见，通常将条件α和结果β表示为一些二制特征
  - 特征$f_1, f_2, \ldots, f_n$定义如下：

    $f_i$:ε->{0,1}, ε=A( 预测条件)×B(结论)
  - 这样每个事件都可以表示为一个由特征值{0,1}组成的n维向量，不再直接用条件和结论来描述。

# 最大熵模型：天气预报

- 引入约束
  - “模型分布与样本分布一致”可以描述为：

$$E_p[f_i]=e_{\hat{p}}[f_i], i=1\ldots n, \text{这里 } p \text{ 为样本分布，} \hat{p} \text{ 为模型分布}$$

这个公式含义为：对于任何一个特征，模型和样本应该具有相同的均值（即：根据样本数据统计各种各种出现的次数）。由于特征取值只有{0,1}，因此这个公式实质上可以理解为：模型中任何一个特征为1的概率与样本应相同。

# 最大熵模型：天气预报

- 简单特征（单一特征）
  - 为天气预报引入以下简单特征：
    - f1：昨天天气=晴
    - f2：昨天天气=阴
    - f3：昨天天气=雨
    - f4：昨天风向=无
    - f5：昨天风向=南
    - f6：昨天风向=北
    - f7：今天天气=晴
    - f8：今天天气=阴
    - f9：今天天气=雨

# 最大熵模型：天气预报

■用简单特征表示样本：样本表示

| 昨天风向 | | | 昨天天气 | | | 今天天气 | | |
|---|---|---|---|---|---|---|---|---|
| 无 | 南 | 北 | 晴 | 阴 | 雨 | 晴 | 阴 | 雨 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

# 最大熵模型：天气预报

- 用简单特征表示样本：建模目标（模型能预测所有的可能，并给出概率）

所有的可能性

| 昨天风向 | | | 昨天天气 | | | 今天天气 | | | 概率 |
|---|---|---|---|---|---|---|---|---|---|
| 无 | 南 | 北 | 晴 | 阴 | 雨 | 晴 | 阴 | 雨 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.003 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0.004 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0.001 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| … | … | … | … | … | … | … | … | … | …… |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

概率之和为一

# 最大熵模型：天气预报

- 用简单特征建模
  - 利用上述简单特征，采用最大熵原理为天气预报建模，得到的模型是：
    - 模型中所有特征出现概率与样本一致：
      - 模型中“昨天天气为晴”概率与样本一致
      - 模型中“昨天天气为阴”概率与样本一致
      - ……
      - 模型中“今天天气为雨”概率与样本一致
    - 模型熵最大
  - 这个模型显然不是我们想要的模型，我们希望：
    - **模型中“昨天天气为晴”的情况下今天各种天气的概率**与样本一致
    - 模型中“昨天风向为南”的情况下今天各种天气的概率与样本一致
    - 模型中昨天天气和风向给定情况下今天各种天气的概率与样本一致
    - ……

# 最大熵模型：天气预报

- 复合特征（组合特征）
  - 为了使得到的模型满足我们所期望的约束条件，可以重新定义特征：
    - f1: 昨天天气为晴，且今天天气为晴
    - f2: 昨天天气为晴，且今天天气为阴
    - ……
    - fn: 昨天风向为北，天气为雨，且今天天气为雨
  - 按照这种方式定义特征，那么所有的特征都反映了某种预测条件和结果之间的依赖关系，这样得到的模型才是我们所期望的模型。

# 最大熵模型：天气预报

■用复合特征表示样本

所有的可能性

| $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | … | $f_{n-2}$ | $f_{n-1}$ | $f_n$ | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | … | 0 | 0 | 0 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 0 | 1 | 0.003 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 1 | 0 | 0.004 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 1 | 1 | 0.002 |
| 0 | 0 | 0 | 0 | 0 | … | 1 | 0 | 0 | 0.001 |
| 0 | 0 | 0 | 0 | 0 | … | 1 | 0 | 1 | 0.003 |
| … | … | … | … | … | … | … | … | … | …… |
| 1 | 1 | 1 | 1 | 1 | … | 1 | 0 | 1 | 0.003 |
| 1 | 1 | 1 | 1 | 1 | … | 1 | 1 | 0 | 0.007 |
| 1 | 1 | 1 | 1 | 1 | … | 1 | 1 | 1 | 0.010 |

概率之和为一

# 最大熵模型：天气预报

- 模型应用
  - 假设我们已经得到一个满足上述约束条件且熵最大的模型，如何预测天气？
  - 给定预测条件α（今天风向和今天天气），需要预测明天天气β∈{晴,阴,雨}，我们只要根据模型分别将(α,晴) / (α,阴) / (α,雨)转化为特征值，然后根据模型计算出相应的概率，取概率最大者即可。

# 最大熵模型：天气预报

- 最优化：学习过程
  - 根据前面的定义，最大熵模型的参数估计可以表示为一个约束条件下的极值问题
    - 在以下约束条件下：

$$E_p[f_i]=e_{\hat{p}}[f_i], i=1\ldots n,$$

这里 $p$ 为样本分布，$\hat{p}$ 为模型分布

    - 求解熵最大的模型：

$$\hat{p}=\arg\max_{p\in C} H(p)$$

$C$为所有可能的模型组成的集合

# 最大熵模型：天气预报

- 最优化：学习过程
  - 经推导（过程略），满足上述约束条件的最大熵模型具有如下形式：

$$p(\beta|\alpha)=\frac{\exp\left(\sum_{(i=1)}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$

  - 其中，Z为规范化因子。$\lambda_i$ 是一组参数（权重），其中每个参数对应于一个特征
  - 学习参数：一组$\lambda_i$

# 最大熵模型：天气预报

- 用最大熵模型预测
  - 给定条件α，结论为β的概率可以表示为：

$$p(\beta|\alpha)=\frac{\exp\left(\sum_{(i=1)}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$

  - 给定条件α，最优的β可以表示为：

$$\hat{\beta}=\underset{\beta}{\operatorname{argmax}}(\beta|\alpha)=\underset{\beta}{\operatorname{argmax}}\frac{\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$
$$=\underset{\beta}{\operatorname{argmax}}\sum_{i=1}^{n}\lambda_i f_i(\alpha,\beta)$$

  - 可以看出，一旦得到这组参数$\lambda_i$，那么对于给定的条件α，对所有的结论β，只要将其所有值为1特征$f_i$对应的$\lambda_i$加起来，取和最大的β即可

# 最大熵模型

- 工具
  - 最普遍使用的工具：Maximum Entropy Modeling Toolkit for Python and C++
  - 作者：张乐
  - 主页：http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

# 基于最大熵模型的中文分词

- 采用最大熵模型对每个汉字标注BMES标记
- 假设当前字$C_0$，当前标记是$T_0$
- 常用最大熵特征模板（当前字是$C_0$）：
  - $C_n T_0 (n = -2, -1, 0, 1, 2)$：汉字
    - 例如：$C_{-1}T_0$表示当前字的前一个字和当前标记
      $C_2T_0$表示当前字的前一个字
  - $C_n C_{n+1} T_0 (n = -2, -1, 0, 1)$：两字组
  - $C_{-1}C_1$ T0 ：当前字左右两个字
  - $D(C_0)T_0$ ：当前字是否数字
  - $A(C_0)T_0$ ：当前字是否字母
  - $P(C_0)T_0$ ：当前字是否标点

# 基于最大熵模型的中文分词

- 生成最大熵训练实例
  - 假设给定训练样本：

    <s> 今天 是 星期三 。 </s>
  - 生成对应标注序列：

    今|B 天|E 是|S 星|B 期|M 三|E 。|S
  - 对于每一个字生成一个训练实例：上例中天字生成的实例

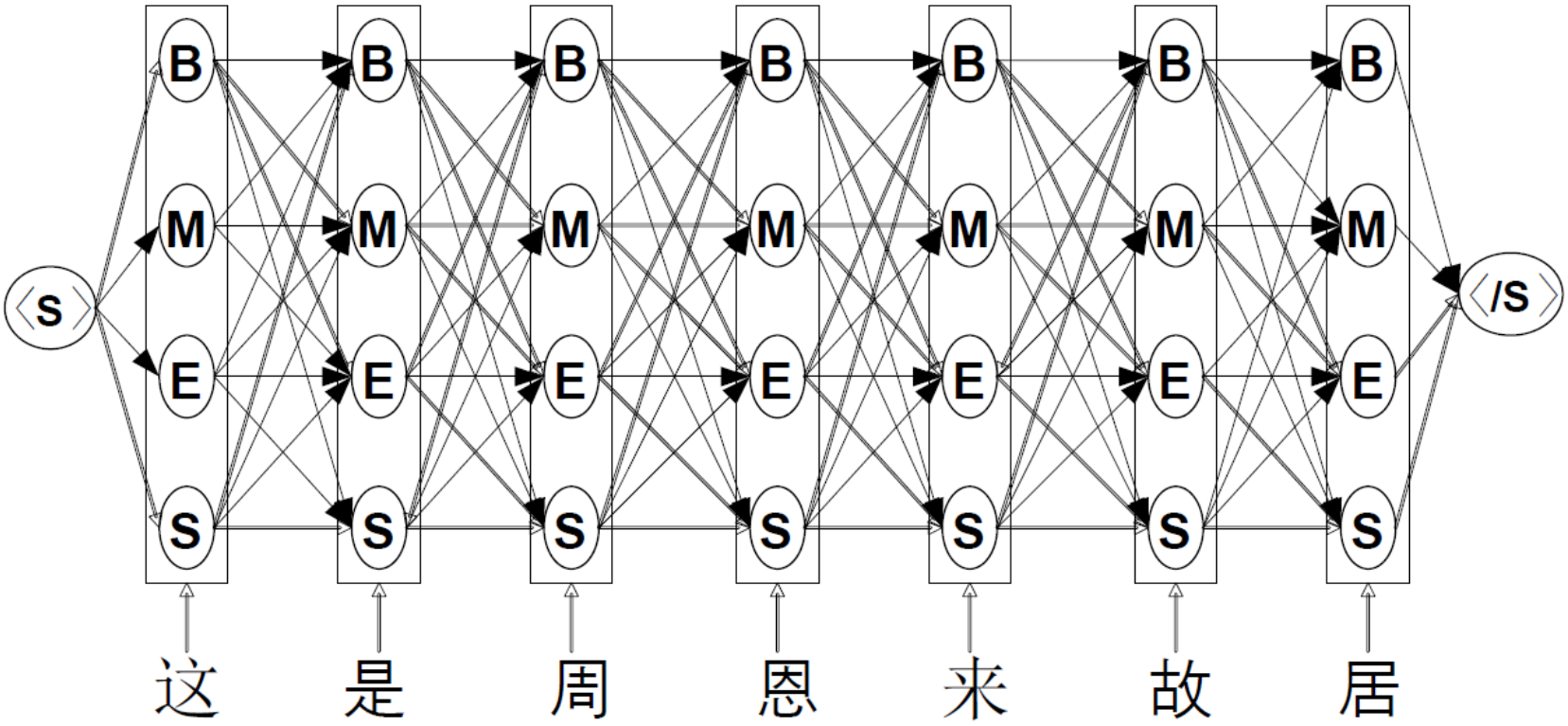
| | | | |
|---|---|---|---|
| $C_{-2}T_0$=<s>E | $C_{-1}T_0$= 今 E | $C_0T_0$= 天 E | $C_1T_0$= 是 E |
| $C_2T_0$= 星 E | $C_{-2}C_{-1}T_0$=<s> 今 E | $C_{-1}C_0T_0$= 今天 E | |
| $C_0C_1T_0$= 天是 E | $C_1C_2T_0$= 是星 E | $C_{-1}C_1T_0$= 今是 E | |
| $D(C_0)T_0$= 否 | $A(C_0)T_0$= 否 | $P(C_0)T_0$= 否 | |

# 基于最大熵模型的中文分词

- 最大熵模型的参数训练
  - 对训练语料库中每个汉字生成一个实例
  - 把所有实例的列表送给最大熵模型的训练工具
  - 最大熵模型的训练工具将学习到模型的参数
- 最大熵模型的测试
  - 对于输入串中每一个汉字以及该汉字的每一个可能的标记，生成一个实例，这样对每个汉字就生成了四个实例
  - 对这每个实例，生成其所有激活的特征，计算这些特征所对应的概率
  - 选取可能性最大的标记为该字的标记

# 基于最大熵模型的中文分词

■搜索最优标注路径

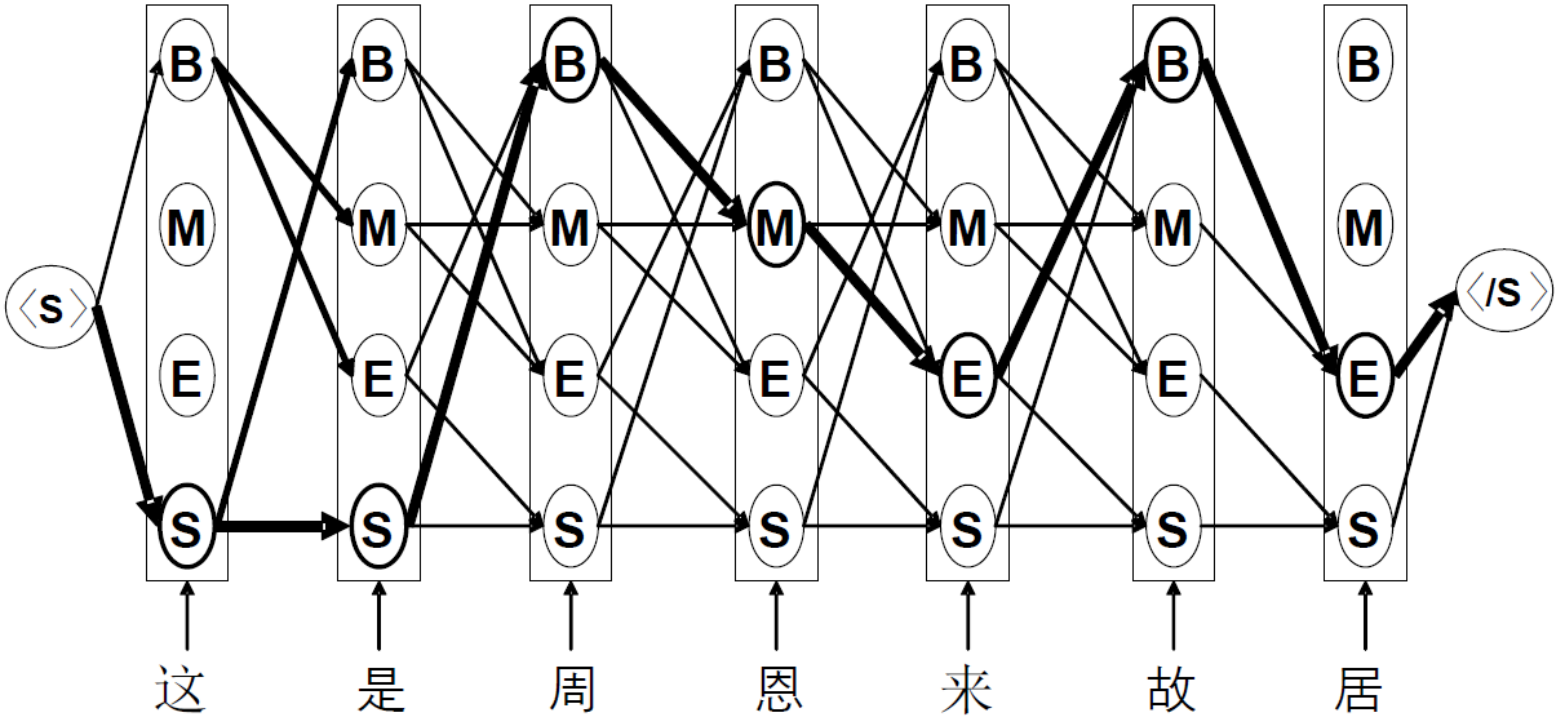


# 基于最大熵模型的中文分词

- 搜索最优标注路径
  - 有些边是非法的，比如M→B，E→M，E→E，S→M，S→E，等等
  - 每条路径的概率是路径上各节点标注概率P(标记|汉字)之积，边上没有转移概率
  - 搜索算法与HMM模型解码的Viterbi算法相同

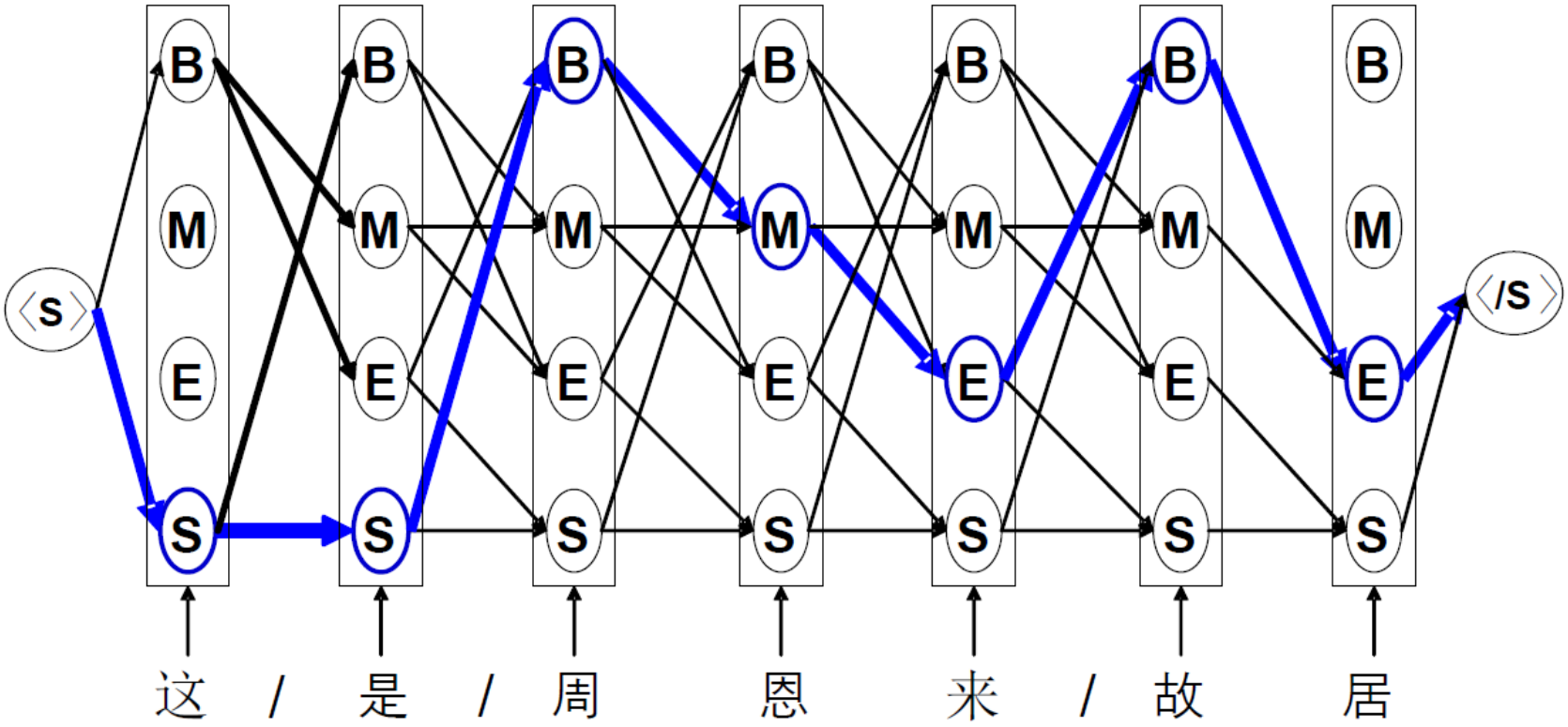
# 基于最大熵模型的中文分词

■搜索最优标注路径



# 基于最大熵模型的中文分词

■搜索最优标注路径



# 基于统计的分词方法：判别式方法

- 优点
  - 分词精度高
  - 新词识别率较高
- 缺点
  - 训练速度慢
  - 性能与特征紧密相连
  - 需要人工标注训练语料

# 现有分词方法总结

| 比较项目 | 基于词典的分词 | 产生式分词 | 判别式分词 |
| --- | --- | --- | --- |
| 分词精度 | 80% | 90% | 95% |
| 分词速度 | 最快 | 最慢 | 较快 |
| 理论基础 | 没有 | 不扎实 | 扎实 |
| 学习素材 | 不需要 | 大规模 | 中小规模 |
| 词典 | 需要 | 需要 | 不需要 |
| 学习效率 | 无需学习 | 高 | 中低 |
| 歧义处理能力 | 有限 | 中等 | 强 |
| 新词处理能力 | 不能 | 中等 | 强 |

# 中文分词的技术水平

- 2003年国家863评测部分结果
  - 最好成绩：P=93.44%，R=93.69%，F1=93.46%
  - 最差成绩：P=91.42%，R=89.27%，F1=90.33%
- 2005年SIGHAN汉语分词评测结果（使用MSR语料）

| 评测方式 | 系统排名 | 性能指标 | | | | |
|---|---|---|---|---|---|---|
| | | 召回率 | 精确率 | $F1$ | $R_{oov}$ | $R_{iv}$ |
| 封闭测试 | 最好 | 0.962 | 0.966 | 0.964 | 0.717 | 0.968 |
| | 最差 | 0.898 | 0.896 | 0.897 | 0.327 | 0.914 |
| 开放测试 | 最好 | 0.980 | 0.965 | 0.972 | 0.59 | 0.99 |
| | 最差 | 0.788 | 0.818 | 0.803 | 0.37 | 0.8 |

$R_{oov}$表示集外词的召回率，$R_{iv}$表示集内词的召回率。

# 中文分词的技术水平

■2010年SIGHAN评测结果

| | | | Closed Track | | | | | | Open Track | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | OOV | ID | R | P | F | $R_{oov}$ | $R_{iv}$ | ID | R | P | F | $R_{oov}$ | $R_{iv}$ |
| S | L | 0.069 | S5 | 0.945 | 0.946 | 0.946 | 0.816 | 0.954 | S6 | 0.958 | 0.953 | 0.955 | 0.655 | 0.981 |
| | C | 0.152 | S6 | 0.953 | 0.95 | 0.951 | 0.827 | 0.975 | S9 | 0.95 | 0.95 | 0.95 | 0.82 | 0.973 |
| | M | 0.11 | S6 | 0.942 | 0.936 | 0.939 | 0.75 | 0.965 | S9 | 0.94 | 0.936 | 0.938 | 0.768 | 0.962 |
| | | | | | | | | | S18 | 0.941 | 0.935 | 0.938 | 0.787 | 0.96 |
| | F | 0.087 | S6 | 0.959 | 0.96 | **0.959** | **0.827** | 0.972 | S9 | 0.96 | 0.96 | **0.96** | **0.847** | 0.971 |
| T | L | 0.094 | S10 | 0.942 | 0.942 | 0.942 | 0.788 | 0.958 | S1 | 0.905 | 0.9 | 0.902 | **0.775** | 0.918 |
| | C | 0.094 | S10 | 0.948 | 0.957 | 0.952 | 0.666 | 0.977 | S1 | 0.911 | 0.924 | 0.918 | 0.698 | 0.933 |
| | M | 0.075 | S10 | 0.953 | 0.957 | 0.955 | 0.798 | 0.966 | S1 | 0.903 | 0.903 | 0.903 | 0.729 | 0.917 |
| | F | 0.068 | S10 | 0.964 | 0.962 | **0.963** | **0.812** | 0.975 | S1 | 0.903 | 0.916 | **0.91** | **0.721** | 0.916 |

# 可选的分词工具

| | 分词系统 | 标识 |
|---|---|---|
| 1 | BosonNLP | BOSON |
| 2 | IKAnalyzer | 中文分词库 IKAnalyzer |
| 3 | NLPIR | NLPIR 汉语分词系统 又名：ICTCLAS 2015 |
| 4 | SCWS | SCWS 中文分词 |
| 5 | 结巴分词 | jieba |
| 6 | 盘古分词 | 盘古分词-开源中文分词组件 |
| 7 | 庖丁解牛 | paoding Lucene中文分词"庖丁解牛" Paoding Analysis |
| 8 | 搜狗分词 | Sogou搜狗 云服务 |
| 9 | 腾讯文智 | 互联网+ 腾讯云 |
| 10 | 新浪云 | 新浪云 |
| 11 | 语言云 | 語言 |

- http://ictclas.nlpir.org/中科院计算所(ICTCLAS)
- http://www.openpr.org.cn中科院自动化所(Urheen)
- http://www.fnlp.org/复旦大学
- http://nlp.stanford.edu/software/tagger.shtml Stanford University

# 当前分词技术存在的主要问题

- 训练语料规模小：分词模型过于依赖训练样本，而标注大规模训练样本费时费力，由此导致分词系统对新词的识别能力差，往往在与训练样本差异较大的测试集上性能大幅度下降

- 训练语料领域少：现有的训练样本主要在新闻领域，而实际应用千差万别：网络新闻、微博/微信/QQ等对话文本、不同的专业领域(中医药、生物、化学、能源……)。

- 对实体和专有名词的识别性能较低

# 提纲

- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 命名实体

- 命名实体的定义
  - 狭义地讲,命名实体指现实世界中具体或抽象的实体,如人（张三）、机构（中国中文信息学会、阿里巴巴网络技术有限公司）、地点等,通常用唯一的标志符(专有名称)表示,如人名、机构名、地名等。
  - 广义地讲,命名实体还可以包含时间（12：00）、日期（2017年10月17日）、数量表达式（100）、金钱（一亿美金）等。至于命名实体的确切含义,只能根据具体应用来确定。比如,在具体应用中,可能需要把住址、电子信箱地址、电话号码、舰船编号、会议名称等作为命名实体。

# 命名实体识别的任务

- 一般而言，主要是识别出待处理文本中七类（人名、机构名、地名、时间、日期、货币和百分比）命名实体
- 两个子任务：实体边界识别和确定实体类别

2011年4月11日17点16分，日本东北部的福岛和茨城地区发生里氏7.0级强烈地震（震中北纬36.9度、东经140.7度，即福岛西南30公里左右的地方，震源深度10公里，属于浅层地震）当局已经发布海啸预警震后约30分钟后在日本海地区发生巨型海啸，同时造成福岛核电站出现核泄漏震后第十天，国际原子能机构对于日本政府反应迟钝进行了谴责

→

**2011年4月11日17点16分**，日本东北部的**福岛和茨城地区**发生里氏7.0级强烈地震（**震中北纬36.9度、东经140.7度**，即福岛西南30公里左右的地方，震源深度10公里，属于浅层地震）当局已经发布海啸预警震后约30分钟后在日本海地区发生巨型海啸，同时造成**福岛核电站**出现核泄漏震后第十天，**国际原子能机构**对于**日本政府**反应迟钝进行了谴责

# 命名实体识别的特点

- 时间、日期、货币和百分比的构成有比较明显的规律，识别起来相对容易
- 人名、地名、机构名的用字灵活，识别的难度很大
  - 在不同领域、 场景下, 命名实体的外延有差异 。
  - 数量巨大,不能枚举,难以全部收录在词典中。
  - 某些类型的实体名称变化频繁,并且没有严格的规律可以遵循。
  - 表达形式多样。
  - 首次出现后往往采用缩写形式。

# 中文人名识别

- 人名识别在英文中已得到很好的研究,
  - 因为英文本身具有一些明显特征(如大小写)
    - Barack Obama
    - Donald Trump
    - ……
  - 不存在切分造成的错误，使得人名识别在英文中变得较为容易。

# 中文人名识别

- 中文人名识别的难点:
  - 名字用字范围广，分布松散，规律不很明显
  - 姓氏和名字都可以单独使用用于特指某一人
    - 王二小，小王，二小
  - 许多姓氏用字和名字用字(词)可以作为普通用字或词被使用
    - 姓氏为普通词：于(介词)，张(量词)，江(名词)等
    - 名字为普通词：建国，国庆，胜利，文革，计划
    - 全名也是普通词汇，如：万里，温馨，高山，高升，高飞，周密，江山，夏天等。
  - 缺乏可利用的启发标记（人名与其上下文组合成词）
    - 祝贺老总百战百胜。
    - 这里有关天培的壮烈

# 中文人名识别

- 中文人名的特点:
  - 中国人名一般由以下部分组合而成：
    - 姓：张、王、李、刘、诸葛、西门、范徐丽泰
    - 名：李素丽，张华平，王杰、诸葛亮
    - 前缀：老王，小李
    - 后缀：王老，赵总
  - 中国人名各组成部分用字比较有规律

根据统计, 汉语姓氏大约有1000多个, 姓氏中使用频度最高的是“王”姓, “王, 陈, 李,张, 刘” 等5个大姓覆盖率达32%, 姓氏频度表中的前14个高频度的姓氏覆盖率为50%,前400个姓氏覆盖率达99%。 人名的用字也比较集中。 频度最高的前6个字覆盖率达10.35%, 前10个字的覆盖率达14.936%, 前15个字的覆盖率达19.695%, 前400个字的覆盖率达90%。

# 中文人名识别

- 中文人名的特点:
  - 中国人名各组成部分的组合规律
    - 姓 + 名
    - 姓
    - 名
    - 前缀 + 姓
    - 姓 + 后缀
    - 姓 + 姓 + 名（海外已婚妇女）
  - 中国人名的上下文构成规律
    - 身份词：
      - 前：工人、教师、影星、犯人
      - 后：先生、同志
      - 前后：女士、教授、经理、小姐、总理

# 中文人名识别

- 中文人名的特点:
  - 地名或机构名：
    - 前：静海县大丘庄禹作敏
  - 的字结构
    - 前：年过七旬的王贵芝
  - 动作词：
    - 前：批评，逮捕，选举
    - 后：说，表示，吃，结婚

# 中文地名识别

- 中文地名识别的难点:
  - 地名数量大，缺乏明确、规范的定义。

    中国地名委员会编写了《中华人民共和国地名录》，收集了全国乡镇以上（含乡镇）各级行政区域的名称，以乡镇人民政府所在地为主的居民聚落名称，山、河、湖、海、岛、高原、盆地、沙溪等自然地理实体名称，名胜古迹、纪念地、古遗址、水库、桥梁、电站等名称。共收录地名10万多条。这个地名录中使用的汉字共2662个，频度最高的前65个汉字占总频度的50.22%，前622个汉字占总频度的90.01%，前1872个汉字占总频度的99%。
  - 真实语料中地名出现情况复杂。如地名简称、地名用词与其他普通词冲突、地名是其他专用名词的一部分，地名长度不一等。

# 中文地名识别

- 中文地名识别的特点:
  - 地名中使用的多字词中绝大部分是两个字
    - 山口
    - 水库
    - 遗址
    - 城镇
    - .....
  - 经常与方位词和介词连用
    - 东、南、西、北
    - 周边、附近
    - 在、于
    - .....

# 中文机构名识别

- 中文地名识别的难点:
  - 机构名中含有大量的人名、地名、企业字号等专有名称。在这些专有名称中，地名所占的比例最大，其中未登录地名又占了相当一部分的比例。所以机构名识别应在人名、地名等其他专名识别之后进行，其他专名识别的正确率对机构名识别正确率有较大影响。
    - 地名：山东大学，上海大学，北京大学
    - 人名：哈弗大学，逸夫图书馆，中山大学
    - 学科、专业和部分系统：公安部，教育部
    - 研究、生产或经营等活动的对像：软件研究所，卫星制造厂
    - 上述情况的综合：白求恩医科大学
    - 大机构、团体、组织和职业的名称：中国人民解放军洛阳外国语学院，中国发明家学会等

# 中文机构名识别

- 中文机构名识别的难点:
  - 用词广泛。通过对人民日报1998年1月中的10817个机构名所含的19986个词进行统计，共计27种词，其中名词最多（9941个），地名其次（5023个），以下依次为简称（1169个）、专有名词（1125个）、动词（848个）以及机构名（714个）等
  - 机构名长度极其不固定
    - 新疆维吾尔自治区乌鲁木齐国家高新技术产业开发区社会管理综合治理委员会学校及周边治安综合治理工作领导小组
    - 北大
  - 机构名很不稳定。随着社会发展，新机构不断涌现，旧机构不断被淘汰、改组或更名

# 中文机构名识别

- 中文机构名识别的特点:
  - 词法角度：偏正式(修饰格式)的复合词

    {名词|形容词|数量词|动词} + 名词

  - 句法角度：“定语 + 名词性中心语”型的名词短语(定名型短语)

  - 中心语：机构称呼词，如：大学，学院，研究所，学会，公司，股份公司等

# 音译名识别

- 音译名用字非常集中《英语姓名译名手册》中共收英语姓氏, 教名约4万个, 经计算机统计得出英语姓名译名用字表共476 个：

“ 啊阿埃艾爱昂奥巴白柏拜班邦包保堡鲍北贝倍本比彼边别滨宾玻波博勃伯卜布采蔡藏策查察昌彻陈楚垂茨慈次聪存措达大戴代丹当道德得登邓迪底地蒂第帝丁东杜敦顿多厄恩耳尔法凡范方菲费芬丰冯佛夫福弗辅富盖甘冈高哥戈葛格各根贡古顾瓜圭郭果哈海罕翰汉杭豪赫黑亨洪侯胡华怀惠霍基吉季计嘉佳加贾简姜焦杰捷金津京久居喀卡开凯坎康考柯科可克肯孔扣寇库夸匡奎魁坤昆阔拉腊莱来赖兰朗劳勒乐雷黎理李里礼荔丽历利立莲连廉良列琳林霖龄留刘流柳龙隆卢鲁露路吕略伦萝罗洛玛马麦迈满曼芒茅梅门蒙孟米密敏明名摩莫墨默姆木穆拿娜纳乃奈南内嫩能妮尼年涅宁牛纽农努女诺欧帕派潘庞培佩彭蓬皮匹平泼朴普漆奇齐契恰钱强乔切钦琴青琼丘邱屈让热仁日荣茹儒瑞若撒萨塞赛三缮桑瑟森莎沙珊山尚绍舍申生盛圣施诗石什史士寿舒朔斯思丝松孙索所塔泰坦汤唐陶特藤提惕田铁汀廷亭通透图托脱娃瓦万旺威韦为维伟魏卫温文翁沃乌武伍西锡希悉席霞夏显香向晓肖歇谢欣辛兴幸姓雄休修雪逊雅亚延扬阳尧耀耶叶依易意因英永尤雨约宰赞早泽曾扎詹湛章张哲者珍真芝知智治朱卓兹子宗祖佐丕谟葆薇岑弼娅缪珀瑙赉滕斐熙鸠窦艮麟黛”。

# 音译名识别

- 音译名内部很难划分出结构，但有一些常见音节，如“斯基、斯坦”等
- 不同语言的音译规律不尽相同，如法语、俄语、蒙古语译名用字与英语就有较大区别（蒙语人名举例：“那顺乌日图、青格勒图”），如果按不同的语言训练不同的模型可能会比使用统一的模型效果更好
- 音译名可以是人名、地名或其他专名，上下文规律差别较大

# 命名实体识别的方法

- 方法分类
  - 有词典切分/无词典切分

    在分词的过程中使用词典的方法是有词典切分，反之是无词典切分，有字典切分的方法一般是基于规则的，无词典切分的方法一般是基于统计的

  - 基于规则的方法/基于统计的方法

    基于规则的方法不需要标注训练语料，能直接根据词典和规则进行分词，基于统计的方法需要标注训练语料训练模型。基于统计的方法可以分为生成式统计分词和判别式统计分词

# 基于字典的命名实体识别方法

- 方法概述：按照一定的策略将待分析的汉字串与一个充分大的词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功

- 典型方法
  - 正向最大匹配法
  - 反向最大匹配法
  - 最短路径法（最少分词法）

- 方法与之前介绍的分词中的方法类似

# 基于统计的命名实体识别方法

- 生成式方法
  - 原理：首先建立学习样本的生成模型，再利用模型对预测结果进行间接推理
  - 典型算法：

    HMM, PCFG等
- 判别式方法
  - 原理：由字构词的命名实体识别理念，将NER问题转化为判别式分类问题(序列标注问题)
  - 典型算法：

    Maxent　SVM　CRF　CNN　RNN　LSTM+CRF

# LSTM+CRF命名实体识别（Lample NAACL16）

■动机1

语言领域内资源或特征的获取耗时耗力（例如地理资源集gazetteers），避免使用语言领域内的资源或特征

■动机2

大多数领域内的命名实体识别标注语料很少，但是存在很多未标注语料，为了在少量的标注语料上取得较好的效果

■动机3

传统的特征提起过程中会用到复杂的NLP工具，会造成误差传递

# LSTM+CRF命名实体识别（Lample NAACL16）

- 特征等级1

  词级别（token-level）特征，基础的命名实体识别特征，包含每个词在当前语义中属于哪个实体类别

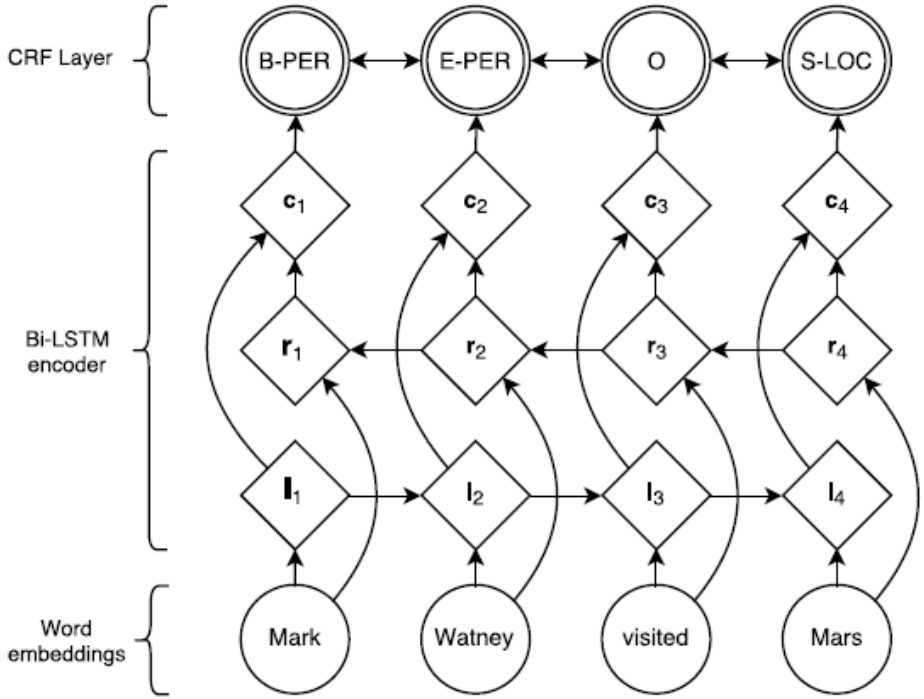
- 特征等级2

  字符级别（character-level）特征，词语的拼写特征、形态特征、词序特征等也包含实体类别的信息

# LSTM+CRF命名实体识别（Lample NAACL16）

■LSTM+CRF命名实体识别联合模型

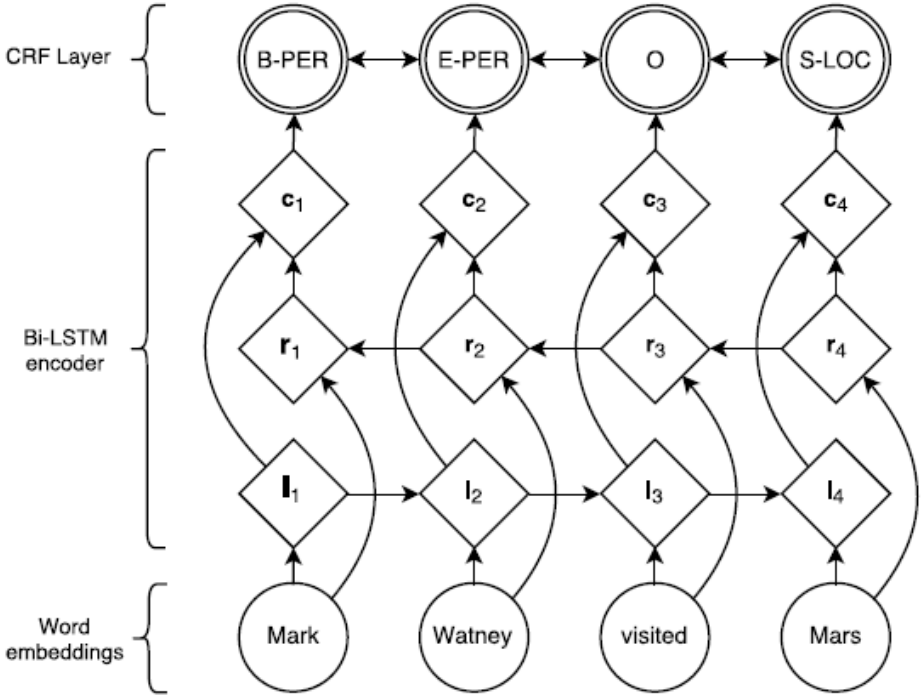
每个句子按照词序逐个输入双向LSTM中，结合正反向隐层输出得到包含每个词类别信息的表示，输入CRF中，优化目标函数，从而得到每个词所属的实体类别



# LSTM+CRF命名实体识别（Lample NAACL16）

## ■LSTM+CRF命名实体识别联合模型

每个词对应的LSTM输出为包含该词语实体类别信息的$1 \times k$ 维向量（k为实体tag的个数），拼接成矩阵P，作为CRF中特征矩阵输入




$$(x_1, \ldots\ldots, x_n)$$

$$\downarrow$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i)$$

$$\mathbf{c}_t = (1 - \mathbf{i}_t) \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

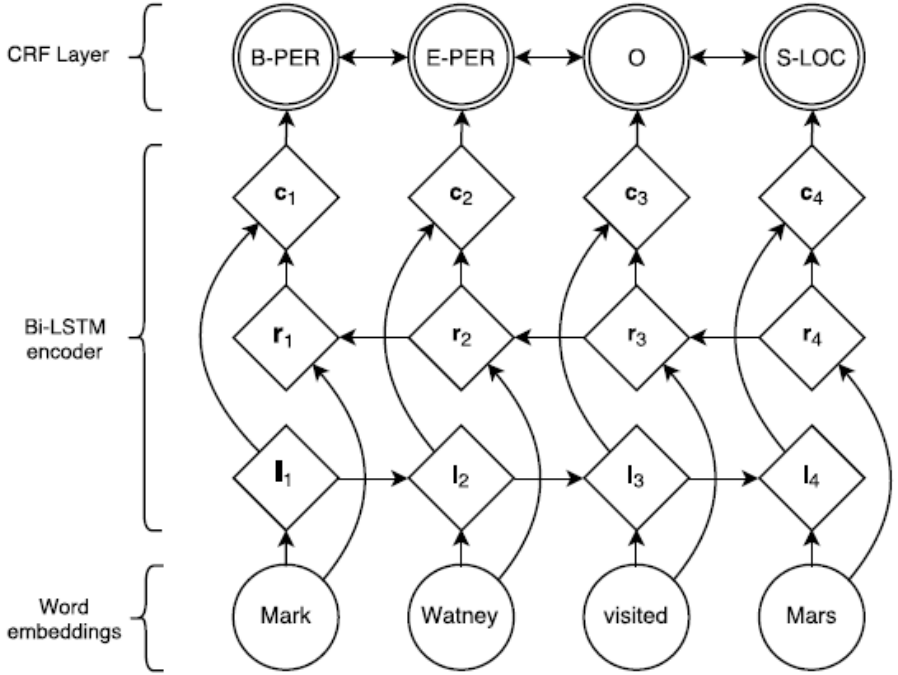
$$\downarrow$$

$$(h_1, \ldots\ldots, h_n) \quad h_t = [\vec{h}_t; \overleftarrow{h}_t] \rightarrow P^{n \times k}$$

# LSTM+CRF命名实体识别（Lample NAACL16）

## ■LSTM+CRF命名实体识别联合模型

由转移概率矩阵A（标签i转移到标签j的概率）和特征矩阵P构成的得分函数，即为CRF的特征函数。对每一个句子序列X，给定任意一组标签Y，通过特征得分函数S计算条件概率，进而通过对条件概率求log似然得到目标函数，进行联合训练，同时优化LSTM与CRF的参数




$$\forall(y_1,\ldots\ldots,y_n)$$

$$\downarrow$$

$$s(\mathbf{X},\mathbf{y}) = \sum_{i=0}^{n} A_{y_i,y_{i+1}} + \sum_{i=1}^{n} P_{i,y_i}$$

$$\downarrow$$

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\widetilde{\mathbf{y}}\in\mathbf{Y}_\mathbf{X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}}.$$

$$\downarrow$$

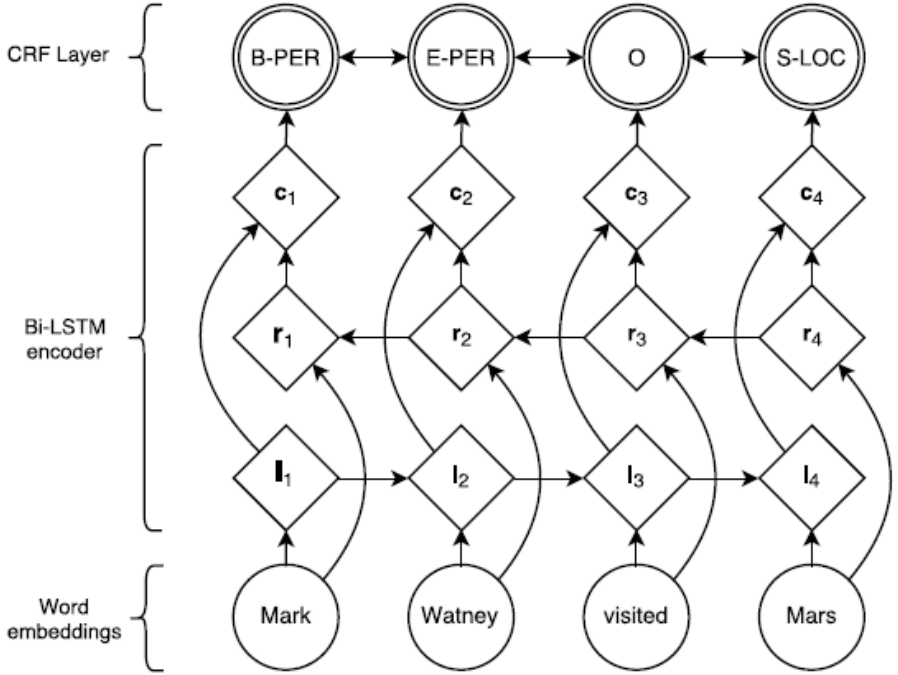
$$\log(p(\mathbf{y}|\mathbf{X})) = s(\mathbf{X},\mathbf{y}) - \log\left(\sum_{\widetilde{\mathbf{y}}\in\mathbf{Y}_\mathbf{X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}\right)$$

$$= s(\mathbf{X},\mathbf{y}) - \operatorname*{logadd}_{\widetilde{\mathbf{y}}\in\mathbf{Y}_\mathbf{X}} s(\mathbf{X},\widetilde{\mathbf{y}}), \quad ($$

# LSTM+CRF命名实体识别（Lample NAACL16）

## ■LSTM+CRF命名实体识别联合模型

训练得到模型参数后，在该参数条件下，对于一给定句子序列X，使 $S(X,Y)$ 得分最高的标签序列Y就为该句子序列的实体标签序列Y

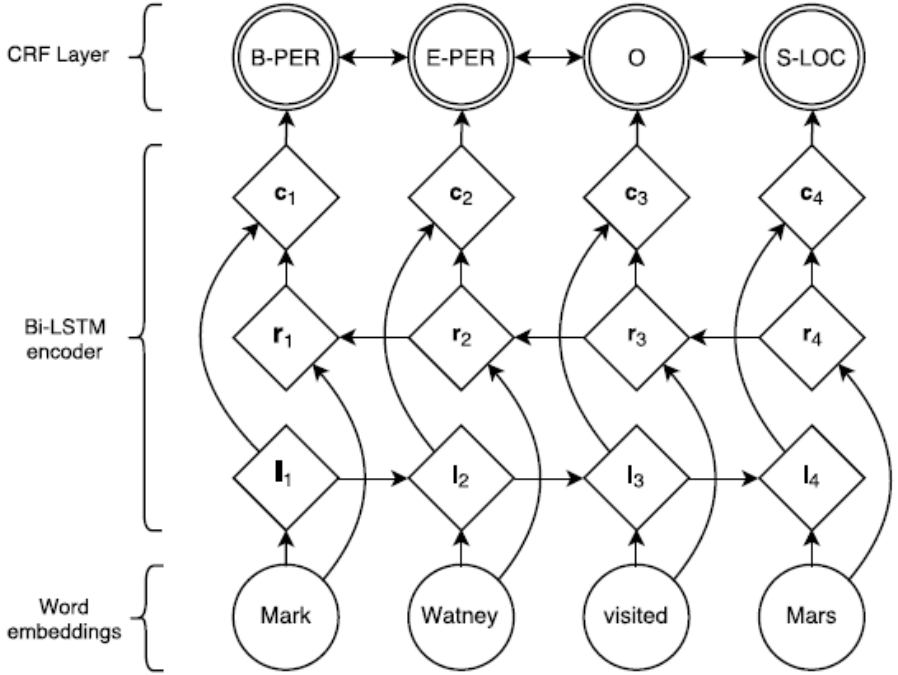


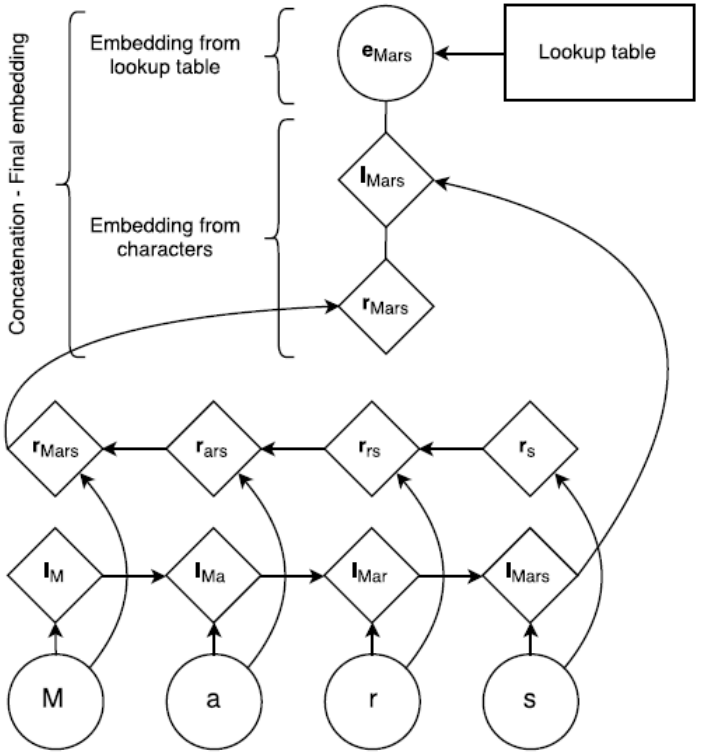

$$\mathbf{y}^* = \underset{\tilde{\mathbf{y}} \in \mathbf{Y}_{\mathbf{X}}}{\operatorname{argmax}}\, s(\mathbf{X}, \tilde{\mathbf{y}}).$$

# LSTM+CRF命名实体识别（Lample NAACL16）

## ■词级别特征与字符级别特征

词级别的特征即每个词的词向量作为词的表示，字符级别特征是通过将每个词拆成字符序列，通过双向LSTM得到该词的表示，将两种表示结合，作为该词的LSTM+CRF模型输入



# LSTM+CRF命名实体识别（Lample NAACL16）

## ■数据集与实验

数据集：CoNLL2003

| Model | $F_1$ |
|---|---|
| Collobert et al. (2011)* | 89.59 |
| Lin and Wu (2009) | 83.78 |
| Lin and Wu (2009)* | 90.90 |
| Huang et al. (2015)* | 90.10 |
| Passos et al. (2014) | 90.05 |
| Passos et al. (2014)* | 90.90 |
| Luo et al. (2015)* + gaz | 89.9 |
| Luo et al. (2015)* + gaz + linking | **91.2** |
| Chiu and Nichols (2015) | 90.69 |
| Chiu and Nichols (2015)* | 90.77 |
| LSTM-CRF (no char) | 90.20 |
| LSTM-CRF | **90.94** |

**Table 1:** English NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Florian et al. (2003)* | 72.41 |
| Ando and Zhang (2005a) | 75.27 |
| Qi et al. (2009) | 75.72 |
| Gillick et al. (2015) | 72.08 |
| Gillick et al. (2015)* | 76.22 |
| LSTM-CRF – no char | 75.06 |
| LSTM-CRF | **78.76** |
| S-LSTM – no char | 65.87 |
| S-LSTM | 75.66 |

**Table 2:** German NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Carreras et al. (2002) | 77.05 |
| Nothman et al. (2013) | 78.6 |
| Gillick et al. (2015) | 78.08 |
| Gillick et al. (2015)* | **82.84** |
| LSTM-CRF – no char | 73.14 |
| LSTM-CRF | **81.74** |

**Table 3:** Dutch NER (CoNLL-2002 test set).

# LSTM+CRF命名实体识别（Lample NAACL16）

## ■数据集与实验

数据集：CoNLL2003

| Model | $F_1$ |
|---|---|
| Collobert et al. (2011)* | 89.59 |
| Lin and Wu (2009) | 83.78 |
| Lin and Wu (2009)* | 90.90 |
| Huang et al. (2015)* | 90.10 |
| Passos et al. (2014) | 90.05 |
| Passos et al. (2014)* | 90.90 |
| Luo et al. (2015)* + gaz | 89.9 |
| Luo et al. (2015)* + gaz + linking | **91.2** |
| Chiu and Nichols (2015) | 90.69 |
| Chiu and Nichols (2015)* | 90.77 |
| LSTM-CRF (no char) | 90.20 |
| LSTM-CRF | **90.94** |

**Table 1:** English NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Florian et al. (2003)* | 72.41 |
| Ando and Zhang (2005a) | 75.27 |
| Qi et al. (2009) | 75.72 |
| Gillick et al. (2015) | 72.08 |
| Gillick et al. (2015)* | 76.22 |
| LSTM-CRF – no char | 75.06 |
| LSTM-CRF | **78.76** |
| S-LSTM – no char | 65.87 |
| S-LSTM | 75.66 |

**Table 2:** German NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Carreras et al. (2002) | 77.05 |
| Nothman et al. (2013) | 78.6 |
| Gillick et al. (2015) | 78.08 |
| Gillick et al. (2015)* | **82.84** |
| LSTM-CRF – no char | 73.14 |
| LSTM-CRF | **81.74** |

**Table 3:** Dutch NER (CoNLL-2002 test set).

# 汉语分词与实体识别联合模型（PengACL16）

- 动机1

  词边界能提供实体类别的有用信息，所以将词边界信息融入到命名实体识别中

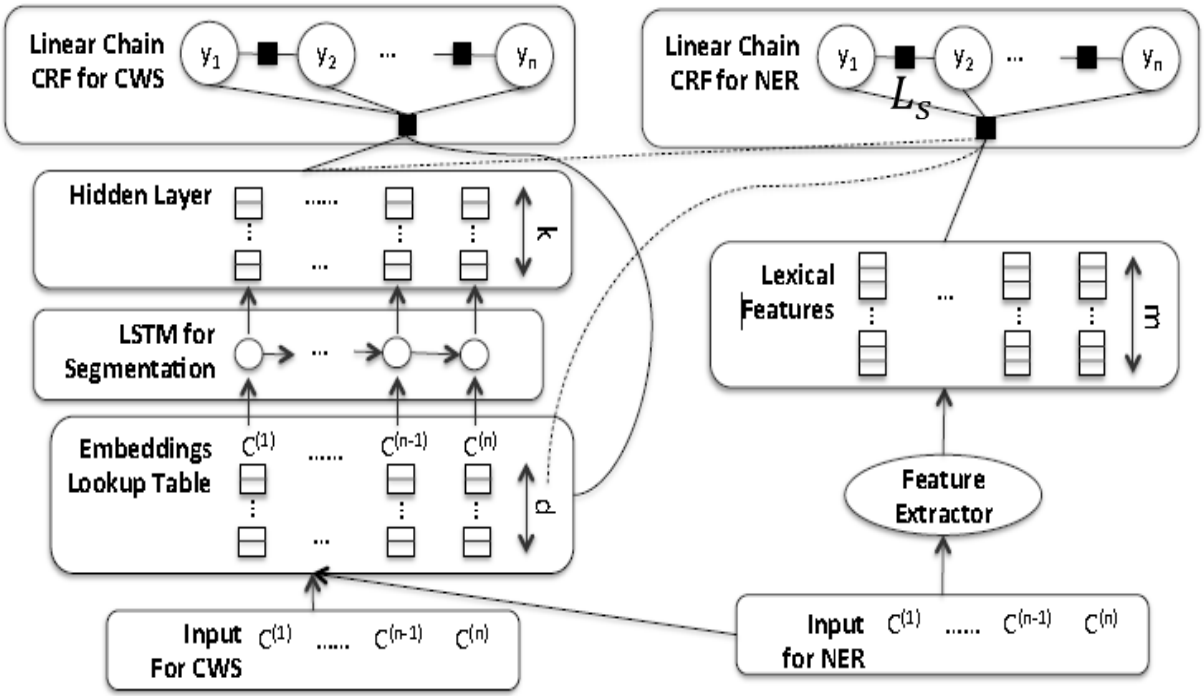
- 动机2

  汉语分词困难，与英语相比较分词准确率低，先分词，后实体识别，分词时的误差会传递到实体识别中，所以将分词与实体识别联合训练

# 汉语分词与实体识别联合模型（PengACL16）

## ■汉语分词与实体识别联合模型

汉语分词模型：LSTM+CRF 将句子序列输入双向LSTM中，通过LSTM+CRF模型进行汉语分词训练。通过改模型得到分词阶段的损失函数 $L_S$

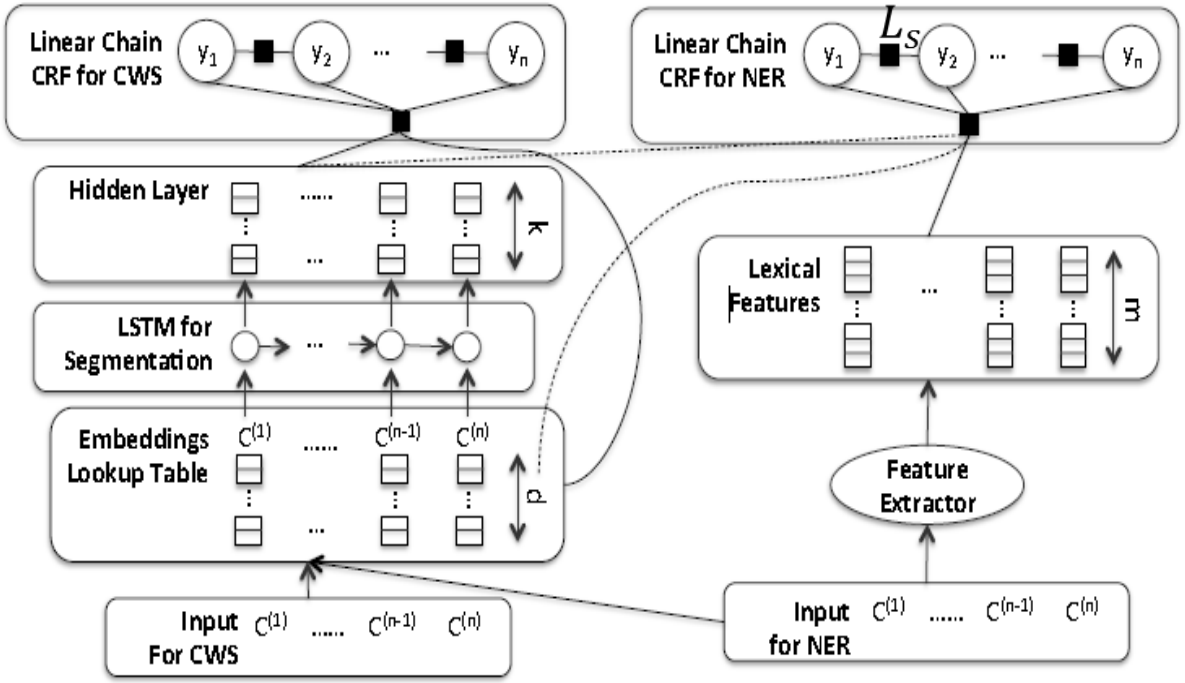



$$\mathcal{L}_s(\boldsymbol{y}_s;\boldsymbol{x}_s,\Theta)=\frac{1}{K}\sum_k\Big[\log\frac{1}{Z(\boldsymbol{x}_s)^k}+\sum_i\Big(T_s(y^k_{i-1},y^k_i)+s(y^k_i;\boldsymbol{x}^k_s,\boldsymbol{\Lambda}_s)\Big)\Big]$$

# 汉语分词与实体识别联合模型（PengACL16）

## ■汉语分词与实体识别联合模型

实体识别模型：链式CRF 将句子序列输入特征提取器中，得到词汇特征，分词阶段LSTM隐层输出得到分词特征，分词阶段词向量为词语特征，三个特征融合为CRF的总特征，得到实体识别阶段损失函数 $L_n$，最后两个损失函数共同联合优化




$$\mathcal{L}_n(\boldsymbol{y}_n; \boldsymbol{x}_n, \Theta) = \frac{1}{K}\sum_k \Big[\log \frac{1}{Z(\boldsymbol{x}_n)^k} + \sum_j \Lambda_j F_j(\boldsymbol{y}_n^k, \boldsymbol{x}_n^k, \boldsymbol{e}_w, \boldsymbol{h}_w)\Big],$$

# 汉语分词与实体识别联合模型（PengACL16）

## ■结果

实体识别模型中，除了基础的词汇特征外，还有词语特征与LSTM隐层特征，9/10/11三组实验分别代表单独使用词语特征、单独使用LSTM隐层特征和共同使用词语特征和隐层特征，可以看出，三个特种共同使用效果最好

| | | Named Entity | | | | | | Nominal Mention | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dev | | | Test | | | Dev | | | Test | | |
| | Method | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| 1 | CRF with baseline features | 60.27 | 25.43 | 35.77 | 57.47 | 25.77 | 35.59 | 72.06 | 32.56 | 44.85 | 59.84 | 23.55 | 33.80 |
| 2 | + Segment Features | 62.34 | 27.75 | 38.40 | 58.06 | 27.84 | 37.63 | 58.50 | 38.87 | 46.71 | 47.43 | 26.77 | 34.23 |
| 3 | P & D best NER model | 57.41 | 35.84 | 44.13 | 57.98 | 35.57 | 44.09 | 72.55 | 36.88 | 48.90 | 63.84 | 29.45 | 40.38 |
| 4 | + Segment Features | 47.40 | 42.20 | 44.65 | 48.08 | 38.66 | 42.86 | 76.38 | 36.54 | 49.44 | 63.36 | 26.77 | 37.64 |
| 5 | P & D w/ Char Embeddings | 58.76 | 32.95 | 42.22 | 57.89 | 34.02 | 42.86 | 66.88 | 35.55 | 46.42 | 55.15 | 29.35 | 38.32 |
| 6 | + Segment Features | 51.47 | 40.46 | 45.31 | 52.55 | 37.11 | 43.50 | 65.43 | 40.86 | 50.31 | 54.01 | 32.58 | 40.64 |
| 7 | Pipeline Seg. Repr. + NER | 64.71 | 38.14 | **48.00** | 64.22 | 36.08 | 46.20 | 69.36 | 39.87 | 50.63 | 56.52 | 33.55 | 42.11 |
| 8 | Jointly LSTM w/o feat. | 59.22 | 35.26 | 44.20 | 60.00 | 35.57 | 44.66 | 60.10 | 39.53 | 47.70 | 56.90 | 31.94 | 40.91 |
| 9 | Jointly Train Char. Emb. | 64.21 | 35.26 | 45.52 | 63.16 | 37.11 | 46.75 | 73.55 | 37.87 | 50.00 | 65.33 | 31.61 | 42.61 |
| 10 | Jointly Train LSTM Hidden | 61.86 | 34.68 | 44.44 | 63.03 | 38.66 | 47.92 | 67.23 | 39.53 | 49.79 | 60.00 | 33.87 | 43.30 |
| 11 | Jointly Train LSTM + Emb. | 59.29 | 38.73 | 46.85 | 63.33 | 39.18 | **48.41** | 61.61 | 43.19 | **50.78** | 58.59 | 37.42 | **45.67** |

# 基于词典跨语言命名实体识别（Mayhew EMNLP17）

- 动机1

  不同语言实体识别的标注语料资源丰富程度差别很大，寻找一种简单通用的多语言实体识别模型

- 动机2

  多语言实体识别任务中平行语料标注耗时耗力，寻找到一种“廉价”的基于多语言词典的实体识别模型

# 基于词典跨语言命名实体识别（Mayhew EMNLP17）

- 基于词典的跨语言翻译方法

  英语和土耳其语举例：

  源语言英语，目标语言西班牙语英语词典中有短语和词语，西班牙语中有短语和词语，通过简单的词-词，词-短语，短语-短语翻译，将两语言词典对齐，再通过显著性得分（prominence）进行消歧，从而得到双语对齐词典

**Algorithm 1** Our translation algorithm

**Input**

$D_E$: Annotated data in $E$

$L$: Lexicon between $E$–$F$

**Returns**

$D_F$: Annotated data in $F$

```
for ∀w_i ∈ D_E do
    p = w_i w_{i+1} ... w_{i+j}    ▷ Window of size j
    while p not in L and j ≥ 0 do
        Decrement j
        p = w_i w_{i+1} ... w_{i+j}
    end while
    if p in L then
        if |L[p]| > 1 then
            resolve with LM and prominence
        end if
        D_F add (L[p], labels of p)
    else
        D_F add (w_i, label of w_i)
    end if
    Increment i by length of p
end for
```

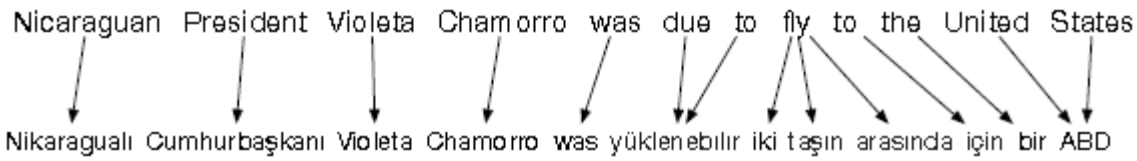
# 基于词典跨语言命名实体识别（Mayhew EMNLP17）

## ■基于词典的跨语言翻译方法

英语和土耳其语举例：得到对齐词典之后，给定一土耳其语句子，将句子中的词和短语通过词典匹配翻译，得到英语句子，虽然语序有时候问题，但是与实体相关的词和短语能准确的识别出来

如：尼加拉瓜总统维罗拉·夏莫罗将要飞往美国

尼加拉瓜总统、维罗拉·夏莫罗、美国都翻译准确



# 基于词典跨语言命名实体识别（Mayhew EMNLP17）

## ■实验结果

本实验将把句子通过Google翻译进行实体识别的结果作为“上限”，通过实验发现，利于词典的翻译方法结合维基百科中提取的不同语言的特征进行实体识别，能达到或者超过直接用Google翻译进行实体识别的结果

| Method | Dutch | German | Spanish | Turkish | Bengali | Tamil | Yoruba | Avg |
|---|---|---|---|---|---|---|---|---|
| Lexicon Coverage | 88.01 | 89.97 | 90.94 | 83.80 | 83.34 | 73.84 | 74.60 | – |
| E-L Dict size | 961K | 1.36M | 1.25M | 578K | 217K | 182K | 334K | – |
| Baseline | 43.10 | 22.61 | 45.77 | 34.63 | 6.40 | 4.60 | 37.70 | 27.83 |
| Google Translate ceiling | 65.71 | 56.65 | 53.65 | 45.63 | 37.84 | 29.11 | 39.18 | 46.82 |
| Wiki (Tsai et al., 2016) | 61.56 | 48.12 | 60.55 | 47.12 | 43.27 | 29.64 | 36.65 | 46.70 |
| Cheap Translation | 53.94 | 50.96 | 51.82 | 46.37 | 30.47† | 25.91† | 37.58 | 42.43 |
| Cheap Translation+Wiki | 63.37 | 57.23 | 64.10 | 51.79 | 46.28† | 33.10† | 38.52 | 50.62 |

# 命名实体识别的评测

- 国际会议：MUC、SigHAN、CoNLL、IEER 和ACE
  - MUC-6和MUC-7设立的命名实体识别专项评测大大推动了英语命名实体识别技术的发展
  - MUC-6和MUC-7还设立了多语言实体识别评测任务MET，对日语、西班牙语、汉语等多种语言命名实体识别任务进行评测
  - SigHAN从2003年开始举办第一届中文分词评测BAKEOFF，2006年和2008年举行的BAKEOFF-3和BAKEOFF-4设立了命名实体识别专项评测
  - 2003年和2004年举办的863计划“中文信息处理与智能人机接口技术评测”中设立了中文命名实体识别评测任务
  - 评测指标：正确率P，召回率R和 F值

# 英文命名实体识别的技术水平

- 英文：Language Technology Group Summary开发的英语命名实体识别系统在MUC-7评测中取得第一名，其准确率和召回率分别达到95%和92%（吴友政，2006）

- 许多英语命名实体系统已经具备了相当程度的大规模文本处理能力

# 汉语命名实体识别的技术水平

- 参加MET-2评测的汉语命名实体识别系统对人名、地名、机构名识别的最优性能指标（准确率，召回率）只有(66%，92%)、(89%，91%)和(89%，88%)（吴友政，2006）

# 汉语命名实体识别的技术水平：BAKEOFF-3

| 数据来源 | 测试类别 | 测试性能 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | P | R | F | ORG- F | LOC-F | PER-F | GPE-F |
| MSRC（简） | Closed | 0.8894 | 0.8420 | 0.8651 | 0.8310 | 0.8545 | 0.9009 | ~ |
| | Open | 0.9220 | 0.9018 | 0.9118 | 0.8590 | 0.9034 | 0.9604 | ~ |
| LDC（简） | Closed | 0.8026 | 0.7265 | 0.7627 | 0.6585 | 0.3046 | 0.7884 | 0.8204 |
| | Open | 0.7616 | 0.6621 | 0.7084 | 0.5209 | 0.2857 | 0.7422 | 0.7930 |
| CITYU（繁） | Closed | 0.9143 | 0.8676 | 0.8903 | 0.8046 | 0.9211 | 0.9087 | ~ |
| | Open | 0.8692 | 0.7498 | 0.8051 | 0.6801 | 0.8604 | 0.8098 | ~ |

| 数据来源 | 简繁体类别 | 训练集规模 | 测试集规模 |
|---|---|---|---|
| MSRC | 简体 | 1.3M/63K（词/词次） | 100K/13K（词/词次） |
| LDC | 简体 | 632K（词） | 61K（词） |
| CITYU | 繁体 | 1.6M/76K（词/词次） | 220K/23K（词/词次） |

(Levow, 2006 )

# 汉语命名实体识别的技术水平：863

| 数据来源 | 测试类别 | 测试性能 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ORG | | | LOC | | | PER | | |
| | | P | R | F | P | R | F | P | R | F |
| SXU (简) | 开放 | .6464 | .5741 | .6081 | .8702 | .7843 | .8251 | .8138 | .8847 | .8478 |
| CITYU(繁) | 开放 | .3986 | .2532 | .3097 | .6839 | .7004 | .6921 | .3986 | .2532 | .3097 |

| 数据来源 | 简繁体类别 | 训练集规模（词/词次） | 测试集规模（字） |
|---|---|---|---|
| SXU | 简体 | NONE | 约400K |
| CITYU | 繁体 | NONE | 约400K |

# 从评测结果看汉语命名实体识别的技术水平

- 在BAKEOFF-3 MSRC语料和BAKEOFF-3 CITYU语料上的评测结果要好于BAKEOFF-3 LDC语料上的评测结果以及863语料上的评测结果
- 其中一个很重要原因是：BAKEOFF-3 MSRC和CITYU评测提供了相当规模的训练集，而BAKEOFF-3 LDC只提供了小规模的训练集，而863评测根本不提供训练集
- 因为训练集和测试集在题材和体裁方面比较类似，可能使得各个系统在BAKEOFF-3 MSRC语料和BAKEOFF-3 CITYU语料上的评测性能较高
- **在真实的应用环境中，命名实体识别的性能会大打折扣**

# 可选的NER工具

| | 分词系统 | 标识 |
|---|---|---|
| 1 | BosonNLP | BOSON |
| 2 | IKAnalyzer | 中文分词库 IKAnalyzer |
| 3 | NLPIR | NLPIR 汉语分词系统 又名：ICTCLAS 2015 |
| 4 | SCWS | SCWS 中文分词 |
| 5 | 结巴分词 | jieba |
| 6 | 盘古分词 | 盘古分词-开源中文分词组件 |
| 7 | 庖丁解牛 | paoding Lucene中文分词"庖丁解牛" Paoding Analysis |
| 8 | 搜狗分词 | Sogou搜狗 云服务 |
| 9 | 腾讯文智 | 互联网+ 腾讯云 |
| 10 | 新浪云 | 新浪云 |
| 11 | 语言云 | 語言 |

- http://ictclas.nlpir.org/中科院计算所(ICTCLAS)
- http://www.openpr.org.cn中科院自动化所(Urheen)
- http://www.fnlp.org/复旦大学
- http://nlp.stanford.edu/software/tagger.shtml Stanford University

# 小结

- 受限于训练语料规模，系统的自适应能力不强
  - 网页信息：不规范、存在很多噪音，有些根本就不构成自然语言句子，因此通常的命名实体识别模型所依赖的上下文特征发生了明显变化，使得识别性能剧烈下降
- 类别数限定，不满足实际的应用
  - 摩托罗拉V8088折叠手机、第6届苏迪曼杯羽毛球混合团体赛、胆结石腹腔镜手术等
- 需要开放域实体抽取
  - 实体类型更多、更细，而且有些实体类别是未知、或者是随时间演化的
- 基于深度学习的分词和命名实体识别

# 提纲

- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 开放域实体抽取

- 不限定实体类别
- 不限定目标文本
- 任务：给定某一类别的实体实例，从网页中抽取同一类别其他实体实例
  - 给出<中国，美国，俄罗斯>（称为“种子”）
  - 找出其他国家<德国，英国，法国……>

Input

中国

美国

俄罗斯

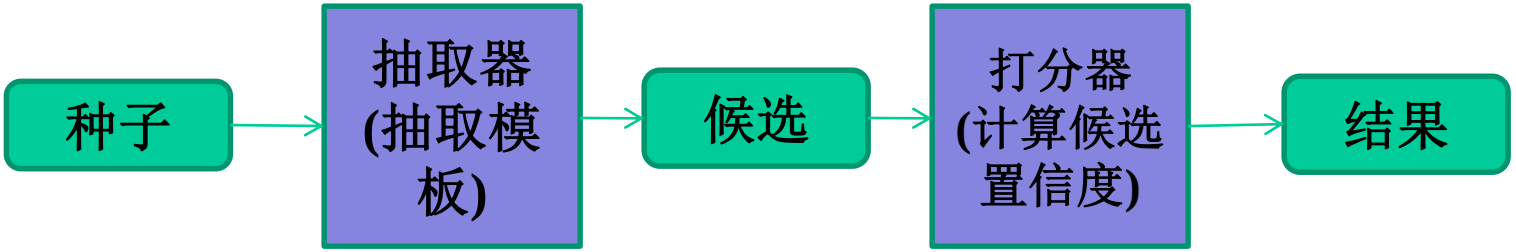
Google sets labs

Output

| Predicted Items |
| --- |
| 中国 |
| 俄罗斯 |
| 美国 |
| 日本 |
| 德国 |
| 英国 |
| 法国 |
| 韩国 |
| 印度 |
| 香港 |
| 意大利 |
| 巴西 |
| 加拿大 |
| 新加坡 |

# 开放域实体抽取的主要方法

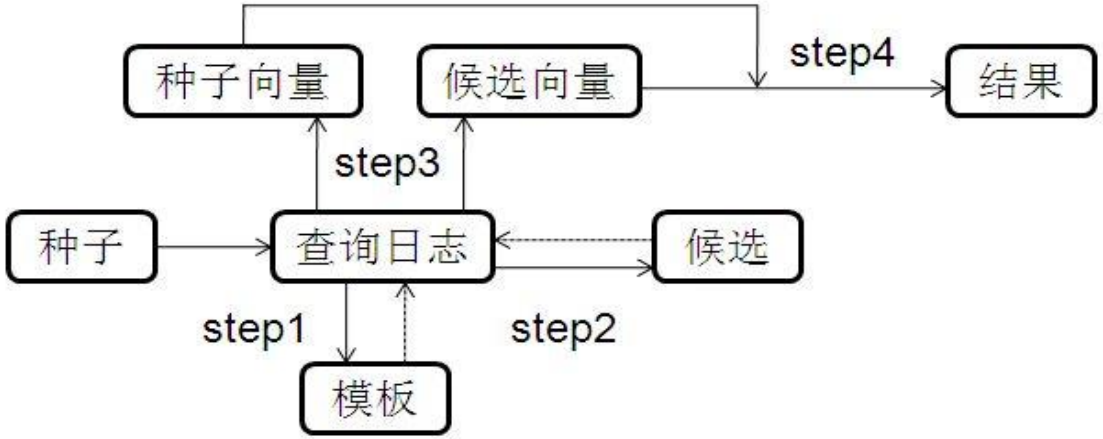
- 基本思路
  - 种子词与目标词在网页中具有相同或者类似的上下文
    - 网页结构
    - 上下文
  - Step1：种子词→模板
  - Step2：模板→更多同类实体
- 处理实例扩展问题的主流框架




- 利用不同数据源（例如查询日志、网页文档、知识库文档等）的不同特点，设计方法

# 开放域实体抽取的主要方法: Query Log (Pasca CIKM 2007)

- 通过分析种子实例在查询日志中的上下文学得模板，再利用模板找到同类别的实例
  - 联想 笔记本 如何
  - 苹果 笔记本 如何
  - 戴尔 笔记本 如何
- 构造候选与种子上下文向量，计算相似度



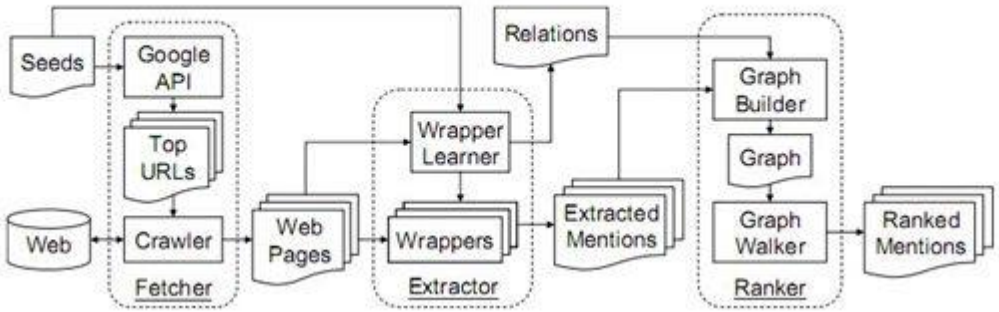
# 开放域实体抽取的主要方法: Web Page (Wang ICDM 2007) (1/2)

- Motivation
  - 处理列表型网页
  - 在列表中，种子与目标实体具有相同的网页结构

```
<li>·<a target="_self" href='#brand_A_2'>奥迪</a></li>
<li>·<a target="_self" href='#brand_B_3'>宝马</a></li>
<li>·<a target="_self" href='#brand_B_4'>保时捷</a></li>
<li>·<a target="_self" href='#brand_B_5'>奔驰</a></li>
<li>·<a target="_self" href='#brand_B_7'>本田</a></li>
<li>·<a target="_self" href='#brand_B_8'>比亚迪</a></li>
<li>·<a target="_self" href='#brand_B_9'>标致</a></li>
<li>·<a target="_self" href='#brand_B_10'>别克</a></li>
<li>·<a target="_self" href='#brand_C_15'>长城</a></li>
<li>·<a target="_self" href='#brand_D_18'>大众</a></li>
<li>·<a target="_self" href='#brand_F_24'>丰田</a></li>
```

# 开放域实体抽取的主要方法: Web Page (Wang ICDM 2007) (2/2)

- 系统框架




一个模板抽取出的正确实例越多，则该模板的质量越高（反之亦然）

- 爬取模块（Fetcher）
  把种子送到搜索引擎，把返回的前100个网页抓取下来作为语料

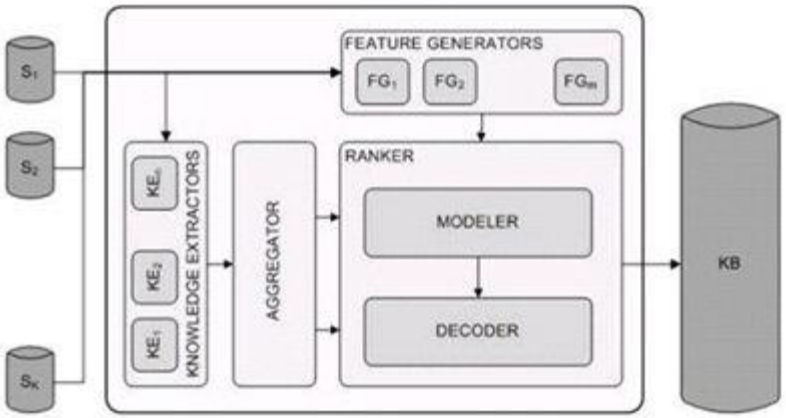
一个网页包含的高质量模板越多，则该网页质量越高（反之亦然）

- 抽取模块（Extractor）
  针对单个网页学习模板，再使用模板抽取候选实例

- 排序模块（Ranker）
  利用种子、网页、模板、候选构造一个图，综合考虑网页和模板的质量，使用Random Walk算法为候选打分并排序

# 开放域实体抽取的主要方法: 融合多个数据源 (Pennacchiotti EMNLP 2009)

- 针对不同数据源，选取不同特征分别进行实例扩展，对结果进行融合
  - 6亿个网页
  - 1年的查询日志
  - Wikipedia
- 针对不同数据源选取不同的模板和特征
- 使用不同特征计算候选的置信度
  - 结果融合



# 评价指标与技术水平

- 针对实例扩展问题，目前缺少公认的评测，研究者在自己构建的数据上进行测试
- 评价方法：以平均准确度MAP为主；召回无法评价，覆盖度替代
  - 因为系统输出是一个ranked list，单纯考察准确率无法体现出rank的作用
  - 采用TREC中常用的MAP（平均正确率均值）

$$\mathrm{AvgPrec}(L) = \frac{\sum_{r=1}^{|L|} \mathrm{Prec}(r) \times \mathrm{NewEntity}(r)}{\#\ \mathrm{True\ Entities}}$$

  - Prec: 列表中到目前为止的准确率；NewEntity(r)当前实体是否正确的0-1函数
  - 每个类别做N次实验，每次都随机选取种子，对其求平均

# 评价指标与技术水平

取前100个网页作为语料

- Wang 2007在12个自制数据集的结果，

Table 5. Experimental results

| | | Max. 100 Results | | | Max. 200 | Max. 300 |
|---|---|---|---|---|---|---|
| English | G.Sets | E1+EF | E2+EF | E2+GW | E2+GW | E2+GW |
| classic-disney | 37.62% | 79.36% | 74.45% | 84.42% | 88.20% | 89.39% |
| cmu-buildings | 0.00% | 87.85% | 87.75% | 87.83% | 87.83% | 87.83% |
| common-diseases | 1.12% | 17.94% | 52.84% | 57.46% | 75.79% | 76.87% |
| constellations | 10.45% | 89.61% | 99.97% | 100.00% | 100.00% | 100.00% |
| countries | 14.24% | 95.95% | 97.86% | 98.17% | 98.67% | 98.53% |
| mlb-teams | 70.06% | 98.61% | 99.50% | 99.80% | 99.84% | 99.81% |
| nba-teams | 90.73% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| nfl-teams | 94.26% | 99.22% | 99.98% | 100.00% | 100.00% | 100.00% |
| periodic-comets | 0.22% | 69.24% | 79.04% | 84.78% | 84.77% | 84.77% |
| popular-car-makers | 73.61% | 79.18% | 88.23% | 95.16% | 96.23% | 96.95% |
| us-presidents | 56.77% | 91.64% | 97.07% | 99.99% | 100.00% | 100.00% |
| us-states | 76.00% | 99.96% | 93.55% | 100.00% | 100.00% | 100.00% |
| **Average** | **43.76%** | **84.05%** | **89.19%** | **92.30%** | **94.28%** | **94.51%** |

# 小结

- 针对不同数据源的特点设计方法，其针对性、灵活性很强
- 方法一般分为模板抽取和实例候选置信度计算两个模块，两部分迭代进行，相互依赖
- 以无监督的方法为主
- 缺少公认的数据集或相关评测

# 总结

- 自然语言处理技术的历史
- 信息抽取技术概述：四个任务
- 中文分词
  - 任务介绍
  - 难点
  - 典型方法
  - 评测指标
  - 技术水平

# 总结

- 命名实体识别
  - 任务介绍
  - 难点
  - 典型方法
  - 技术水平

- 开放域实体识别
  - 任务介绍
  - 难点
  - 典型方法
  - 评测指标
  - 技术水平

谢谢

jzhao@nlpr.ia.ac.cn

# 实 体 识 别

赵 军（jzhao@nlpr.ia.ac.cn）

中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

# 基于统计的分词方法：判别式方法

- 原理：
  - 在有限样本条件下建立对于预测结果的判别函数，直接对预测结果进行判别
  - 由字构词的分词理念，将分词问题转化为判别式分类问题

- 典型算法：

  Maxent SVM CRF Perceptron

# 基于统计的分词方法

- 判别式分词
  - 流程：
    - 把分词问题转化为确定句中每个字在词中位置问题
    - 每个字在词中可能的位置可以分为以下三种
      - 词首B（日本 占领 了 东三省）
      - 词中M（游泳 比赛 菲尔普斯 独占鳌头）
      - 词尾E（中国队 抢占 了 风头）
      - 独字S （男生 占 全班 人数 的 百分之八十）
  - 分词结果展示：
    - 分词结果：毛/B 新/M 年/E 2/B 0/M 0/M 0/M 年/E 毕/B 业/E 于/S 东/B 北/M 大/M 学/E

# 最大熵模型：熵

- 什么是熵？
  - 什么是熵？ 没有什么问题在科学史的进程中曾被更为频繁地讨论过

    （比利时物理化学家，诺贝尔奖，普里高津）

  - 熵定律是自然界一切定律中的最高定律

    （里夫金&霍华德，《熵:一种新的世界观》）

# 最大熵模型：熵的提出

- 热力学第二定律：物理过程总是自发地从有序走向无序，最后达“热寂”。
- 热寂是猜想宇宙终极命运的一种假说。作为一个“孤立”的系统，宇宙的熵会随着时间的流逝而增加，由有序向无序，当宇宙的熵达到最大值时，宇宙中的其他有效能量已经全数转化为热能，所有物质温度达到热平衡。这样的宇宙中再也没有任何可以维持运动或是生命的能量存在。

# 最大熵模型：熵的提出

- 德国物理学家克劳修斯（Rudolph J.E clausius）从热力学第二定律出发，于1865提出熵的概念用来描述一个系统的无序度。 因此热力学第二定律又被称为“增熵原理”，即系统的演进总是指向熵增加的方向。

- 克劳修斯的熵概念这是在热力学角度提出的，后被Boltzmann通过统计物理学的角度重新诠释。

# 最大熵模型：熵与信息

- 熵表示了一个事件的不确定性
- 信息可以理解为事件不确定性的减少
  - 原来不确定的事件现在确定下来，就是获得了信息
  - 原来不确定性越大的事情发生了，获得的信息越多

    例如：

    狗咬人不是新闻，人咬狗才是新闻

# 最大熵模型：信息熵

- 熵：1948年电气工程师香农( Shannon)创立了信息论，将信息量与熵联系起来。
- 他用非常简洁的数学公式定义了信息时代的基本概念：信息熵（简称熵），定量地描述事件的不确定性
- 设随机变量ξ ，它有A1，A2，…，An共n个可能的结局，每个结局出现的概率分别为$p_1$,$p_2$，… ，$p_n$，则ξ 的不确定程度，即信息熵为:

$$H(\xi)=-\sum_{i=1}^{n} p_i \log p_i$$

- 熵越大，越不确定；熵等于0，事件是确定的；通常对数底取2，熵的单位为比特（bit）

# 最大熵模型：例子

- 抛硬币
  - X={正面，反面}
  - p(正面)=p(反面)=0.5
  - H(X)=-(0.5log0.5+0.5log0.5)=1
- 掷骰子
  - X={1，2，3，4，5，6}
  - p(1)=p(2)=p(3)=p(4)=p(5)=p(6)=1/6
  - H(X)=log(6)=2.585
- 不对称的硬币
  - p(正面)=0.3, p(反面)=0.7
  - H(X)=-(0.3log0.3+0.7log0.7)=0.881

# 最大熵模型：通信中的信息量和熵

- 表示“是” 和 “否”
  - 1 = 是，0 =否
- 表示“是” 、 “否”和“可能是”
  - 11 =是，00 = 否 ，10(01) = 可能是
- 一条消息的熵就是编码这条消息所需二进制位即比特的个数
- 通信中，熵越大，需要的编码越长

# 最大熵模型：最大熵理论

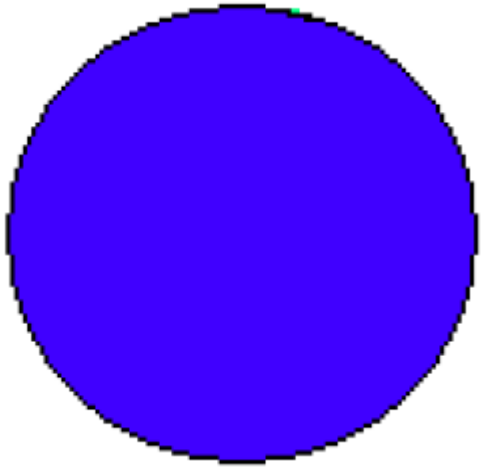
- 熵增原理
- 在无外力作用下，事物总是朝着最混乱的方向发展
- 事物是约束和自由的统一体，事物总是在约束下争取最大的自由权，这其实也是自然界的根本原则
- 在已知条件下，熵最大的事物，最可能接近它的真实状态

# 最大熵模型

- 基于最大熵原理的模型选择
  - 任务：研究某个随机事件，根据已知信息，预测其未来行为。
  - 方法：当无法获得随机事件的真实分布时，构造统计模型对随机事件进行模拟。
  - 难点：满足已知信息要求的模型可能有很多个，用哪个模型来预测最合适呢？
    - 选择熵最大的模型
    - Jaynes证明：对随机事件的所有相容的预测中，熵最大的预测出现的概率占绝对优势
    - Tribus证明：正态分布、伽玛分布、指数分布等，都是最大熵原理的特殊情况

# 最大熵模型

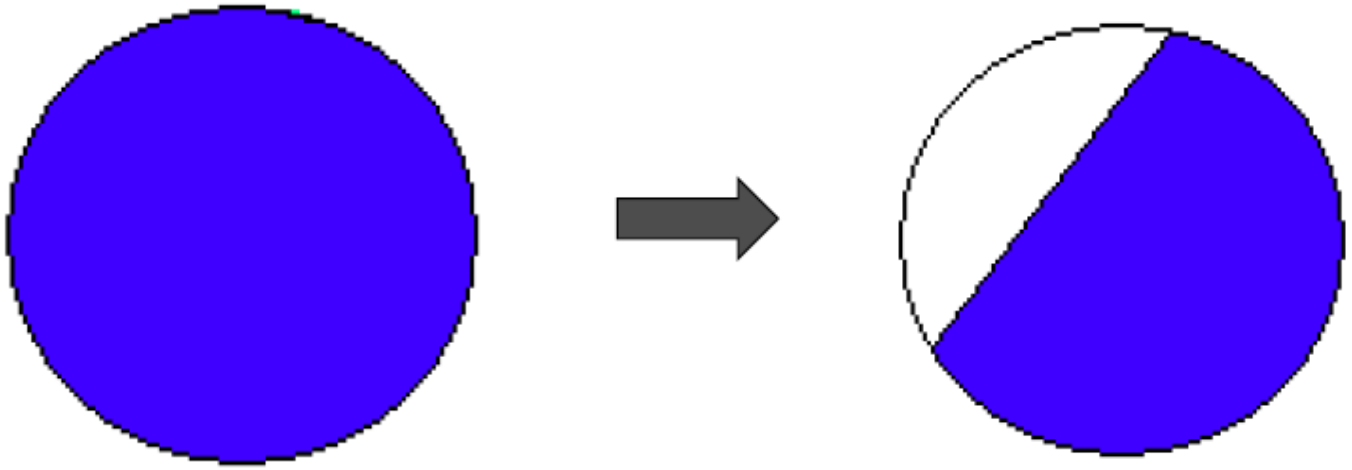
- 最大熵原则下点的分布



  - 对一个随机过程，如果没有任何观察量，既没有任何约束，则解为均匀分布

# 最大熵模型

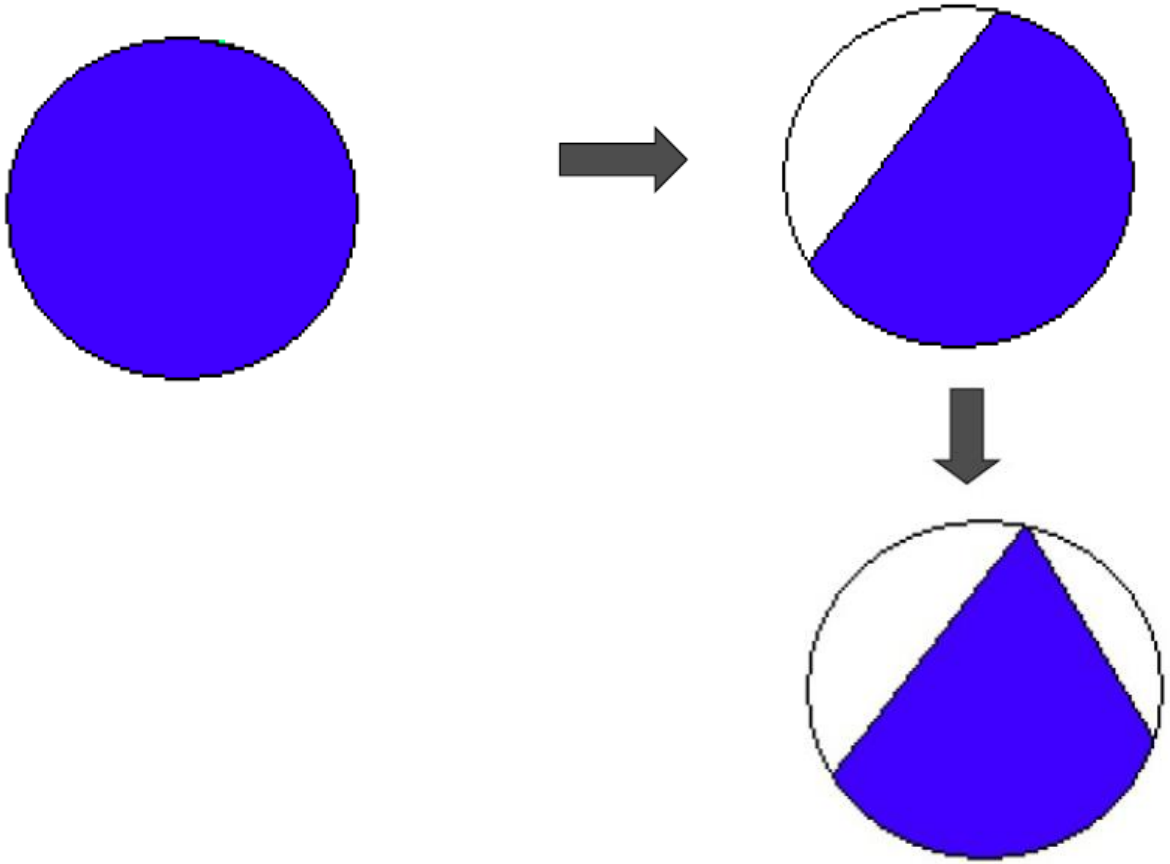
- 最大熵原则下点的分布



  - 加上一个约束条件，则在满足约束的范围内均匀分布

# 最大熵模型

- 最大熵原则下点的分布



- 加上更多约束条件，则在满足约束的范围内均匀分布

# 最大熵模型：天气预报

- 假设要用今天的天气预测明天的天气
  - 天气∈{晴、阴、雨}
  - 风向∈{无、南、北}
  - 已知今天的天气和风向，要预测明天的天气

- 已知历史天气数据（见下页表）
- 已知今天天气为阴，风向北，明天天气最可能是什么？

# 最大熵模型：天气预报

■样本数据：

| 昨天风向 | 昨天天气 | 今天天气 |
|---|---|---|
| 无 | 雨 | 晴 |
| 南 | 阴 | 阴 |
| 北 | 晴 | 晴 |
| 北 | 晴 | 晴 |
| 南 | 阴 | 雨 |
| 无 | 晴 | 晴 |
| 北 | 晴 | 阴 |
| 无 | 雨 | 晴 |
| 南 | 阴 | 雨 |
| 北 | 晴 | 阴 |

# 最大熵模型：天气预报

- 问题定义
  - 用A表示条件集合，B表示结论集合。
    随机事件 $X=(\alpha,\beta)\in\xi=A\times B,\ \alpha\in A,\ \beta\in B$
  - 已经有一些X的样本：

$$X_1=\{\alpha_1,\ \beta_1\},\ X_2=\{\alpha_2,\ \beta_2\},\ \ldots\ ,X_n=\{\alpha_n,\ \beta_n\}$$

  - 假设已知 $\alpha_{n+1}$，如何预测 $\beta_{n+1}$ ？
  - 求解一个X的概率分布，使得：
    - X的分布与已知的样本分布一致
    - X的熵最大

# 最大熵模型：天气预报

- 建模
  - 建立天气预报模型：给出所有可能的条件下结论的概率 $p(\beta|\alpha)$，$\alpha \in A$ ，$\beta \in B$
  - 为简化问题起见，通常用联合概率模型取代上述的条件概率模型$p(\alpha,\beta)$
  - 二者关系： $p(\beta|\alpha)= p(\alpha,\beta)/p(\alpha)$
  - 由于条件和结论都是离散量，理论上，所有的可能性是可以穷举的，因此只要给出所有可能性的概率即可

# 最大熵模型：天气预报

■建模目标：模型能预测所有的可能，并给出概率

所有的可能性

| 昨天风向 | 昨天天气 | 今天天气 | 概率 |
|---|---|---|---|
| 无 | 晴 | 晴 | 0.011 |
| 无 | 晴 | 阴 | 0.003 |
| 无 | 晴 | 雨 | 0.002 |
| 无 | 阴 | 晴 | 0.008 |
| 无 | 阴 | 阴 | 0.013 |
| 无 | 阴 | 雨 | 0.005 |
| …… | …… | …… | |
| 北 | 雨 | 晴 | 0.009 |
| 北 | 雨 | 阴 | 0.001 |
| 北 | 雨 | 雨 | 0.003 |

概率之和为一

# 最大熵模型：天气预报

- 建模
  - 上述模型需要满足以下两个条件：
    - 模型的概率分布应尽可能与样本一致
      - 模型中，“晴、阴、雨”天的概率分布应与样本一致
      - 模型中，昨天为“晴”、“阴”、“雨”时第二天的天气概率分布应与样本一致
      - 模型中，昨天为“无风”、“南风”、“北风”时第二天的天气概率分布应与样本一致
      - 模型中，昨天天气与风向组合给定的情况下，第二天的天气概率分布应与样本一致
      - ……
    - 模型的熵最大！

# 最大熵模型：天气预报

- 优化目标
  - X熵最大：可以表示为：

$$\hat{p} = \underset{p}{\operatorname{argmax}}\, H(p)$$

  - X的分布与已知的样本分布一致：如何表示？
  - 实际上，由于A（预测的条件）的可能性数量可能极其巨大，穷举所有可能性是不现实的，如何对任意的条件和结论给出其概率？

# 最大熵模型：天气预报

- 引入特征
  - 实际问题中，由于条件α和结果β取值多样化，为模型表示方便起见，通常将条件α和结果β表示为一些二制特征
  - 特征$f_1, f_2, \ldots, f_n$定义如下：

    $f_i$:ε->{0,1}, ε=A( 预测条件)×B(结论)
  - 这样每个事件都可以表示为一个由特征值{0,1}组成的n维向量，不再直接用条件和结论来描述。

# 最大熵模型：天气预报

- 引入约束
  - “模型分布与样本分布一致”可以描述为：

$$E_p[f_i]=e_{\hat{p}}[f_i], i=1\ldots n, \text{这里 } p \text{ 为样本分布，} \hat{p} \text{ 为模型分布}$$

这个公式含义为：对于任何一个特征，模型和样本应该具有相同的均值（即：根据样本数据统计各种特征出现的次数）。由于特征取值只有{0,1}，因此这个公式实质上可以理解为：模型中任何一个特征为1的概率与样本应相同。

# 最大熵模型：天气预报

- 简单特征（单一特征）
  - 为天气预报引入以下简单特征：
    - f1：昨天天气=晴
    - f2：昨天天气=阴
    - f3：昨天天气=雨
    - f4：昨天风向=无
    - f5：昨天风向=南
    - f6：昨天风向=北
    - f7：今天天气=晴
    - f8：今天天气=阴
    - f9：今天天气=雨

# 最大熵模型：天气预报

■用简单特征表示样本：样本表示

| 昨天风向 | | | 昨天天气 | | | 今天天气 | | |
|---|---|---|---|---|---|---|---|---|
| 无 | 南 | 北 | 晴 | 阴 | 雨 | 晴 | 阴 | 雨 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

# 最大熵模型：天气预报

- 用简单特征表示样本：建模目标（模型能预测所有的可能，并给出概率）

所有的可能性

| 昨天风向 | | | 昨天天气 | | | 今天天气 | | | 概率 |
|---|---|---|---|---|---|---|---|---|---|
| 无 | 南 | 北 | 晴 | 阴 | 雨 | 晴 | 阴 | 雨 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.003 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0.004 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0.006 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0.002 |
| … | … | … | … | … | … | … | … | … | … |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0.15 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0.005 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

概率之和为一

# 最大熵模型：天气预报

- 用简单特征建模
  - 利用上述简单特征，采用最大熵原理为天气预报建模，得到的模型是：
    - 模型中所有特征出现概率与样本一致：
      - 模型中“昨天天气为晴”概率与样本一致
      - 模型中“昨天天气为阴”概率与样本一致
      - ……
      - 模型中“今天天气为雨”概率与样本一致
    - 模型熵最大
  - 这个模型显然不是我们想要的模型，我们希望：
    - **模型中“昨天天气为晴”的情况下今天各种天气的概率**与样本一致
    - 模型中“昨天风向为南”的情况下今天各种天气的概率与样本一致
    - 模型中昨天天气和风向给定情况下今天各种天气的概率与样本一致
    - ……

# 最大熵模型：天气预报

- 复合特征（组合特征）
  - 为了使得到的模型满足我们所期望的约束条件，可以重新定义特征：
    - f1: 昨天天气为晴，且今天天气为晴
    - f2: 昨天天气为晴，且今天天气为阴
    - ……
    - fn: 昨天风向为北，天气为雨，且今天天气为雨
  - 按照这种方式定义特征，那么所有的特征都反映了某种预测条件和结果之间的依赖关系，这样得到的模型才是我们所期望的模型。

# 最大熵模型：天气预报

- 用复合特征表示样本：建模目标（模型能预测所有的可能，并给出概率）

所有的可能性

| $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | … | $f_{n-2}$ | $f_{n-1}$ | $f_n$ | 概率 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | … | 0 | 0 | 0 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 0 | 1 | 0.003 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 1 | 0 | 0.004 |
| 0 | 0 | 0 | 0 | 0 | … | 0 | 1 | 1 | 0.005 |
| 0 | 0 | 0 | 0 | 0 | … | 1 | 0 | 0 | 0.006 |
| 0 | 0 | 0 | 0 | 0 | … | 1 | 0 | 1 | 0.002 |
| … | … | … | … | … | … | … | … | … | … |
| 0 | 0 | 0 | 0 | 0 | … | 1 | 0 | 0 | 0.15 |
| 1 | 1 | 1 | 1 | 1 | … | 1 | 1 | 0 | 0.005 |
| 1 | 1 | 1 | 1 | 1 | … | 1 | 1 | 1 | 0 |

概率之和为一

# 最大熵模型：天气预报

- 模型应用
  - 假设我们已经得到一个满足上述约束条件且熵最大的模型，如何预测天气？
  - 给定预测条件α（今天风向和今天天气），需要预测明天天气β∈{晴,阴,雨}，我们只要：根据模型分别将(α,晴)/(α,阴)/(α,雨)转化为特征值，然后根据模型计算出相应的概率，取概率最大者即可。

# 最大熵模型：天气预报

- 最优化：学习过程
  - 根据前面的定义，最大熵模型的参数估计可以表示为一个约束条件下的极值问题
    - 在以下约束条件下：

$$E_p[f_i]=e_{\hat{p}}[f_i], i=1\ldots n,$$

模型中任何一个特征为1的概率与样本应相同

这里 $p$ 为样本分布，$\hat{p}$ 为模型分布

    - 求解熵最大的模型：

$$\hat{p}=\arg\max_{p\in C} H(p)$$

$C$为所有可能的模型组成的集合

# 最大熵模型：天气预报

- 最优化：学习过程
  - 经推导（过程略），满足上述约束条件的最大熵模型具有如下形式：

$$p(\beta|\alpha)=\frac{\exp\left(\sum_{(i=1)}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$

  - 其中，Z为规范化因子。$\lambda_i$ 是一组参数（权重），其中每个参数对应于一个特征
  - 学习参数：一组$\lambda_i$

# 最大熵模型：天气预报

- 用最大熵模型预测
  - 给定条件α，结论为β的概率可以表示为：

$$p(\beta|\alpha)=\frac{\exp\left(\sum_{(i=1)}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$

  - 给定条件α，最优的β可以表示为：

$$\hat{\beta}=\underset{\beta}{\operatorname{argmax}}(\beta|\alpha)=\underset{\beta}{\operatorname{argmax}}\frac{\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\alpha,\beta)\right)}{Z}$$
$$=\underset{\beta}{\operatorname{argmax}}\sum_{i=1}^{n}\lambda_i f_i(\alpha,\beta)$$

  - 可以看出，一旦得到这组参数$\lambda_i$，那么对于给定的条件α，对所有的结论β，只要将其所有值为1特征$f_i$对应的$\lambda_i$加起来，取和最大的β即可

# 最大熵模型

- 工具
  - 最普遍使用的工具：Maximum Entropy Modeling Toolkit for Python and C++
  - 作者：张乐
  - 主页：http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

# 基于最大熵模型的中文分词

- 采用最大熵模型对每个汉字标注BMES标记
- 假设当前字$C_0$，当前标记是$T_0$
- 常用最大熵特征模板（当前字是$C_0$）：
  - $C_n T_0 (n = -2, -1, 0, 1, 2)$：汉字
    - 例如：$C_{-1}T_0$表示当前字的前一个字和当前标记
      $C_2T_0$表示当前字的前一个字
  - $C_n C_{n+1} T_0 (n = -2, -1, 0, 1)$：两字组
  - $C_{-1}C_1$ T0 ：当前字左右两个字
  - $D(C_0)T_0$ ：当前字是否数字
  - $A(C_0)T_0$ ：当前字是否字母
  - $P(C_0)T_0$ ：当前字是否标点

# 基于最大熵模型的中文分词

- 生成最大熵训练实例
  - 假设给定训练样本：

    <s> 今天 是 星期三 。 </s>
  - 生成对应标注序列：

    今|B 天|E 是|S 星|B 期|M 三|E 。|S
  - 对于每一个字生成一个训练实例：上例中天字生成的实例

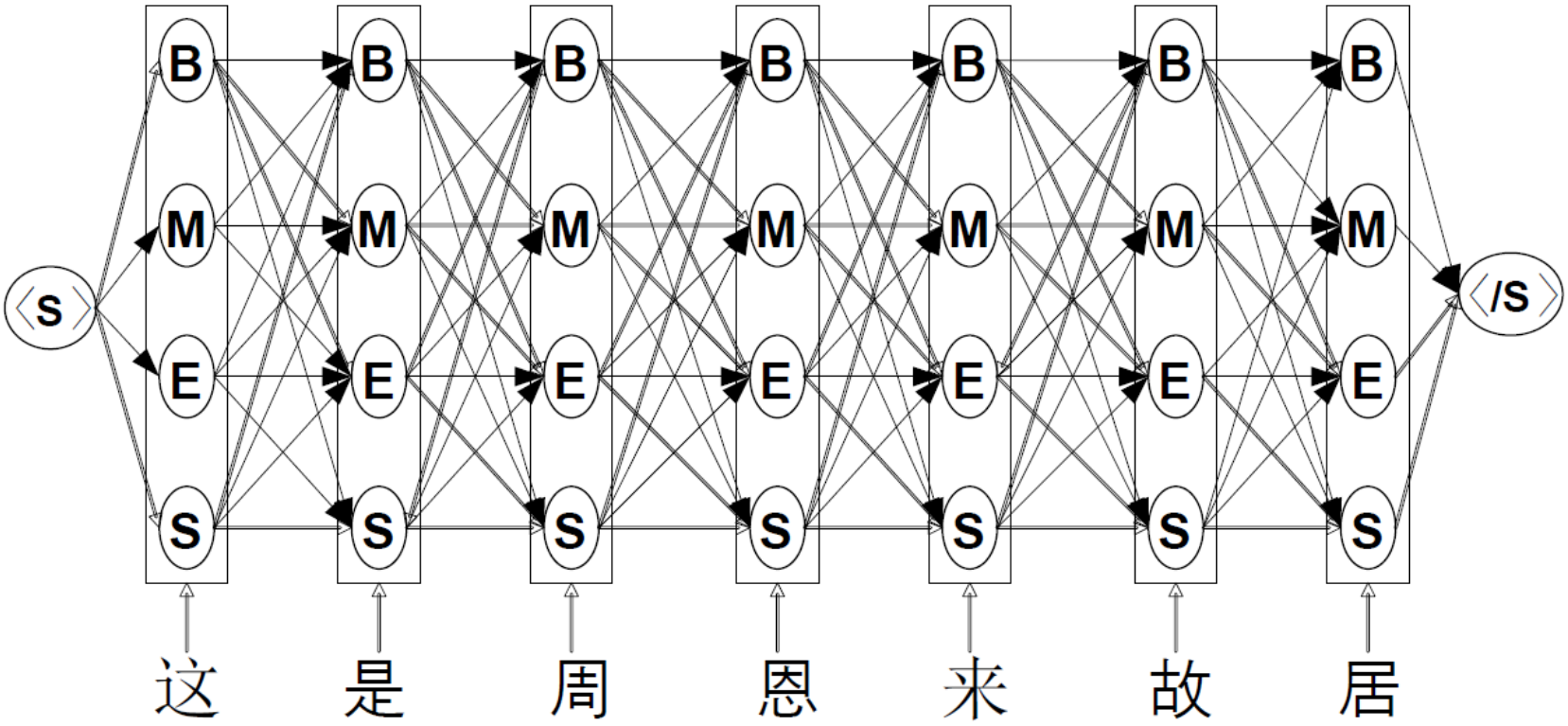
| | | | |
|---|---|---|---|
| $C_{-2}T_0$=<s>E | $C_{-1}T_0$= 今 E | $C_0T_0$= 天 E | $C_1T_0$= 是 E |
| $C_2T_0$= 星 E | $C_{-2}C_{-1}T_0$=<s> 今 E | $C_{-1}C_0T_0$= 今天 E | |
| $C_0C_1T_0$= 天是 E | $C_1C_2T_0$= 是星 E | $C_{-1}C_1T_0$= 今是 E | |
| $D(C_0)T_0$= 否 | $A(C_0)T_0$= 否 | $P(C_0)T_0$= 否 | |

# 基于最大熵模型的中文分词

- 最大熵模型的参数训练
  - 对训练语料库中每个汉字生成一个实例
  - 把所有实例的列表送给最大熵模型的训练工具
  - 最大熵模型的训练工具将学习到模型的参数
- 最大熵模型的测试
  - 对于输入串中每一个汉字以及该汉字的每一个可能的标记，生成一个实例，这样对每个汉字就生成了四个实例
  - 对每个实例，生成其所有激活的特征，以及这些特征所对应的概率
  - 选取可能性最大的标记为该字的标记

# 基于最大熵模型的中文分词

■搜索最优标注路径

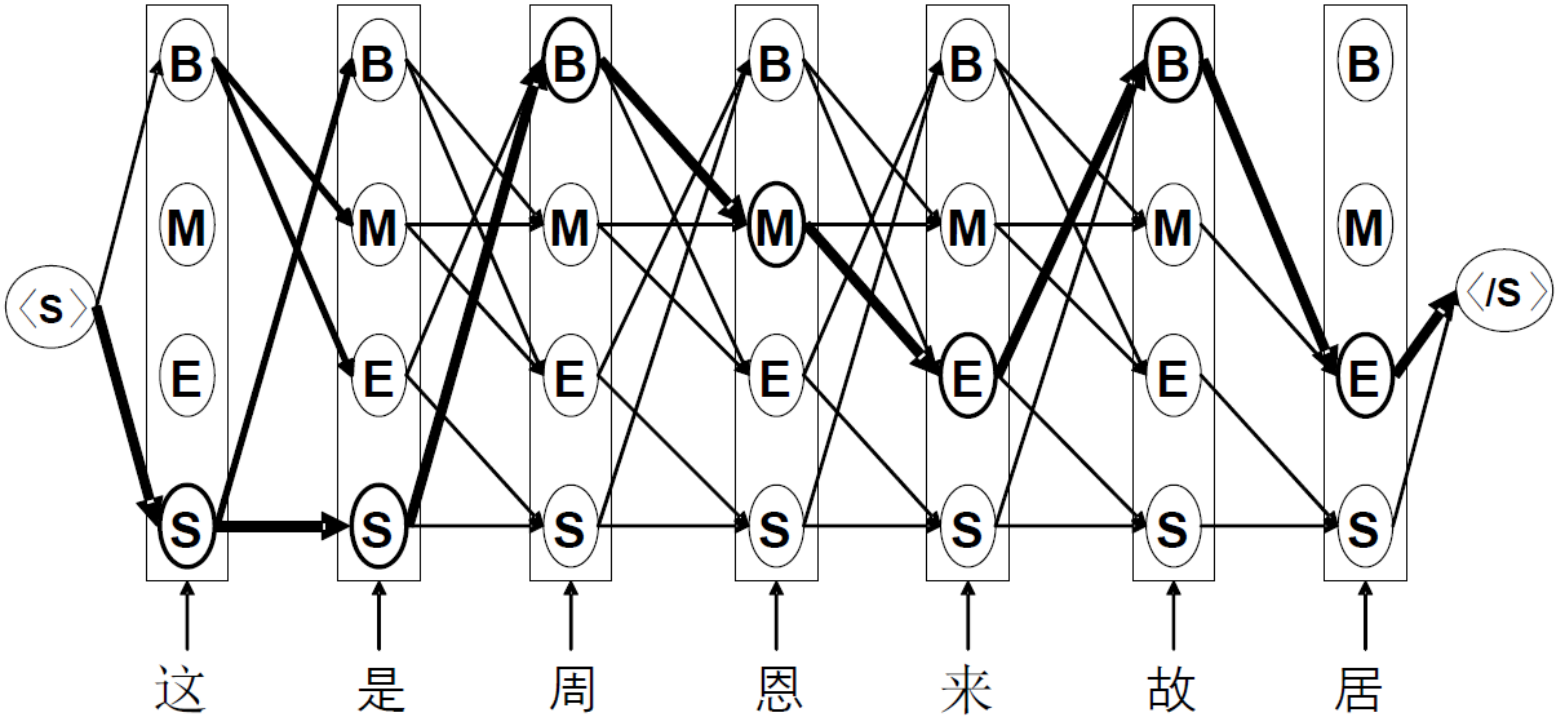


# 基于最大熵模型的中文分词

- 搜索最优标注路径
  - 有些边是非法的，比如M→B，E→M，E→E，S→M，S→E，等等
  - 每条路径的概率是路径上各节点标注概率P(标记|汉字)之积，边上没有转移概率
  - 搜索算法与HMM模型解码的Viterbi算法相同

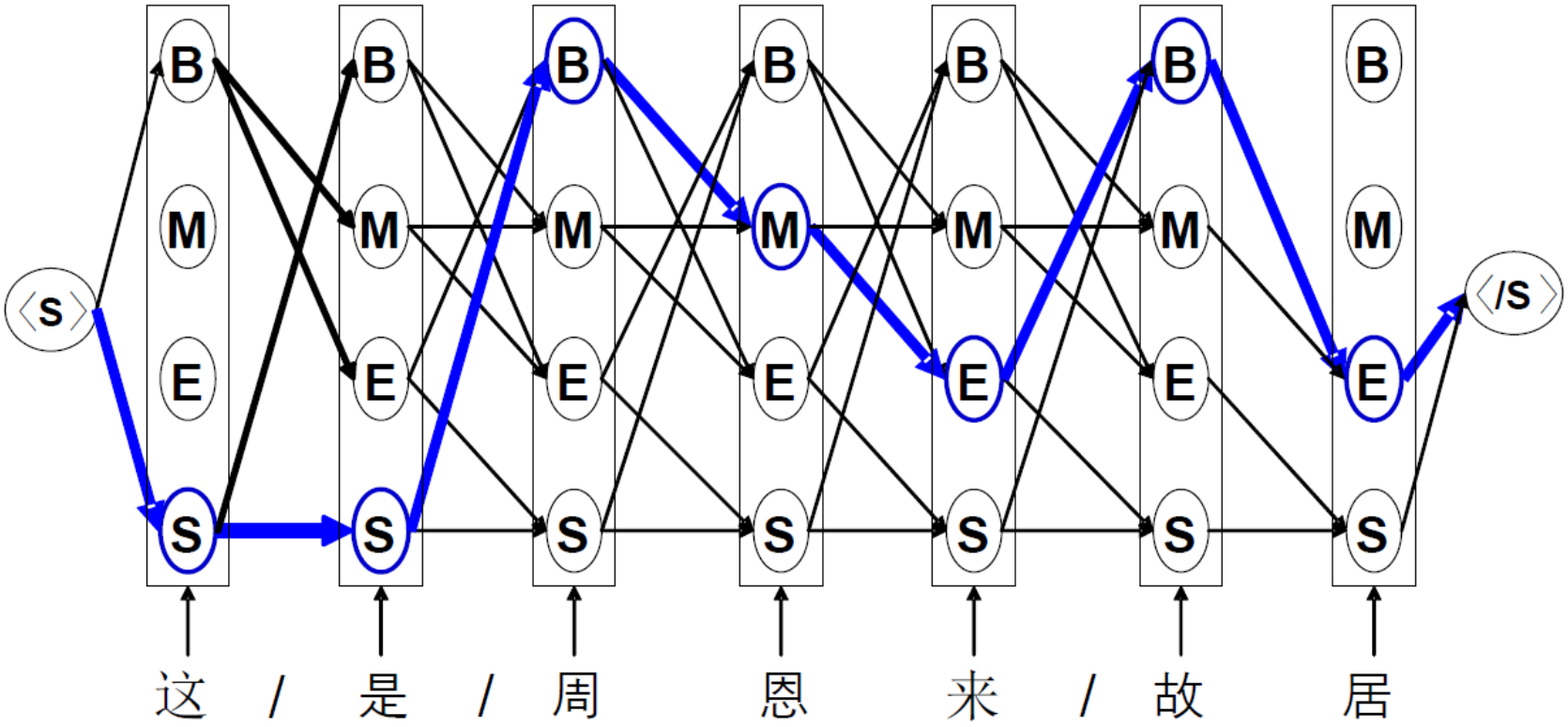
# 基于最大熵模型的中文分词

■搜索最优标注路径



# 基于最大熵模型的中文分词

■搜索最优标注路径



# 基于统计的分词方法：判别式方法

- 优点
  - 分词精度高
  - 新词识别率较高
- 缺点
  - 训练速度慢
  - 需要人工设计特征
  - 性能与特征紧密相连
  - 需要人工标注训练语料

# 现有分词方法总结

| 比较项目 | 基于词典的分词 | 产生式分词 | 判别式分词 |
| --- | --- | --- | --- |
| 分词精度 | 80% | 90% | 95% |
| 分词速度 | 最快 | 最慢 | 较快 |
| 理论基础 | 没有 | 扎实 | 扎实 |
| 学习素材 | 不需要 | 大规模 | 中小规模 |
| 词典 | 需要 | 需要 | 不需要 |
| 学习效率 | 无需学习 | 高 | 中低 |
| 歧义处理能力 | 有限 | 中等 | 强 |
| 新词处理能力 | 不能 | 中等 | 强 |

# 中文分词的技术水平

- 2003年国家863评测部分结果
  - 最好成绩：P=93.44%，R=93.69%，F1=93.46%
  - 最差成绩：P=91.42%，R=89.27%，F1=90.33%
- 2005年SIGHAN汉语分词评测结果（使用MSR语料）

| 评测方式 | 系统排名 | 性能指标 | | | | |
|---|---|---|---|---|---|---|
| | | 召回率 | 精确率 | $F1$ | $R_{oov}$ | $R_{iv}$ |
| 封闭测试 | 最好 | 0.962 | 0.966 | 0.964 | 0.717 | 0.968 |
| | 最差 | 0.898 | 0.896 | 0.897 | 0.327 | 0.914 |
| 开放测试 | 最好 | 0.980 | 0.965 | 0.972 | 0.59 | 0.99 |
| | 最差 | 0.788 | 0.818 | 0.803 | 0.37 | 0.8 |

$R_{oov}$表示集外词的召回率，$R_{iv}$表示集内词的召回率。

# 中文分词的技术水平

■2010年SIGHAN评测结果

| | | | Closed Track | | | | | | Open Track | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | OOV | ID | R | P | F | $R_{oov}$ | $R_{iv}$ | ID | R | P | F | $R_{oov}$ | $R_{iv}$ |
| S | L | 0.069 | S5 | 0.945 | 0.946 | 0.946 | 0.816 | 0.954 | S6 | 0.958 | 0.953 | 0.955 | 0.655 | 0.981 |
| | C | 0.152 | S6 | 0.953 | 0.95 | 0.951 | 0.827 | 0.975 | S9 | 0.95 | 0.95 | 0.95 | 0.82 | 0.973 |
| | M | 0.11 | S6 | 0.942 | 0.936 | 0.939 | 0.75 | 0.965 | S9 | 0.94 | 0.936 | 0.938 | 0.768 | 0.962 |
| | | | | | | | | | S18 | 0.941 | 0.935 | 0.938 | 0.787 | 0.96 |
| | F | 0.087 | S6 | 0.959 | 0.96 | **0.959** | **0.827** | 0.972 | S9 | 0.96 | 0.96 | **0.96** | **0.847** | 0.971 |
| T | L | 0.094 | S10 | 0.942 | 0.942 | 0.942 | 0.788 | 0.958 | S1 | 0.905 | 0.9 | 0.902 | **0.775** | 0.918 |
| | C | 0.094 | S10 | 0.948 | 0.957 | 0.952 | 0.666 | 0.977 | S1 | 0.911 | 0.924 | 0.918 | 0.698 | 0.933 |
| | M | 0.075 | S10 | 0.953 | 0.957 | 0.955 | 0.798 | 0.966 | S1 | 0.903 | 0.903 | 0.903 | 0.729 | 0.917 |
| | F | 0.068 | S10 | 0.964 | 0.962 | **0.963** | **0.812** | 0.975 | S1 | 0.903 | 0.916 | **0.91** | **0.721** | 0.916 |

开始于2002年，第一届SIGHAN是在COLING2002上举办的。 后面几次分别在ACL、COLING 和 IJCNLP 上举办。中间停办过几次，SIGHAN-8是在ACL2015上举办的。最新的是SIGHAN-9也就是2017年和IJCNLP在今年的12月1日在台湾举办。

# 可选的分词工具

| | 分词系统 | 标识 |
|---|---|---|
| 1 | BosonNLP | BOSON |
| 2 | IKAnalyzer | 中文分词库 IKAnalyzer |
| 3 | NLPIR | NLPIR 汉语分词系统 又名：ICTCLAS 2015 |
| 4 | SCWS | SCWS 中文分词 |
| 5 | 结巴分词 | jieba |
| 6 | 盘古分词 | 盘古分词-开源中文分词组件 |
| 7 | 庖丁解牛 | paoding Lucene中文分词"庖丁解牛" Paoding Analysis |
| 8 | 搜狗分词 | Sogou搜狗 云服务 |
| 9 | 腾讯文智 | 互联网+ 腾讯云 |
| 10 | 新浪云 | 新浪云 |
| 11 | 语言云 | 語言 |

- http://ictclas.nlpir.org/中科院计算所(ICTCLAS)
- http://www.openpr.org.cn中科院自动化所(Urheen)
- http://www.fnlp.org/复旦大学
- http://nlp.stanford.edu/software/tagger.shtml Stanford University

# 当前分词技术存在的主要问题

- 训练语料规模小：分词模型过于依赖训练样本，而标注大规模训练样本费时费力，由此导致分词系统对新词的识别能力差，往往在与训练样本差异较大的测试集上性能大幅度下降

- 训练语料领域少：现有的训练样本主要在新闻领域，而实际应用千差万别：网络新闻、微博/微信/QQ等对话文本、不同的专业领域(中医药、生物、化学、能源……)。

- 对实体和专有名词的识别性能较低

# 提纲

- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 命名实体

- 命名实体的定义
  - 狭义地讲，命名实体指现实世界中具体或抽象的实体，如人（张三）、机构（中国中文信息学会、阿里巴巴网络技术有限公司）、地点等，通常用唯一的标志符(专有名称)表示，如人名、机构名、地名等。
  - 广义地讲，命名实体还可以包含时间（12：00）、日期（2017年10月17日）、数量表达式（100）、金钱（一亿美金）等。至于命名实体的确切含义，只能根据具体应用来确定。比如，在具体应用中，可能需要把住址、电子信箱地址、电话号码、舰船编号、会议名称等作为命名实体。

# 命名实体识别的任务

- 一般而言，主要是识别出待处理文本中七类（人名、机构名、地名、时间、日期、货币和百分比）命名实体
- 两个子任务：实体边界识别和确定实体类别

2011年4月11日17点16分，日本东北部的福岛和茨城地区发生里氏7.0级强烈地震（震中北纬36.9度、东经140.7度，即福岛西南30公里左右的地方，震源深度10公里，属于浅层地震）当局已经发布海啸预警震后约30分钟后在日本海地区发生巨型海啸，同时造成福岛核电站出现核泄漏震后第十天，国际原子能机构对于日本政府反应迟钝进行了谴责

→

**2011年4月11日17点16分**，日本东北部的**福岛和茨城地区**发生里氏7.0级强烈地震（**震中北纬36.9度、东经140.7度**，即福岛西南30公里左右的地方，震源深度10公里，属于浅层地震）当局已经发布海啸预警震后约30分钟后在日本海地区发生巨型海啸，同时造成**福岛核电站**出现核泄漏震后第十天，**国际原子能机构**对于**日本政府**反应迟钝进行了谴责

# 命名实体识别的特点

- 时间、日期、货币和百分比的构成有比较明显的规律，识别起来相对容易
- 人名、地名、机构名的用字灵活，识别的难度很大
  - 在不同领域、 场景下, 命名实体的外延有差异 。
  - 数量巨大,不能枚举,难以全部收录在词典中。
  - 某些类型的实体名称变化频繁，并且没有严格的规律可以遵循。
  - 表达形式多样。
  - 首次出现后往往采用缩写形式。

# 英文人名识别

- 人名识别在英文中已得到很好的研究,
- 因为英文本身具有一些明显特征(如大小写)
  - Barack Obama
  - Donald Trump
  - ……

- 切分造成的错误很少，使得人名识别在英文中变得较为容易。

# 中文人名识别：难点

- 名字用字范围广，分布松散，规律不很明显
- 姓氏和名字都可以单独使用用于特指某一人
  - 王二小，小王，二小
- 许多姓氏用字和名字用字(词)可以作为普通用字或词被使用
  - 姓氏为普通词：于(介词)，张(量词)，江(名词)等
  - 名字为普通词：建国，国庆，胜利，文革，计划
  - 全名也是普通词汇，如：万里，温馨，高山，高升，高飞，周密，江山，夏天等。
- 缺乏可利用的启发标记（人名与其上下文组合成词）
  - 祝贺老总百战百胜。
  - 这里有关天培的壮烈

# 中文人名识别：特点

- 中国人名一般由以下部分组合而成：
  - 姓：张、王、李、刘、诸葛、西门、范徐丽泰
  - 名：李素丽，王杰、诸葛亮
  - 前缀：老王，小李
  - 后缀：王老，赵总

# 中文人名识别：特点

- 中国人名各组成部分用字比较有规律
  - 姓：汉语姓氏大约有1000多个, 姓氏中使用频度最高的是“王”姓, “王, 陈, 李, 张, 刘” 等5个大姓覆盖率达32%, 姓氏频度表中的前14个高频度的姓氏覆盖率为50%，前400个姓氏覆盖率达99%。
  - 名：人名的用字也比较集中。 频度最高的前6个字覆盖率达10.35%, 前10个字的覆盖率达14.936%, 前15个字的覆盖率达19.695%, 前400个字的覆盖率达90%。

# 中文人名识别：特点

- 中国人名各组成部分的组合规律
  - 姓 + 名
  - 姓
  - 名
  - 前缀 + 姓
  - 姓 + 后缀
  - 姓 + 姓 + 名（海外已婚妇女）
- 中国人名的上下文构成规律
  - 身份词：
    - 前：工人、教师、影星
    - 后：先生、同志
    - 前后：女士、教授、经理、小姐、总理

# 中文人名识别：特点

- 地名或机构名：
  - 前：静海县大丘庄禹作敏
- 的字结构
  - 前：年过七旬的王贵芝
- 动作词：
  - 前：批评，逮捕，选举
  - 后：说，表示，吃，结婚

# 中文地名识别：难点

- 地名数量大，缺乏明确、规范的定义。

  中国地名委员会编写了《中华人民共和国地名录》，收集了全国乡镇以上（含乡镇）各级行政区域的名称，以乡镇人民政府所在地为主的居民聚落名称，山、河、湖、海、岛、高原、盆地、沙溪等自然地理实体名称，名胜古迹、纪念地、古遗址、水库、桥梁、电站等名称，共收录地名10万多条。这个地名录中使用的汉字共2662个，频度最高的前65个汉字占总频度的50.22%，前622个汉字占总频度的90.01%，前1872个汉字占总频度的99%。

- 真实语料中地名出现情况复杂。
  - 地名简称、地名用词与其他普通词冲突
  - 地名是其他专用名词的一部分
  - 地名长度不一

# 中文地名识别：特点

- 地名中使用的多字词中绝大部分是两个字
  - 山口
  - 水库
  - 遗址
  - 城镇
  - .....
- 经常与方位词和介词连用
  - 东、南、西、北
  - 周边、附近
  - 在、于
  - .....

# 中文机构名识别：难点

- 机构名中含有大量的人名、地名、企业字号等专有名称。在这些专有名称中，地名所占的比例最大，其中未登录地名又占了相当一部分的比例。所以机构名识别应在人名、地名等其他专名识别之后进行，其他专名识别的正确率对机构名识别正确率有较大影响。
  - 地名：山东大学，上海大学，北京大学
  - 人名：哈佛大学，逸夫图书馆，中山大学
  - 学科、专业和部分系统：公安部，教育部
  - 研究、生产或经营等活动的对像：软件研究所，卫星制造厂
  - 上述情况的综合：白求恩医科大学
  - 大机构、团体、组织和职业的名称：中国人民解放军洛阳外国语学院，中国发明家学会等

# 中文机构名识别：难点

- 用词广泛
  - 通过对人民日报1998年1月中的10817个机构名所含的19986个词进行统计，共计27种词，其中名词最多（9941个），地名其次（5023个），以下依次为简称（1169个）、专有名词（1125个）、动词（848个）以及机构名（714个）等
- 机构名长度极其不固定
  - 新疆维吾尔自治区乌鲁木齐国家高新技术产业开发区社会管理综合治理委员会
  - 学校及周边治安综合治理工作领导小组
  - 北大
- 机构名很不稳定
  - 随着社会发展，新机构不断涌现，旧机构不断被淘汰、改组或更名

# 中文机构名识别：特点

- 词法角度：偏正式(修饰格式)的复合词

  {名词|形容词|数量词|动词} + 名词

- 句法角度：“定语 + 名词性中心语”型的名词短语(定名型短语)

- 中心语：机构称呼词，如：大学，学院，研究所，学会，公司，股份公司等

# 音译名识别

- 音译名用字非常集中《英语姓名译名手册》中共收英语姓氏, Given Names约4万个, 经计算机统计得出英语姓名译名用字表共476 个：

“ 啊阿埃艾爱昂奥巴白柏拜班邦包保堡鲍北贝倍本比彼边别滨宾玻波博勃伯卜布采蔡藏策查察昌彻陈楚垂茨慈次聪存措达大戴代丹当道德得登邓迪底地蒂第帝丁东杜敦顿多厄恩耳尔法凡范方菲费芬丰冯佛夫福弗辅富盖甘冈高哥戈葛格各根贡古顾瓜圭郭果哈海罕翰汉杭豪赫黑亨洪侯胡华怀惠霍基吉季计嘉佳加贾简姜焦杰捷金津京久居喀卡开凯坎康考柯科可克肯孔扣寇库夸匡奎魁坤昆阔拉腊莱来赖兰朗劳勒乐雷黎理李里礼荔丽历利立莲连廉良列琳林霖龄留刘流柳龙隆卢鲁露路吕略伦萝罗洛玛马麦迈满曼芒茅梅门蒙孟米密敏明名摩莫墨默姆木穆拿娜纳乃奈南内嫩能妮尼年涅宁牛纽农努女诺欧帕派潘庞培佩彭蓬皮匹平泼朴普漆奇齐契恰钱强乔切钦琴青琼丘邱屈让热仁日荣茹儒瑞若撒萨塞赛三缮桑瑟森莎沙珊山尚绍舍申生盛圣施诗石什史士寿舒朔斯思丝松孙索所塔泰坦汤唐陶特藤提惕田铁汀廷亭通透图托脱娃瓦万旺威韦为维伟魏卫温文翁沃乌武伍西锡希悉席霞夏显香向晓肖歇谢欣辛兴幸姓雄休修雪逊雅亚延扬阳尧耀耶叶依易意因英永尤雨约宰赞早泽曾扎詹湛章张哲者珍真芝知智治朱卓兹子宗祖佐丕谟葆薇岑弼娅缪珀瑙赉滕斐熙鸠窦艮麟黛”。

# 音译名识别：难点

- 音译名内部很难划分出结构，但有一些常见音节，如“斯基、斯坦”等
- 不同语言的音译规律不尽相同，如法语、俄语、蒙古语译名用字与英语就有较大区别（蒙语人名举例：“那顺乌日图、青格勒图”），如果按不同的语言训练不同的模型比使用统一的模型效果更好
- 音译名可以是人名、地名或其他专名，上下文规律差别较大

# 命名实体识别的方法

- 方法分类
  - 有词典切分/无词典切分

    在分词的过程中使用词典的方法是有词典切分，反之是无词典切分，有字典切分的方法一般是基于规则的，无词典切分的方法一般是基于统计的

  - 基于规则的方法/基于统计的方法

    基于规则的方法不需要标注训练语料，能直接根据词典和规则进行分词，基于统计的方法需要标注训练语料训练模型。基于统计的方法可以分为生成式统计命名实体识别和判别式命名实体识别

# 基于字典的命名实体识别方法

- 方法概述：按照一定的策略将待分析的汉字串与一个充分大的词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功

- 典型方法
  - 正向最大匹配法
  - 反向最大匹配法
  - 最短路径法（最少分词法）

- 方法与之前介绍的分词中的方法类似

# 基于统计的命名实体识别方法

- 生成式方法
  - 原理：首先建立学习样本的生成模型，再利用模型对预测结果进行间接推理
  - 典型算法：

    HMM, PCFG等
- 判别式方法
  - 原理：由字构词的命名实体识别理念，将NER问题转化为判别式分类问题(序列标注问题)
  - 典型算法：

    Maxent　SVM　CRF　CNN　RNN　LSTM+CRF

# 条件随机场（CRF）

- 定义：条件随机场（Conditional Random Field，CRF）是给定一组输入随机变量条件下另一组输出随机变量的条件概率分布模型
- 特点：假设输出随机变量构成马尔可夫随机场
- 应用：可以应用于不同类型的标注问题，例如：
  - 单个目标的标注、序列结构的标注、图结构的标注
- 自然语言处理的很多任务可以转化为序列结构的标注问题。本课程主要介绍线性链条件随机场，这时，问题变成了由输入序列对输出序列标注的判别模型。
- CRF和HMM
  - HMM：生成式模型，难以考虑复杂的特征
  - CRF：判别式模型，可以考虑复杂的特征

# 条件随机场（CRF）

- 内容：
  - 概率图模型、概率无向图模型
  - 马尔可夫随机场
  - 条件随机场的表示形式
  - 条件随机场的三个基本问题
    - 概率计算问题
    - 解码问题
    - 学习问题

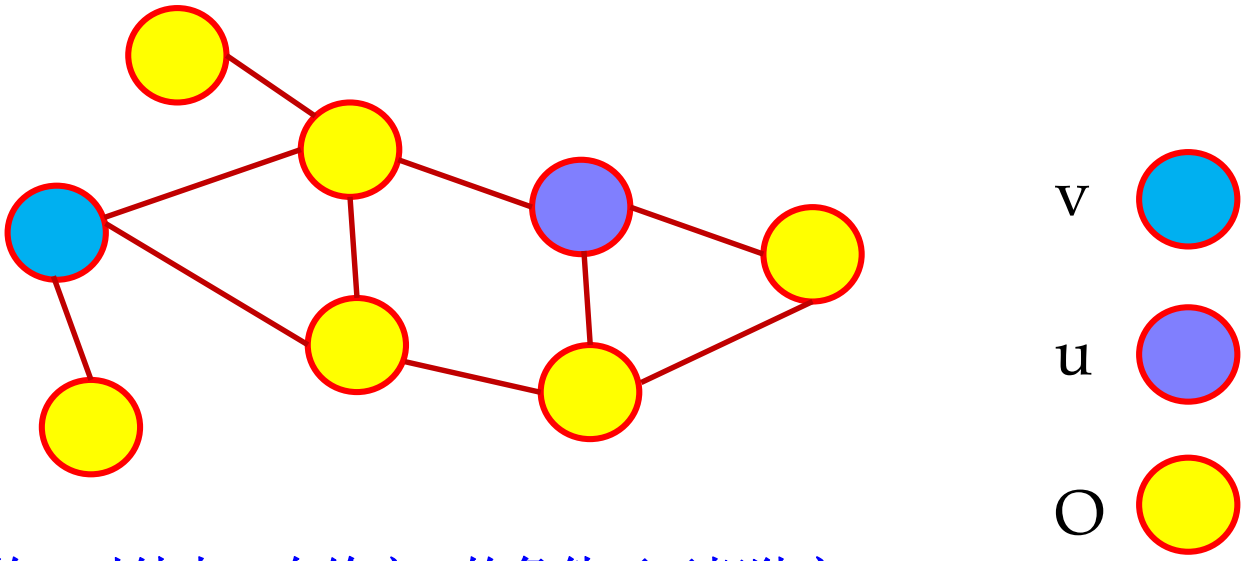
# 概率无向图模型

- **图**是由结点及连接结点的边组成的集合。结点和边分别记作v和e，结点和边的集合分别记作V和E，图记作G=（V,E）。
- **无向图**是指边没有方向的图
- **概率图模型**是由图表示的概率分布：设有联合概率分布P(Y)，Y是一组随机变量。由图G=（V,E）表示概率分布P(Y)，即在图G中，一个结点v表示一个随机变量$Y_v$，边e表示随机变量之间的概率依赖关系。
  - 根据边的性质，不同概率图模型可以分为两类
    - **概率有向图模型**：图G是有向图, 例如：隐马尔可夫模型
    - **概率无向图模型**：图G是无向图, 例如：条件随机场模型

# 概率无向图模型：成对马尔可夫性

- 设u和v是无向图G中任意两个没有连接的结点，结点u和v分别对应随机变量$Y_u$和$Y_v$。其它所有结点为O，对应的随机变量组是$Y_o$，成对马尔可夫性是指给定随机变量组$Y_o$的条件下$Y_u$和$Y_v$是条件独立的，即：

$$P(Y_u, Y_v \mid Y_o) = P(Y_u \mid Y_o) P(Y_v \mid Y_o)$$

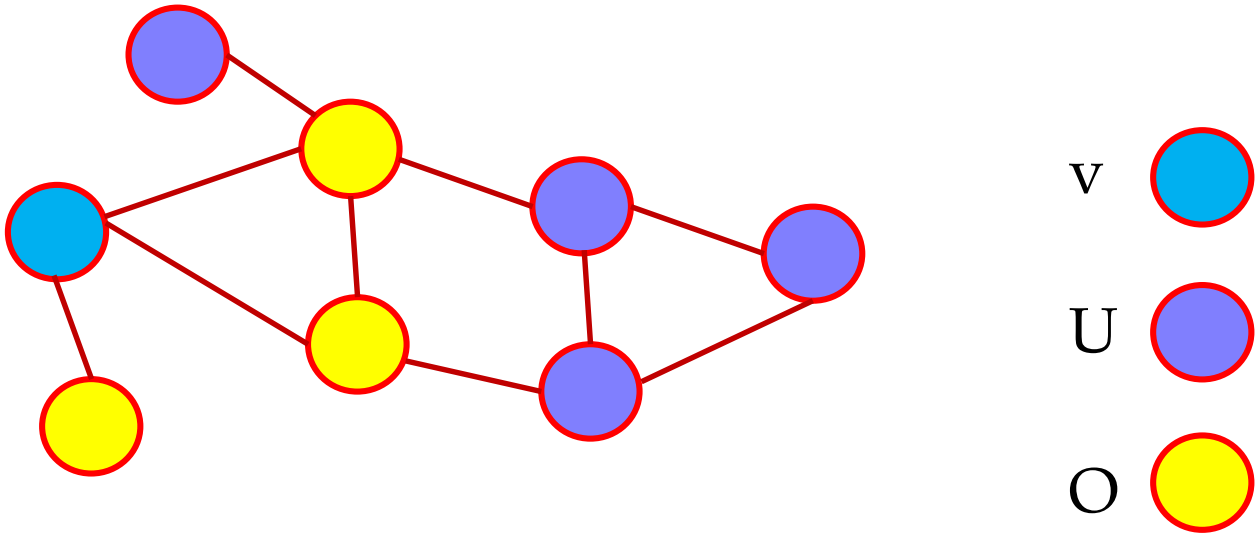



任意没有直接相连的一对结点，在给定O的条件下互相独立

# 概率无向图模型：局部马尔可夫性

- 设v是无向图G中任意一点，O是与v有边连接的所有点，U是v, O以外的其他所有结点。v表示的随机变量是$Y_v$，O表示的随机变量组是$Y_O$，U表示的随机变量组是$Y_U$。局部马尔可夫性是指在给定随机变量组$Y_O$的条件下随机变量$Y_v$和$Y_O$是独立的。即

$$P(Y_v, Y_U|Y_O) = P(Y_v|Y_O)\ P(Y_U|Y_O)$$

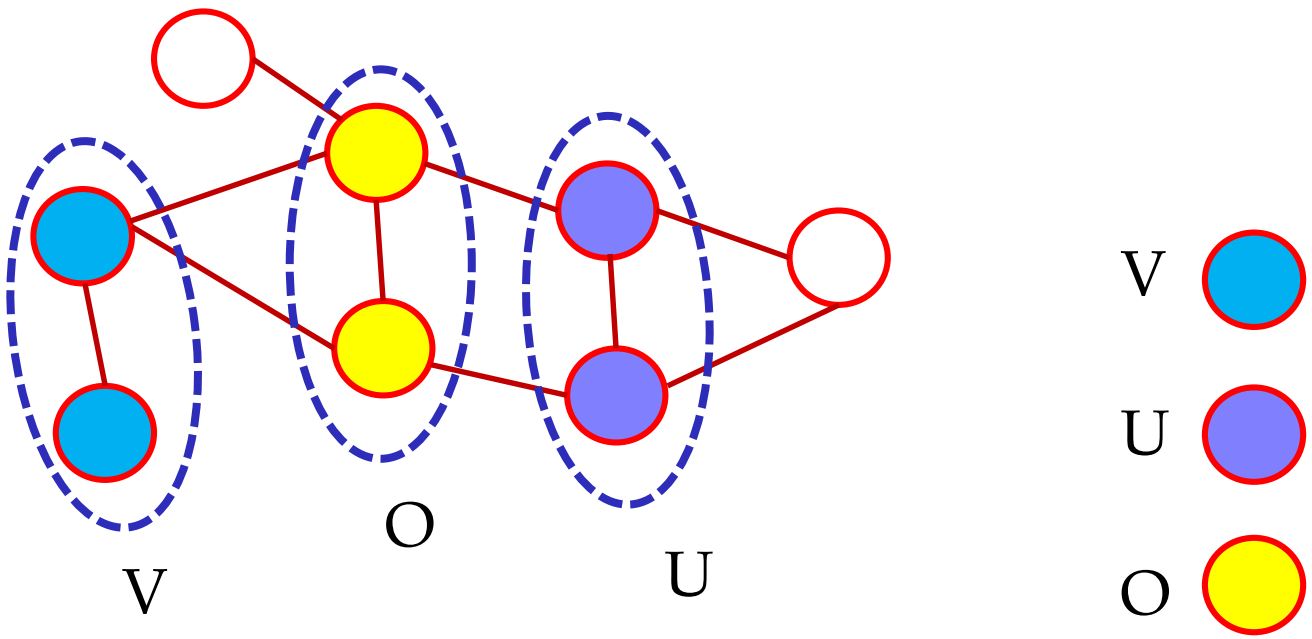



任意一个结点与所有不与它直接相连的结点，在给定的条件下互相独立

# 概率无向图模型：全局马尔可夫性

- 设结点集合V, U是在无向图G中被结点集合O分开的任意结点集合。结点集合V, U和O所对应的随机变量组分别是$Y_V$, $Y_U$ 和$Y_O$。全局马尔可夫性是指给定随机变量$Y_O$条件下随机变量组$Y_V$和$Y_U$是条件独立的，即

$$P(Y_V, Y_U|Y_O) = P(Y_V|Y_O)\ P(Y_U|Y_O)$$



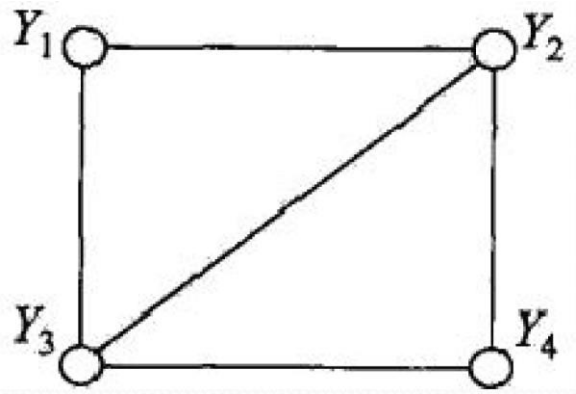

**被结点集合O分开的任意两个结点集合A和B，在给定O的条件下互相独立**

# 马尔可夫随机场

- 如果概率无向图模型的联合概率分布P(Y)满足三种马尔科夫性，就称此联合概率分布为马尔可夫随机场
  - 成对马尔科夫性
  - 局部马尔科夫性
  - 全局马尔可夫性
- 关键问题：求联合概率P(Y)
- 对给定的马尔可夫随机场模型，希望将整体的联合概率写成若干子联合概率的乘积的形式，也就是将联合概率进行因子分解

# 马尔可夫随机场：因子分解

- 团：无向图G中任何两个结点均有边连接的结点子集称为团(clique)。
- 最大团：若C是无向图G的一个团，并且不能再加进任何一个c的结点使其成为一个更大的团，则称此C为最大团(maximal clique).




- 两个结点组成的团有5个{$Y_1$，$Y_2$}, {$Y_2$，$Y_3$}, {$Y_1$，$Y_3$}, {$Y_3$，$Y_4$}, {$Y_2$，$Y_4$},有两个最大团：{$Y_1$，$Y_2$，$Y_3$}， {$Y_2$，$Y_3$，$Y_4$}

# 马尔可夫随机场：因子分解

- 将马尔可夫随机场模型的联合概率分布表示为其最大团上的随机变量的函数的乘积形式的操作，称为马尔可夫随机场模型的因子分解(Factorization)
- 给定马尔可夫随机场模型，设其无向图为G，C为G上的最大团，Yc表示C对应的随机变量，那么马尔可夫随机场模型的联合概率分布P(Y)可写作图中所有最大团C上的函数 $\Psi_c(Y_c)$ 的乘积形式，即

$$P(Y)=\frac{1}{Z}\prod_c \Psi_c(Y_c)$$

$$\Psi_c(Y_c)=\exp\{-E(Yc)\}$$

势函数

- Z是归一化因子(normalization factor)

$$Z=\sum_Y \prod_c \Psi_c(Y_c)$$

# 条件随机场：定义

- 设X与Y是随机变量，P(Y|X)是在给定X的条件下Y的条件概率分布。若随机变量Y构成一个由无向图G=(V,E)表示的马尔可夫随机场，则称条件概率分布P(Y|X)为**条件随机场**。

$$P(Y_v \mid X, Y_w, w \neq v) = P(Y_v \mid X, Y_w, w \sim v)$$

其中，w~v表示在图G=(V,E)中与结点v有边连接的所有结点w，w≠v表示结点v以外的所有结点。

# 线性链条件随机场：定义

- 设 $X=(X_1,X_2,\cdots,X_n)$，$Y=(Y_1,Y_2,\cdots,Y_n)$ 均为线性链表示的随机变量序列，若在给定随机变量序列X的条件下，随机变量序列Y的条件概率分布P(Y|X)构成条件随机场，即满足马尔可夫性

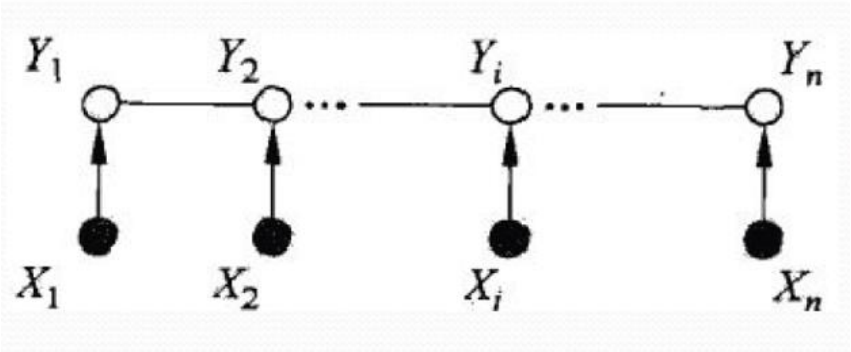
$$P(Y_i|X,Y_1,\cdots,Y_{i-1},Y_{i+1},Y_n)=P(Y_i|X,Y_{i-1},Y_{i+1})$$

$$i=1,2,\cdots,n$$ （在i=1和n时只考虑单边）

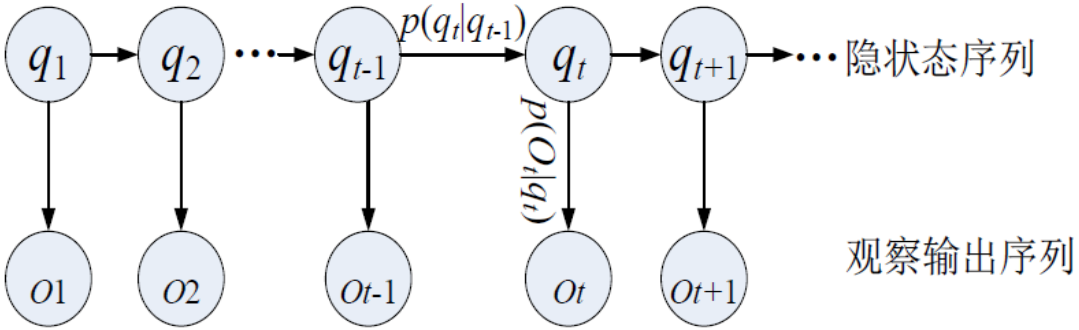
则称P(Y |X)为线性链条件随机场。

# 线性链条件随机场：图示

- 在标注问题中，X表示输入观察序列，Y表示对应的输出标记序列或状态序列

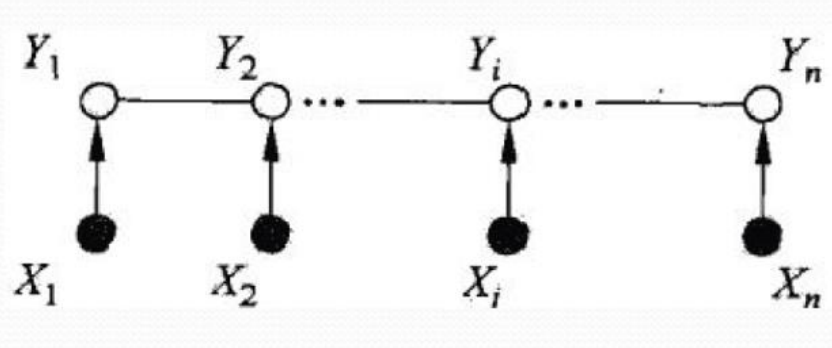



- 对比：HMM




- 本课程中，接下来条件随机场指线性条件随机场

# CRF的参数化形式




- 设P(Y|X)为线性链CRF，则在随机变量X取值为x的条件下，随机变量Y取值为y的条件概率具有如下形式：

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

观察为x，在i和i-1位置标记为$y_{i-1}$和$y_i$的可能性

其中：x可以为观察样本的任意特征（例如：词本身、大小写、POS、词典等），这里假设有k个特征，特征的重要程度由lamda控制，训练得到。

tk：定义在边上的特征函数，也称转移特征，依赖于前一个和当前位置

sl: 定义在结点上的特征函数，也称状态特征，依赖于当前位置。

$$Z(x)=\sum_{y}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

# 条件随机场的参数化形式：例子

- 输入观测为：$X=(X_1,X_2,X_3)$，输出标记为$Y=(Y_1,Y_2,Y_3)$，$Y_1$, $Y_2$, $Y_3$ 取值于{1,2}
- 假设特征和对应权值，只注明特征取值为1，为0省略

$$\boxed{t_1=t_1(y_{i-1}=1,\ y_i=2,x,i),\quad i=2,3,\quad \lambda_1=1}$$

$$t_1(y_{i-1},\ y_i,x,i)=\begin{cases}1, & y_{i-1}=1,\ y_i=2,x,i,(i=2,3)\\ 0, & \text{其它}\end{cases}$$

$$\boxed{\begin{array}{ll} t_2=t_2(y_1=1,\ y_2=1,x,2), & \lambda_2=0.6\\ t_3=t_3(y_2=2,\ y_3=1,x,3), & \lambda_3=1\\ t_4=t_4(y_1=2,\ y_2=1,x,2), & \lambda_4=1\\ t_5=t_5(y_2=2,\ y_3=2,x,3), & \lambda_5=0.2\end{array}}$$

$$\boxed{\begin{array}{lll} s_1=s_1(y_1=1,\ x,1), & & \mu_1=1\\ s_2=s_2(y_i=2,\ x,i), & i=1,2 & \mu_2=0.5\\ s_3=s_3(y_i=1,\ x,i), & i=2,3 & \mu_3=0.8\\ s_4=s_4(y_3=2,\ x,3), & & \mu_4=0.5\end{array}}$$

- 对给定的观测序列x，求标记序列Y=(1, 2, 2)的非规范化条件概率

# 条件随机场的参数化形式：例子

- 解：根据如下求解公式

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

代入求得：

$$P(y|x)\propto\exp\left[\sum_{k=1}^{5}\lambda_k\sum_{i=2}^{3}t_k(y_{i-1},y_i,x,i)+\sum_{k=1}^{4}\mu_k\sum_{i=2}^{3}s_k(y_i,x,i)\right]$$

$$\lambda_1 t_1+\lambda_5 t_5+\mu_1 s_1+\mu_2 s_2+\mu_1 s_1+\mu_4 s_4=1+0.2+1+0.5+0.5=3.2$$

$$P(y1=1,y2=2,y3=2|x)\propto\exp(3.2)$$

# 条件随机场的简化形式

- 简化的目的：在参数化形式表示中，同一特征在各个位置都有定义，可以对同一个特征在各个位置求和 ，将局部特征函数转化为一个全局特征函数，这样就可以将条件随机场写成权值向量和特征向量的内积形式，即条件随机场的简化形式。
- 首先将转移特征和状态特征及其权值用统一的符号表示。设有$K_1$个转移特征，$K_2$个状态特征，$K=K_1+K_2$

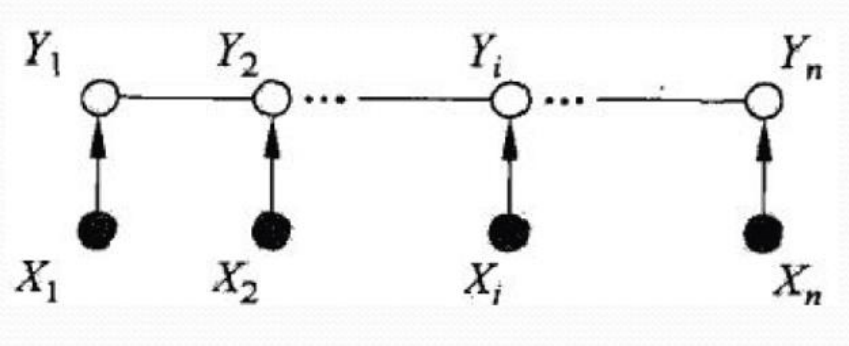
$$f_k(y_{i-1},y_i,x,i)=\begin{cases} t_k(y_{i-1},y_i,x,i), & k=1,2,\cdots,K_1 \\ s_l(y_i,x,i), \quad k=K_1+1; & l=1,2,\cdots,K_2 \end{cases}$$

特征的分段函数

- 然后，对转移特征与状态特征在各个位置i求和，记作

$$f_k(y,x)=\sum_{i=1}^{n} f_k(y_{i-1},y_i,x,i), \quad k=1,2,\cdots,K$$

# 条件随机场的简化形式




- 用$w_k$表示特征$f_k$（x，y）的权值，即

$$w_k = \begin{cases} \lambda_k, & k = 1,2,\cdots,K_1 \\ \mu_l, & k = K_1 + l;\ l = 1,2,\cdots,K_2 \end{cases}$$

权重的分段函数

- 于是，条件随机场可以表示为:

$$P(y|x) = \frac{1}{Z(x)} \exp\left( \sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \right)$$

$$\Rightarrow \quad P(y|x) = \frac{1}{Z(x)} \exp \boxed{\sum_{k=1}^{K} w_k f_k(y, x)}$$

$$Z(x) = \sum_{y} \exp \sum_{k=1}^{K} w_k f_k(y, x)$$

# 条件随机场的简化形式

$$P(y|x)=\frac{1}{Z(x)}\exp\boxed{\sum_{k=1}^{K}}w_k f_k(y,x)$$

$$Z(x)=\sum_{y}\exp\sum_{k=1}^{K}w_k f_k(y,x)$$

把上面公式中的求和转变为内积形式

- 若w表示权值**向量**：

$$w=(w_1,w_2,\cdots,w_K)^T$$

- 以F(y,x)表示全局特征**向量**，即

$$F(y,x)=(f_1(y,x),f_1(y,x),\cdots,f_K(y,x))^T$$

- 条件随机场写成**内积**：

$$P_w(y|x)=\frac{\exp(w\bullet F(y,x))}{Z_w(x)}$$

$$Z_w(x)=\sum_{y}\exp(w\bullet F(y,x))$$

# 条件随机场的矩阵形式

- 以上是CRF的参数化及其简化形式，为了计算方便，可以表示为矩阵形式
- 首先引进特殊的起点和终点状态标记$Y_0$=start, $Y_{n+1}$=stop，这时$P_w(y|x)$可以通过矩阵形式表示
- 需要对观察序列x的每一个位置i=1,2,..,n+1，定义一个m阶矩阵(m是标记$Y_i$取值的个数)

$$M_i(x) = \left[M_i(y_{i-1}, y_i | x)\right]$$

$$M_i(y_{i-1}, y_i | x) = \exp\left(W_i(y_{i-1}, y_i | x)\right)$$

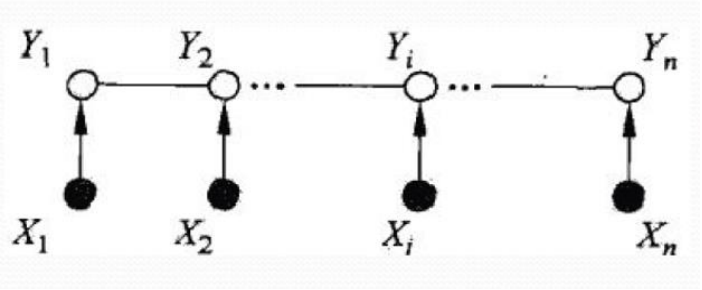
$$W_i(y_{i-1}, y_i | x) = \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, x, i)$$

直观上，$M_i$表示从位置i-1到位置i时，状态从$y_{i-1}$转移到$y_i$的概率
该概率由一组（K）转移特征、状态特征和对应的权重决定。

# 条件随机场的矩阵形式

■例子：

**假设y可以取N和V两种标签，如公式所示，y1取N，y2取N的概率是由K个特征函数决定的，其中有K1个转移特征，K2个状态特征。**




i=2时

| | $N$ | $V$ |
|---|---|---|
| $N$ | $M_2(N-N)=\exp(\sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, \mathrm{x}, i))$ | $M_2(N-V)$ |
| $V$ | $M_2(V-N)$ | $M_2(V-V)$ |

# 条件随机场的矩阵形式

- 定义为矩阵形式之后，条件概率$P_w(y|x)$：

$$\boxed{P_w(y|x)=\frac{1}{Z_w(x)}\prod_{i=1}^{n+1}M_i(y_{i-1},y_i|x)}$$

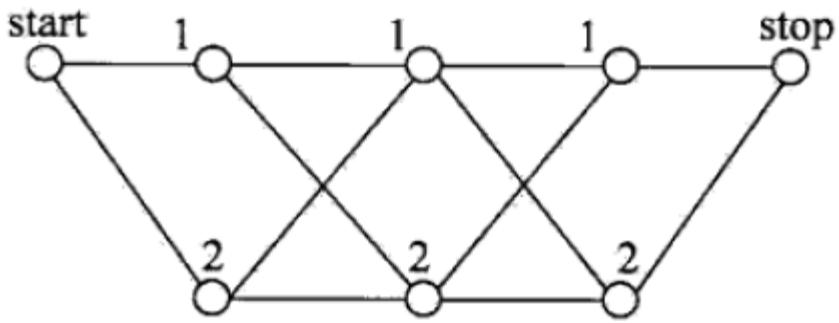
- $Z_w(x)$为规范化因子，是n+1个矩阵的乘积的(start, stop)元素。具体地，规范化因子$Z_w(x)$是以start为起点stop为终点通过状态的所有路径$y_1y_2y_3\ldots y_n$的非规范化概率$\prod_{i=1}^{n+1}M_i(y_{i-1},y_i|x)$之和

$$Z_w(x)=\left(M_1(x)M_2(x)\cdots M_{n+1}(x)\right)_{start,stop}$$

乘积的形式，方便递推计算

# 条件随机场的矩阵形式：例子

- 给定一个下图所示的线性链条件随机场，观测序列x，状态序列y，i=1,2,3, n=3，标记$y_i \in \{1,2\}$，假设$y_0$=start=1，$y_4$=stop=1，各个位置的随机矩阵$M_1(x)$， $M_2(x)$， $M_3(x)$，$M_4(x)$分别是：




- 试求状态序列y以start为起点stop为终点所有路径的非规范概率及规范化因子

$$M_1(x)=\begin{bmatrix} a_{01} & a_{02} \\ 0 & 0 \end{bmatrix} \quad M_2(x)=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad M_3(x)=\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \quad M_4(x)=\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

# 条件随机场的矩阵形式：例子

$$M_1(x)=\begin{bmatrix} a_{01} & a_{02} \\ 0 & 0 \end{bmatrix} \qquad M_2(x)=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$M_3(x)=\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \qquad M_4(x)=\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

- 首先计算图中从start到stop对应于y=（1,1,1），y=（1,1,2）,…,y=(2,2,2)各路径的非规范化概率分别是：

$$a_{01}b_{11}c_{11} \qquad a_{01}b_{11}c_{12} \qquad a_{01}b_{12}c_{21} \qquad a_{01}b_{12}c_{22}$$

$$a_{02}b_{21}c_{11} \qquad a_{02}b_{21}c_{12} \qquad a_{02}b_{22}c_{21} \qquad a_{02}b_{22}c_{22}$$

- 通过计算矩阵乘积$M_1(x)$ $M_2(x)$ $M_3(x)M_4(x)$可知，其第一行第一列的元素如下，恰好等于从start到stop的所有路径的非规范化概率之和

$$a_{01}b_{11}c_{11}+a_{02}b_{21}c_{11}+a_{01}b_{12}c_{21}+a_{02}b_{22}c_{22}+a_{01}b_{11}c_{12}+a_{02}b_{21}c_{12}+a_{01}b_{12}c_{22}+a_{02}b_{22}c_{21}$$

# 条件随机场：概率计算问题

- 问题描述：给定条件随机场P(Y|X)，输入序列x和输出序列y计算条件概率：

$$P\left(Y_i \middle| X\right)$$

标记序列在位置i为$y_i$的概率

$$P\left(Y_{i-1} = y_{i-1}, Y_i = y_i \middle| x\right)$$

标记序列在位置i-1为$y_{i-1}$且在位置i为$y_i$概率

- 为了提高计算效率，像HMM那样，引进前向-后向向量，进行递推计算（前向后向算法）

# CRF：概率计算问题

$$P_w(y|x)=\frac{1}{Z_w(x)}\prod_{i=1}^{n+1}M_i(y_{i-1},y_i|x)$$

- 前向-后向算法：
  - 对每个位置i=0,1,...,n+1，定义前向向量 $\alpha_i(x)$

$$\alpha_0(y|x)=\begin{cases}1, & y=start\\ 0, & 否则\end{cases}$$

  - 递推公式：

一个值 $$\alpha_i^T(y_i|x)=\alpha_{i-1}^T(y_{i-1}|x)M_i(y_{i-1},y_i|x),\quad i=1,2,\cdots,n+1$$

表示：在位置i的标记是$y_i$，且到位置i的前部分标记序列($y_0$ $y_1$ $y_2$.....$y_{i-1}$)的非规范化概率

  - 又可表示为：

$$\alpha_i^T(x)=\alpha_{i-1}^T(x)M_i(x)$$

一个m维的列向量

因为$y_i$可取的值m个，所以$\alpha_i(x)$ 是m维列向量

# 条件随机场的概率计算问题

- 前向-后向算法：
  - 同样，对每个指标i=0,1,…,n+1，定义后向向量 $\beta_i(x)$

$$\beta_{n+1}(y_{n+1}|x)=\begin{cases}1, & y_{n+1}=stop\\ 0, & \text{否则}\end{cases}$$

$$\beta_i(y_i|x) = M_{i+1}(y_i, y_{i+1}|x)\beta_{i+1}(y_{i+1}|x)$$

  - 表示：在位置i的标记是$y_i$，且从位置i+1到n+1的后部分标记序列($y_{i+1}$ $y_{i+2}$……$y_{n+1}$)的非规范化概率
  - 又可表示为：

$$\beta_i(x)=M_{i+1}(x)\beta_{i+1}(x)$$

$$Z(x)=\alpha_n^T(x)\bullet 1=1^T\bullet\beta_1(x)$$

# 条件随机场：概率计算问题

- 前向-后向算法：
  - 根据上述前向和后向变量的定义可以计算标记序列在位置i是标记$y_i$的条件概率

$$P(Y_i = y_i|x) = \frac{\alpha_i^T(y_i|x)\beta_i(y_i|x)}{Z(x)}$$

  - 和在位置i-1与i是标记$y_{i-1}$和$y_i$的条件概率：

$$P(Y_{i-1} = y_{i-1}, Y_i = y_i|x) = \frac{\alpha_{i-1}^T(y_{i-1}|x)M_i(y_{i-1}, y_i|x)\beta_i(y_i|x)}{Z(x)}$$

  - 其中

$$Z(x) = \alpha_n^T(x) \cdot \mathbf{1}$$

# 条件随机场：解码问题

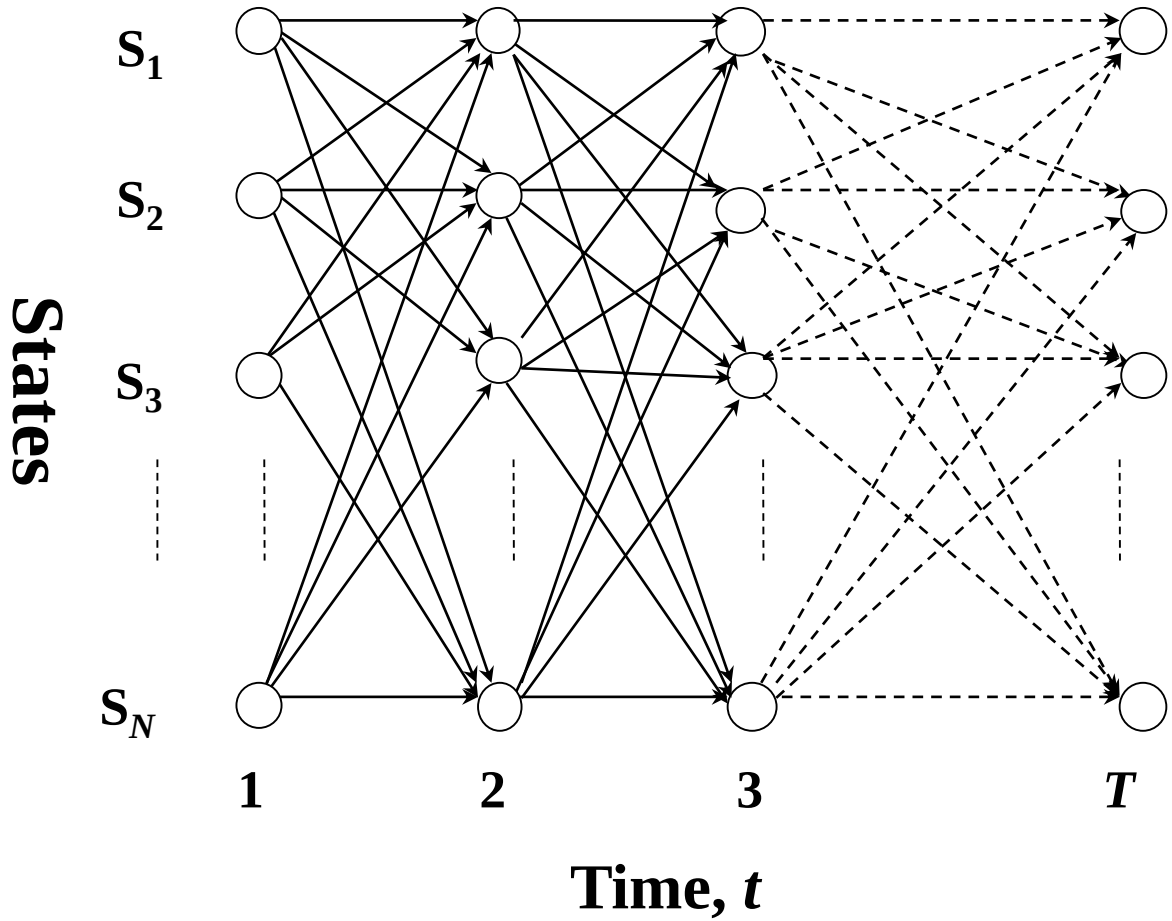
- 问题描述：给定条件随机场P(Y|X)和输入序列(观察序列)x，求条件概率最大的输出序列(标记序列)y*
- Viterbi算法：

$$P_w(y|x) = \frac{\exp(w \bullet F(y,x))}{Z_w(x)}$$

$$\begin{aligned} y^* &= \arg\max_y P_w(y|x) \\ &= \arg\max_y \frac{\exp(w \bullet F(y,x))}{Z_w(x)} \\ &= \arg\max_y \exp(w \bullet F(y,x)) \\ &= \arg\max_y (w \bullet F(y,x)) \end{aligned}$$

# 条件随机场：解码问题

- Viterbi 搜索算法



# 条件随机场：解码问题

- 根据上述公式，条件随机场的预测问题成为求非规范化概率最大的最优路径问题

$$\max_{y}\left(w \bullet F(y,x)\right)$$

- 其中，路径表示标记序列，其中

$$w=\left(w_1,w_2,\cdots,w_K\right)^T$$

$$F(y,x)=\left(f_1(y,x),f_1(y,x),\cdots,f_K(y,x)\right)^T$$

$$f_k(y,x)=\sum_{i=1}^{n} f_k\left(y_{i-1},y_i,x,i\right),\quad k=1,2,\cdots,K$$

01函数

# 条件随机场的解码问题

- 注意，为了提高计算效率，只需计算非规范化概率，而不必计算概率，于是，为了求解最优路径，上述式子可以写成如下形式

$$\max \sum_{i=1}^{n} w \bullet F_i(\mathrm{y}_{i-1}, \mathrm{y}_i, x)$$

- 其中

$$F_i(y_{i-1}, y_i, x) = \left(f_1(y_{i-1}, y_i, x, i), f_2(y_{i-1}, y_i, x, i), \cdots, F_K(y_{i-1}, y_i, x, i)\right)^T$$

每个位置的各种特征，每一个是一个01函数

# 条件随机场：解码算法

- Viterbi算法：
  - 初始化：首先求出位置1的各个标记j=1,2..m的非规范化概率：

$$\delta_i(j) = w \bullet F_i(y_0 = start, y_1 = j, x), \quad j = 1,2,\cdots,m$$

  - 递归计算：

$$\delta_i(l) = \max_{1 \le f \le m} \{\delta_{i-1}(j) + w \bullet F_i(y_{i-1} = j, y_i = l, x)\}, \quad l = 1,2,\cdots,m$$

$$\Psi_i(l) = \arg\max_{1 \le f \le m} \{\delta_{i-1}(j) + w \bullet F_i(y_{i-1} = j, y_i = l, x)\}, \quad l = 1,2,\cdots,m$$

# 条件随机场：解码算法

- Viterbi算法：
  - 终止：直到i=n时终止，这时求得非规范化概率的最大值为

$$\max_{y}\left(w \bullet F\left(y, x\right)\right) = \max_{1 \le j \le m} \delta_n\left(j\right)$$

$$y_n^* = \arg\max_{1 \le j \le m} \delta_n\left(j\right)$$

  - 回溯路径：

$$y_i^* = \Psi_{i+1}\left(y_{i+1}^*\right), \quad i = n-1, n-2, \cdots, 1$$

$$y^* = \left(y_1^*, y_2^*, \cdots, y_n^*\right)^T$$

# 条件随机场：解码算法的例子

- 输入观测为：$X=(X_1,X_2,X_3)$，输出标记为$Y=(Y_1,Y_2,Y_3)$，$Y_1$, $Y_2$, $Y_3$ 取值于{1,2}
- 假设特征和对应权值，只注明特征取值为1，为0省略

$$\boxed{t_1=t_1(y_{i-1}=1,\ y_i=2,x,i),\quad i=2,3,\quad \lambda_1=1}$$

$$t_1(y_{i-1},\ y_i,x,i)=\begin{cases}1, & y_{i-1}=1,\ y_i=2,x,i,(i=2,3)\\ 0, & \text{其它}\end{cases}$$

$$\boxed{\begin{array}{ll} t_2=t_2(y_1=1,\ y_2=1,x,2), & \lambda_2=0.6\\ t_3=t_3(y_2=2,\ y_3=1,x,3), & \lambda_3=1\\ t_4=t_4(y_1=2,\ y_2=1,x,2), & \lambda_4=1\\ t_5=t_5(y_2=2,\ y_3=2,x,3), & \lambda_5=0.2\end{array}}$$

$$\boxed{\begin{array}{lll} s_1=s_1(y_1=1,\ x,1), & & \mu_1=1\\ s_2=s_2(y_i=2,\ x,i), & i=1,2 & \mu_2=0.5\\ s_3=s_3(y_i=1,\ x,i), & i=2,3 & \mu_3=0.8\\ s_4=s_4(y_3=2,\ x,3), & & \mu_4=0.5\end{array}}$$

- 对给定的观测序列x，求对应的最优输出序列 $y^*$

# 条件随机场：解码算法的例子

- 解：
  - 利用Viterbi算法求最优路径问题

$$\max \sum_{i=1}^{3} w \bullet F_i(\mathrm{y}_{i-1}, \mathrm{y}_i, x)$$

  - 初始化

$$i = 1\text{时} \quad \delta_1(j) = w \bullet F_1(\mathrm{y}_0 = \mathrm{start}, \mathrm{y}_1 = j, x), \qquad j = 1,2$$

$$\delta_1(1) = \mu_1 s_1 = 1$$

$$\delta_1(2) = \mu_2 s_2 = 0.5$$

# 条件随机场：解码算法的例子

- 解：
  - 递推

$i=2$时 $\quad \delta_2(l)=\max\{\delta_1(j)+w\bullet F_2(j,l,x)\}$

$$\delta_2(1)=\max\{\delta_1(1)+\lambda_2 t_2+\mu_3 s_3,\delta_1(2)+\lambda_4 t_4+\mu_3 s_3\}$$
$$=\max\{1+0.6+0.8,0.5+1+0.8\}=2.4$$
$$\psi_2(1)=1$$

$$\delta_2(2)=\max\{\delta_1(1)+\lambda_1 t_1+\mu_2 s_2,\delta_1(2)+\mu_2 s_2\}$$
$$=\max\{1+1+0.5,0.5+0.5\}=2.5$$
$$\psi_2(2)=1$$

# 条件随机场：解码算法的例子

- 解：
  - 递推

$i=3$时 $\delta_3(l)=\max\{\delta_2(j)+w\bullet F_3(j,l,x)\}$

$$\delta_3(1)=\max\{\delta_2(1)+\mu_3 s_3,\delta_2(2)+\lambda_3 t_3+\mu_3 s_3\}$$

$$=\max\{2.4+0.8,2.5+1+0.8\}=4.3$$

$$\psi_3(1)=2$$

$$\delta_3(2)=\max\{\delta_1(1)+\lambda_1 t_1+\mu_4 s_4,\delta_1(2)+\lambda_5 t_5+\mu_4 s_4\}$$

$$=\max\{2.4+1+0.5,2.5+0.2+0.5\}=3.9$$

$$\psi_3(2)=1$$

# 条件随机场：解码算法的例子

- 解：
  - 终止

$$\max_{y}(w \bullet F(x,y)) = \max \delta_3(l) = \delta_3(1) = 4.3$$

$$y_3^* = \arg\max_{l} \delta_3(l) = 1$$

  - 返回

$$y_2^* = \psi_3(y_3^*) = \psi_3(1) = 2$$

$$y_1^* = \psi_2(y_2^*) = \psi_2(2) = 1$$

  - 最优标记序列

$$y^* = (y_1^*, y_2^*, y_3^*) = (1,2,1)$$

# 条件随机场：学习问题

参考《统计机器学习》附录

■拟牛顿法：

$$P_w(y|x) = \frac{\exp(\sum_{i=1}^{n} w_i f_i(x,y))}{\sum_y \exp(\sum_{i=1}^{n} w_i f_i(x,y))}$$

■学习的优化目标函数：

$$\min_{w \in R^n} f(w) = \sum_x \tilde{P}(x) \log \sum_y \exp(\sum_{i=1}^{n} w_i f_i(x,y)) - \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^{n} w_i f_i(x,y)$$

■梯度函数：

$$g(w) = \sum_{x,y} \tilde{P}(x) P_w(y|x) f(x,y) - E_{\tilde{P}}[f]$$

有监督的学习

$$g(w) = E_P[f] - E_{\tilde{P}}[f]$$

# HMM vs ME vs CRF

- HMM是一种生成式模型，利用转移矩阵和生成矩阵建模相邻状态的转移概率和状态到观察的生成概率。无法利用复杂特征（只有两个矩阵建模）

- ME是一种判别式模型，可以使用任意的复杂特征（特征函数），但是ME只能建模观察序列和某一状态的关系，状态之间的关系无法得到充分利用。

- CRF是一种判别式模型，可以使用任意的复杂特征（特征函数）（虽然每个位置有一个矩阵，但是矩阵的元素是由特征函数和权重计算得到，我们可以任意地定义特征函数从而考虑各种特征），可以建模观察序列和多个状态的关系，考虑了状态之间的关系。

# CRF工具

- 参考项目：python-crfsuite
  - https://github.com/scrapinghub/python-crfsuite
  - 安装： pip install python-crfsuite

scrapinghub / python-crfsuite

Watch 18 | Star 251 | Fork 94

Code | Issues 10 | Pull requests 2 | Projects 0 | Wiki | Insights

Branch: master | python-crfsuite / examples / CoNLL 2002.ipynb | Find file | Copy path

samgalen Added trainer.logparser snippet to example notebook 45b9ab3 on 23 Jun 2016

2 contributors

854 lines (853 sloc) | 22.3 KB | Raw | Blame | History

In [1]:
```
from itertools import chain
import nltk
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import LabelBinarizer
import sklearn
import pycrfsuite

print(sklearn.__version__)
```

```
0.16.1
```

# CRF工具

- 参考任务(Chunking)：https://www.clips.uantwerpen.be/conll2000/chunking/
- 示例：He reckons the current account deficit will narrow to only # 1.8 billion in September.

[NP He ] [VP reckons ] [NP the current account deficit ] [VP will narrow ] [PP to ] [NP only # 1.8 billion ] [PP in ] [NP September ] .

标注：He\B-NP reckons \B-VP the \B-NP current \I-NP account \I-NP deficit \I-NP will \B-VP narrow \I-NP to \B-PP only \B-NP # \I-NP 1.8 \I-NP billion \I-NP in \B-PP September \B-NP . \O

- 数据集：
  - http://www.clips.uantwerpen.be/conll2000/chunking/train.txt.gz
  - http://www.clips.uantwerpen.be/conll2000/chunking/test.txt.gz

# CRF工具

## ■特征设计：

- w[t-2], w[t-1], w[t], w[t+1], w[t+2],
- w[t-1]|w[t], w[t]|w[t+1],
- pos[t-2], pos[t-1], pos[t], pos[t+1], pos[t+2],
- pos[t-2]|pos[t-1], pos[t-1]|pos[t], pos[t]|pos[t+1], pos[t+1]|pos[t+2],
- pos[t-2]|pos[t-1]|pos[t], pos[t-1]|pos[t]|pos[t+1], pos[t]|pos[t+1]|pos[t+2]

```
      He PRP B-NP
      reckons VBZ B-VP
t —> the DT B-NP
      current JJ I-NP
      account NN I-NP
```

- w[-2]=He, w[-1]=reckons, w[0]=the, w[1]=current, w[2]=account
- w[-1]|w[0]=reckons|the, w[0]|w[1]=the|current
- pos[-2]=PRP, pos[-1]=VBZ, pos[0]=DT, pos[1]=JJ, pos[2]=NN
- pos[-2]|pos[-1]=PRP|VBZ, pos[-1]|pos[0]=VBZ|DT, pos[0]|pos[1]=DT|JJ, pos[1]|pos[2]=JJ|NN
- pos[-2]|pos[-1]|pos[0]=PRP|VBZ|DT, pos[-1]|pos[0]|pos[1]=VBZ|DT|JJ, pos[0]|pos[1]|pos[2]=DT|JJ|NN

# 基于CRF的命名实体识别

- 采用**CRF**模型对每个汉字标注对应实体类型的BIO标记
- 例如：句子“毛泽东是中国伟大的主席”
  - 观察序列： 毛1 泽2 东3 是4 中5 国6 伟7 大8 的9 主10 席11
  - 标注序列：

    B-PER1 I-PER2 I-PER3 O4 B-LOC5 I-LOC6 O7 O8 O9 O10 O11
- 特征函数：可以构建不同种类的特征
  - 上下文特征：{$y_{i-1}$= B-LOC，$y_i$= I-LOC，$x_{i-2}$=是，i=6}

    {$y_{i-1}$= #，$y_i$= I-LOC，$x_{i-2}$=是，i=6}
  - 词性特征：{$x_{i-2}$=v，i=6}{$x_{i-1}$=n，i=6} {$x_0$=n，i=6}
  - 词表特征：{$x_{i-1}x_i$=1，i=6}

# 基于CRF的命名实体识别

- CRF模型的参数训练
  - 对训练语料库中每句话生成一个实例
  - 把所有实例的列表送给CRF模型的训练工具
  - CRF的训练工具将学习到模型的参数（每个特征的权重）
- CRF模型的测试
  - 对于输入串整体作为一个实例
  - 对这每个实例，生成其所有激活的特征，计算这些特征所对应的概率
  - 选取可能性最大的标记序列为该句子的标记

# LSTM+CRF命名实体识别（Lample NAACL16）

■动机1

领域内资源（例如地理资源集gazetteers）或特征的获取耗时耗力，避免使用领域内的资源或特征

■动机2

传统的特征提取过程中会用到复杂的NLP工具，会造成误差传递

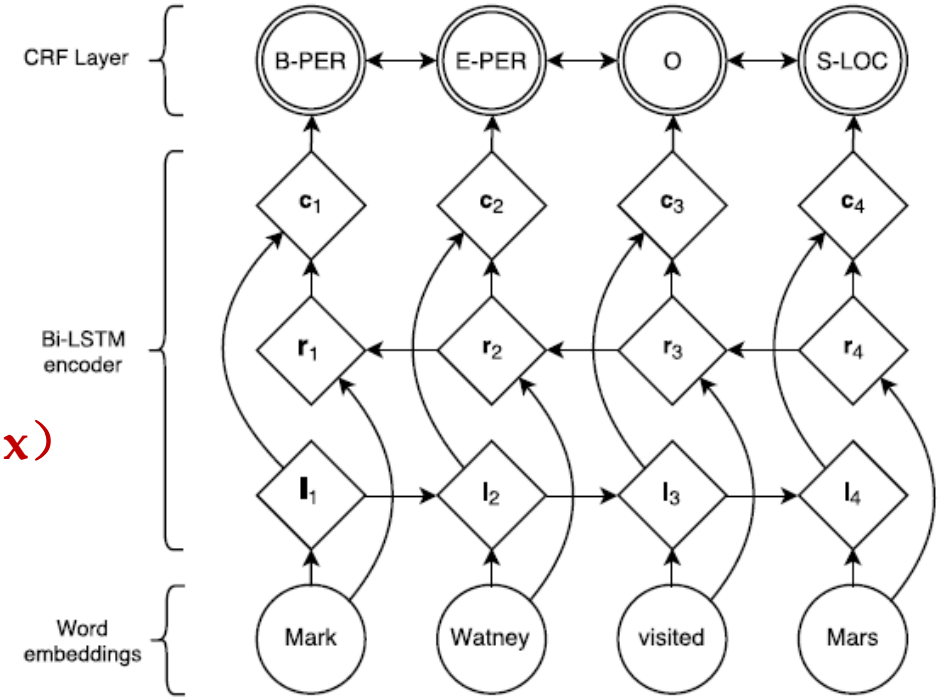
利用神经网络自动从文本中学习特征

# LSTM+CRF命名实体识别（Lample NAACL16）

## ■LSTM+CRF命名实体识别联合模型

每个句子按照词序逐个输入双向LSTM中，结合正反向隐层输出得到包含每个词类别信息的表示，输入CRF中，优化目标函数，从而得到每个词所属的实体类别

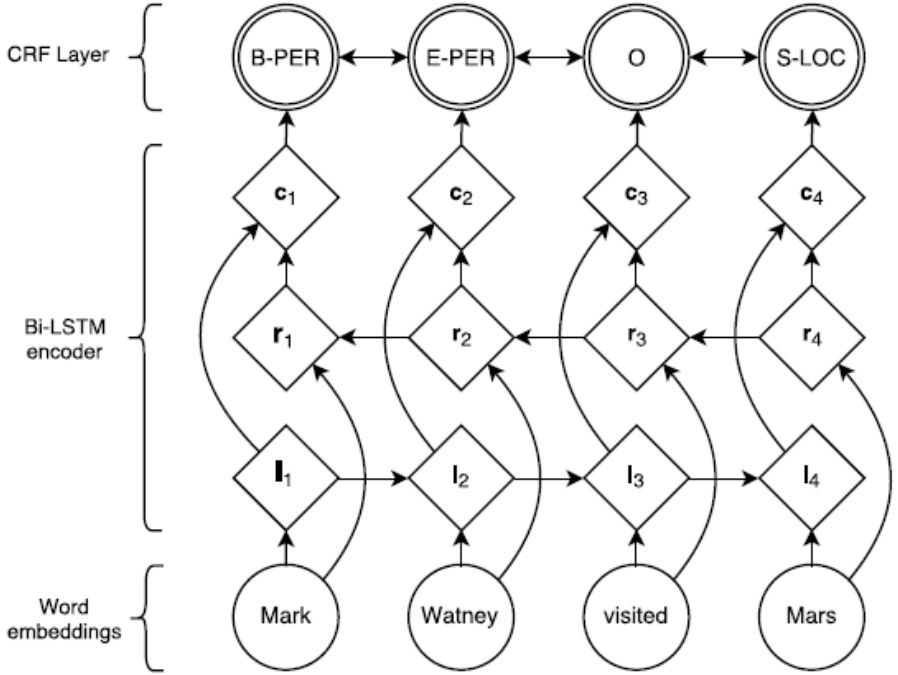
**LSTM学习得到的特征相当于传统的CRF中的状态特征S（$y_i|x$）**

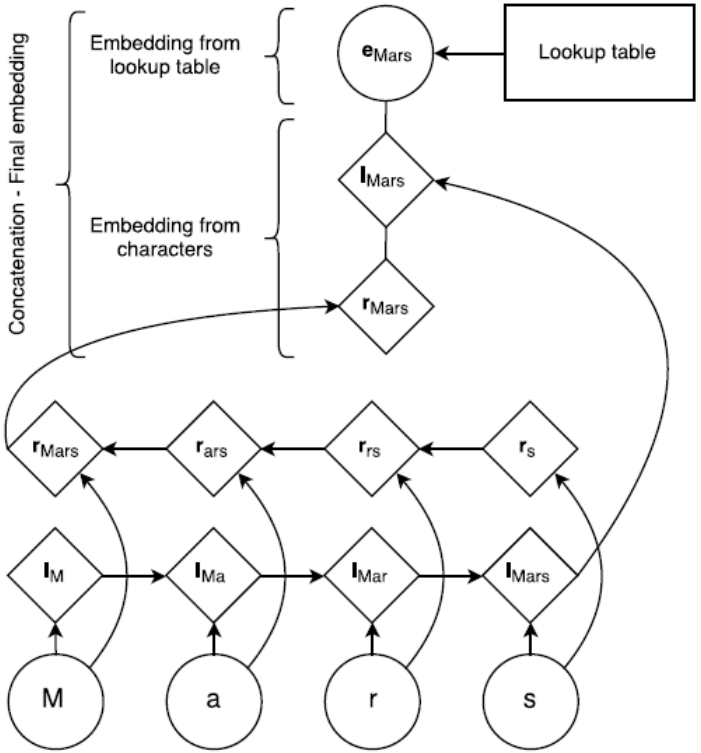


# LSTM+CRF命名实体识别（Lample NAACL16）

## ■词级别特征与字符级别特征

词级别的特征即每个词的词向量作为词的表示，字符级别特征是通过将每个词拆成字符序列，通过双向LSTM得到该词的表示，将两种表示结合，作为该词的LSTM+CRF模型输入



# LSTM+CRF命名实体识别（Lample NAACL16）

## ■数据集与实验

数据集：CoNLL2003

| Model | $F_1$ |
|---|---|
| Collobert et al. (2011)* | 89.59 |
| Lin and Wu (2009) | 83.78 |
| Lin and Wu (2009)* | 90.90 |
| Huang et al. (2015)* | 90.10 |
| Passos et al. (2014) | 90.05 |
| Passos et al. (2014)* | 90.90 |
| Luo et al. (2015)* + gaz | 89.9 |
| Luo et al. (2015)* + gaz + linking | **91.2** |
| Chiu and Nichols (2015) | 90.69 |
| Chiu and Nichols (2015)* | 90.77 |
| LSTM-CRF (no char) | 90.20 |
| LSTM-CRF | **90.94** |

**Table 1:** English NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Florian et al. (2003)* | 72.41 |
| Ando and Zhang (2005a) | 75.27 |
| Qi et al. (2009) | 75.72 |
| Gillick et al. (2015) | 72.08 |
| Gillick et al. (2015)* | 76.22 |
| LSTM-CRF – no char | 75.06 |
| LSTM-CRF | **78.76** |
| S-LSTM – no char | 65.87 |
| S-LSTM | 75.66 |

**Table 2:** German NER results (CoNLL-2003 test set).

| Model | $F_1$ |
|---|---|
| Carreras et al. (2002) | 77.05 |
| Nothman et al. (2013) | 78.6 |
| Gillick et al. (2015) | 78.08 |
| Gillick et al. (2015)* | **82.84** |
| LSTM-CRF – no char | 73.14 |
| LSTM-CRF | **81.74** |

**Table 3:** Dutch NER (CoNLL-2002 test set).

# 汉语分词与实体识别联合模型（PengACL16）

- 动机1

  词边界能提供实体类别的有用信息，所以将词边界信息融入到命名实体识别中

- 动机2

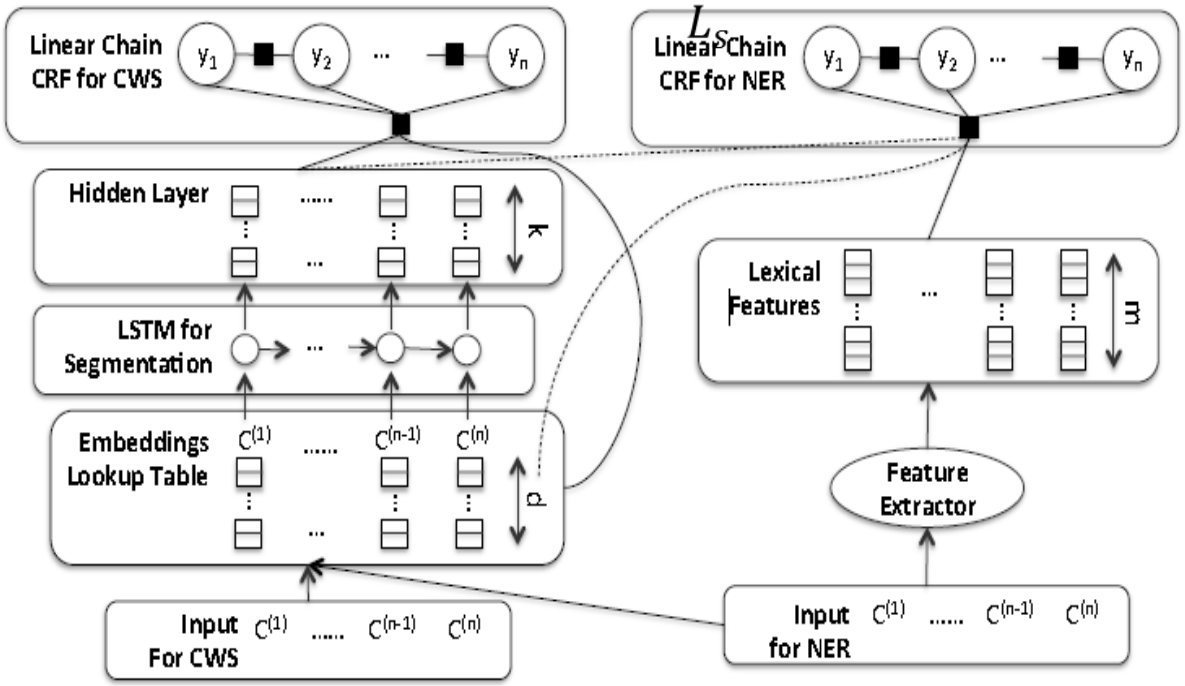
  汉语分词困难，与英语相比较分词准确率低，先分词，后实体识别，分词时的误差会传递到实体识别中，所以将分词与实体识别联合训练

# 汉语分词与实体识别联合模型（PengACL16）

## ■汉语分词与实体识别联合模型

首先，利用和上一篇类似的模型去做分词

然后，将分词阶段LSTM隐层输出得到分词特征、词向量特征和传统的NER特征融合，输入到CRF模型中，进行实体识别，分词和实体识别分别定义损失函数，最后将两个损失函数融合，实现两个任务的联合学习。



# 汉语分词与实体识别联合模型（PengACL16）

## ■结果

实体识别模型中，除了基础的词汇特征外，还有词语特征与LSTM隐层特征，9/10/11三组实验分别代表单独使用词语特征、单独使用LSTM隐层特征和共同使用词语特征和隐层特征，可以看出，三个特种共同使用效果最好

| | | Named Entity | | | | | | Nominal Mention | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dev | | | Test | | | Dev | | | Test | | |
| | Method | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| 1 | CRF with baseline features | 60.27 | 25.43 | 35.77 | 57.47 | 25.77 | 35.59 | 72.06 | 32.56 | 44.85 | 59.84 | 23.55 | 33.80 |
| 2 | + Segment Features | 62.34 | 27.75 | 38.40 | 58.06 | 27.84 | 37.63 | 58.50 | 38.87 | 46.71 | 47.43 | 26.77 | 34.23 |
| 3 | P & D best NER model | 57.41 | 35.84 | 44.13 | 57.98 | 35.57 | 44.09 | 72.55 | 36.88 | 48.90 | 63.84 | 29.45 | 40.38 |
| 4 | + Segment Features | 47.40 | 42.20 | 44.65 | 48.08 | 38.66 | 42.86 | 76.38 | 36.54 | 49.44 | 63.36 | 26.77 | 37.64 |
| 5 | P & D w/ Char Embeddings | 58.76 | 32.95 | 42.22 | 57.89 | 34.02 | 42.86 | 66.88 | 35.55 | 46.42 | 55.15 | 29.35 | 38.32 |
| 6 | + Segment Features | 51.47 | 40.46 | 45.31 | 52.55 | 37.11 | 43.50 | 65.43 | 40.86 | 50.31 | 54.01 | 32.58 | 40.64 |
| 7 | Pipeline Seg. Repr. + NER | 64.71 | 38.14 | **48.00** | 64.22 | 36.08 | 46.20 | 69.36 | 39.87 | 50.63 | 56.52 | 33.55 | 42.11 |
| 8 | Jointly LSTM w/o feat. | 59.22 | 35.26 | 44.20 | 60.00 | 35.57 | 44.66 | 60.10 | 39.53 | 47.70 | 56.90 | 31.94 | 40.91 |
| 9 | Jointly Train Char. Emb. | 64.21 | 35.26 | 45.52 | 63.16 | 37.11 | 46.75 | 73.55 | 37.87 | 50.00 | 65.33 | 31.61 | 42.61 |
| 10 | Jointly Train LSTM Hidden | 61.86 | 34.68 | 44.44 | 63.03 | 38.66 | 47.92 | 67.23 | 39.53 | 49.79 | 60.00 | 33.87 | 43.30 |
| 11 | Jointly Train LSTM + Emb. | 59.29 | 38.73 | 46.85 | 63.33 | 39.18 | **48.41** | 61.61 | 43.19 | **50.78** | 58.59 | 37.42 | **45.67** |

# 命名实体识别的评测

- 国际会议：MUC、SigHAN、CoNLL、IEER 和ACE
  - MUC-6和MUC-7设立的命名实体识别专项评测大大推动了英语命名实体识别技术的发展
  - MUC-6和MUC-7还设立了多语言实体识别评测任务MET，对日语、土耳其语、汉语等多种语言命名实体识别任务进行评测
  - SigHAN从2003年开始举办第一届中文分词评测BAKEOFF，2006年和2008年举行的BAKEOFF-3和BAKEOFF-4设立了命名实体识别专项评测
  - 2003年和2004年举办的863计划“中文信息处理与智能人机接口技术评测”中设立了中文命名实体识别评测任务
  - 评测指标：正确率P，召回率R和 F值

# 英文命名实体识别的技术水平

- 不同语言，不同数据集，不同的方法能达到不同的性能
- 英文：Language Technology Group Summary开发的英语命名实体识别系统在MUC-7评测中取得第一名，其准确率和召回率分别达到95%和92%（吴友政，2006）
- 许多英语命名实体系统已经具备了相当程度的大规模文本处理能力

# 汉语命名实体识别的技术水平

- 参加MET-2评测的汉语命名实体识别系统对人名、地名、机构名识别的最优性能指标（准确率，召回率）只有(66%，92%)、(89%，91%)和(89%，88%)（吴友政，2006）

# 汉语命名实体识别的技术水平：BAKEOFF-3

| 数据来源 | 测试类别 | 测试性能 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | P | R | F | ORG- F | LOC-F | PER-F | GPE-F |
| MSRC（简） | Closed | 0.8894 | 0.8420 | 0.8651 | 0.8310 | 0.8545 | 0.9009 | ~ |
| | Open | 0.9220 | 0.9018 | 0.9118 | 0.8590 | 0.9034 | 0.9604 | ~ |
| LDC（简） | Closed | 0.8026 | 0.7265 | 0.7627 | 0.6585 | 0.3046 | 0.7884 | 0.8204 |
| | Open | 0.7616 | 0.6621 | 0.7084 | 0.5209 | 0.2857 | 0.7422 | 0.7930 |
| CITYU（繁） | Closed | 0.9143 | 0.8676 | 0.8903 | 0.8046 | 0.9211 | 0.9087 | ~ |
| | Open | 0.8692 | 0.7498 | 0.8051 | 0.6801 | 0.8604 | 0.8098 | ~ |

| 数据来源 | 简繁体类别 | 训练集规模 | 测试集规模 |
|---|---|---|---|
| MSRC | 简体 | 1.3M/63K（词/词次） | 100K/13K（词/词次） |
| LDC | 简体 | 632K（词） | 61K（词） |
| CITYU | 繁体 | 1.6M/76K（词/词次） | 220K/23K（词/词次） |

(Levow, 2006 )

# 汉语命名实体识别的技术水平：863

| 数据来源 | 测试类别 | 测试性能 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ORG | | | LOC | | | PER | | |
| | | P | R | F | P | R | F | P | R | F |
| SXU (简) | 开放 | .6464 | .5741 | .6081 | .8702 | .7843 | .8251 | .8138 | .8847 | .8478 |
| CITYU(繁) | 开放 | .3986 | .2532 | .3097 | .6839 | .7004 | .6921 | .3986 | .2532 | .3097 |

| 数据来源 | 简繁体类别 | 训练集规模（词/词次） | 测试集规模（字） |
|---|---|---|---|
| SXU | 简体 | NONE | 约400K |
| CITYU | 繁体 | NONE | 约400K |

# 从评测结果看汉语命名实体识别的技术水平

- 在BAKEOFF-3 MSRC语料和BAKEOFF-3 CITYU语料上的评测结果要好于BAKEOFF-3 LDC语料上的评测结果以及863语料上的评测结果
- 其中一个很重要原因是：BAKEOFF-3 MSRC和CITYU评测提供了相当规模的训练集，而BAKEOFF-3 LDC只提供了小规模的训练集，而863评测根本不提供训练集
- 因为训练集和测试集在题材和体裁方面比较类似，可能使得各个系统在BAKEOFF-3 MSRC语料和BAKEOFF-3 CITYU语料上的评测性能较高
- **在真实的应用环境中，命名实体识别的性能会大打折扣**

# 可选的NER工具

| | 分词系统 | 标识 |
|---|---|---|
| 1 | BosonNLP | BOSON |
| 2 | IKAnalyzer | 中文分词库 IKAnalyzer |
| 3 | NLPIR | NLPIR 汉语分词系统 又名：ICTCLAS 2015 |
| 4 | SCWS | SCWS 中文分词 |
| 5 | 结巴分词 | jieba |
| 6 | 盘古分词 | 盘古分词-开源中文分词组件 |
| 7 | 庖丁解牛 | paoding Lucene中文分词"庖丁解牛" Paoding Analysis |
| 8 | 搜狗分词 | Sogou搜狗 云服务 |
| 9 | 腾讯文智 | 互联网+ 腾讯云 |
| 10 | 新浪云 | 新浪云 |
| 11 | 语言云 | 語言 |

- http://ictclas.nlpir.org/中科院计算所(ICTCLAS)
- http://www.openpr.org.cn中科院自动化所(Urheen)
- http://www.fnlp.org/复旦大学
- http://nlp.stanford.edu/software/tagger.shtml Stanford University

# 小结

- 受限于训练语料规模，系统的自适应能力不强
  - 网页信息：不规范、存在很多噪音，有些根本就不构成自然语言句子，因此通常的命名实体识别模型所依赖的上下文特征发生了明显变化，使得识别性能剧烈下降
- 类别数限定，不满足实际的应用
  - 摩托罗拉V8088折叠手机、第6届苏迪曼杯羽毛球混合团体赛、胆结石腹腔镜手术等
- 需要开放域实体抽取
  - 实体类型更多、更细，而且有些实体类别是未知、或者是随时间演化的
- 基于深度学习的分词和命名实体识别

# 提纲

- 引言
- 信息抽取概述
- 中文分词
- 命名实体识别
- 开放域实体识别

# 开放类别实体抽取

- 不限定实体类别
- 不限定目标文本
- 任务：给定某一类别的实体实例，从网页中抽取同一类别其他实体实例
  - 给出<中国，美国，俄罗斯>（称为“种子”）
  - 找出其他国家<德国，英国，法国……>

Input

中国

美国

俄罗斯

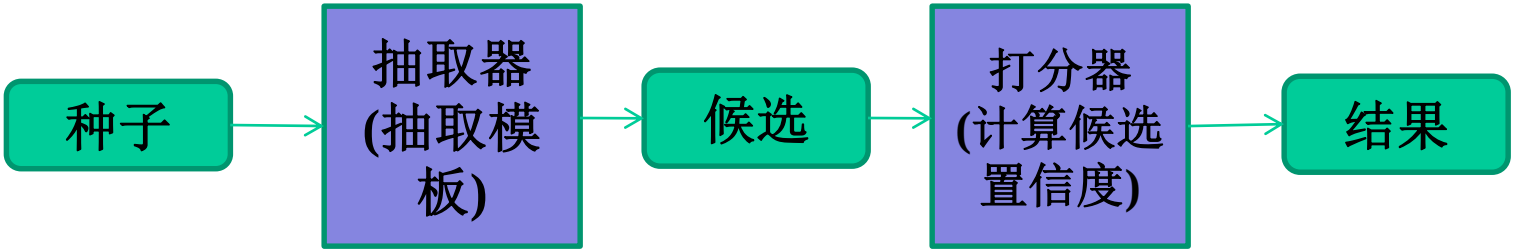
Google sets labs

Output

| Predicted Items |
| --- |
| 中国 |
| 俄罗斯 |
| 美国 |
| 日本 |
| 德国 |
| 英国 |
| 法国 |
| 韩国 |
| 印度 |
| 香港 |
| 意大利 |
| 巴西 |
| 加拿大 |
| 新加坡 |

# 开放类别实体抽取：主要方法

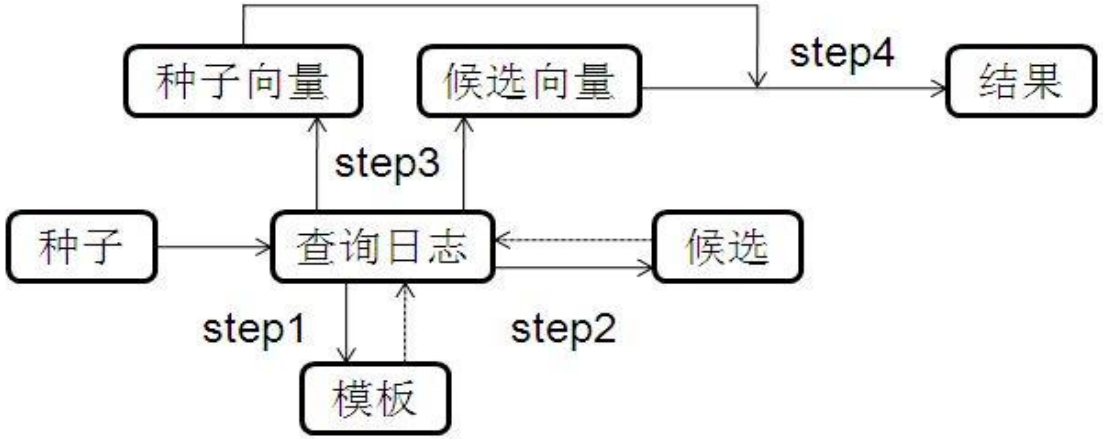
- 基本思路
  - 种子词与目标词在网页中具有相同或者类似的上下文
    - 网页结构
    - 上下文
  - Step1：种子词→模板
  - Step2：模板→更多同类实体
- 处理实例扩展问题的主流框架




- 利用不同数据源（例如查询日志、网页文档、知识库文档等）的不同特点，设计方法

# 开放类别实体抽取: Query Log (Pasca CIKM 2007)

- 通过分析种子实例在查询日志中的上下文学得模板，再利用模板找到同类别的实例
  - 联想 笔记本 如何
  - 苹果 笔记本 如何
  - 戴尔 笔记本 如何
- 构造候选与种子上下文向量，计算相似度



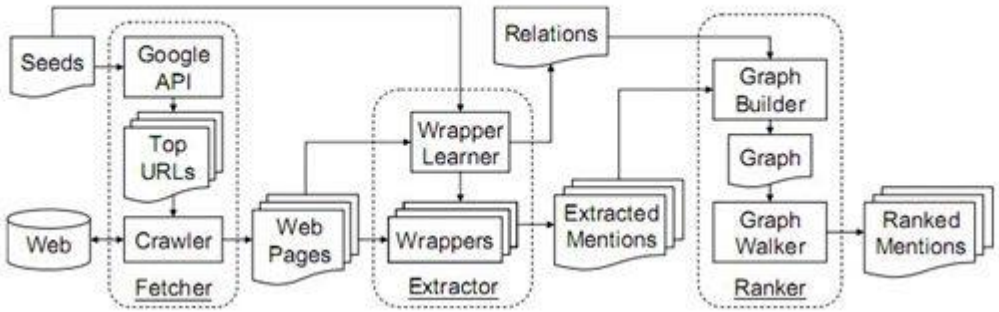
# 开放类别实体抽取: Web Page (Wang ICDM 2007) (1/2)

- Motivation
  - 处理列表型网页
  - 在列表中，种子与目标实体具有相同的网页结构

```
<li>·<a target="_self" href='#brand_A_2'>奥迪</a></li>
<li>·<a target="_self" href='#brand_B_3'>宝马</a></li>
<li>·<a target="_self" href='#brand_B_4'>保时捷</a></li>
<li>·<a target="_self" href='#brand_B_5'>奔驰</a></li>
<li>·<a target="_self" href='#brand_B_7'>本田</a></li>
<li>·<a target="_self" href='#brand_B_8'>比亚迪</a></li>
<li>·<a target="_self" href='#brand_B_9'>标致</a></li>
<li>·<a target="_self" href='#brand_B_10'>别克</a></li>
<li>·<a target="_self" href='#brand_C_15'>长城</a></li>
<li>·<a target="_self" href='#brand_D_18'>大众</a></li>
<li>·<a target="_self" href='#brand_F_24'>丰田</a></li>
```

# 开放类别实体抽取的主要方法: Web Page (Wang ICDM 2007) (2/2)

- 系统框架




一个模板抽取出的正确实例越多，则该模板的质量越高（反之亦然）

- 爬取模块（Fetcher）
  把种子送到搜索引擎，把返回的前100个网页抓取下来作为语料

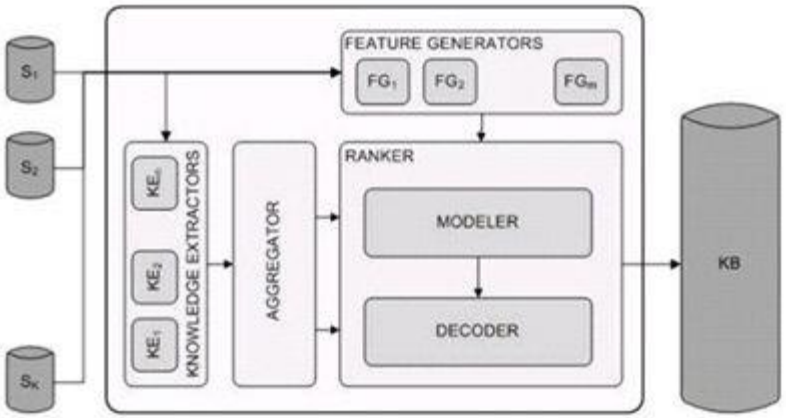
一个网页包含的高质量模板越多，则该网页质量越高（反之亦然）

- 抽取模块（Extractor）
  针对单个网页学习模板，再使用模板抽取候选实例

- 排序模块（Ranker）
  利用种子、网页、模板、候选构造一个图，综合考虑网页和模板的质量，使用Random Walk算法为候选打分并排序

# 开放类别实体抽取的主要方法: 融合多个数据源 (Pennacchiotti EMNLP 2009)

- 针对不同数据源，选取不同特征分别进行实例扩展，对结果进行融合
  - 6亿个网页
  - 1年的查询日志
  - Wikipedia
- 针对不同数据源选取不同的模板和特征
- 使用不同特征计算候选的置信度
  - 结果融合



# 评价指标与技术水平

- 针对实例扩展问题，目前缺少公认的评测，研究者在自己构建的数据上进行测试
- 评价方法：以平均准确度MAP为主；召回无法评价，覆盖度替代
  - 因为系统输出是一个ranked list，单纯考察准确率无法体现出rank的作用
  - 采用TREC中常用的MAP（平均正确率均值）

$$\mathrm{AvgPrec}(L) = \frac{\sum_{r=1}^{|L|} \mathrm{Prec}(r) \times \mathrm{NewEntity}(r)}{\#\ \mathrm{True\ Entities}}$$

  - Prec: 列表中到目前为止的准确率；NewEntity(r)当前实体是否正确的0-1函数
  - 每个类别做N次实验，每次都随机选取种子，对其求平均

# 评价指标与技术水平

取前100个网页作为语料

- Wang 2007在12个自制数据集的结果，

Table 5. Experimental results

| English | G.Sets | Max. 100 Results E1+EF | Max. 100 Results E2+EF | Max. 100 Results E2+GW | Max. 200 E2+GW | Max. 300 E2+GW |
|---|---|---|---|---|---|---|
| classic-disney | 37.62% | 79.36% | 74.45% | 84.42% | 88.20% | 89.39% |
| cmu-buildings | 0.00% | 87.85% | 87.75% | 87.83% | 87.83% | 87.83% |
| common-diseases | 1.12% | 17.94% | 52.84% | 57.46% | 75.79% | 76.87% |
| constellations | 10.45% | 89.61% | 99.97% | 100.00% | 100.00% | 100.00% |
| countries | 14.24% | 95.95% | 97.86% | 98.17% | 98.67% | 98.53% |
| mlb-teams | 70.06% | 98.61% | 99.50% | 99.80% | 99.84% | 99.81% |
| nba-teams | 90.73% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| nfl-teams | 94.26% | 99.22% | 99.98% | 100.00% | 100.00% | 100.00% |
| periodic-comets | 0.22% | 69.24% | 79.04% | 84.78% | 84.77% | 84.77% |
| popular-car-makers | 73.61% | 79.18% | 88.23% | 95.16% | 96.23% | 96.95% |
| us-presidents | 56.77% | 91.64% | 97.07% | 99.99% | 100.00% | 100.00% |
| us-states | 76.00% | 99.96% | 93.55% | 100.00% | 100.00% | 100.00% |
| **Average** | **43.76%** | **84.05%** | **89.19%** | **92.30%** | **94.28%** | **94.51%** |

# 小结

- 针对不同数据源的特点设计方法，其针对性、灵活性很强
- 方法一般分为模板抽取和实例候选置信度计算两个模块，两部分迭代进行，相互依赖
- 以无监督的方法为主
- 缺少公认的数据集或相关评测

# 总结

- 自然语言处理技术的历史
- 信息抽取技术概述：四个任务
- 中文分词和命名实体识别
  - 任务介绍
  - 难点
  - 典型方法
  - 评测指标
  - 技术水平

# 总结

- 命名实体识别
  - 任务介绍
  - 难点
  - 典型方法
  - 技术水平
- 开放域实体识别
  - 任务介绍
  - 难点
  - 典型方法
  - 评测指标
  - 技术水平

谢谢

jzhao@nlpr.ia.ac.cn

# 实体消歧

赵 军（jzhao@nlpr.ia.ac.cn）

中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

# 实体消歧定义

- 命名实体的歧义指的是一个实体指称项可对应到多个真实世界实体，例如，给定如下的四个实体指称项“*Michael Jordan*”

  *MJ1: Michael Jordan is a researcher in machine learning.*

  *MJ2: Learning in Graphical Models: Michael Jordan*

  *MJ3: M. Jordan wins NBA MVP.*

  *MJ4 : Michael Jordan plays basketball in Chicago Bulls.*

- 确定一个实体指称项所指向的真实世界实体，这就是命名实体消歧

# 网络媒体中的实体歧义

Chen Guangcheng

| Morph | Target | Motivation |
|---|---|---|
| Blind Man (瞎子) | Chen Guangcheng(陈光诚) | Sensitive |
| Kimchi Country (泡菜国) | Korea (韩国) | Vivid |
| Rice Country (米国) | United States (美国) | Pronunciation |
| Miracle Brother (奇迹哥) | Wang Yongping (王勇平) | Irony |

# 实体消歧分类

- 基于聚类的实体消歧
  - 把所有实体指称项按其指向的目标实体进行聚类
  - 每一个实体指称项对应到一个单独的类别

*MJ1: Michael Jordan is a researcher in machine learning.*

*MJ2: Research in Graphical Models: Michael Jordan*

*MJ3: M. Jordan wins NBA MVP.*

*MJ4 : Michael Jordan plays basketball in Chicago Bulls*

- 基于实体链接的实体消歧
  - 将实体指称项与目标实体列表中的对应实体进行链接实现消歧

*MJ4 : Michael Jordan plays basketball in Chicago Bulls*

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

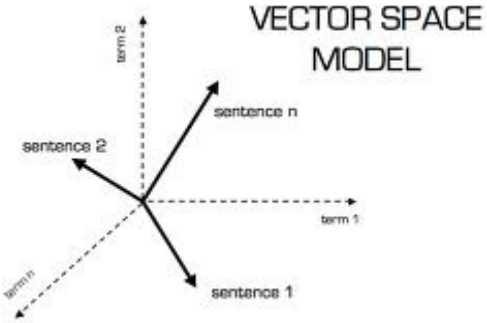
# 基于聚类的实体消歧

- 基本思路
  - 同一指称项具有近似的上下文
  - 利用聚类算法进行消歧
  - 核心问题：选取何种特征对于指称项进行表示
    - 词袋模型(Bagga et al., COLING, 1998)
    - 语义特征(Pederson et al., CLITP, 2005)
    - 社会化网络(Bekkerman et al., WWW, 2005)
    - 维基百科的知识(Han and Zhao, CIKM, 2009)
    - 多源异构语义知识融合(Han and Zhao, ACL, 2010)

# 基于聚类的实体消歧:词袋模型
## (Bagga et al. COLING 1998)

- 利用待消歧实体周边的词来构造向量

- 利用向量空间模型来计算两个实体指称项的相似度，进行聚类




MJ1: Michael Jordan is a researcher in machine learning.

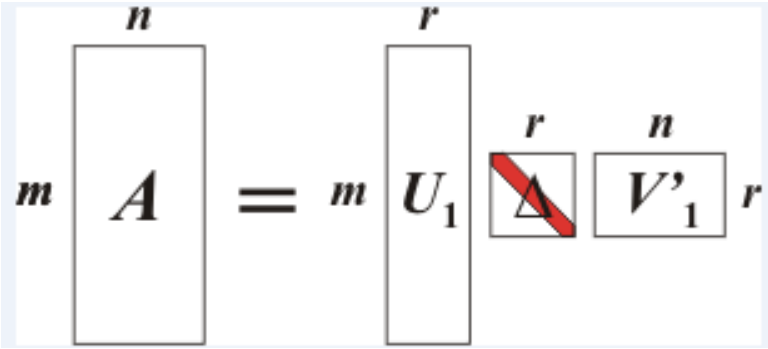
MJ1 → researcher → machine → learning

MJ1: Michael Jordan plays basketball in Chicago Bulls

MJ4 → plays → basketball → Chicago → Bulls

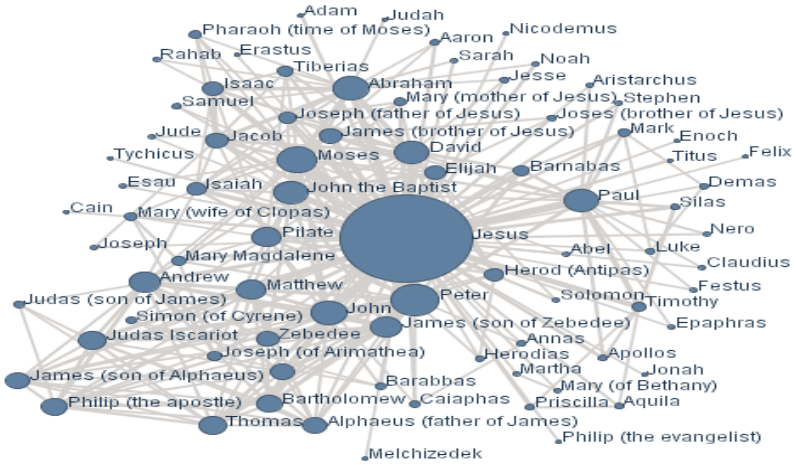
# 基于聚类的实体消歧: 语义特征
(Pederson et al. CLITP 2005)

- 词袋模型，没有考虑词的语义信息
- 利用SVD分解挖掘词的语义信息
- 利用词袋和浅层语义特征，共同表示指称项，利用余弦相似度来计算两个指称项的相似度



# 基于聚类的实体消歧：社会化网络
## (Bekkerman et al. WWW 2005)

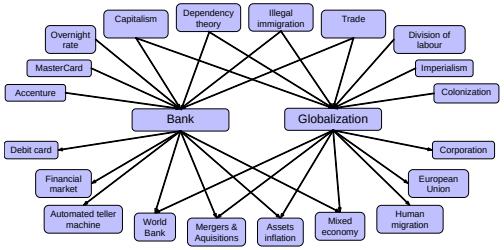
- 不同的人具有不同的社会关系
  - MJ（BasketBall）: Pippen, Buckley, Ewing, Kobe…
  - MJ（Machine Learning）: Liang, Mackey, Wauthier…
- MJ, Pippen, Buckley, Ewing, Kobe等的社会化关联信息所表现出来的网页链接特征，对网页进行聚类，从而实现网页内的人名聚类消歧。



# 基于聚类的实体消歧: Wikipedia
## (Han CIKM 2009) (1/3)

- Wikipedia中相关实体具有链接关系

- 这种链接关系反映条目之间的语义相关度
  - D. Milne and Ian H. Witten 2008: The higher semantic related Wikipedia concepts will share more semantic related concepts.




The concepts has links with a and b

$$sr(a,b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))}$$

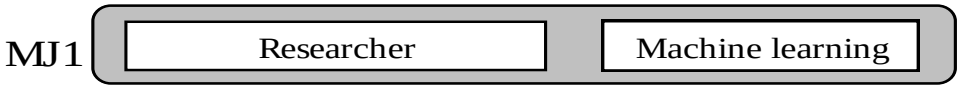
The Whole Wikipedia concepts

The concepts has links with a and b

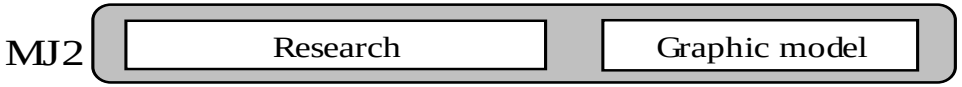
# 基于聚类的实体消歧: Wikipedia
## (Han CIKM 2009) (2/3)

- 用实体上下文的维基条目对于实体进行向量表示

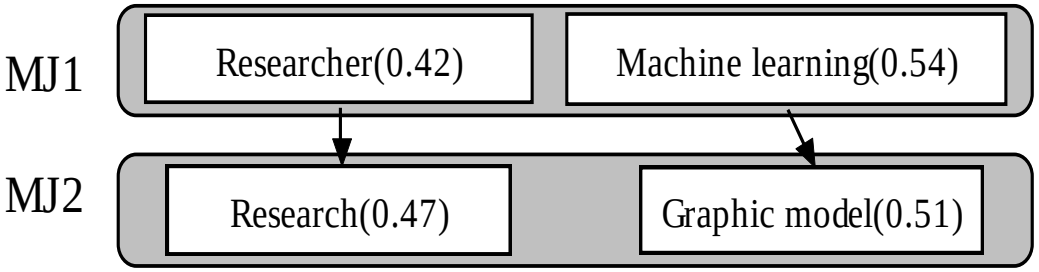
*MJ1: Michael Jordan is a Researcher in machine learning.*




*MJ2: Research in Graphical Models: Michael Jordan*




- 利用维基条目之间的相关度计算指称项之间的相似度（解决数据稀疏问题）



# 实验比较
## (Han CIKM 2009) (3/3)

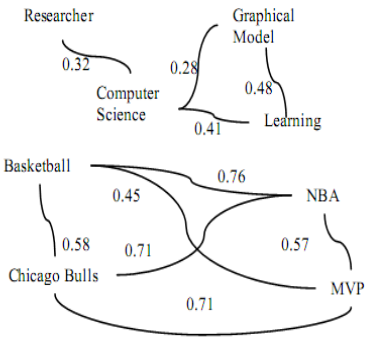
- 使用WePS数据集测试
- 使用结构化关联语义核的实体相似度能够提升10.7%的消歧性能

| Method | WePS1_training | | |
|---|---|---|---|
| | Pur | Inv_Pur | F |
| *BOW* | 0.71 | 0.88 | **0.78** |
| *SocialNetwork* | 0.66 | 0.98 | **0.76** |
| *WikipediaConcept* | 0.80 | 0.88 | **0.82** |
| ***WS-SameWeight*** | 0.84 | 0.89 | **0.85** |
| ***WS*** | 0.88 | 0.89 | **0.87** |

# 基于聚类的实体消歧: 多源异构知识
## (Han ACL 2010) (1/3)

- 仅仅考虑Wikipedia一种知识源，覆盖度有限
- 多源异构知识的挖掘与集成
  - 知识源中存在大量的多源异构知识
  - 挖掘和集成多源异构知识可以提高实体消歧的性能
  - Wikipedia
    - 用于捕捉概念之间的语义关联
  - WordNet
    - 用于捕捉词语之间的语言学关联
  - Web网页库
    - 用于捕捉命名实体之间的社会化关联

# 基于聚类的实体消歧：多源异构知识
## (Han ACL 2010) (2/3)




- 多源异构知识的表示框架：语义图
  - 等同概念识别
  - 概念连接
    - 同时捕捉显式语义知识和结构化语义知识
      - **语义图的边（显式语义知识）**——建模了所有从知识源中直接抽取出的概念之间的显式语义关联
      - **语义图的结构（结构化语义知识）**——建模了概念之间的隐藏语义关联
- 语义图中语义知识的挖掘和融合算法
  - 计算原则：“如果一个概念的邻居概念与另一个概念存在语义关联，则这个概念也与另一个概念存在语义关联”
  - 语义关联在图中的传递性

$$S_{ij} = \boxed{\lambda \sum_{l \in N_i} \frac{A_{il}}{d_i} S_{lj}} + \boxed{\mu A_{ij}}$$

邻居节点传递　　显式语义关联

# 基于聚类的实体消歧:多源异构知识
## (Han ACL 2010) (3/3)

- 实验比较
  - 使用WePS数据集测试
  - 使用多源知识能够有效提高消歧的准确度

| | WePS2_test | | |
|---|---|---|---|
| | **Pur** | **Inv_Pur** | **F** |
| *BOW* | 0.80 | 0.80 | **0.77** |
| *SocialNetwork* | 0.62 | 0.93 | **0.70** |
| *SSR-NoKnowledge* | 0.84 | 0.80 | **0.80** |
| *SSR-NoStructure* | 0.84 | 0.83 | **0.81** |
| ***SSR-NE*** | 0.78 | 0.88 | **0.80** |
| ***SSR-WordNet*** | 0.85 | 0.82 | **0.83** |
| ***SSR-Wikipedia*** | 0.84 | 0.81 | **0.82** |
| ***SSR*** | 0.90 | 0.86 | **0.88** |

# 基于聚类的实体消歧：评测(1/2)

- WePS： Web People Search Evaluation
  - WePS1是SEMEVAL2007的子任务
  - WePS2是WWW的一个workshop
  - 任务：Web环境中的人名消歧，即给定一个包含某个歧义人名的网页集合，按照网页中人名指称项所指向的人物概念来对网页进行聚类，以及抽取一个网页中关于某个人的特定属性来辅助进行人名消歧
  - 评测方法

$$\text{Purity} = \sum_i \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j)$$

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)$$

$$\text{F} = \frac{1}{\alpha \frac{1}{\text{Purity}} + (1-\alpha) \frac{1}{\text{Inverse Purity}}}$$

# 基于聚类的实体消歧：评测(2/2)

| | | Macro-averaged Scores | | | |
|---|---|---|---|---|---|
| | | F-measures | | | |
| rank | team-id | $\alpha = ,5$ | $\alpha = ,2$ | Pur | Inv_Pur |
| 1 | CU_COMSEM | ,78 | ,83 | ,72 | ,88 |
| 2 | IRST-BP | ,75 | ,77 | ,75 | ,80 |
| 3 | PSNUS | ,75 | ,78 | ,73 | ,82 |
| 4 | UVA | ,67 | ,62 | ,81 | ,60 |
| 5 | SHEF | ,66 | ,73 | ,60 | ,82 |
| 6 | FICO | ,64 | ,76 | ,53 | ,90 |
| 7 | UNN | ,62 | ,67 | ,60 | ,73 |
| *8* | *ONE-IN-ONE* | *,61* | *,52* | *1,00* | *,47* |
| 9 | AUG | ,60 | ,73 | ,50 | ,88 |
| 10 | SWAT-IV | ,58 | ,64 | ,55 | ,71 |
| 11 | UA-ZSA | ,58 | ,60 | ,58 | ,64 |
| 12 | TITPI | ,57 | ,71 | ,45 | ,89 |
| 13 | JHU1-13 | ,53 | ,65 | ,45 | ,82 |
| 14 | DFKI2 | ,50 | ,63 | ,39 | ,83 |
| 15 | WIT | ,49 | ,66 | ,36 | ,93 |
| 16 | UC3M_13 | ,48 | ,66 | ,35 | ,95 |
| 17 | UBC-AS | ,40 | ,55 | ,30 | ,91 |
| *18* | *ALL-IN-ONE* | *,40* | *,58* | *,29* | *1,00* |

WePS 1

| | | Macro-averaged Scores | | | |
|---|---|---|---|---|---|
| | | F-measures | | | |
| rank | run | $\alpha = ,5$ | $\alpha = ,2$ | Pur | Inv_Pur |
| | *BEST-HAC-TOKENS* | ,90 | ,89 | ,93 | ,88 |
| | *BEST-HAC-BIGRAMS* | ,90 | ,87 | ,94 | ,86 |
| 1 | PolyUHK | ,88 | ,87 | ,91 | ,86 |
| 2 | UVA_1 | ,87 | ,87 | ,89 | ,87 |
| 3 | ITC-UT_1 | ,87 | ,83 | ,95 | ,81 |
| | *CHEAT_SYS* | ,87 | ,94 | ,78 | 1,00 |
| 4 | UMD_4 | ,81 | ,76 | ,95 | ,72 |
| 5 | XMEDIA_3 | ,80 | ,76 | ,91 | ,73 |
| 6 | UCI_2 | ,80 | ,84 | ,75 | ,89 |
| 7 | LANZHOU_1 | ,80 | ,78 | ,85 | ,77 |
| 8 | FICO_3 | ,80 | ,76 | ,90 | ,73 |
| | *HAC-BIGRAMS* | ,78 | ,64 | ,96 | ,67 |
| 9 | UGUELPH_1 | ,74 | ,84 | ,64 | ,95 |
| 10 | CASIANED_4 | ,73 | ,77 | ,72 | ,83 |
| | *HAC-TOKENS* | ,71 | ,64 | ,96 | ,60 |
| 11 | AUG_4 | ,69 | ,68 | ,79 | ,68 |
| 12 | UPM-SINT_4 | ,67 | ,70 | ,69 | ,74 |
| | *ALL_IN_ONE* | ,67 | ,79 | ,56 | 1,00 |
| 13 | UNN_2 | ,64 | ,59 | ,80 | ,57 |
| 14 | ECNU_1 | ,53 | ,56 | ,60 | ,63 |
| 15 | PRIYAVEN | ,53 | ,49 | ,71 | ,48 |
| 16 | UNED_3 | ,51 | ,48 | ,71 | ,48 |
| 17 | BUAP_1 | ,37 | ,30 | ,89 | ,27 |
| | *ONE_IN_ONE* | ,34 | ,27 | 1,00 | ,24 |

WePS 2

# 小结

- 主要研究集中在实体指称项的语义表示
- 已有工作大多是通过扩展特征，增加更多的知识来提高消歧精度
- 挑战
  - 消歧目标难以确定
  - 缺乏实体的显式表示

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

# 实体链接的任务

- 任务
  - 给定实体指称项和它所在的文本，将其链接到给定知识库中的相应实体上

*Michael Jordan*
*is an NBA player.*

*Michael Jordan*
*is Computer Science professor in UC Berkeley.*

| 实体 ID | 实体 |
|---|---|
| … | … |
| E00010 | Michael Jordan(NBA player) |
| E00011 | Michael Jordan(Berkeley Professor) |
| … | … |
| E10000 | Michael Jordan(football player) |
| E100001 | Michael Jackson |
| … | … |

目标实体列表

# 实体链接主要步骤

- 主要步骤
  - 候选实体的发现
    - 给定实体指称项，链接系统根据知识、规则等信息找到实体指称项的候选实体
  - 候选实体的链接
    - 系统根据指称项和候选实体之间的相似度等特征，选择实体指称项的目标实体

实体指称项文本：Michael Jordan is a former NBA player, active businessman and majority owner of the Charlotte Bobcats.

候选实体：
Michael Jordan (basketball player)
Michael Jordan (mycologist )
Michael Jordan (footballer)
Michael B. Jordan
Michael H. Jordan
Michael-Hakim Jordan
Michael Jordan (Irish polotician)
…

# 候选实体发现

- 如何根据实体指称项找出候选实体
  - 利用Wikipedia的信息
  - 利用上下文信息

| 实体指称项 | 候选实体 |
| --- | --- |
| Michael Jordan | Michael Jordan (basketball)<br>Michael Jordan (mycologist)<br>Michael Jordan (football)<br>Michael B. Jordan (American actor)<br>… |
| AI | Artificial intelligence<br>Ai (singer)<br>… |
| … | … |

# 利用Wikipedia信息获取候选实体

- 利用Wikipedia中锚文本的超级链接关系
  - Michal Jordan is a former NBA player

- 利用Wikipedia中的消歧页面

- 利用Wikipedia中的重定向页面

# 利用上下文获取缩略语候选实体
## (Zhang IJCAI 2011)

- 问题
  - 缩略语在实体指称项中十分常见，据统计，在KBP2009的测试数据，在3904个实体指称项中有827个为缩略语
- 动机
  - 缩略语指称项具有很强的歧义性，但它的全称往往是没有歧义的
  - ABC和American Broadcasting Company, AI和Artificial Intelligence等
  - 在实体指称项文本中，缩略语的全称出现过
- 解决方法
  - 利用人工规则抽取实体候选

# 候选实体链接

- 如何进行实体链接
  - 基本方法：计算实体指称项和候选实体的相似度，选择相似度最大的候选实体
  - 单一实体链接
    - BOW模型 (Honnibal TAC 2009, Bikel TAC 2009)
    - 加入候选实体的类别特征（Bunescu et al., EACL 2006）
    - 加入候选实体的流行度等特征（Han et al., ACL 2011）
  - 协同实体链接
    - 利用实体之间类别的共现特征（Cucerzan, EMNLP 2007）
    - 利用实体之间链接关系（Kulkarni et al., KDD 2009）
    - 利用同一篇文档中不同实体之间存在的语义关联特征（Han et al., SIGIR 2011）

# 实体链接的基本方法
## (Honnibal TAC 2009, Bikel TAC 2009)

- 基于词袋子模型计算相似度
  - 将实体指称项上下文文本与候选实体上下文文本表示成词袋子向量形式，通过计算向量间的夹角确定指称项与候选实体相似度，系统选择相似度最大的候选实体进行链接

$$score(q, e_k) = cos(q.T, e_k.T) = \frac{q.T}{\|q.T\|} \frac{e_k.T}{\|e_k.T\|}$$

$$\hat{e} = arg \max_{e_k} score(q, e_k)$$

# 类别特征
## (Bunescu EACL 2006)

- 动机
  - 候选实体的文本内容可能太短，会导致相似度计算的不准确
  - 加入指称项文本中的词与候选实体类别的共现特征
    - 例：除了计算待消歧文本和实体Wikipedia文本John Williams (composer)的相似度外，还考虑当前文本中的词语与**Music, Art** 等类别的共现信息
- 方法
  - 训练SVM分类器对候选实体进行选择
  - 训练数据由Wikipedia中的超级链接获得
  - 所采用的特征
    - 文本相似度
    - 指称项文本中词与候选实体类别的共现信息

John Williams (composer): Category={**Music, Art**…}
John Williams (wrestler): Category={Sport,…}
John Williams (VC): Category={Bank,…}

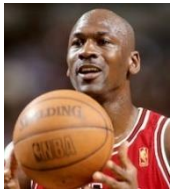
Williams has also composed numerous classical concerti, and he served as the principal conductor of the Boston Pops Orchestra from 1980 to 1993

类别：music

# 实体流行度等特征
## （Han ACL 2011）

- 动机
  - 传统的方法仅仅是计算实体指称项与候选实体的相似度，忽略了候选实体的背景知识与先验信息，如实体本身的流行度、实体与指称项的关系等
- 方法
  - 考虑实体的背景知识，将实体的背景知识融入到实体链接的过程，实体的背景知识和先验信息主要有
    - 实体流行度：实体e在知识库中的概率P(e)
    - 名称的知识：指称项s指向实体e的概率P(s|e)
    - 上下文知识：实体e出现在特定上下文环境c的概率P(c|e)

# 协同实体链接



During his standout career at **Bulls**,
**Jordan** also acts in the movie Space Jam .

- 实体指称项与目标实体的语义相似度
- 目标实体之间的语义相似度

# 协同学习策略

- 动机
  - 同一篇文档中实体之间具有语义相关性
  - 利用Pairwise优化策略

$$\frac{1}{\binom{|S_0|}{2}} \sum_{s \neq s' \in S_0} r(y_s, y_{s'}) + \frac{1}{|S_0|} \sum_{s \in S_0} w^\top f_s(y_s).$$

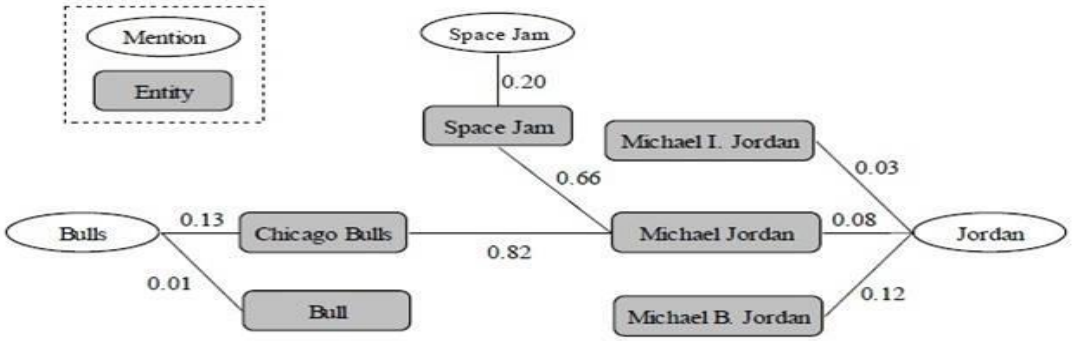
任意两个目标实体之间的语义相关度

实体指称项到目标实体的语义相似度

- 目标实体的语义相关度计算方法：
  - 利用实体类别重合度计算目标实体语义相似度（Cucerzan, EMNLP 2007）
  - 利用实体之间链接关系计算目标实体语义相似度（Kulkarni , KDD 2009 ）

# 基于图的协同链接
## （Han SIGIR 2011）

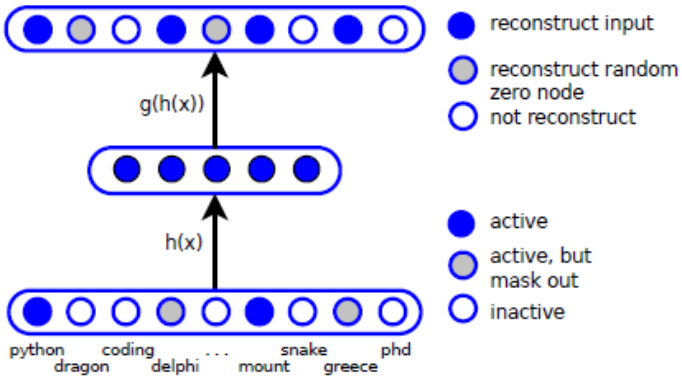
- 动机
  - Pairwise策略只考虑两两实体关系，结果不是全局最优的
  - 采用图方法，全局考虑目标实体之间的语义关联
- 方法：Referent Graph，两种关系构成
  - 指称项与实体之间的关系：该指称项文本与实体文本的相似度，由传统的VSM模型得到
  - 实体之间的语义关系：利用目标实体之间的链接关系计算实体之间的语义相关度

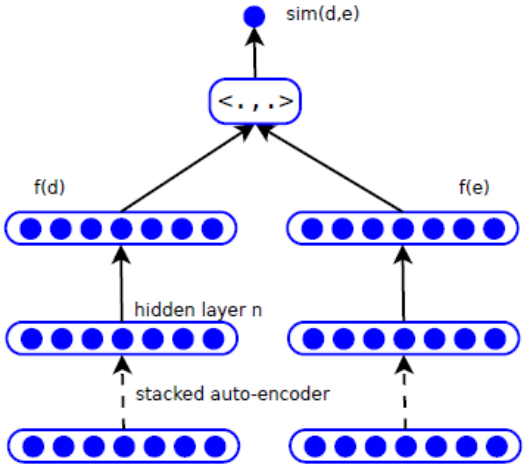


# 基于深度学习的方法
## （He ACL 2013）

- 动机
  - 传统的方法中，计算待消歧实体上下文和目标实体语义相似度的方法（点乘，余弦相似度，KL距离等）可扩展性差，没有考虑各个概念间的内在联系
  - 在协同过滤的方法中，计算待消歧实体上下文和目标实体语义相似度也是基础工作。
- 方法

提出利用深度学习的方法自动联合学习实体和文档的表示，进而完成实体链接任务（数据集：TAC-KBP 2010 和 AIDA）

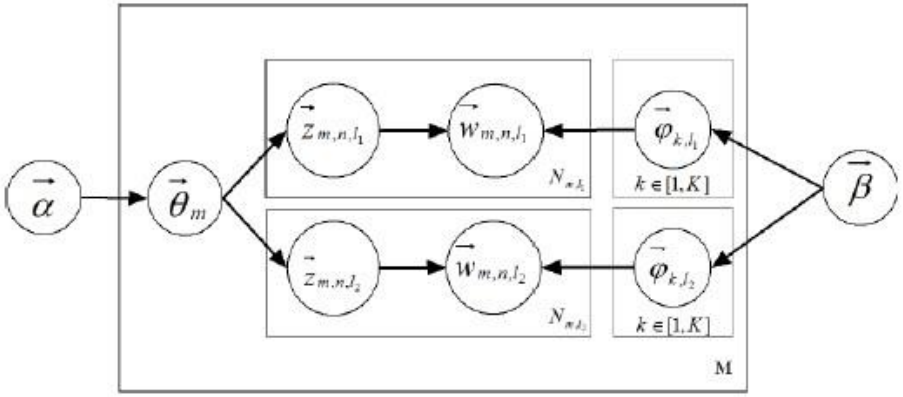


# 跨语言实体链接
## （Zhang IJCAI 2013）

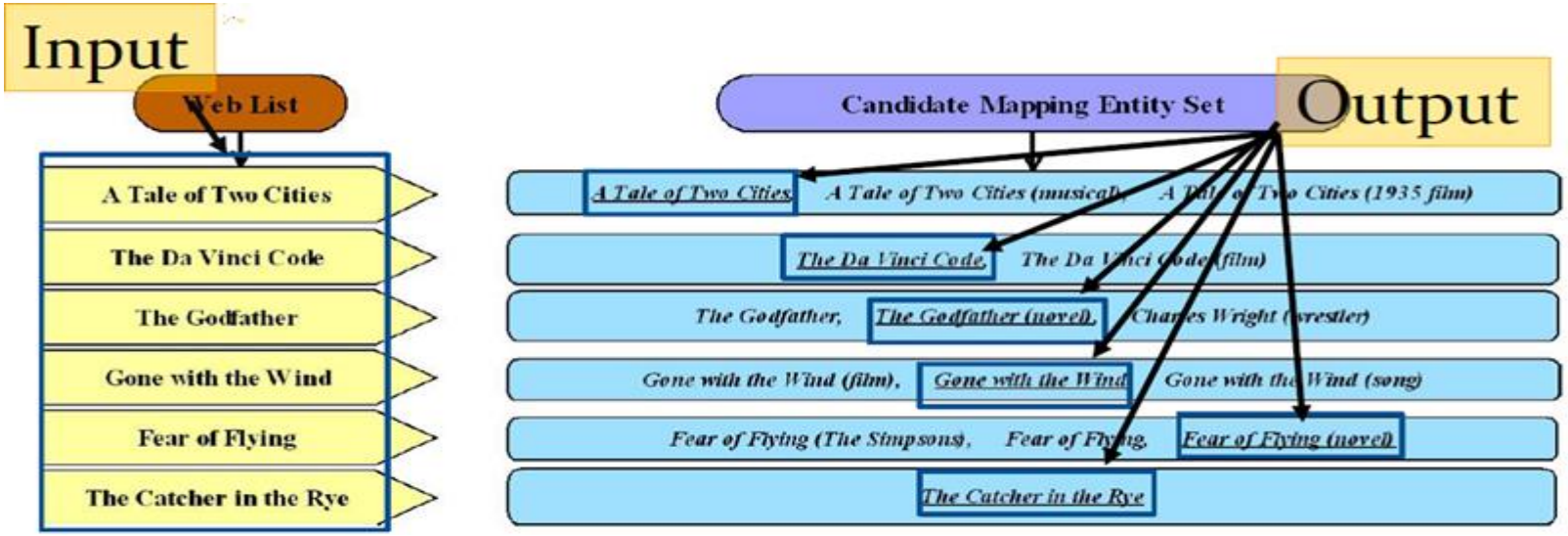
- 动机
  - 给定一种语言的实体指称项和其所在的上下文，将其链接到另外一种语言的知识库中
  - 传统方法要先翻译成目标语言，可能产生错误传递，需要大量的句子级平行的双语训练语料

- 方法：利用双语隐含主题模型将实体指称项与候选实体映射到同一个主题空间中（数据集：TAC-KBP 2011）
  - 每一个隐含主题有两种不同的分布，分别对应两种不同语言
  - 处于同一个主题分布下的两种不同语言的词的分布具有一些共性



# 结构化数据中的实体链接
## ( Shen SIGKDD 2012 )

- 动机
  - 没有上下文
  - 任务与传统的实体链接不同

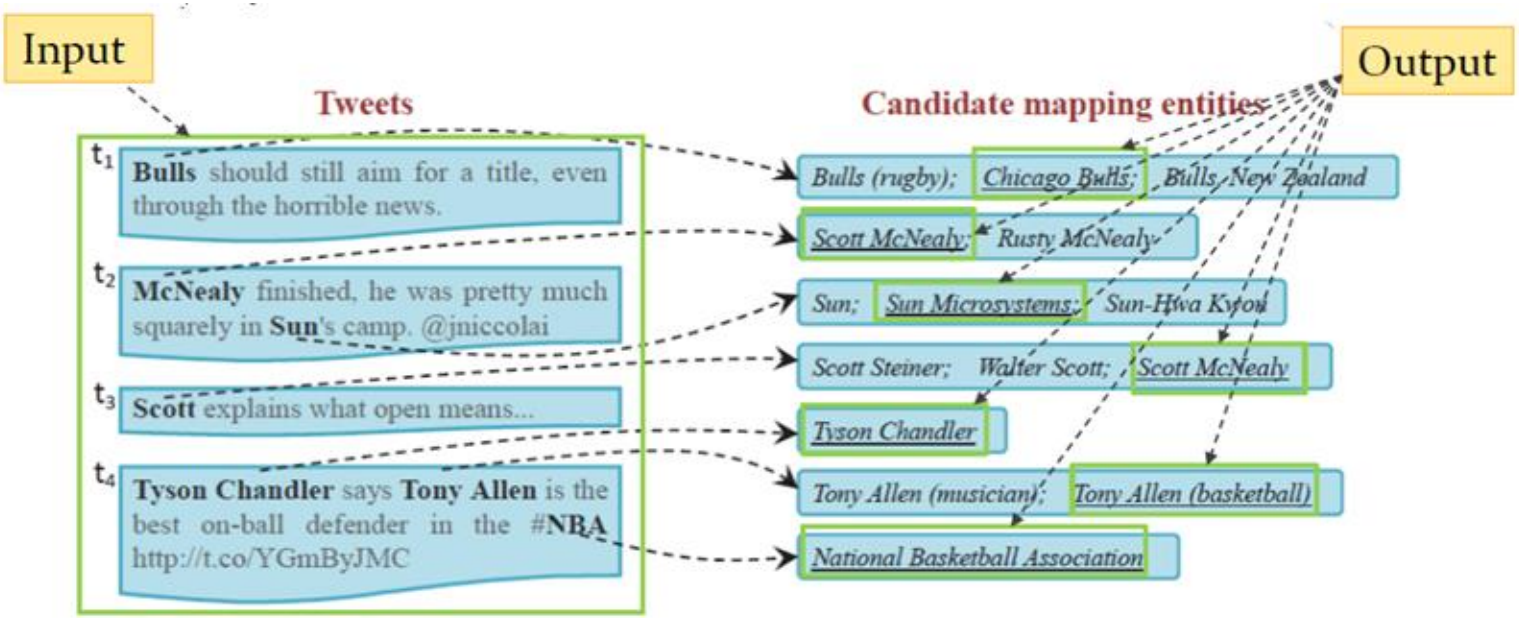



- 方法
  - 主要利用实体的流行度和实体共现类型去消歧

取自王建勇老师报告（第三届中文知识图谱研讨会）

# 社交数据中的实体链接
## （Shen SIGKDD 2013）

- 动机
  - 社交媒体（Twitter）是一种重要的信息来源
  - 社交媒体的上下文较短，语言表述不规范




- 方法
  - 利用tweet的用户信息和tweet的交互信息

取自王建勇老师报告（第三届中文知识图谱研讨会）

# 实体链接评测 (1/2)

- TAC-KBP (2009-Now)：Entity Linking
  - 任务：将文本中的目标实体链接到Wikipedia中的真实概念，达到消歧的目的
  - 评测方法：

以指称项为单位计算的准确率

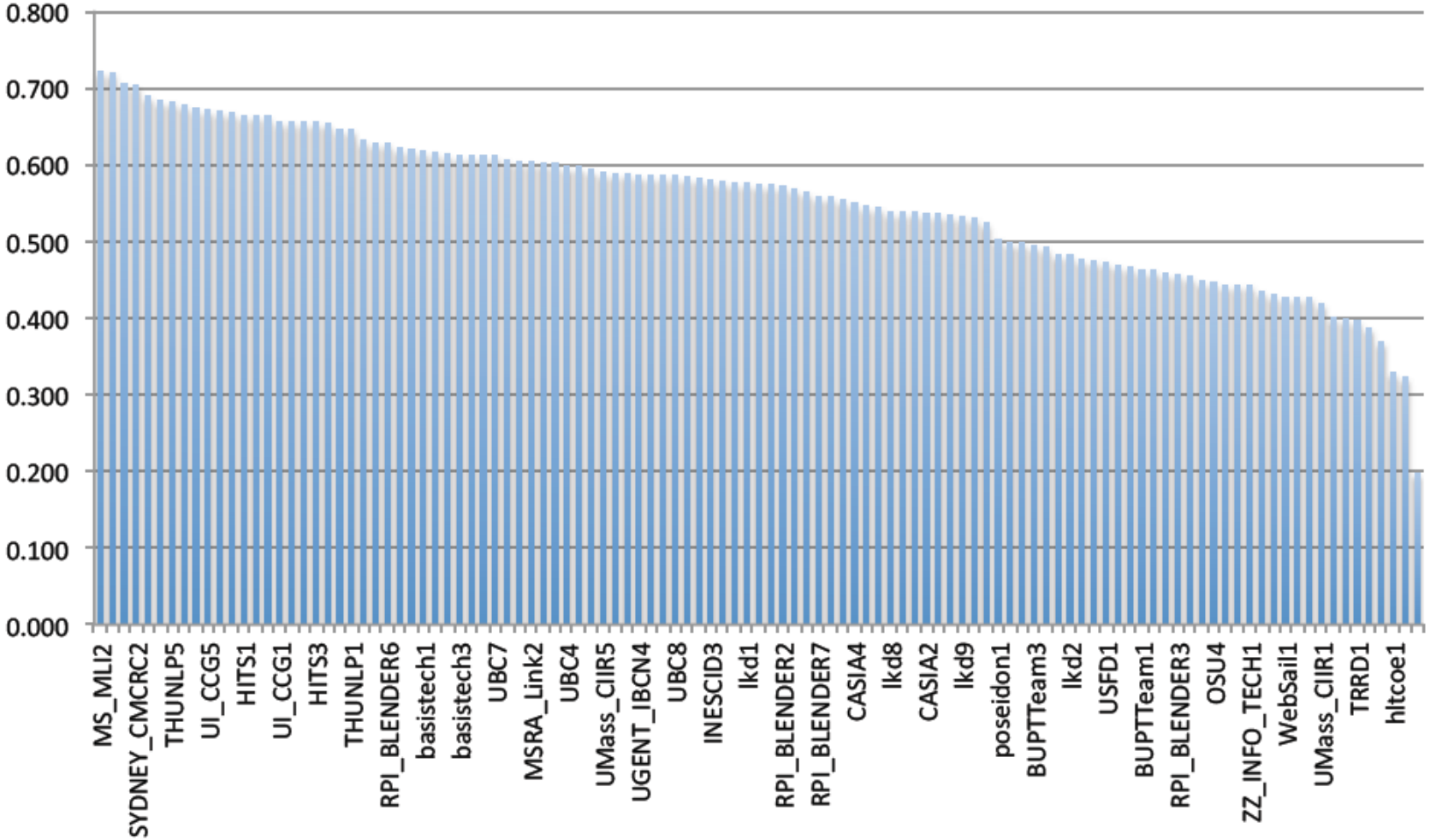
$$Accuracy_{micro} = \frac{NumCorrect}{NumQueries}$$

$$Accuracy_{macro} = \frac{\sum_{i}^{NumEntities} \frac{NumCorrect(E_i)}{NumQueries(E_i)}}{NumEntities}$$

以实体为单位计算的准确率

# 实体链接评测 (2/2)

2013评测结果(Micro Accuracy)



# 小结

- 目前实体链接方法主要是如何更有效挖掘实体指称项信息，如何更准确地计算实体指称项和实体概念之间的相似度
- 由单一实体链接向协同实体链接发展
- 难点：未登录实体的处理

# 关系抽取

# 概述

- 关系抽取任务定义
- 传统关系抽取
- 开放域关系抽取

# 实体间关系抽取的定义

- Alexander Schutz等人认为关系抽取是自动识别由一对概念和联系这对概念的关系构成的相关三元组
  - Example1: 比尔盖茨是微软的CEO
    - CEO(比尔盖茨, 微软)
  - Example2: CMU坐落于匹兹堡
    - Located-in(CMU, 匹兹堡)
  - 高阶关系：Michael Jordan获得1997/98赛季的MVP
    - Award(Michael Jordan, 1997/98赛季, MVP)

# 网络文本信息结构

- 结构化数据（Infobox）
  - 置信度高
  - 规模小
  - 缺乏个性化的属性信息
- 半结构化数据
  - 置信度较高
  - 规模较大
  - 个性化的信息
  - 形式多样
  - 含有噪声
- 纯文本
  - 置信度低
  - 复杂多样
  - 规模大

| | | | |
|---|---|---|---|
| 中文名： | 姚明 | 专业特点： | 20英尺外精确跳投 |
| 外文名： | Yao Ming | 主要奖项： | NBA全明星赛(7次) |
| 别名： | 小巨人 移动长城 | | ESPN全球最有潜力运动员奖(2000) |
| 国籍： | 中国 | | 劳伦斯世界最佳新秀奖(2003) |
| 民族： | 汉族 | | 中国篮球杰出贡献奖 |
| 出生地： | 上海 | 重要事件： | 专题影片《姚明年》发行 |
| 出生日期： | 1980年9月12日 | 祖籍： | 江苏苏州吴江震泽 |
| 身高： | 2.23米（7.32英尺） | 位置： | 中锋 |
| 体重： | 140.6kg | 鞋码： | 18码 |
| 运动项目： | 篮球 | 自传： | 《我的世界我的梦》 |
| 所属运动队： | NBA火箭队 | 生涯最高分： | 41分 |

1984：《一个和八个》摄影

1984：《黄土地》摄影

1986：《大阅兵》摄影

1987：《老井》主演

1987：《红高粱》导演

1989：《古今大战秦俑

1989：《代号美洲豹》

1990：《菊豆》导演

1991：《大红灯笼高高

**栖息环境**

沙狐栖息在广袤荒原及半沙中，行动快而敏捷。狐臭不显。

**饮食习惯**

食物包括各种鼠类、鼠兔、

**生长繁殖**

常在每年1-3月发情，妊娠期常每胎3～6仔。寿命约6年。

**个人简介**

姓名：姚明

**祖籍：**江苏苏州吴江震泽

**出生地：**上海市徐汇区

**出生医院：**上海交通大学附属第六人民医院

**小学：**上海市高安路第一小学

**初中：**上海市第二中学

**大学：**上海交通大学

**星座：**处女座

**曾效力球队：**CBA上海东方大鲨鱼、NBA休斯敦火箭

**前任主教练：**里克·阿德尔曼

**休斯敦火箭队球衣号码：**11

**国家队球衣号码：**13

**2008奥运会号码：**13

**手掌：**19厘米

**净身高：**2.26米

**臂展：**7英尺4.75英寸（2.25米）

**站立摸高：**9英尺7英寸（2.91米）

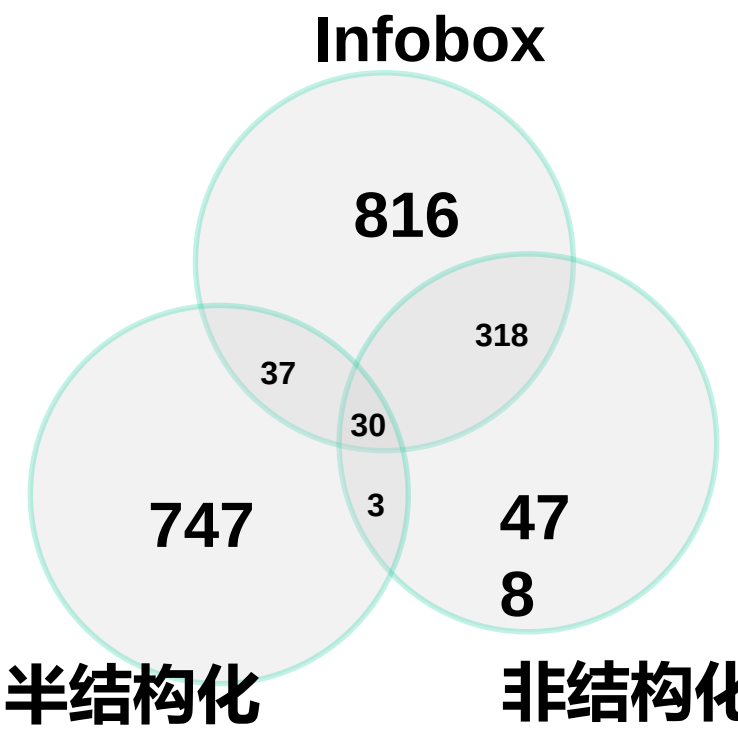
**原地弹跳：**19英寸（约48厘米）

张艺谋，陕西省西安人，中国电影导演，北京奥运会开幕式总导演。在电影学院学的是摄影专业。张艺谋是中国在国际影坛最具影响力的导演，曾多次荣获国际电影节大奖并成功执导北京奥运会开幕式。其代表作《红高粱》被认为是中国电影走向世界的新开始，《英雄》则是开启了中国电影的大片时代。

# 结构化 vs. 半结构化 vs. 非结构化

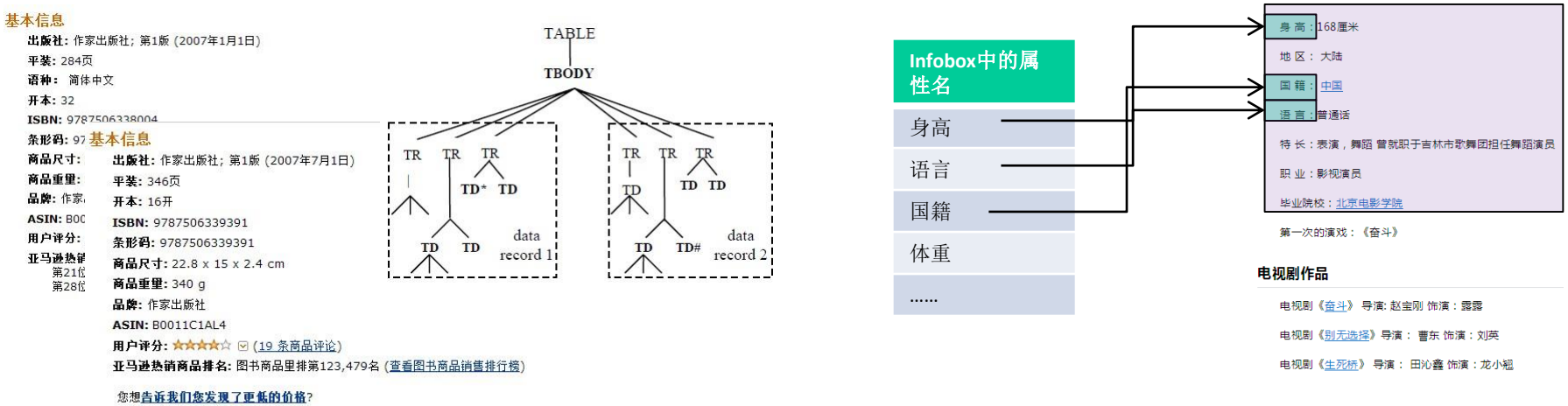
- 随机抽取100篇百科文档（共5类）
  - 对于其中三部分都包含的网页进行了统计

| | 人物 | 地理 | 电影 | 动物 | 图书 |
|---|---|---|---|---|---|
| InfoBox | 87 | 182 | 260 | 183 | 104 |
| 非结构化 | 79 | 147 | 109 | 107 | 36 |
| 半结构化 | 119 | 96 | 327 | 129 | 76 |
| OverLap Info vs. 非 | 62 | 101 | 87 | 72 | 26 |
| OverLap Info vs. 半 | 11 | 11 | 21 | 7 | 17 |
| OverLap 非 vs. 半 | 8 | 7 | 4 | 4 | 10 |
| 三方OverLap | 7 | 7 | 3 | 4 | 9 |



# 结构化与半结构化的关系抽取方法

- 结构化与半结构化文本信息（利用网页结构）
  - 信息块的识别
  - 模板的学习
  - 属性值的抽取




Bootstrapping

# 非结构化文本的关系抽取分类

- 传统关系抽取
  - 评测语料（MUC，ACE, …）
  - 专家制订类别，人工标注语料
- 开放域关系抽取
  - 类别自动获取
  - 语料自动生成
  - 语言证据清晰表达的关系

# 概述

- 关系抽取任务定义
- 传统关系抽取
- 开放域关系抽取

# 传统关系抽取

- 任务
  - 给定实体关系类别，给定语料，抽取目标关系对
- 评测语料（MUC, ACE, KBP, SemEval）
  - 专家标注语料，语料质量高
  - 抽取的目标类别已经定义好
  - 有公认的评价方式

# ACE抽取的目标关系列表

| | PER | ORG | GPE | LOC | FAC | WEA | VEH |
|---|---|---|---|---|---|---|---|
| PER | Per_Social.Bus Per_Social.Family, Per_Social.Lasting, Gen_Aff.Ideology, Gen_Aff.CRRE | Org_Aff.Employment, Org_Aff.Ownership, Org_Aff.Student/Alum, Org_Aff.Sports_Affiliation, Org_Aff.Investor/Shareholder, Org_Aff.Membership, Org_Aff.Founder, Gen_Aff.CRRE | Physical.Located, Physical.Near, Org_Aff.Employment, Org_Aff.Investor/Shareholder, Org_Aff.Founder, Gen_Aff.CRRE | Physical.Located, Physical.Near, Gen_Aff.CRRE | Physical.Located Physical.Near, Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| ORG | | Part_Whole.Subsidiary, Org_Aff.Investor/Shareholder, Org_Aff.Membership | Part_Whole.Subsidiary, Org_Aff.Investor/Shareholder, Gen_Aff.Loc/Origin | Gen_Aff.Loc/Origin | Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| GPE | | Org_Aff.Investor/Shareholder,Org_Aff.Membership, | Physical.Near, Part_Whole.Geographical Org_Aff.Investor/Shareholder | Physical.Near, Part_Whole.Geographical | Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| LOC | | | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | | |
| FAC | | | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | | |
| WEA | | | | | | Part_Whole.Artifact | |
| VEH | | | | | | | Part_Whole.Artifact |

# TAC-KBP抽取的目标关系列表

| Person | Organization | Geo-Political Entity |
|---|---|---|
| alternate names | alternate names | alternate names |
| age | political/religious affiliation | capital |
| birth: date, place | top members/employees | subsidiary orgs |
| death: date, place, cause | number of employees | top employees |
| national origin | members | political parties |
| residences | member of | established |
| spouse | subsidiaries | population |
| children | parents | currency |
| parents | founded by | |
| siblings | founded | |
| other family | dissolved | |
| schools attended | headquarters | |
| job title | shareholders | |
| employee-of | website | |
| member-of | | |
| religion | | |
| criminal charges | | |

# SemEval-2010 Task 8

- 目标关系列表：19种

| 关　系 | 比　例 |
|---|---|
| Other | 280 (31.43%) |
| Cause-Effect | 156 (17.51%) |
| Product-Producer | 114 (12.79%) |
| Entity-Origin | 102 (11.45%) |
| Instrument-Agency | 85 ( 9.54%) |
| Component-Whole | 71 ( 7.97%) |
| Content-Container | 57 ( 6.40%) |
| Entity-Destination | 13 ( 1.46%) |
| Member-Collection | 8 ( 0.90%) |
| Message-Topic | 5 ( 0.56%) |

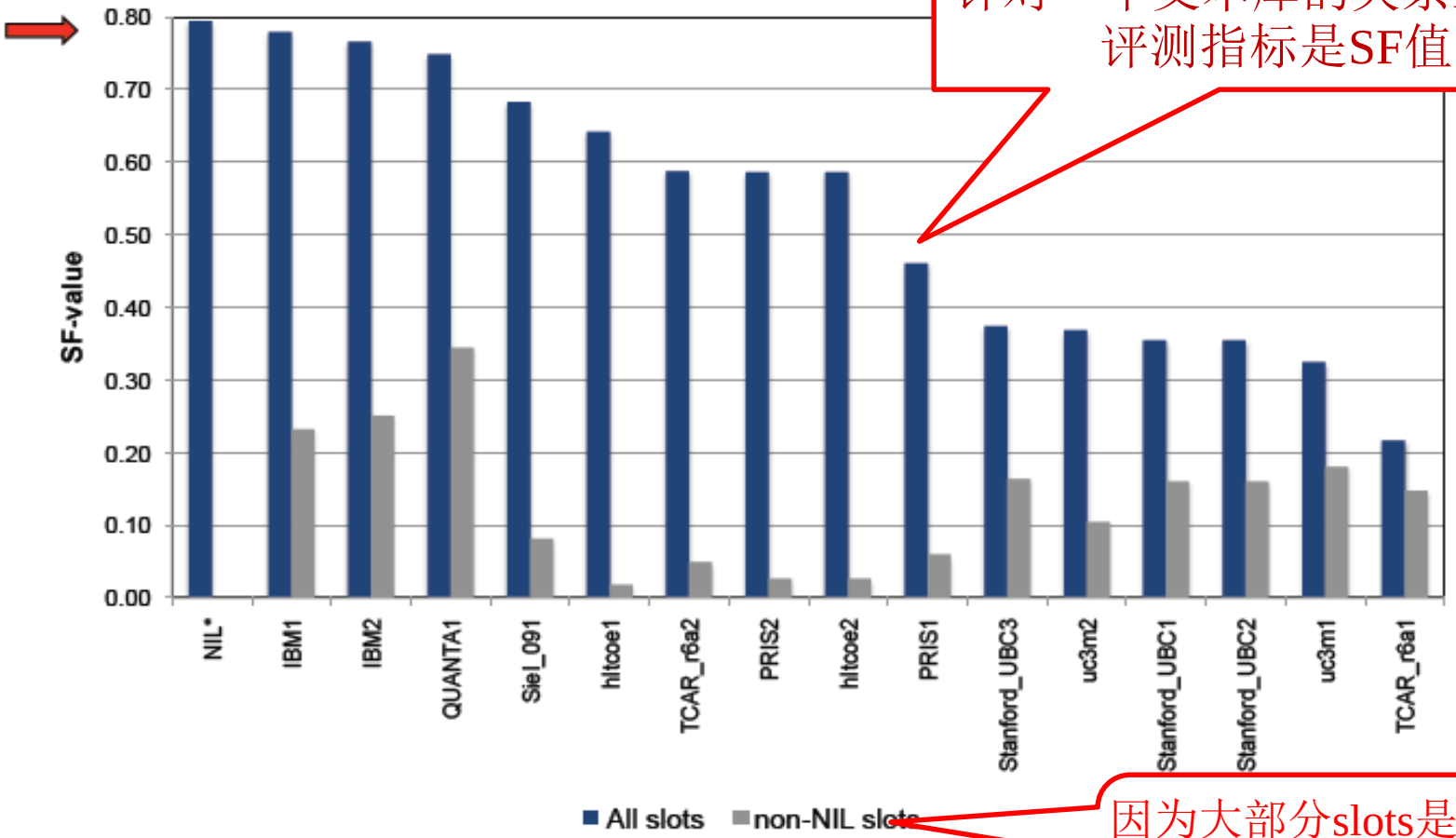
# ACE(2005-2007)评测结果

| 语种 | 语料来源类型 | ACE 2005 | | | | | ACE 2007 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EDR | RDR | TERN | VAL | VDR | EDR | RDR | TERN | VAL | VDR |
| Chinese | Broadcast News | 70.5 | 24.4 | 81.8 | 48.7 | 11.2 | 49.7 | 16.8 | | | |
| | Newswire | 70.1 | 28.6 | 84.3 | 42.1 | 10.8 | 46.9 | 18.7 | 9.9 | | |
| | Weblogs | 67.1 | 26.6 | 86.2 | 71.2 | 2 | 35 | 15.7 | 24.8 | | |
| | 总分 | 69.2 | 26.8 | 83.7 | 49.7 | 10.2 | 45 | 17.6 | 14.8 | | |
| English | Broadcast News | 77.3 | 33.1 | 65.6 | 12.6 | 12.3 | 65.9 | 24.7 | 68.6 | | 12.9 |
| | Newswire | 73.1 | 27.3 | 72.6 | 43.4 | 17.8 | 58.1 | 21.2 | 67.4 | | 15.9 |
| | Weblogs | 71.6 | 20.5 | 62.8 | 38.5 | 15.4 | 52.7 | 18.2 | 54.8 | | 15 |
| | Broadcast Conversation | 72.7 | 16 | 48 | 30 | 6.2 | 50.5 | 11 | [illegible] | | 7.4 |
| | Telephone | 67.7 | 35.6 | 58.6 | 25 | 15.2 | 49.2 | 32.4 | [illegible] | | 11.3 |
| | Usenet | 61.5 | 19.6 | 63.4 | 49.7 | 13.1 | 44 | 19.6 | | | |
| | 总分 | 71.9 | 25.2 | 63.7 | 34.8 | 14.4 | 56.3 | 21.6 | | | |
| Arabic | Broadcast News | | | | | | 51.9 | | | | |
| | Newswire | | | | | | 49.4 | | | | |
| | Weblogs | | | | | | 42.1 | | | | |
| | 总分 | | | | | | 48.8 | | | | |
| Spanish | Newswire | | | | | | 51 | | 46.5 | | |
| | 总分 | | | | | | | | 46.5 | | |

针对每个句子的关系识别，评测指标是F值

(http://www.nist.gov/speech/tests/ace/ace05/doc/acee05eval_official_results_20060110.htm
http://www.nist.gov/speech/tests/ace/ace05/doc/acee05eval_official_results_20070402.htm)

# TAC-KBP 2009 Slot Filling Track评测结果




针对一个文本库的关系抽取，评测指标是SF值

因为大部分slots是空的，所以non-NIL指标反映了系统的性能

(Paul McNamee, 2009)

# KBP 2013 Slot Filling Track评测结果

| | Diagnostic Scores | | | Official Scores | | |
|---|---|---|---|---|---|---|
| | Recall | Precision | F1 | Recall | Precision | F1 |
| lsv | **32.93** | 38.50 | 35.50 | **33.17** | 42.53 | **37.28** |
| ARPANI* | 29.10 | 47.83 | **36.18** | 27.45 | 50.38 | 35.54 |
| RPI-BLENDER | 30.62 | 38.19 | 33.98 | 29.02 | 40.73 | 33.89 |
| PRIS2013 | 27.82 | 35.33 | 31.13 | 27.59 | 38.87 | 32.27 |
| BIT | 22.06 | 57.86 | 31.94 | 21.73 | 61.35 | 32.09 |
| Stanford | 28.46 | 32.30 | 30.26 | 28.41 | 35.86 | 31.70 |
| NYU | 17.35 | 50.70 | 25.85 | 16.76 | 53.83 | 25.56 |
| UWashington | 10.31 | **59.72** | 17.59 | 10.29 | **63.45** | 17.70 |
| CMUML | 10.63 | 28.79 | 15.53 | 10.69 | 32.30 | 16.07 |
| SAFT_KRes | 13.43 | 12.43 | 12.91 | 14.99 | 15.67 | 15.32 |
| UMass_IESL | 18.47 | 9.43 | 12.48 | 18.46 | 10.88 | 13.69 |
| utaustin | 7.91 | 21.85 | 11.62 | 8.11 | 25.16 | 12.26 |
| UNED | 9.11 | 15.08 | 11.36 | 9.33 | 17.59 | 12.19 |
| Compreno | 13.19 | 8.69 | 10.48 | 12.74 | 9.74 | 11.04 |
| TALP_UPC | 9.67 | 6.54 | 7.81 | 9.81 | 7.69 | 8.62 |
| IIRG | 3.20 | 7.38 | 4.46 | 2.86 | 7.72 | 4.17 |
| SINDI | 2.80 | 7.26 | 4.04 | 2.59 | 7.84 | 3.89 |
| CohenCMU | 3.68 | 1.69 | 2.32 | 3.68 | 1.98 | 2.57 |
| LDC | 58.35 | 83.81 | 68.80 | 57.08 | 85.60 | 68.49 |

# SemEval-2010 Task 8 评测结果

| System | TD1 | TD2 | TD3 | TD4 | Acc TD4 | Rank | Best Cat | Worst Cat-9 |
|---|---|---|---|---|---|---|---|---|
| Baseline | 33.04 | 42.41 | 50.89 | 57.52 | 50.0 | - | MC (75.1) | IA (28.0) |
| ECNU-SR-1 | 52.13 | 56.58 | 58.16 | 60.08 | 57.1 | 4 | CE (79.7) | IA (32.2) |
| ECNU-SR-2 | 46.24 | 47.99 | 69.83 | 72.59 | 67.1 | | CE (84.4) | IA (52.2) |
| ECNU-SR-3 | 39.89 | 42.29 | 65.47 | 68.50 | 62.0 | | CE (83.4) | IA (46.5) |
| ECNU-SR-4 | 67.95 | 70.58 | 72.99 | 74.82 | 70.5 | | CE (84.6) | IA (61.4) |
| *ECNU-SR-5* | 49.32 | 50.70 | 72.63 | 75.43 | 70.2 | | CE (85.1) | IA (60.7) |
| ECNU-SR-6 | 42.88 | 45.54 | 68.87 | 72.19 | 65.8 | | CE (85.2) | IA (56.7) |
| ECNU-SR-7 | 58.67 | 58.87 | 72.79 | 75.21 | 70.2 | | CE (86.1) | IA (61.8) |
| FBK_IRST-6C32 | 60.19 | 67.31 | 71.78 | 76.81 | 72.4 | 2 | ED (82.6) | IA (69.4) |
| FBK_IRST-12C32 | 60.66 | 67.91 | 72.04 | 76.91 | 72.4 | | MC (84.2) | IA (68.8) |
| FBK_IRST-12VBC32 | 62.64 | 69.86 | 73.19 | 77.11 | 72.3 | | ED (85.9) | PP (68.1) |
| FBK_IRST-6CA | 60.58 | 67.14 | 71.63 | 76.28 | 71.4 | | CE (82.3) | IA (67.7) |
| FBK_IRST-12CA | 61.33 | 67.80 | 71.65 | 76.39 | 71.4 | | ED (81.8) | IA (67.5) |
| *FBK_IRST-12VBCA* | 63.61 | 70.20 | 73.40 | 77.62 | 72.8 | | ED (86.5) | IA (67.3) |
| *FBK_NK-RES1* | 55.71* | 64.06* | 67.80* | 68.02 | 62.1 | 7 | ED (77.6) | IA (52.9) |
| FBK_NK-RES2 | 54.27* | 63.68* | 67.08* | 67.48 | 61.4 | | ED (77.4) | PP (55.2) |
| FBK_NK-RES3 | 54.25* | 62.73* | 66.11* | 66.90 | 60.5 | | MC (76.7) | IA (56.3) |
| FBK_NK-RES4 | 44.11* | 58.85* | 63.06* | 65.84 | 59.4 | | MC (76.1) | IA/PP (58.0) |
| *ISI* | 66.68 | 71.01 | 75.51 | 77.57 | 72.7 | 3 | CE (87.6) | IA (61.5) |
| *ISTI-1* | 50.49* | 55.80* | 61.14* | 68.42 | 63.2 | 6 | ED (80.7) | PP (53.8) |
| ISTI-2 | 50.69* | 54.29* | 59.77* | 66.65 | 61.5 | | ED (80.2) | IA (48.9) |
| *JU* | 41.62* | 44.98* | 47.81* | 52.16 | 50.2 | 9 | CE (75.6) | IA (27.8) |
| *SEKA* | 51.81 | 56.34 | 61.10 | 66.33 | 61.9 | 8 | CE (84.0) | PP (43.7) |
| TUD-base | 50.81 | 54.61 | 56.98 | 60.50 | 56.1 | 5 | CE (80.7) | IA (31.1) |
| TUD-wp | 55.34 | 60.90 | 63.78 | 68.00 | 63.5 | | ED (82.9) | IA (44.1) |
| TUD-comb | 57.84 | 62.52 | 66.41 | 68.88 | 64.6 | | CE (83.8) | IA (46.8) |
| *TUD-comb-θ* | 58.35 | 62.45 | 66.86 | 69.23 | 65.4 | | CE (83.4) | IA (46.9) |
| *UNITN* | 16.57* | 18.56* | 22.45* | 26.67 | 27.4 | 10 | ED (46.4) | PP (0) |
| *UTD* | **73.08** | **77.02** | **79.93** | **82.19** | 77.9 | 1 | CE (89.6) | IA (68.5) |

# 传统关系抽取方法

- 目前主要采用统计机器学习的方法，将关系实例转换成高维空间中的特征向量或直接用离散结构来表示，在标注语料库上训练生成分类模型，然后再识别实体间关系
  - 基于特征向量方法：最大熵模型(Kambhatla 2004)和支持向量机(Zhao et al., 2005；Zhou et al., 2005; Jiang et al., 2007)等
  - 基于核函数的方法：浅层树核（Zelenko et al., 2003)、依存树核（Culotta et al., 2004)、最短依存树核（Bunescu et al., 2005)、卷积树核（Zhang et al., 2006；Zhou et al., 2007）
  - 基于神经网络的方法：递归神经网络（ Socher et al., 2012)、基于矩阵空间的递归神经网络（ Socher et al., 2012)、卷积神经网络（Zeng et al., 2014）

# 传统关系抽取方法

- 基于特征向量方法:
  - 主要问题：如何获取各种有效的词法、句法、语义等特征，并把它们有效地集成起来，从而产生描述实体语义关系的各种局部特征和简单的全局特征
  - 特征选取：从自由文本及其句法结构中抽取出各种词汇特征以及结构化特征
    - 实体词汇及其上下文特征
    - 实体类型及其组合特征（PER，LOC等）
    - 交叠特征（两个实体或词组块是否在同一个名词短语、动词短语或者介词短语之中、两个实体或者词组块之间单词的个数等）
    - 句法树特征（连接两个实体的语法路径）

# 传统关系抽取方法

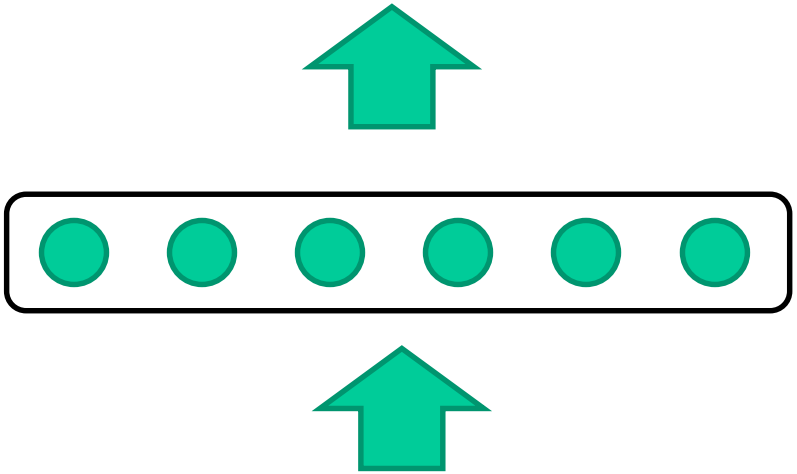
- 基于核函数方法:
  - 主要问题：如何有效挖掘反映语义关系的结构化信息及如何有效计算结构化信息之间的相似度
  - 卷积树核：用两个句法树之间的公共子树的数目来衡量它们之间的相似度
    - 标准的卷积树核(CTK)
      - 在计算两棵子树的相似度时，只考虑子树本身，不考虑子树的上下文信息
    - 上下文相关卷积树核函数(CS-CTK)
      - 在计算子树相似度时，同时考虑子树的祖先信息，如子树根结点的父结点、祖父结点信息，并对不同祖先的子树相似度加权平均

# 传统关系抽取方法

- 基于神经网络的方法:
  - 主要问题：如何设计合理的网络结构，从而捕捉更多的信息，进而更准确地完成关系的抽取
  - 网络结构：不同的网络结构捕捉文本中不同的信息
    - 递归神经网络（RNN）
      - 网络的构建过程更多的考虑到句子的句法结构，但是需要依赖复杂的句法分析工具
    - 卷积神经网络（CNN）
      - 通过卷积操作完成句子级信息的捕获，不需要复杂的NLP工具
    - 循环神经网络（RNN）
      - 通过循环神经网络建模词语之间的依赖关系，自动捕获句子级信息

# 自然语言处理过程

分类, 聚类, （结构）预测, 排序...



伦齐在宪改公投惨败后，宣布有意辞职。

特征
- 符号特征：特征抽取
- 数值特征：表示学习

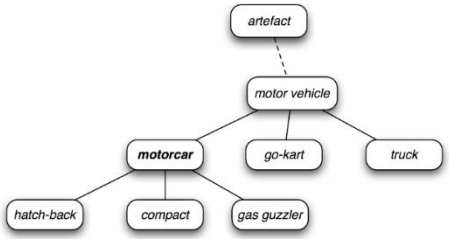
# 自然语言的表示学习

- 基本语言单元的语义表示
  - 词的表示
  - 分布式原则
- 更大语言单元的语义表示
  - 句子的表示
  - 语义组合原则

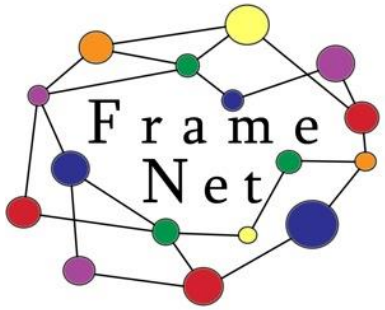
深度学习的本质：用数值（向量，矩阵等）表示各个语言单元的语义（特征）

# 词表示

- 一般认为，词是语言处理中最基本的语言单元
- 词以及词间关系的表示和建模是NLP任务中重要基础工作




WordNet [Miller 1995]



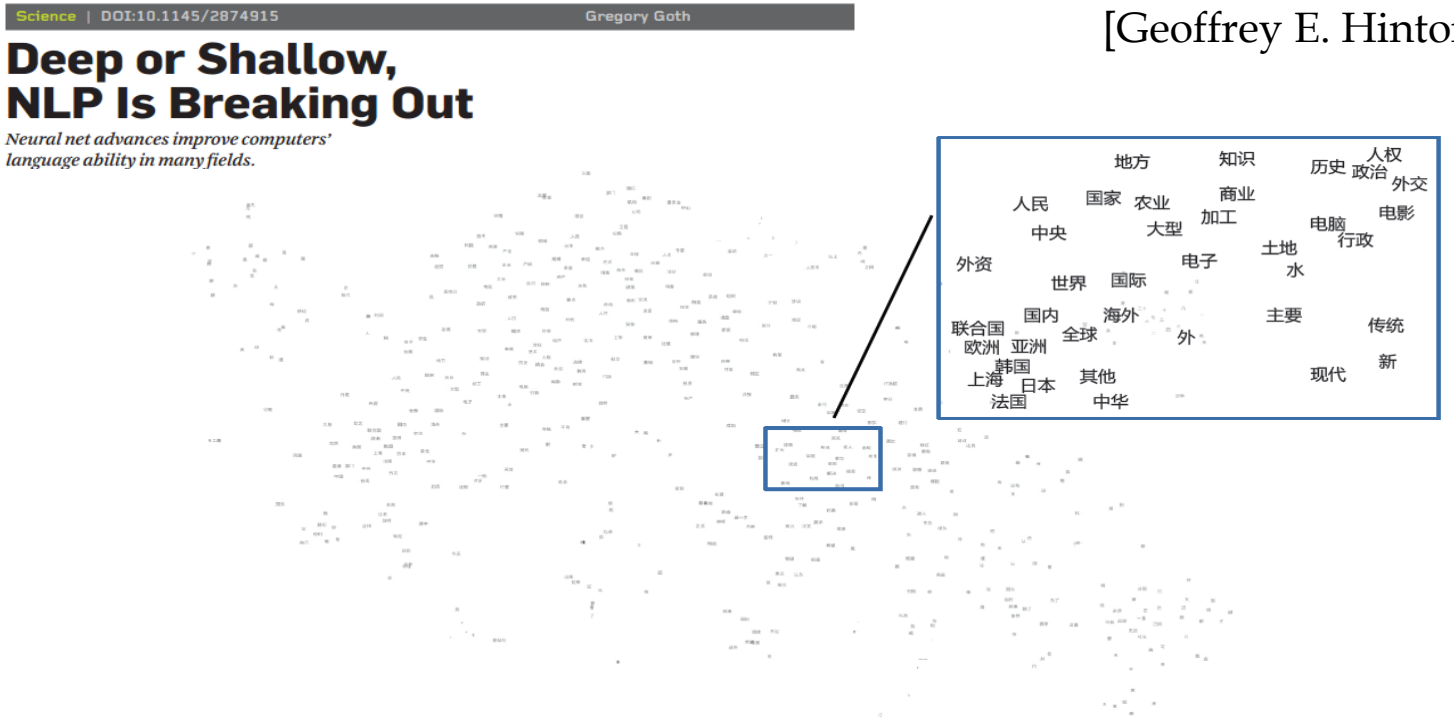

FrameNet [Baker 1998]

人工方法：刻画词之间多维度关系，例如：聚合关系，组合关系
深度学习：用数值化的方法表示词的语义信息

# 词的分布表示

- 深度学习框架下，词的语义表示为向量的形式，也就是分布表示。

The first thing you do with a word symbol is you convert it to a word vector. And you learn to do that, you learn for each word how to turn a symbol into a vector, say, 300 components, and after you’ ve done learning, you’ ll discover the vector for Tuesday is very similar to the vector for Wednesday.




[Geoffrey E. Hinton. 2015]

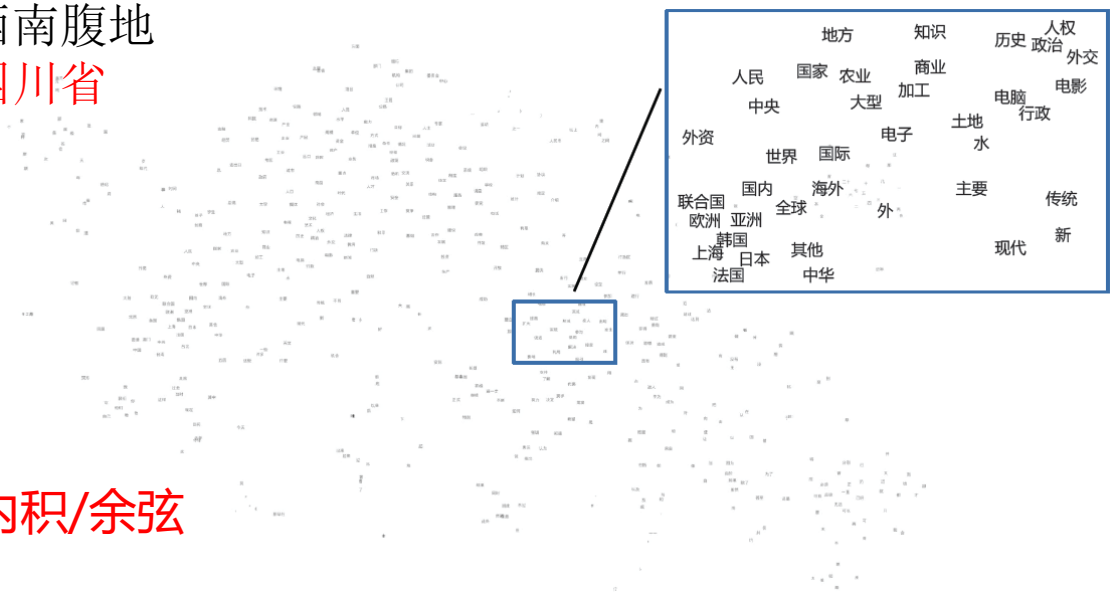
# 词的分布表示

- Harris和firth的分布假说：上下文相似的词 → 词义相似
  - 词的语义可以通过大规模语料库统计获得
  - 词语的语义相似度可以度量

[Harris 1954]

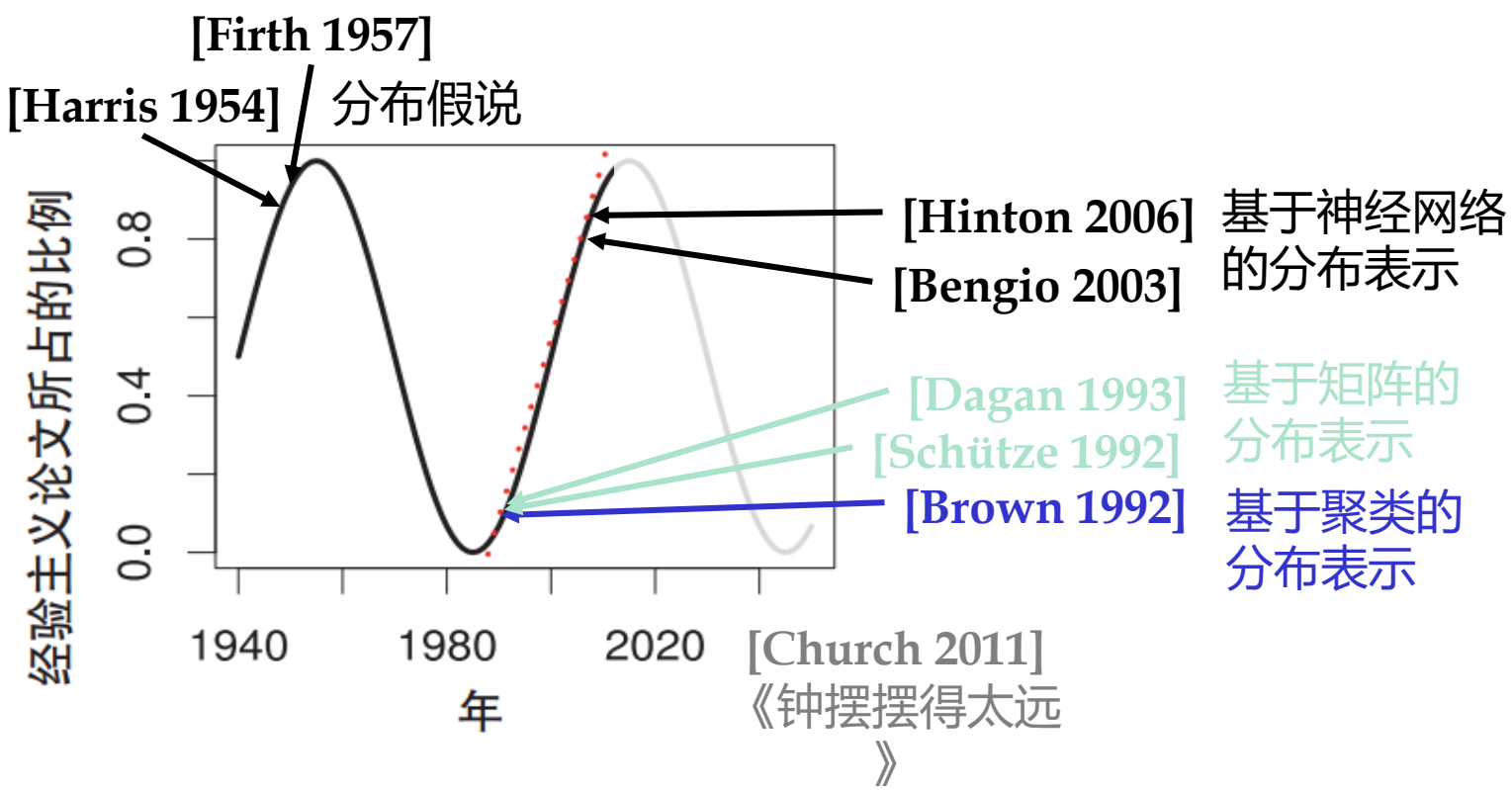
江西省地处中国东南部
携程带你玩转江西省

四川省地处中国西南腹地
携程带你玩转四川省




向量 & 内积/余弦

# 词的分布表示：历史的角度



# 词表示学习

- 上下文的表示：文档、词、n元词组
- 相似度的衡量：向量的内积（余弦）
- 词间关系一般分为聚合关系和组合关系，分布式表示学习常常从聚合关系的角度去建模

Syntagmatic Relation

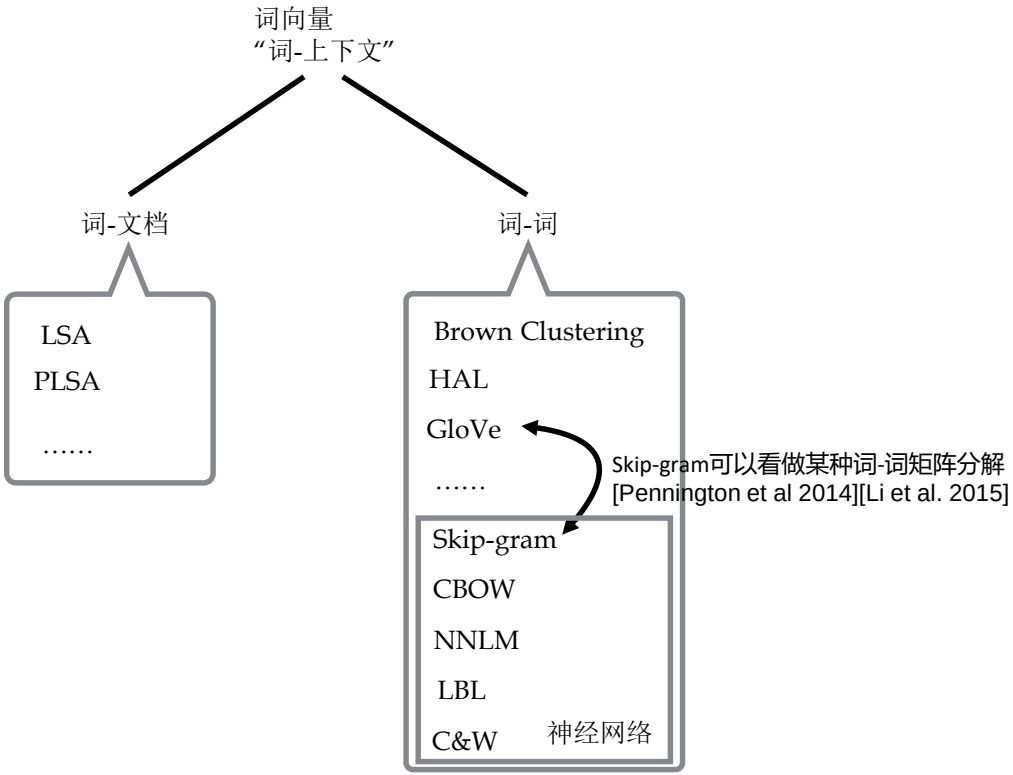
清华 校园 景色 很好

中科院 几乎 没有 校园

Paradigmatic Relation

# 词表示学习方法

- 传统方法：也是从分布的角度建模，但是没有用到神经网络的方法
  - ✓ LSA：矩阵分解
  - ✓ Brown Clustering：聚类
- 神经网络：NNLM, Skip-gram, CBOW, GloVe，LBL，C&W



# NNLM: Neural Network Language Model

- 三层前馈神经网络，以前n-1个词作为输入，第n个词作为输出，进而完成网络参数的优化，实现词向量的表示学习
  - 拼接输入词词向量

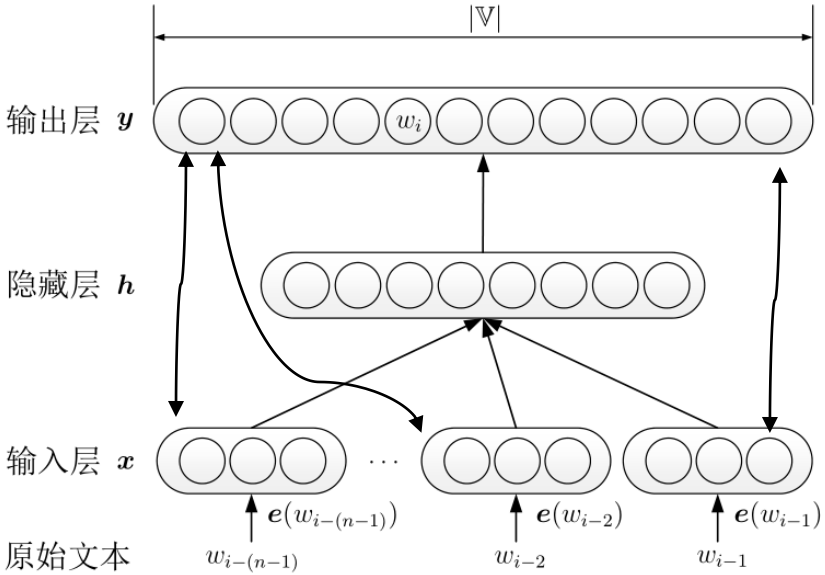
$$\boldsymbol{x} = [\boldsymbol{e}(w_{i-(n-1)}); \ldots; \boldsymbol{e}(w_{i-2}); \boldsymbol{e}(w_{i-1})]$$

  - 隐藏层

$$\boldsymbol{h} = \tanh(\boldsymbol{b}^{(1)} + H\boldsymbol{x})$$

  - 输出层

$$\boldsymbol{y} = \boldsymbol{b}^{(2)} + W\boldsymbol{x} + U\boldsymbol{h}$$



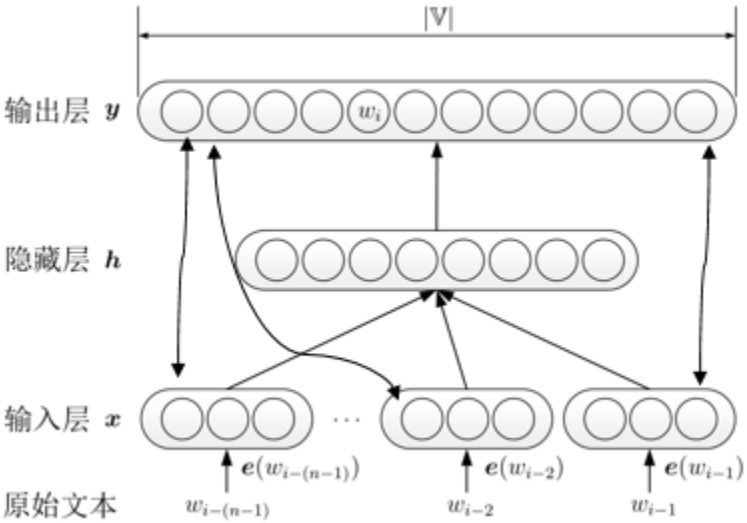

$$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{\exp(\boldsymbol{y}(w_i))}{\sum_{k=1}^{|\mathbb{V}|} \exp(\boldsymbol{y}(v_k))}$$

通过调节输入词向量、输出词向量以及网络的参数，使得网络每次都能正确地预测出第N个词

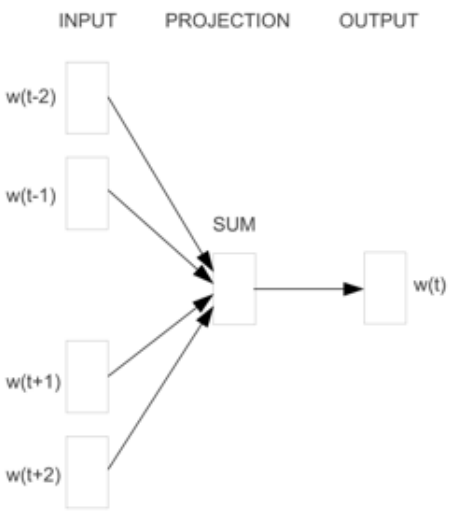
# CBOW & Skip-gram

- Continuous Bag of Word
- 特点：
  - 去除隐藏层
  - 去除词序

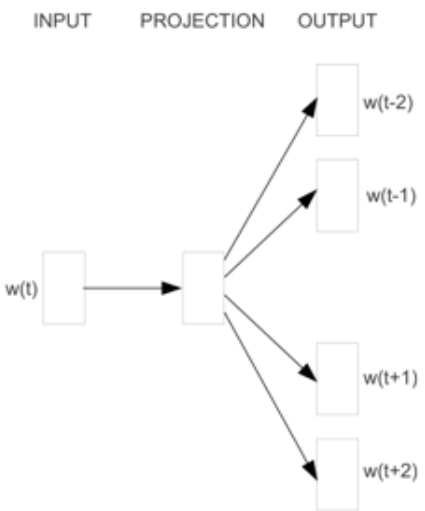
研表究明，汉字序顺并不定一影阅响读！事证实明了也许当你看这完句话之后才发字现都乱是的。




NNLM




CBOW
用周边词预测当前词




Skip-gram
用当前词预测周边的词

# 工具示例

- Word2vec：Google在2013年年中开源的一款将词表征为实数值向量的高效工具，采用的模型有CBOW和Skip-Gram两种。
- 代码链接：https://code.google.com/p/word2vec/
- 输入文本：英文（token后的文本）中文（分词后的文本）
- 输出：指定维度的词向量

```
,
the
.
of
a
and
to
in
''
that
```

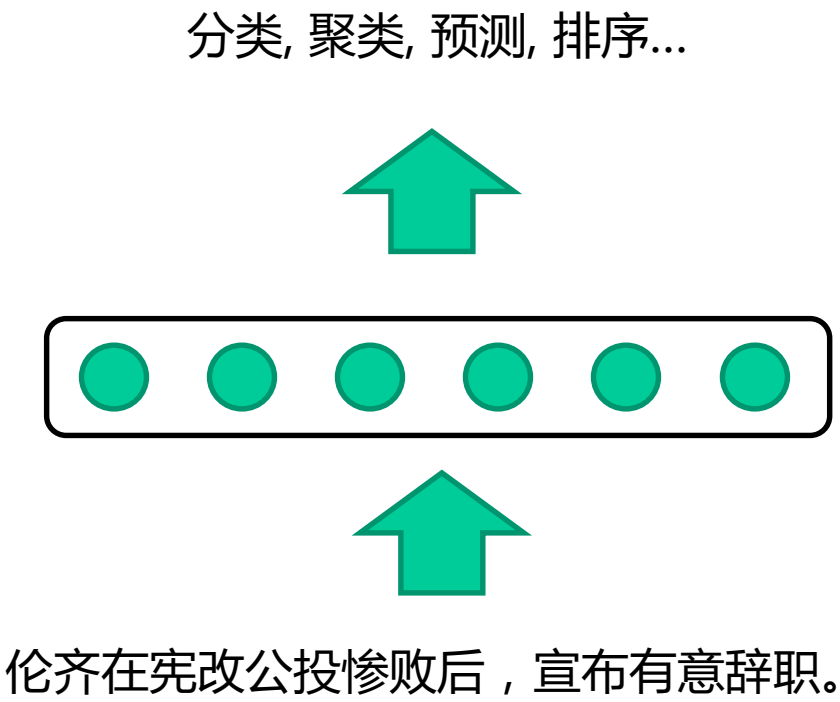
```
0.556314,0.024821,-0.486807,1.029960,0.431357,0.831494,-0.327807,0.066419,-
0.120521,2.321336,-0.976254,2.738440,0.458854,-0.919605,-0.237615,1.961676,
0.389097,0.664733,-1.738388,1.373171,0.987761,0.429780,-0.178071,0.069103,-
0.204649,1.058529,-1.495981,1.302744,-0.883764,-0.671500,1.039746,1.182477,
0.631253,-0.997995,-0.646097,2.619554,-1.440681,-0.128711,-0.336226,2.46306
1.517981,1.651188,-0.554875,0.146960,0.805013,-0.004309,0.383949,1.272646,-
1.078778,1.145040,-2.936579,2.731101,1.569286,0.121215,-1.080063,0.769050,(
0.051939,0.881635,-1.327522,1.980237,0.499068,0.391517,1.666283,2.997930,1.
1.111387,-0.773637,-1.408212,2.719081,0.558309,0.661429,-2.004113,-2.10238
0.892893,1.864152,-2.325571,3.441925,0.562785,0.590744,-1.772657,0.423680,-
```

# 自然语言的表示学习

- 基本语言单元的语义表示
  - 词的表示
  - 分布式原则

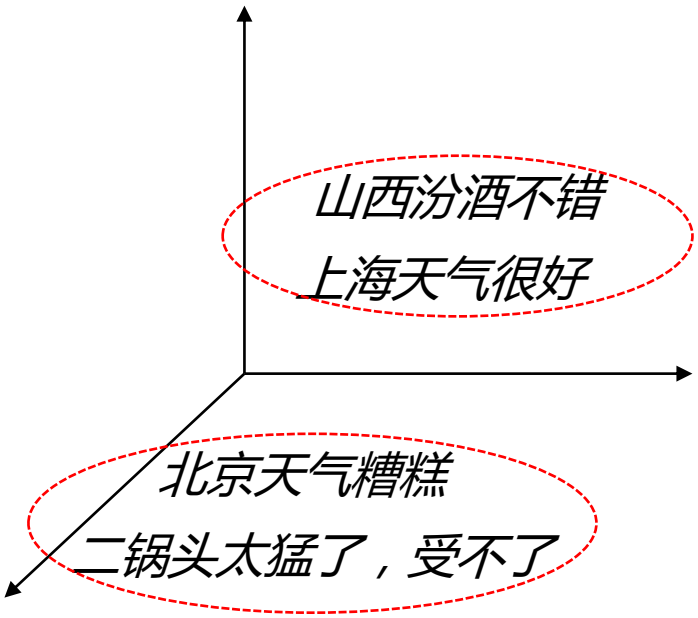
- 更大语言单元的语义表示
  - 句子的表示
  - 语义组合原则

# 组合语义

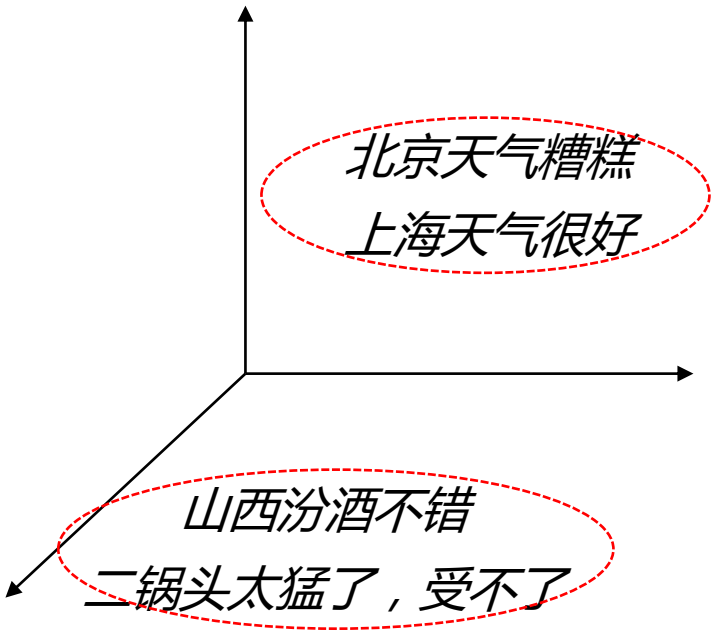



- 一个复杂对象的意义是由其各组成部分的意义以及它们的组合规则来决定
- The meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.

# 意义的组合和具体任务相关




情感分类

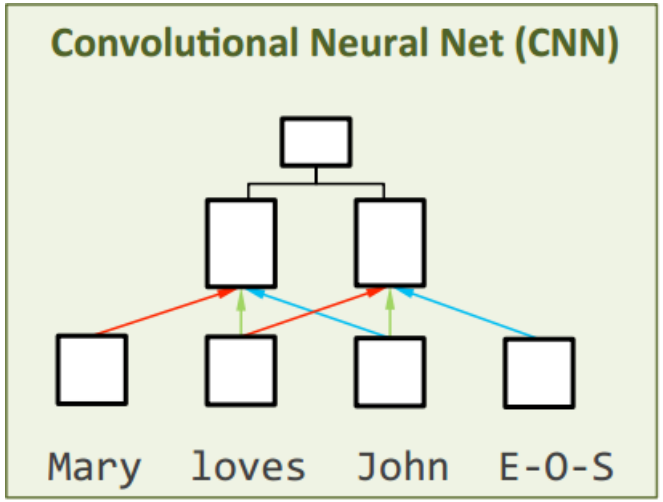



主题分类

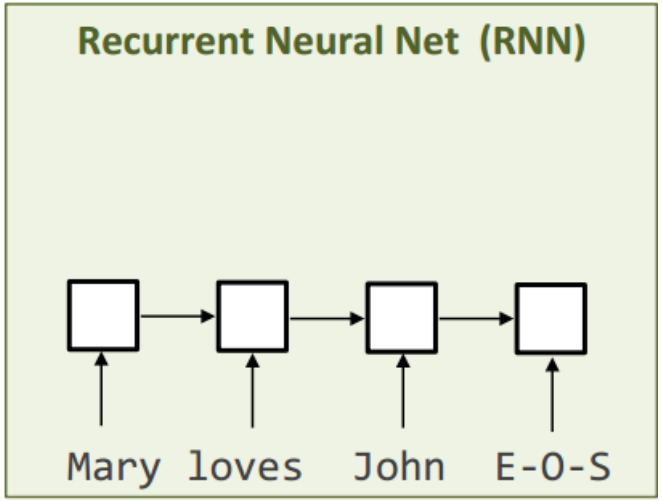
# 语义组合模型

- 基于词语等基本语义单元表示句子等更大语义单元的过程
- 语义组合模型就是要建模句子的句法/语义结构

两类最基本语义组合模型



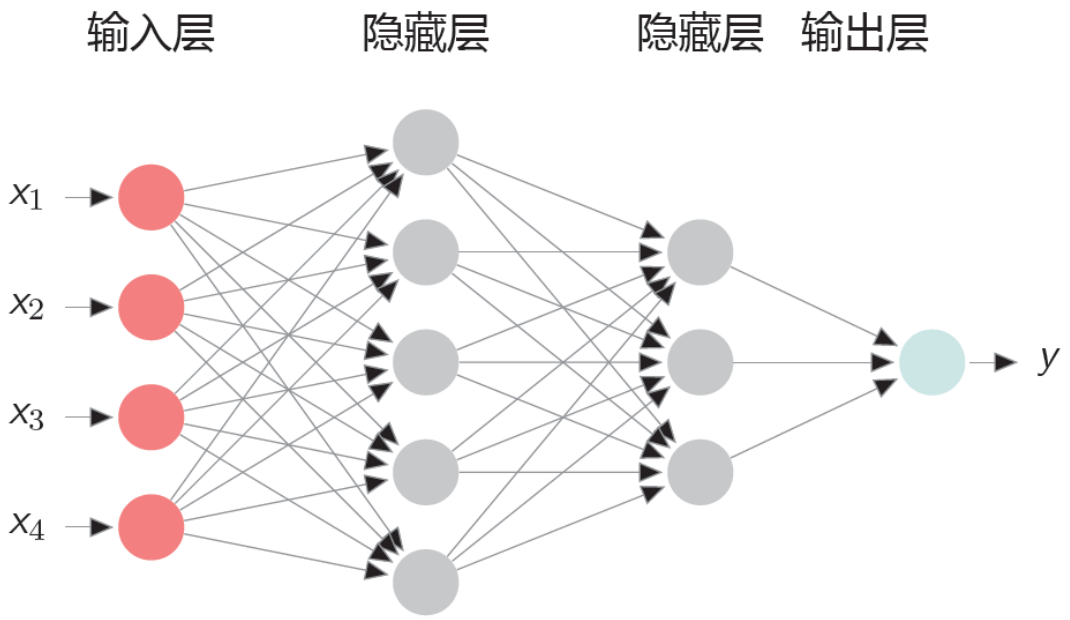

局部空间的特征提取




词语之间的时序关系

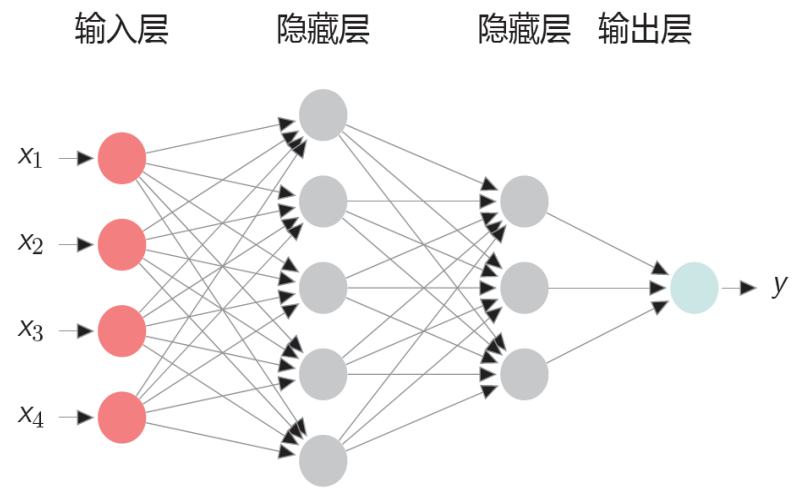
# 基础：多层前馈神经网络

- 在前馈神经网络中，各神经元分别属于不同的层。整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。

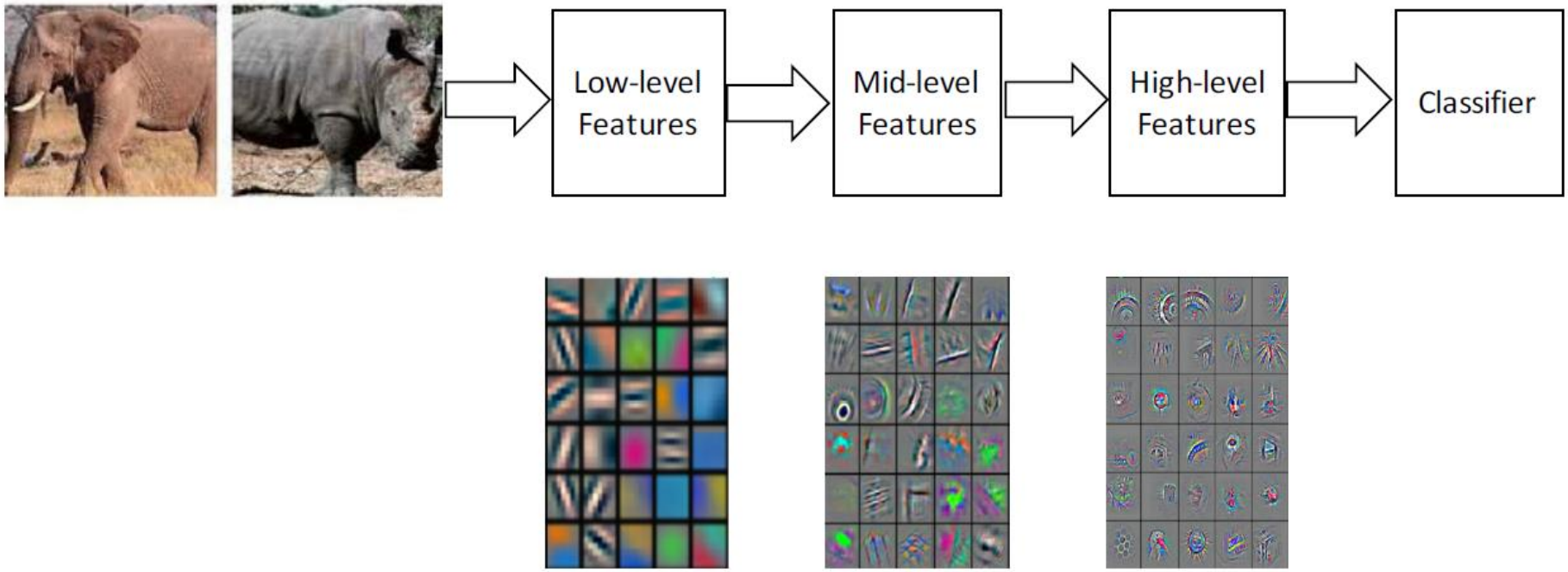



多层前馈神经网络

# 基础：多层前馈神经网络

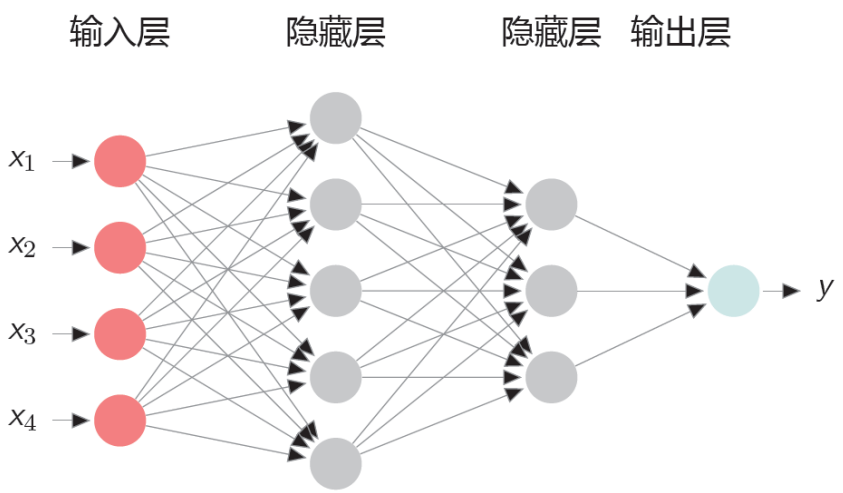



## 示例



# 基础：多层前馈神经网络

- 全连接层：相邻两层的任何两个不同层神经元之间都存在连接。在全连接前馈神经网络中
- 如果第l 层有$n^l$个神经元，*L*表示神经网络的层数
- 则参数个数：$\prod_{l=1}^{L} n^l$
- 权重矩阵参数非常多，训练效率低下
- 数据不足时，欠学习



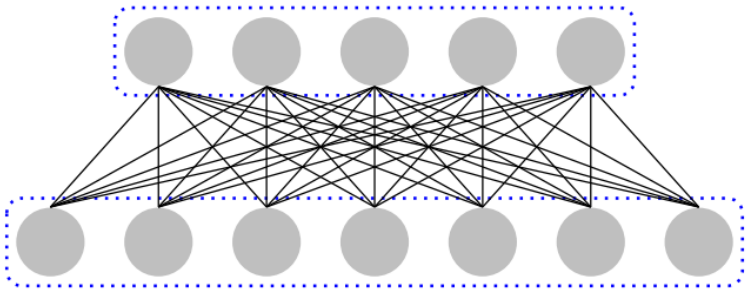

多层前馈神经网络

# 卷积神经网络

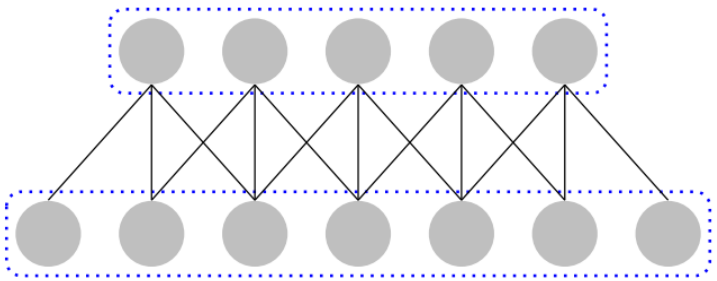
- Convolutional Neural Networks，CNN
- 是一种前馈神经网络
- 是受生物学上感受野（Receptive Field）的机制而提出的。
  - 感受野主要是指听觉系统、感觉系统和视觉系统中神经元的一些性质。
    - 例如：在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。
- 特性：卷积神经网络有三个结构上的特性
  - 局部连接
  - 权重共享
  - 采样

这些特性使得卷积神经网络具有一定程度上的平移、缩放和扭曲不变性

# 全连接 vs 卷积



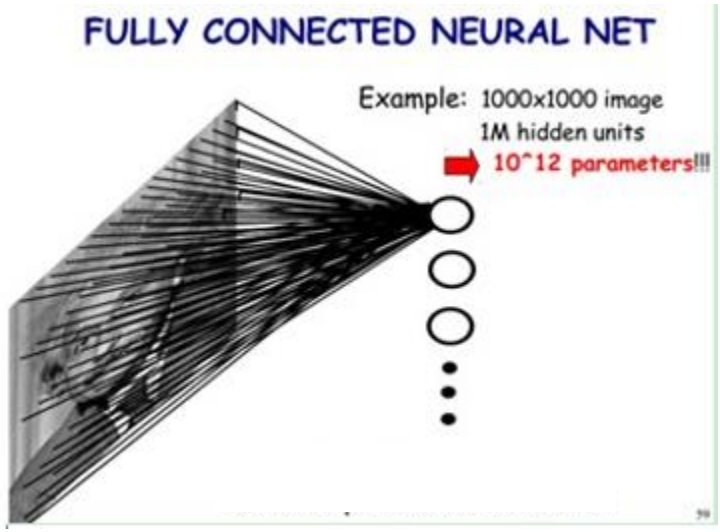
全连接

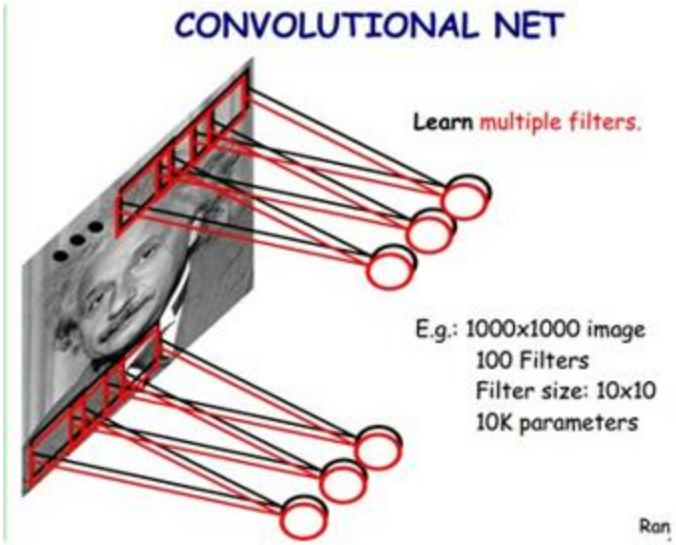


CNN

- 不同
  - 全连接 vs 局部连接
  - 权重独立 vs 权重共享
  - 一般不采样 vs 一般采样

# 例子：参数规模（全连接 vs 卷积）




$1000*1000*1000000=10^{12}$

$10*10*100=10^{4}$

- 左边的图表示全连接，输入为1000*1000，假设下一层神经元的数目为1M，参数规模为$10^{12}$。
- 右边的图是卷积网络示意，当一个滤波器的大小为10*10时，一个滤波器处理图片不同区域时权重共享，所以卷积结果规模为991*991，已经接近1M；即使使用100个滤波器，参数规模也为$10*10*100=10^{4}$，参数个数远小于全连接的情况。

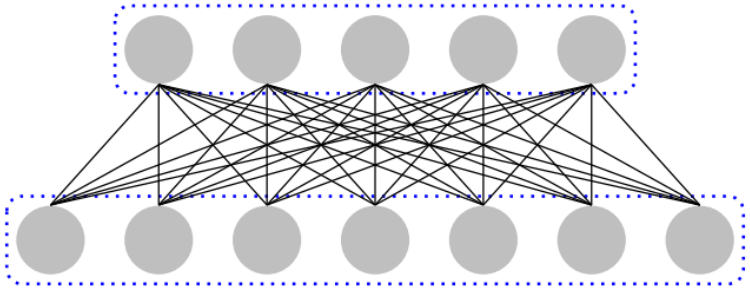
# 一维卷积

- 信号X，信号长度n，X可以表示为（$x_1$，$x_2$，…，$x_n$）
- $x_{i:i+j}$表示（$x_i$，$x_{i+1}$，…，$x_{i+j}$）
- 滤波器（filter）W，滤波器长度为m，W可以表示为（$w_1, w_2, \ldots, w_m$）
- 则卷积后的输出值Y中的每个位置可以如下表示

$$y_i = f(Wx_{i:i+m-1} + b) = f(\sum_{k=1}^{m} w_k x_{\mathrm{i}+k-1} + \mathrm{b})$$

偏置

# 全连接 vs 卷积

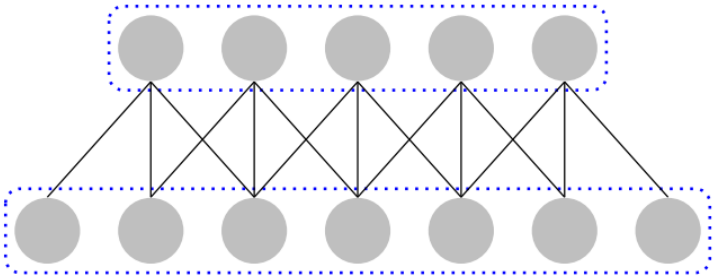


全连接

$$y_i = f(W_i x_{1:n} + b) = f(\sum_{k=1}^{n} w_{ik} x_k + b)$$

n是输入层神经元个数

m是滤波器窗口大小



CNN

局部连接

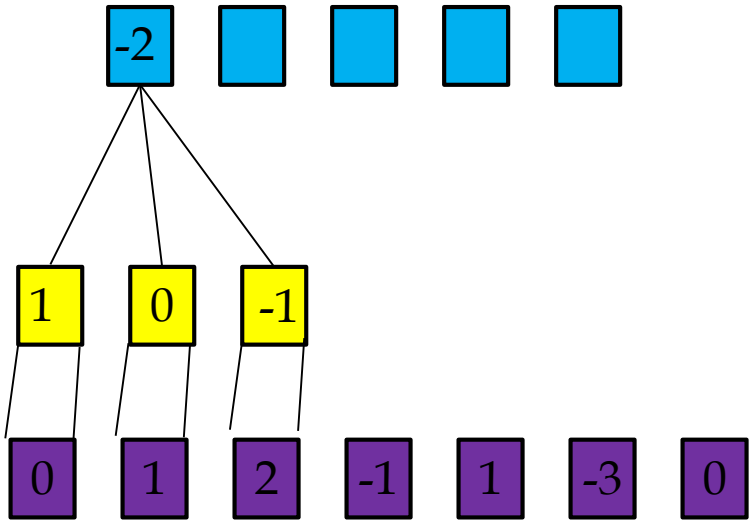
权重共享

$$y_i = f(W x_{i:i+m-1} + b) = f(\sum_{k=1}^{m} w_k x_{i+k-1} + b)$$

f是神经元的激活函数，一般取非线性变换函数

# 一维卷积示例

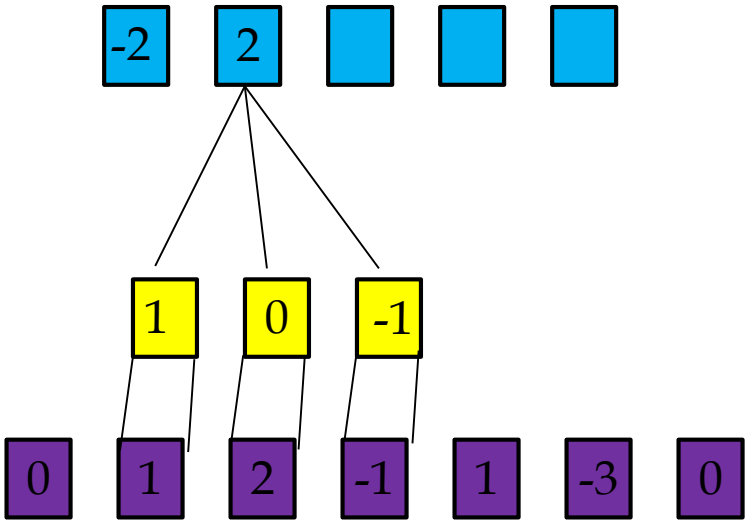
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

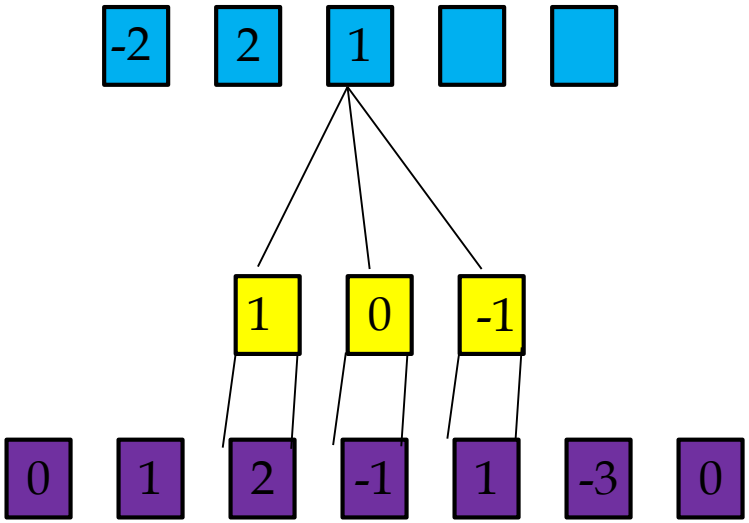
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

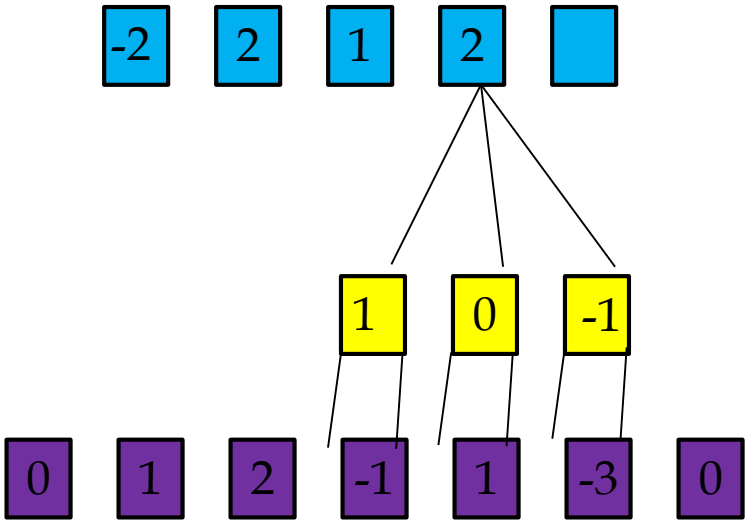
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

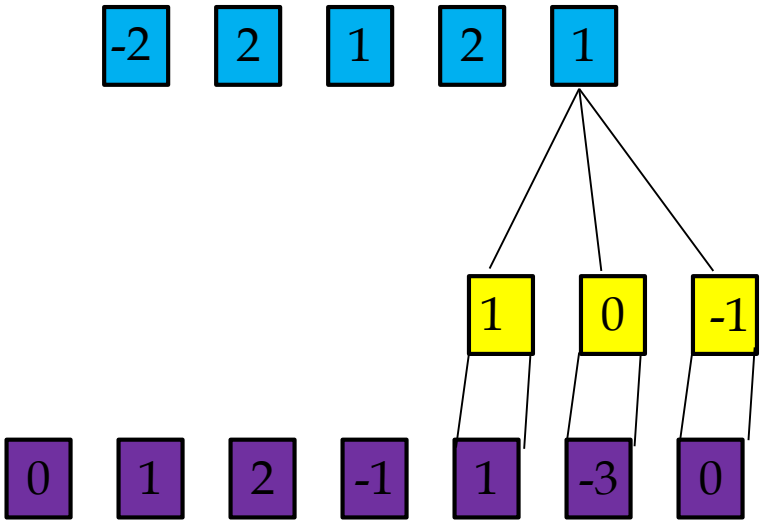
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

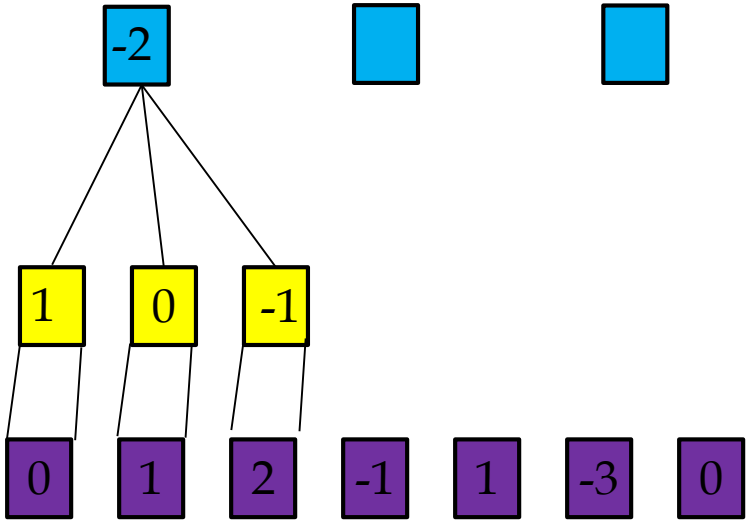
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

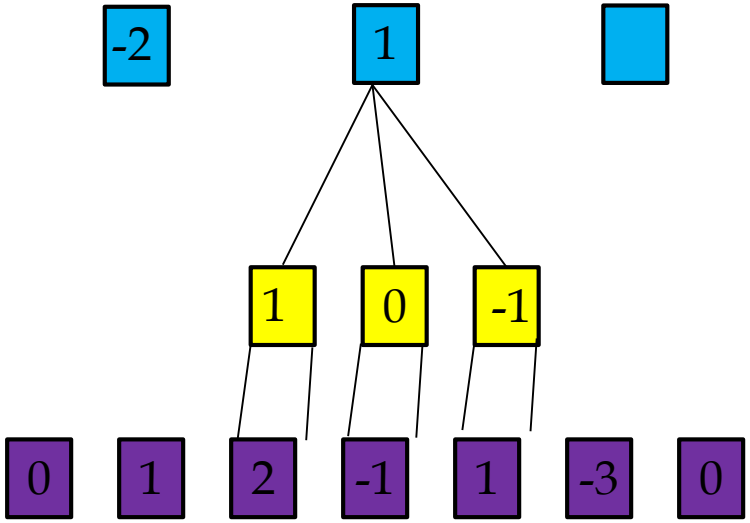
# 卷积的参数

- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2



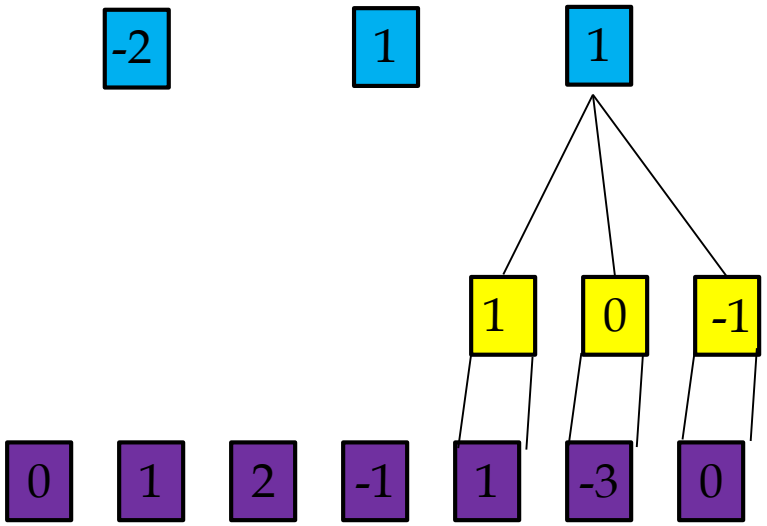
# 卷积步长

- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2



# 卷积步长

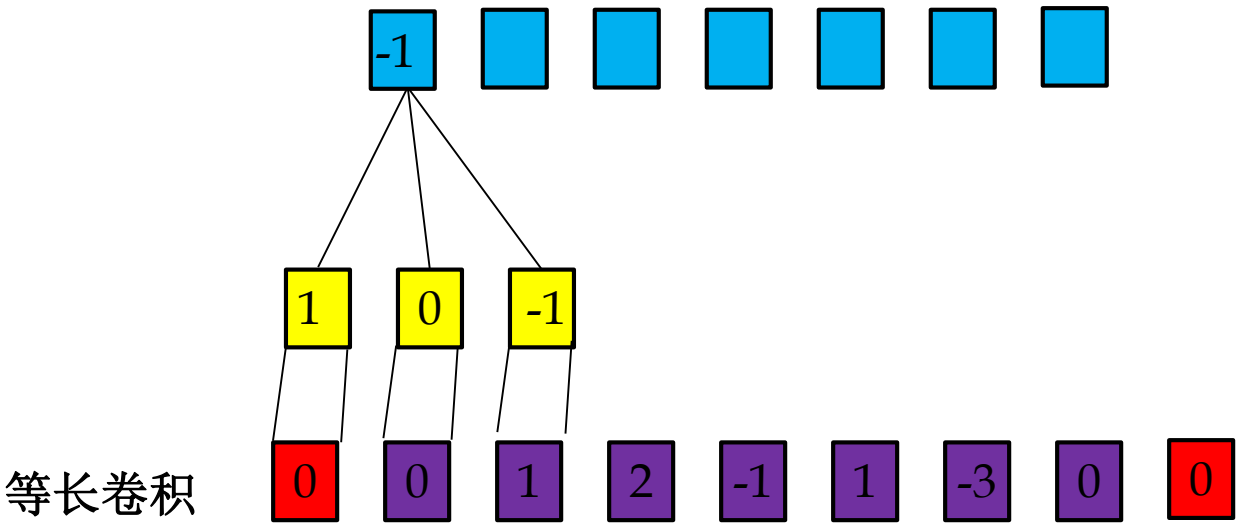
- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2




以下如无特殊说明，步长取1

# 窄卷积、宽卷积、等长卷积

- **窄卷积**：输出长度n-m+1，不补零。
- **宽卷积**：输出长度n + m -1，对于不在[1,n] 范围之外的$x_t$ 用零补齐（zero-padding）。（Padding=m-1）
- **等长卷积**：输出长度n，对于不在[1,n]范围之外的$x_t$ 用零补齐（zero-padding）。(Padding=(m-1)/2)




- 在这里除了特别声明，我们一般说的卷积默认为**窄卷积**。

# 二维卷积

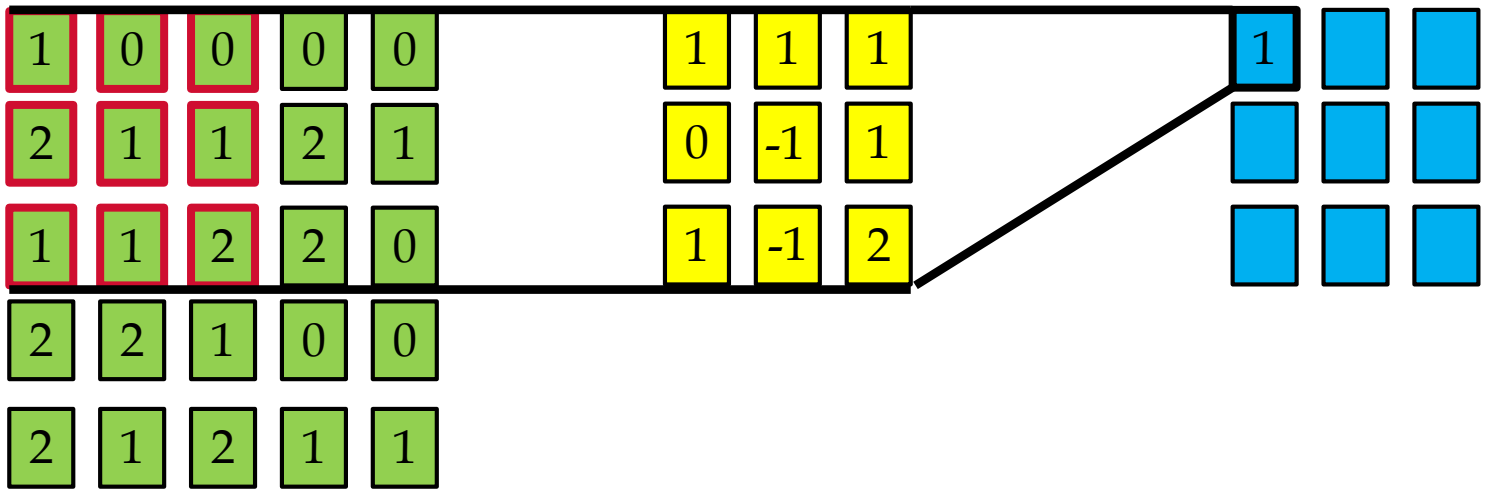
- 两维卷积经常用在图像和自然语言处理中
- 给定一个图像或句子X=[$x_{ij}$], i∈ [1, M], j ∈ [1, $N$]
- 滤波器W=[$w_{ij}$], i∈ [1, m], j ∈ [1, n] 一般$m << M; n << N$
- 则卷积后的输出值Y中的每个位置如下表示

$$y_{\mathrm{ij}} = f\left(W x_{i:i+m-1,j:i+n-1} + b\right)$$

$$= f\left(\sum_{u=1}^{m}\sum_{v=1}^{n} w_{uv} x_{\mathrm{i}+u-1,j+v-1} + \mathrm{b}\right)$$

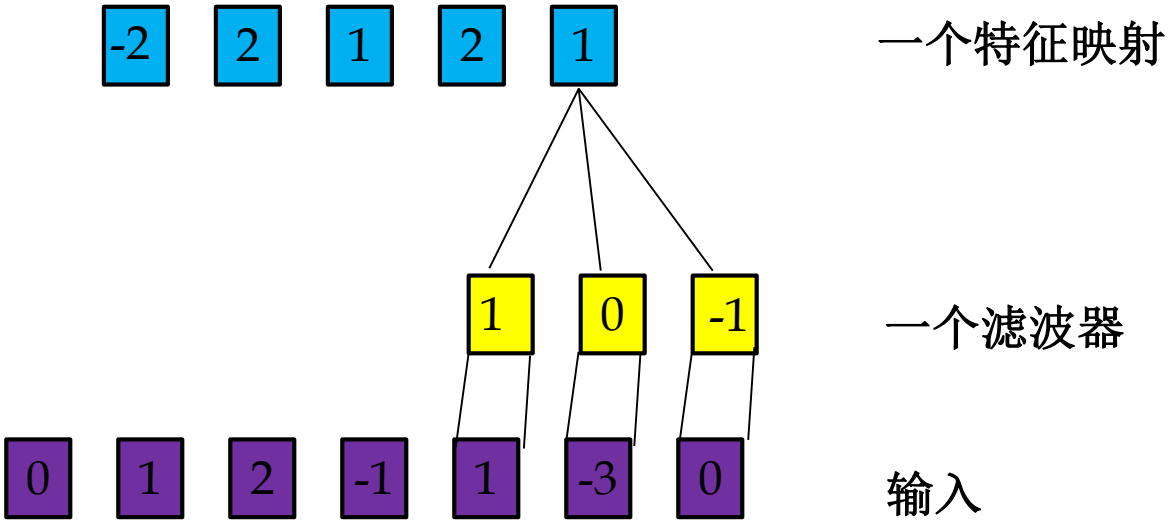
# 二维卷积示例

- 输入X(5，5)
- 滤波器（3，3）



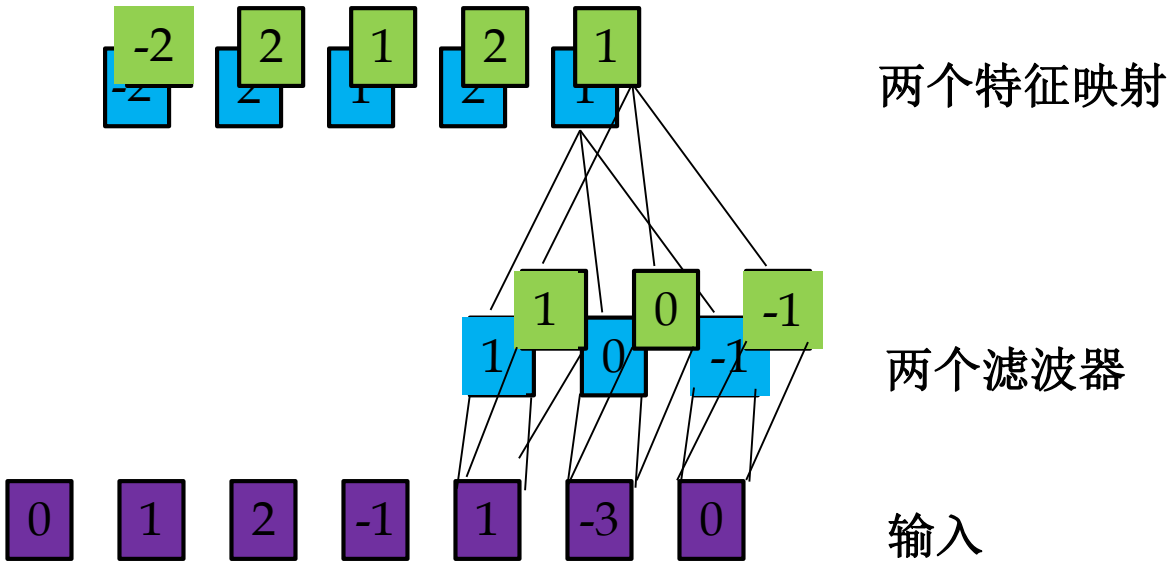
# 滤波器：特征提取器

- 如果把滤波器看成一个特征提取器，每一组输出都可以看成是输入经过一个特征抽取后得到的特征。因此，在卷积神经网络中每一组输出也叫作一个**特征映射**（Feature Map）。



# 多个滤波器

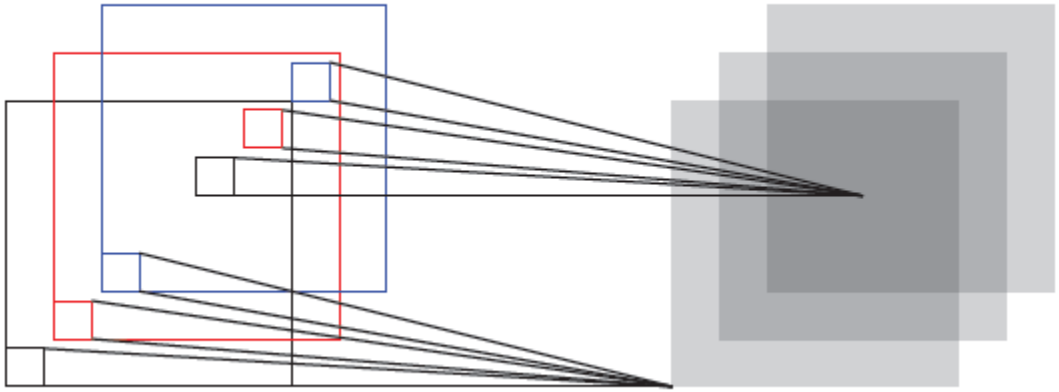
- 为了增强卷积层的表示能力，可以使用*K*个不同的滤波器来得到*K*组输出，即K个特征映射。




每个滤波器的参数随机初始化，滤波器个数是网络中重要的参数

# 卷积层作用

- 提取一个局部区域的特征，每一个滤波器相当于一个特征提取器。



# 子采样层：池化操作

- 虽然卷积层通过局部连接和权重共享显著减少了连接的个数，但是每一个特征映射的维度并没有显著减少。
- 这样，如果后面接一个分类器，分类器的输入维数依然很高，很容易出现过拟合。
- 为了解决这个问题，卷积神经网络一般会在卷积层之后再加一个池化（Pooling）操作，即子采样（Subsampling），构成一个子采样层。通过子采样显著降低特征维数，避免过拟合。

# 子采样层

- 对于卷积层得到的一个特征映射$X^{(l)}$，则采样层输出有：

$$X^{(l+1)} = f(w^{(l+1)} \cdot down(X^{(l)}) + b^{(l+1)})$$

- 其中，$w^{(l+1)}$和 $b^{(l+1)}$分别是可训练的权重和偏置参数
  $down(X^{(l)})$是指子采样后的特征映射

# 子采样层

- 最大值采样（Maximum Pooling）

$$pool_{max}(X^{(l)}) = \max_{i \in X^{(l)}} X_i^{(l)}$$

- 最小值采样（Minimum Pooling）

$$pool_{min}(X^{(l)}) = \min_{i \in X^{(l)}} X_i^{(l)}$$

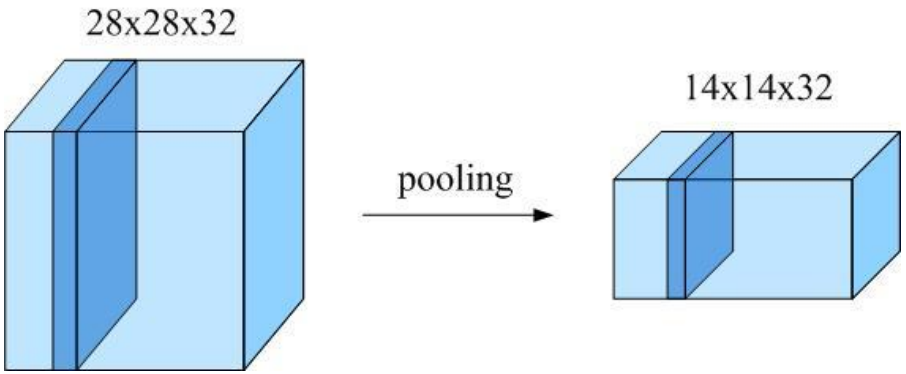
- 平均值（Average Pooling）

$$pool_{avg}(X^{(l)}) = \frac{1}{|X^{(l)}|} \sum_{i \in X^{(l)}}^{|R_k|} X_i^{(l)}$$

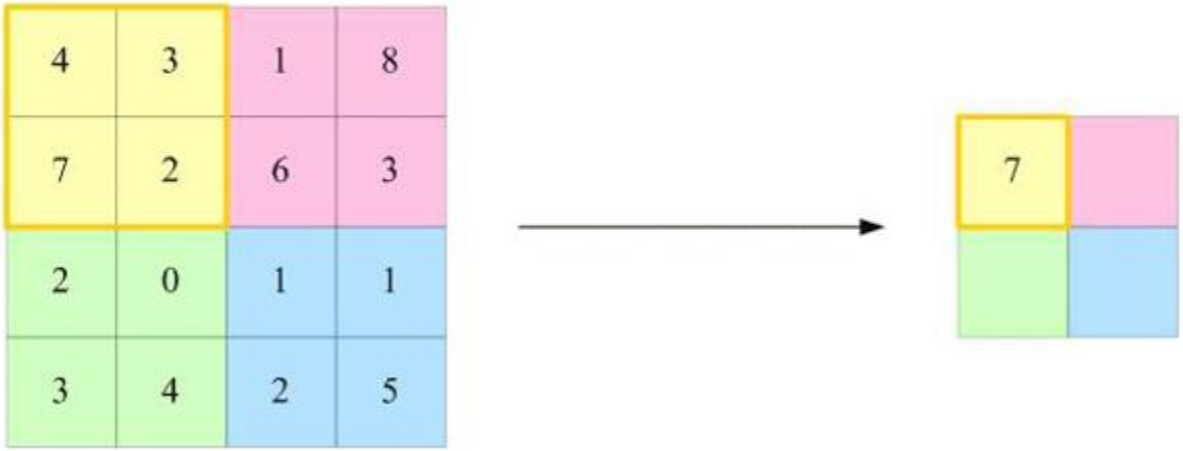
- TopK采样（Average Pooling）

$$pool_k(X^{(l)}) = \operatorname*{top}_{i \in X^{(l)}} X_i^{(l)}$$

# 子采样层示例




采样区域
自行设定



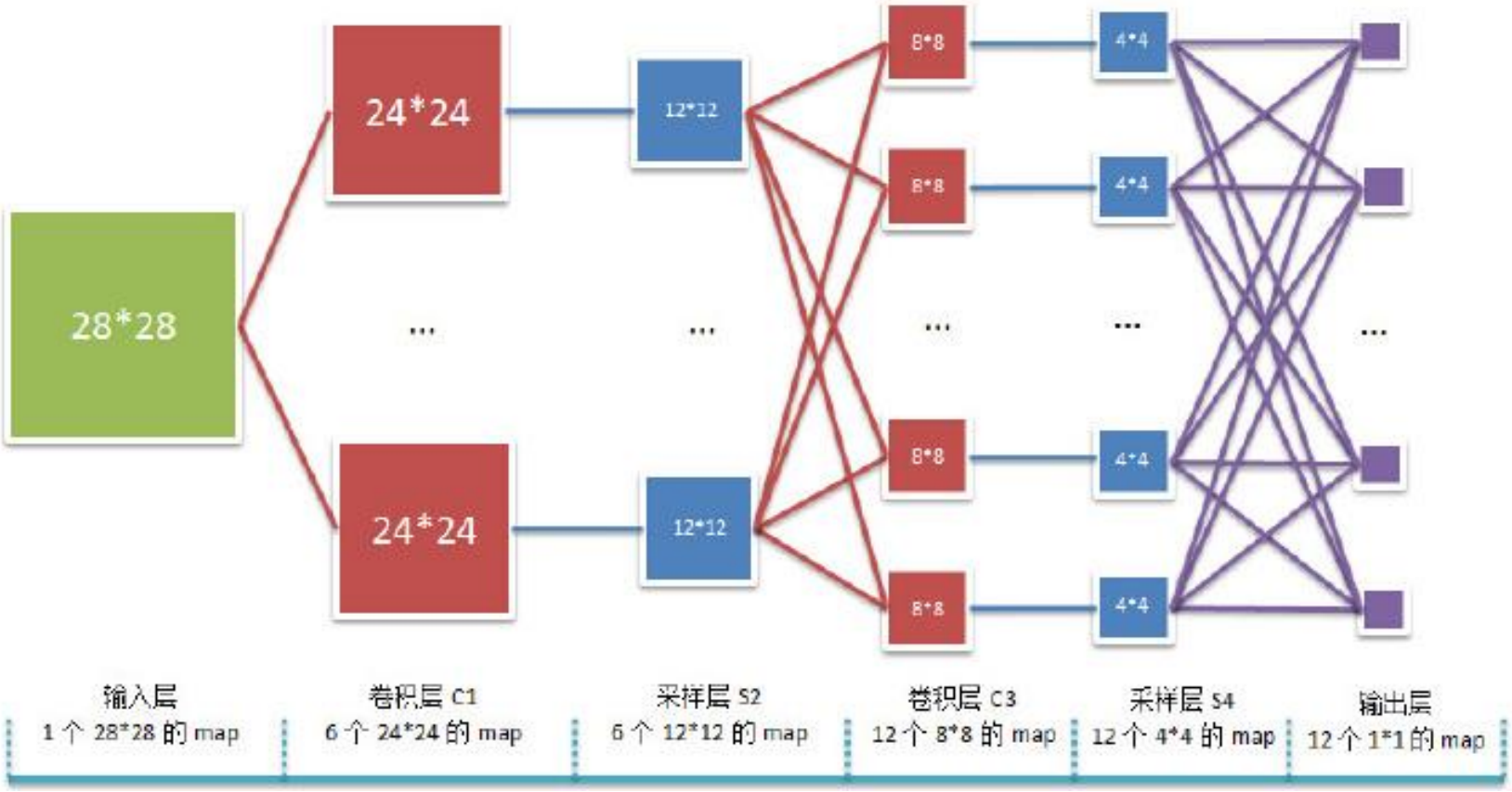
# 参数训练

- 任务：一个滤波器，就是训练W和b

$$y_i = f(Wx_{i:i+m-1} + b) = f(\sum_{k=1}^{m} \boxed{w_k} x_{i+k-1} + \boxed{b})$$

- 采用反向传播算法
  - 选择一个样本 x，信息从输入层经过逐级的变换，传送到输出层
  - 计算该样本的实际输出 o 与相应的理想输出 t 的差
  - 按极小化误差反向传播方法调整权矩阵，也可以调整输入词向量

# 示例


28*28
24*24
24*24
12*12
12*12
8*8
8*8
8*8
8*8
4*4
4*4
4*4
4*4
输入层
1 个 28*28 的 map
卷积层 C1
6 个 24*24 的 map
采样层 S2
6 个 12*12 的 map
卷积层 C3
12 个 8*8 的 map
采样层 S4
12 个 4*4 的 map
输出层
12 个 1*1 的 map

**Softmax层**

# 基于卷积网络的关系抽取
(Zeng et al., Coling 2014)

## 任务：判别句子中实体之间的语义关系

S: 2013年4月20日8时02分四川省雅安市[芦山县]$_{e1}$ 发生了7.0级[地震]$_{e2}$

震中 (e1,e2)

- 语义关系分类

# 基于卷积网络的关系抽取
(Zeng et al., Coling 2014)

## 传统特征提取需要NLP预处理+人工设计的特征

2013年4月20日8时02分四川省雅安市[芦山县]$_{e1}$ 发生了7.0级[地震]$_{e2}$

震中 (e1,e2)

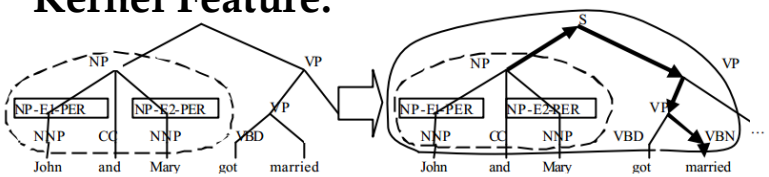
**传统特征：**

**Words：** *芦山县*$_{m11}$, *地震*$_{b1}$, *发生*$_{b2}$, *在*$_{21}$

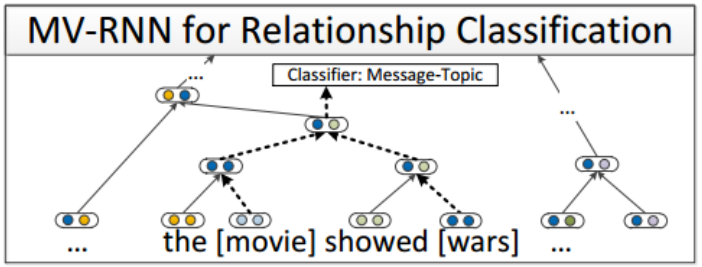
**Entity Type：** *Noun*$_{m1}$ , *Location*$_{m2}$

**Parse Tree：** *Location-VP-PP-Noun*

**Kernel Feature:**




**基于神经网络的特征抽取：**



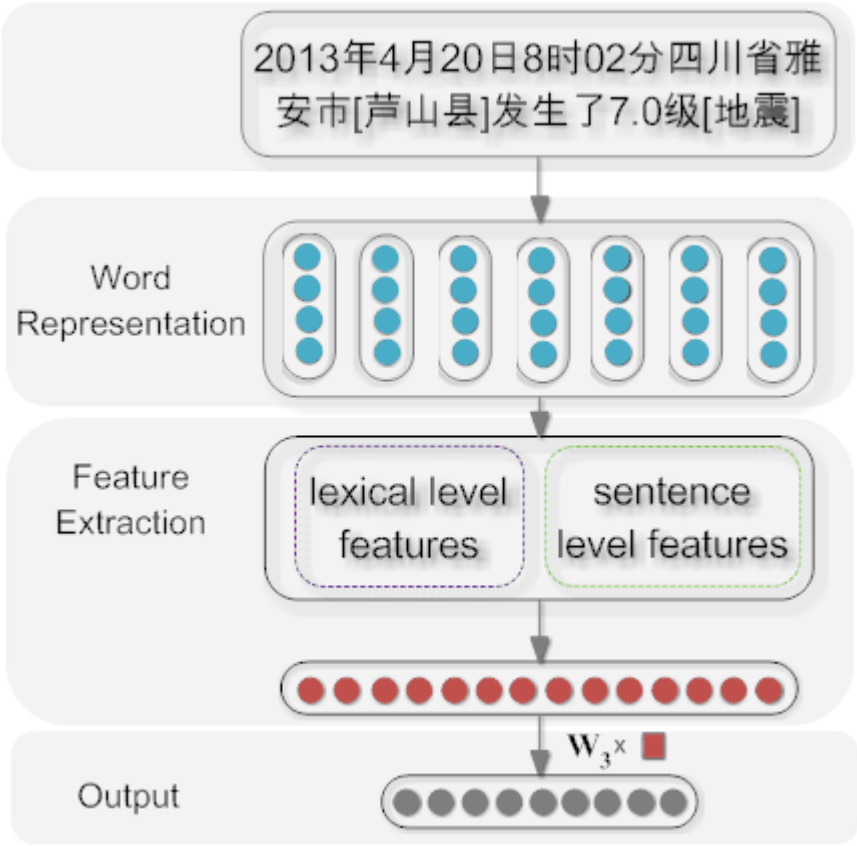

- 问题1：对于缺少NLP处理工具和资源的语言，无法提取文本特征
- 问题2：NLP处理工具引入的“错误累积”
- 问题3：人工设计的特征不一定适合当前任务

# 解决方法

- 基于卷积神经网络的实体关系分类方法
  - 通过CNN学习文本语义特征
  - 不需要人工设计特征

通过Word Embeddings挖掘词汇的语义表示

Lexical Level Features: 实体本身的语义特征
Sentence Level Features: 通过CNN网络挖掘句子级别的文本特征



# 词汇级特征

- 利用词向量（Word Embedding）信息作为词汇级特征

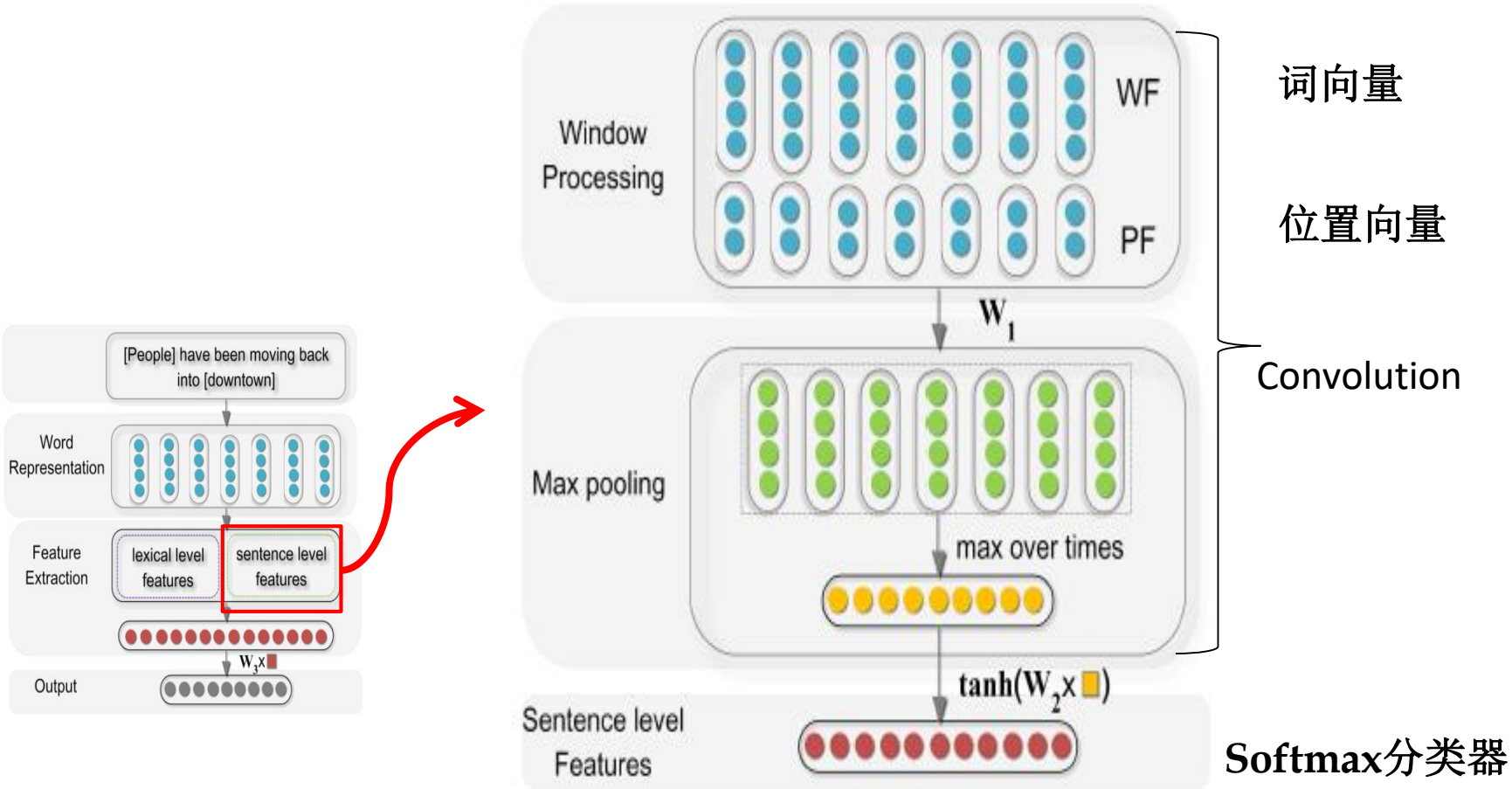
| Features | Remark |
|---|---|
| L1 | Noun 1 |
| L2 | Noun 2 |
| L3 | Left and right tokens of noun 1 |
| L4 | Left and right tokens of noun 2 |
| L5 | WordNet hypernyms of nouns |

The [haft]$_{e1}$ of the [axe]$_{e2}$ is made of yew wood.

(0.5 , 0.2, -0.1, 0.4) , 0.3, -0.1, 0.4) 0.3, 0.1, 0.4)

# 句子级特征

- 利用CNN捕获句子级特征



# 实验结果

| # of training instance | # of test instance | # of relationships |
|---|---|---|
| 8,000 | 2,717 | 19 |

| Classifier | Feature Sets | F1 |
|---|---|---|
| SVM | POS, stemming, syntactic patterns | 60.1 |
| SVM | word pair, words in between | 72.5 |
| SVM | POS, stemming, syntactic patterns, WordNet | 74.8 |
| MaxEnt | POS, morphological, noun compound, thesauri, Google n-grams, WordNet | 77.6 |
| SVM | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | -<br>POS, NER, WordNet, syntactic tree | 74.8<br>77.6 |
| MVRNN | -<br>POS, NER, WordNet, syntactic tree | 79.1<br>82.4 |
| **Proposed** | word pair, WordNet | **82.7** |

# 实验结果

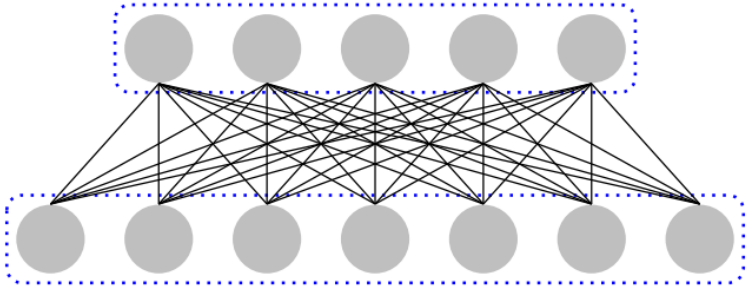
| # of training instance | # of test instance | # of relationships |
|---|---|---|
| 8,000 | 2,717 | 19 |

| | Feature Sets | F1 |
|---|---|---|
| Lexical | L1 | 34.7 |
| | +L2 | 53.1 |
| | +L3 | 59.4 |
| | +L4 | 65.9 |
| | +L5 | 73.3 |
| Sentence | WF | 69.7 |
| | +PF | 78.9 |
| Combination | all | 82.7 |

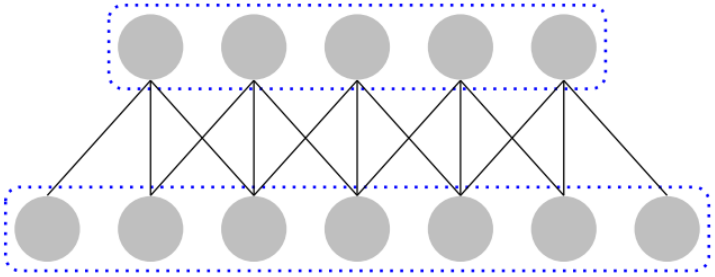
| Classifier | Feature Sets | F1 |
|---|---|---|
| SVM | POS, stemming, syntactic patterns | 60.1 |
| SVM | word pair, words in between | 72.5 |
| SVM | POS, stemming, syntactic patterns, WordNet | 74.8 |
| MaxEnt | POS, morphological, noun compound, thesauri, Google n-grams, WordNet | 77.6 |
| SVM | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | -<br>POS, NER, WordNet, syntactic tree | 74.8<br>77.6 |
| MVRNN | -<br>POS, NER, WordNet, syntactic tree | 79.1<br>82.4 |
| **Proposed** | word pair, WordNet | **82.7** |

# 前馈神经网络的缺点

- 输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据
  - 句子是有长有短，图像大小也不固定
- 连接存在于层与层之间，每层的节点之间是无连接的，假设每次输入都是独立的，无法建模时序或者位置信息



全连接层

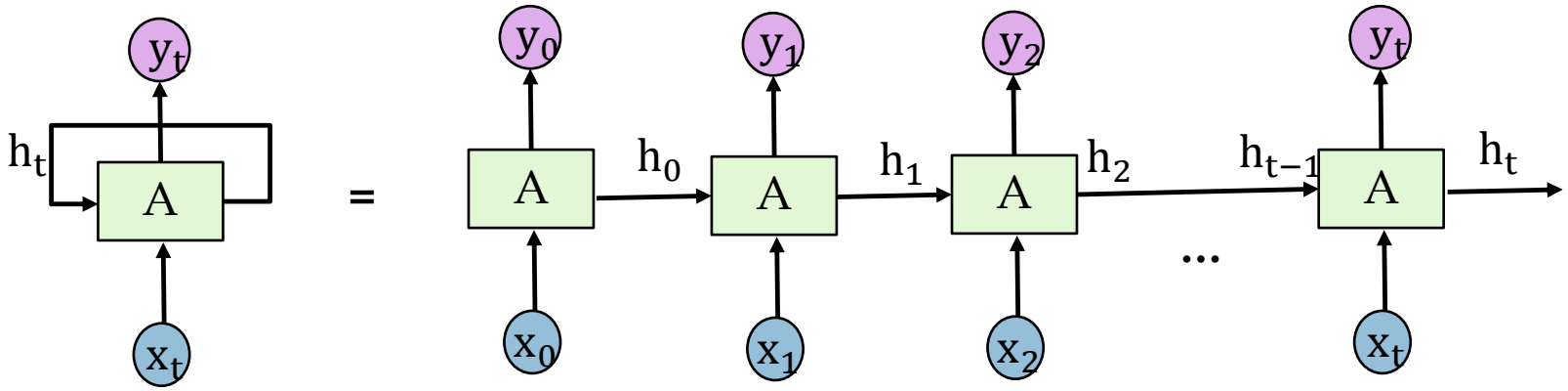


卷积层

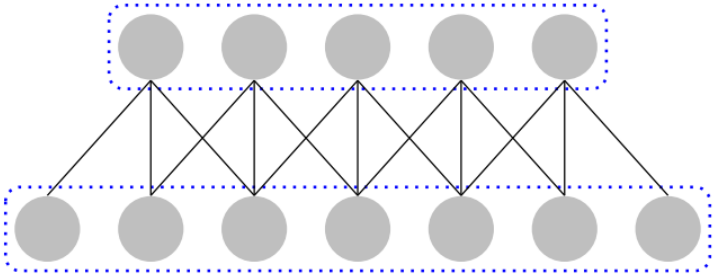
# 循环神经网络

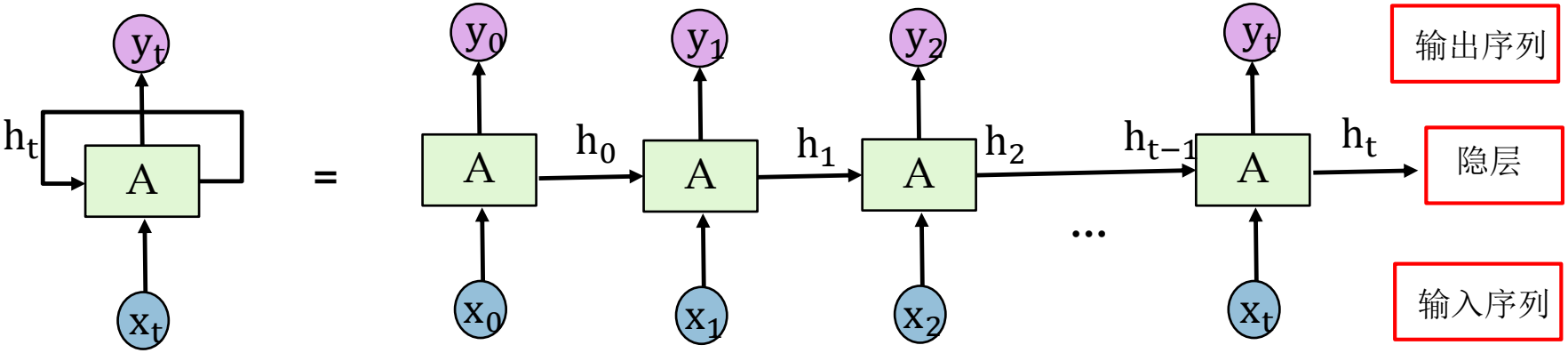
- 是一种反馈神经网络
  - 全部或者部分神经元可以接受来自其它神经元或自身的神经元信号（cf. 前馈神经网络只接受前一层的神经元信号）
  - 可以建立同层神经元的关联，从而每层节点之间有连接，因此可以建模时序或者位置信息，处理变长的序列数据

- 循环神经网络：至少包含一个反馈连接的神经网络结构。因此，网络的激励可以沿着一个loop 进行流动。这种网络结构特别适合于处理时序数据。



# 循环神经网络






- 假设时刻$t$时，输入为$x_t$，隐层状态为$h_t$。 $h_t$不仅和当前时刻的输入相关，也和上一个时刻的隐层状态$h_{t-1}$相关。
- A代表一个RNN单元，由一套参数组成
- 一般使用如下函数：

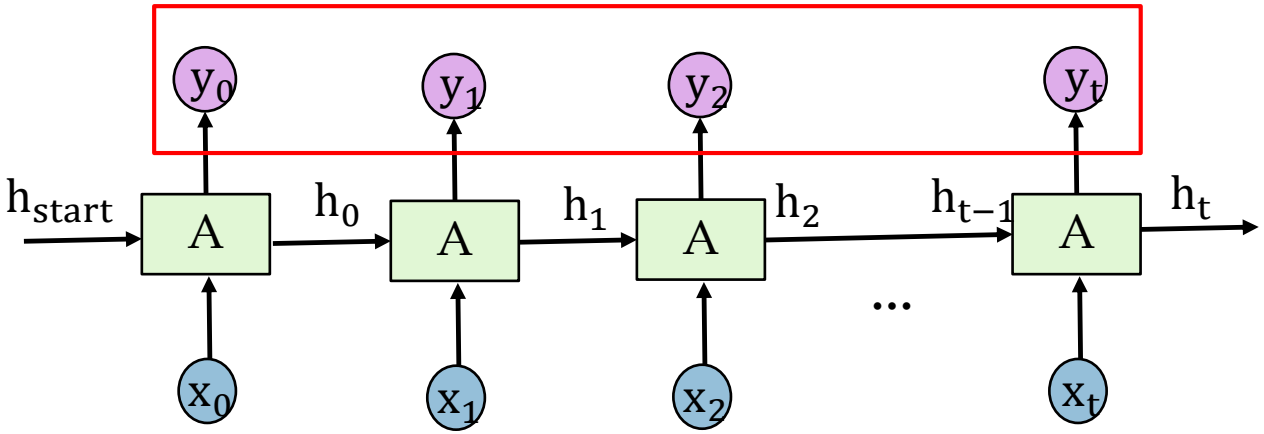
$$\boldsymbol{h}_t = \varphi(\boxed{\boldsymbol{W}}\boldsymbol{x}_t + \boxed{\boldsymbol{U}}\boldsymbol{h}_{t-1} + \boxed{b}) \qquad \boldsymbol{y}_t = softmax(\boxed{V}\boldsymbol{h}_t)$$

这里，$\varphi$是非线性函数，一般为tanh 函数。

- 由上述公式可见，$h_t$的目的是编码从0到t时刻的所有信息

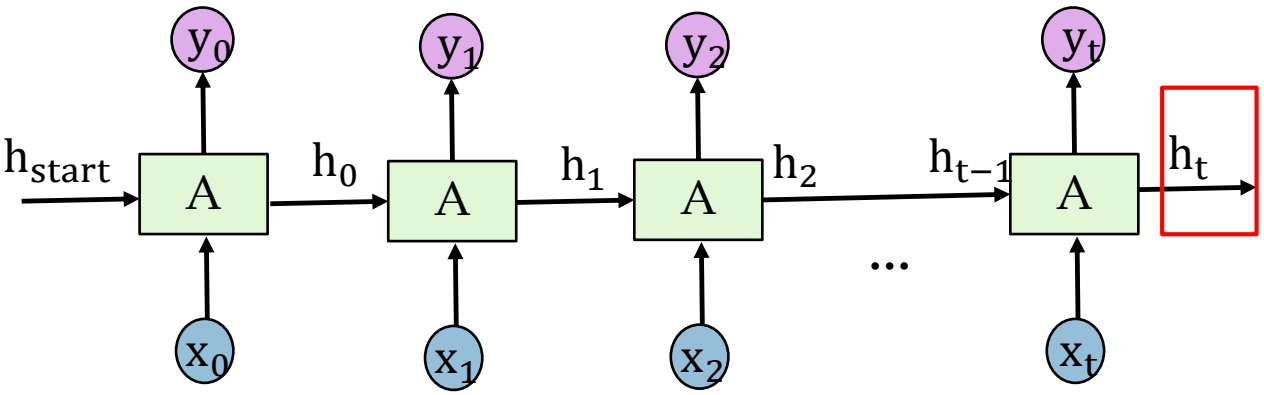
# 循环神经网络：应用场景

- 输入：任意长度的序列
- 输出：同样长度的序列
- 适合：序列标注任务
- 例如：命名实体识别



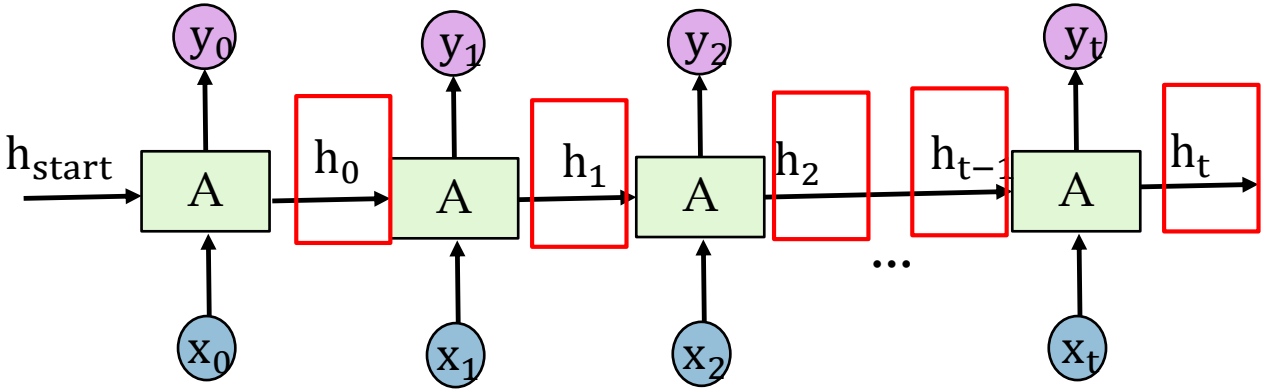
# 循环神经网络：应用场景

- 输入：任意长度的序列
- 输出：关于该序列的一个向量表示
- 适合：输入序列的统一表示
- 应用：文本分类（$h_t$是文本的表示，再加一个分类层）



# 循环神经网络：应用场景

- 输入：任意长度的序列
- 输出：关于该序列的多个向量表示
- 适合：输入序列中每个时刻的表示
- 应用：关系分类
  - $x_t$是句子中的一个词，$h_t$是这个词在当前上下文的表示
  - $h_k$和$h_j$是待分类的两个实体的表示
  - 再加一个分类器完成关系分类



# 参数学习

- 循环神经网络（4个参数），其中U,W,V是矩阵，b是向量

$$\boldsymbol{h}_t = \varphi(\boxed{\boldsymbol{W}}\boldsymbol{x}_t + \boxed{\boldsymbol{U}}\boldsymbol{h}_{t-1} + \boxed{b}) \qquad \boldsymbol{y}_t = softmax(\boxed{V}\boldsymbol{h}_t)$$

- 参数训练可以利用反向传播算法（Backpropagation Through Time，BPTT）
  - 详细参考P. J. Werbos. “Backpropagation through time: what it does and how to do it”. In: Proceedings of the IEEE 10 (1990).

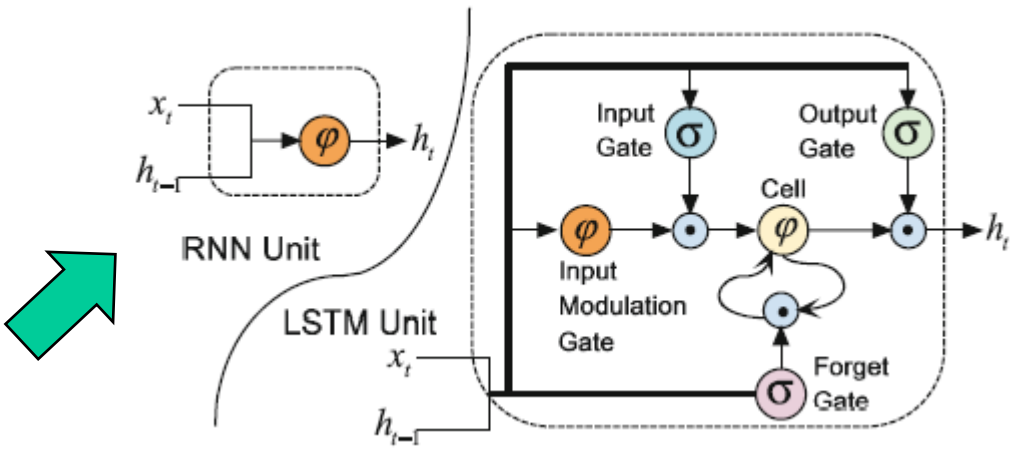
- 问题：序列长度过长时会导致梯度爆炸或者梯度消失问题

# 长短时记忆神经网络：LSTM

- 长短时记忆神经网络：Long Short-Term Memory Neural Network，LSTM）
- RNN的一个变体，可以有效地解决简单循环神经网络的梯度爆炸或消失问题
- 关键：引入了一组记忆单元（Memory Units），允许网络可以学习何时遗忘历史信息，何时用新信息更新记忆单元。具体地，
  - 在时刻 t ，记忆单元 $\mathbf{c}_t$ 记录了到当前时刻为止的所有历史信息，并受三个“门”控制：输入门 $\mathbf{i}_t$，遗忘门$\mathbf{f}_t$和输出门$\mathbf{o}_t$。三个门的元素的值在 [0, 1] 之间。

# 长短时记忆神经网络

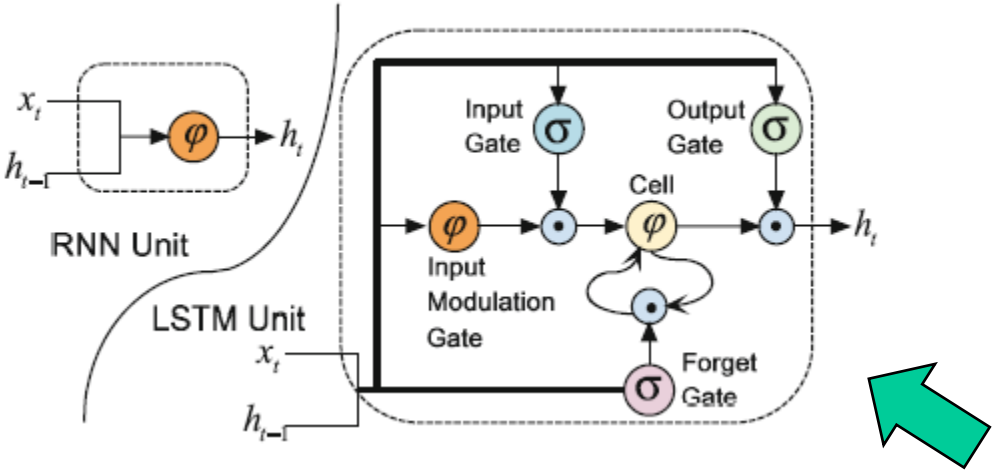
## RNN vs LSTM




RNN： $h_t = \varphi(W_h x_t + U_h h_{t-1} + b_h)$

# 长短时记忆神经网络

## RNN vs LSTM



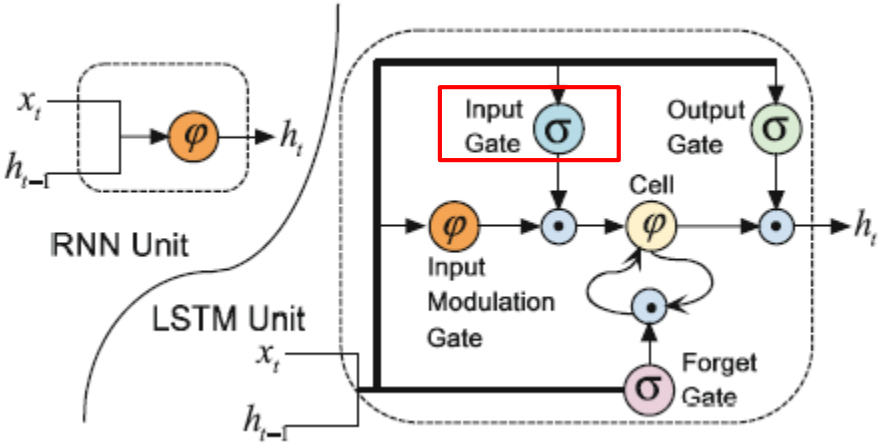

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

# 长短时记忆神经网络

## RNN vs LSTM

$$\boldsymbol{h}_t = \varphi(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1} + b)$$



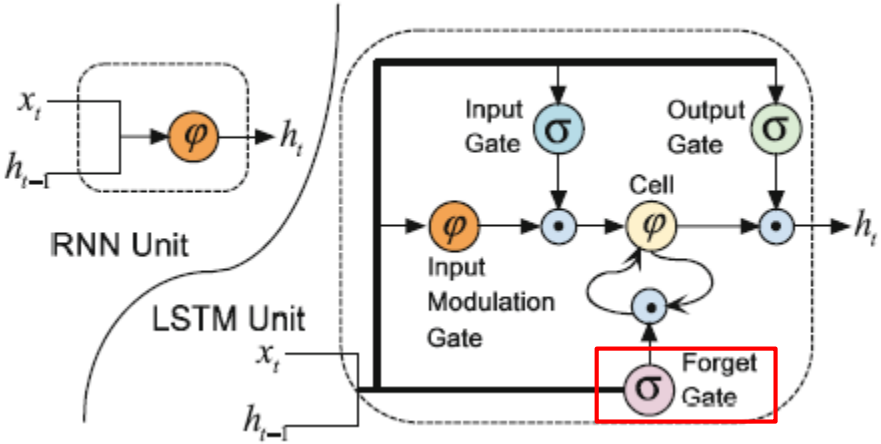

LSTM：

$$\boxed{i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_t)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$u_t = \varphi(W_u x_t + U_u h_{t-1} + b_u)$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot \varphi(c_t)$$

输入门 $\mathbf{i}_t$是一个0-1向量，控制每一个记忆单元加入多少新的信息

# 长短时记忆神经网络

## RNN vs LSTM



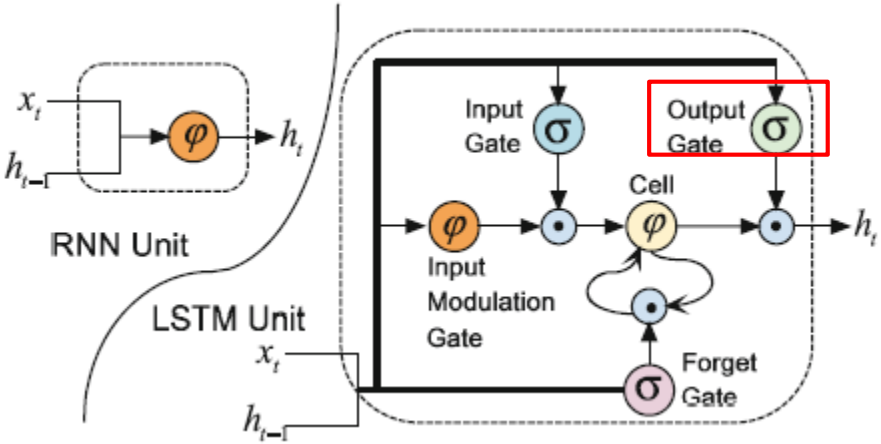

LSTM：

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_t)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$u_t = \varphi(W_u x_t + U_u h_{t-1} + b_u)$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot \varphi(c_t)$$

遗忘门 $\mathbf{f}_t$也是一个0-1向量，控制每一个记忆单元需要遗忘多少信息

# 长短时记忆神经网络

## RNN vs LSTM



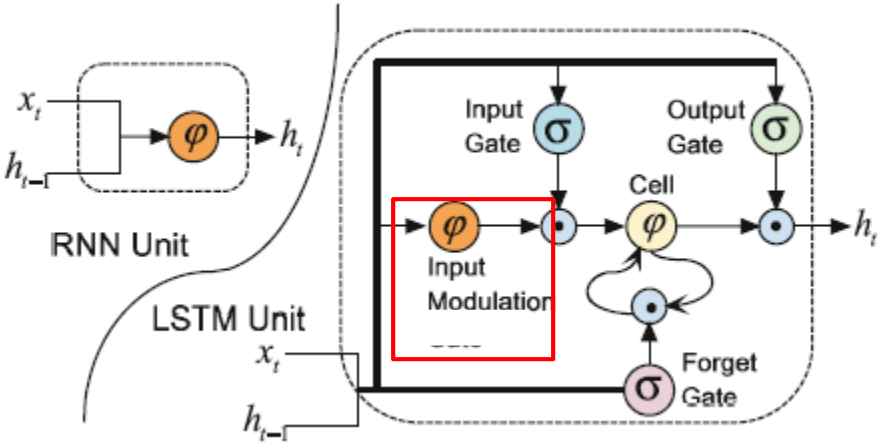

LSTM：

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_t)$$
$$\boxed{o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)}$$
$$u_t = \varphi(W_u x_t + U_u h_{t-1} + b_u)$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot \varphi(c_t)$$

输出门$\mathbf{o}_t$也是一个0-1向量，控制每一个记忆单元输出多少信息。

# 长短时记忆神经网络

## RNN vs LSTM



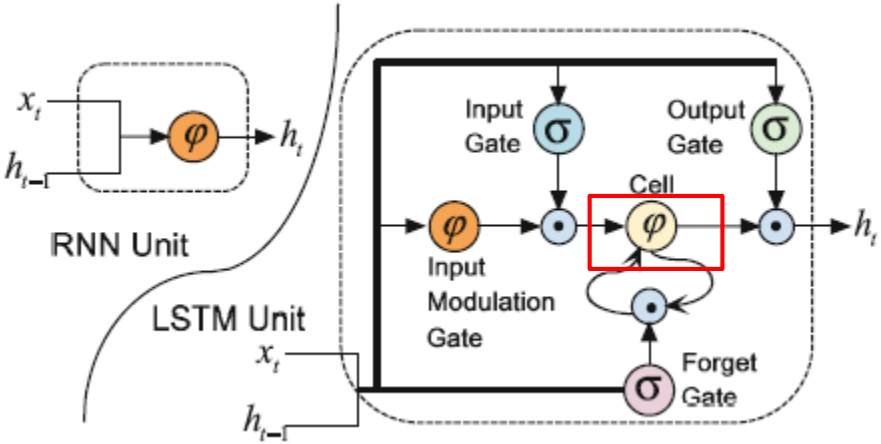

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

输入控制$\mathbf{u}_t$（不是0-1向量）是t时刻当前信息的编码，和输入门$\mathbf{i}_t$共同决策当前输入对记忆单元 $\mathbf{c}_t$ 的贡献

# 长短时记忆神经网络

## RNN vs LSTM

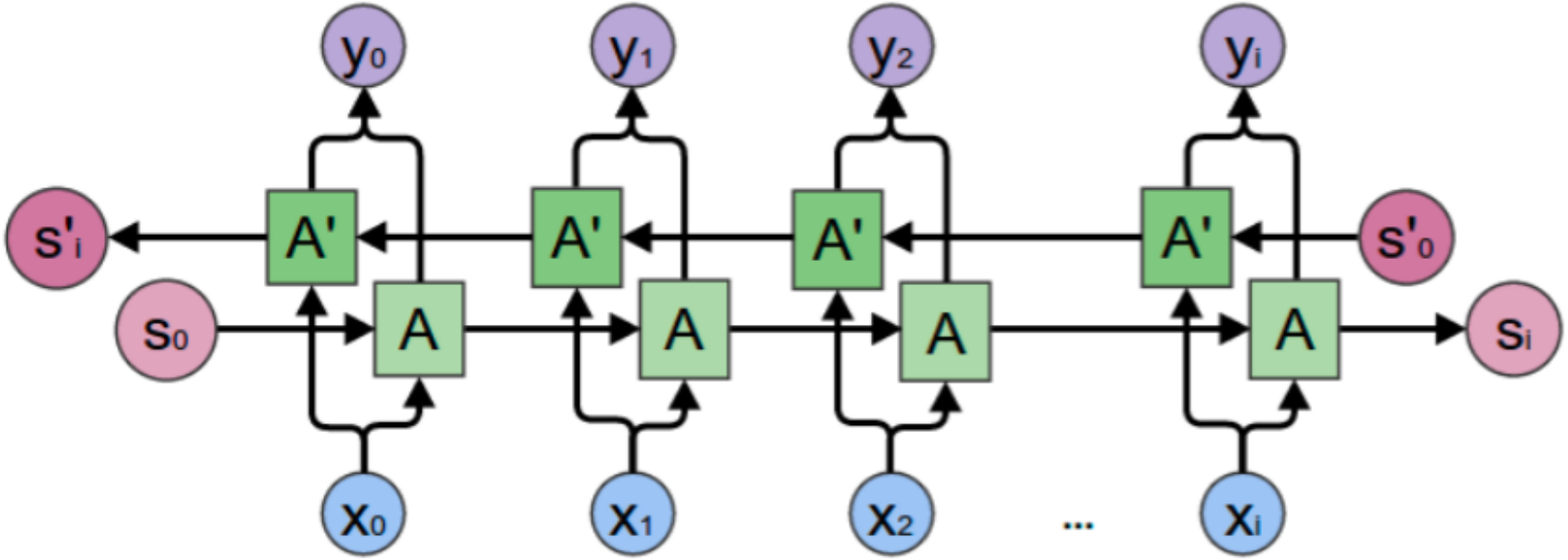



LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

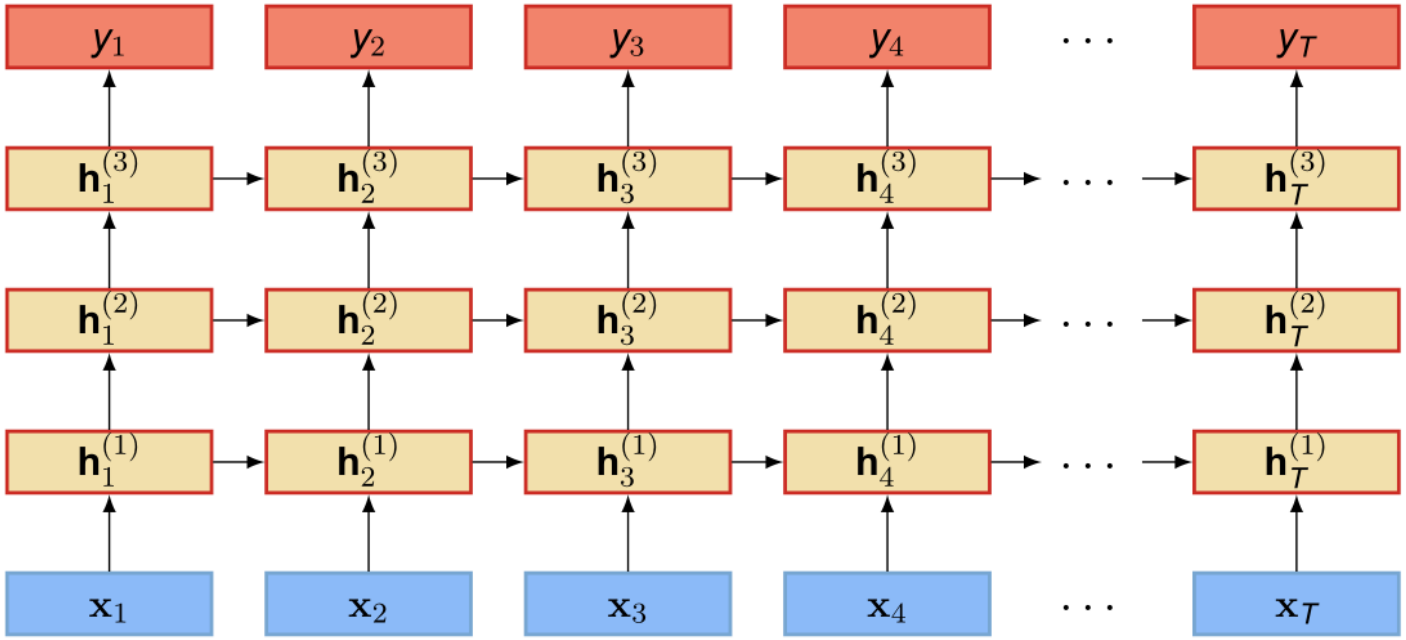
记忆单元 $\mathbf{c}_t$ 记录了到当前时刻为止的所有历史信息

# 双向循环神经网络




A和A′可以为任意具有循环结构的网络，例如：RNN,LSTM,GRU(只有更新门和重置门)

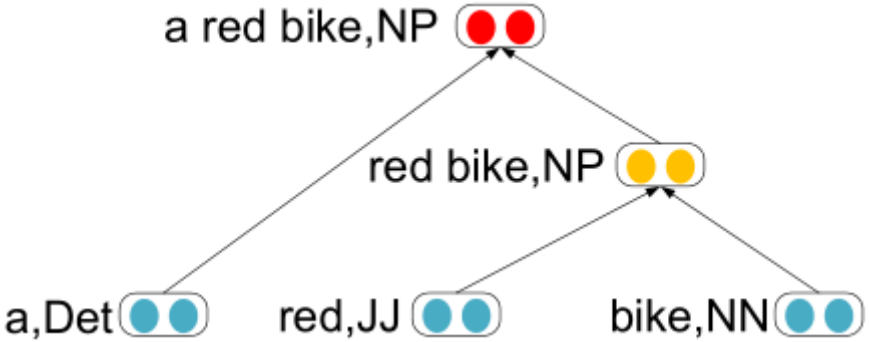
# 层叠(Stack)循环神经网络




本质就是多个循环网络逐层使用

# 递归神经网络

- 递归神经网络：Recursive Neural Network (RecNN)



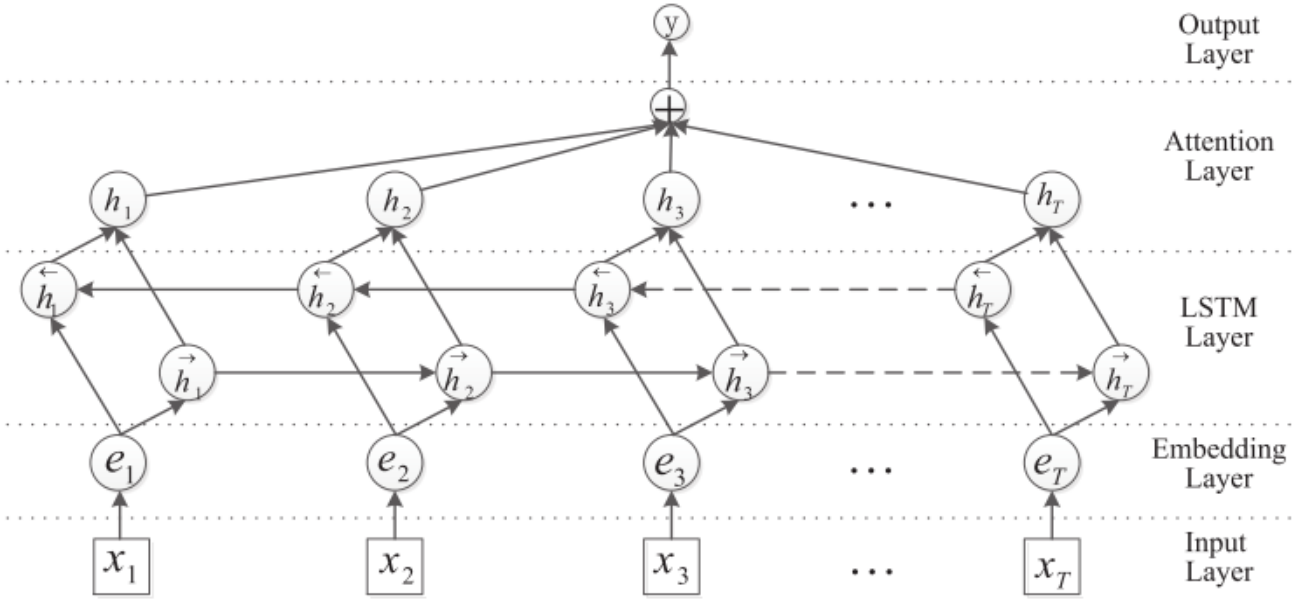

根据一个给定的拓扑结构，
进行词语的语义组合。

# 利用LSTM做关系抽取（Zhou ACL2016）

- 动机：
  - 句子级语义信息对关系抽取任务至关重要
  - 句子中的不同位置的词有不同程度的重要性
  - 传统方法依赖NLP工具，人工提取特征，造成错误传递
  - CNN的方法难以建模句子中长距离的依赖关系

# 利用LSTM做关系抽取（Zhou ACL2016）

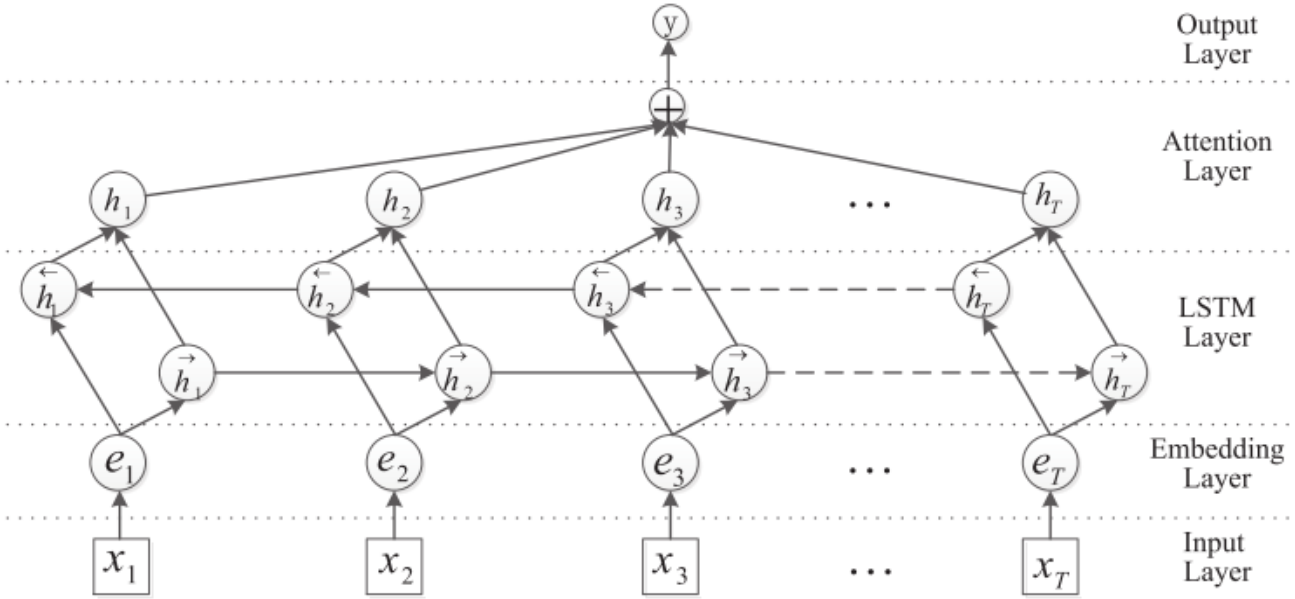
## 方法：




- 利用双向LSTM编码每个词的表示
- $\overleftarrow{h_1}, \overleftarrow{h_2}, \dots, \overleftarrow{h_T} = \overleftarrow{LSTM(x_1, \dots, x_T)}$
- $\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_T} = \overrightarrow{LSTM(x_1, \dots, x_T)}$
- $h_i = [\overleftarrow{h_2} \otimes \overrightarrow{h_2}]$（$\otimes$ 是对应位置求和）

# 利用LSTM做关系抽取（Zhou ACL2016）

- 方法：




- 关注机制（通过对隐层的softmax得到关注的权重）
- $H = [h_1, h_2, \ldots, h_T]$ , $M = \tanh(H)$
- $\alpha = softmax(\omega^T M)$ , $r = H\alpha^T$
- $h^* = \tanh(r)$

# 利用LSTM做关系抽取（ Zhou ACL2016 ）

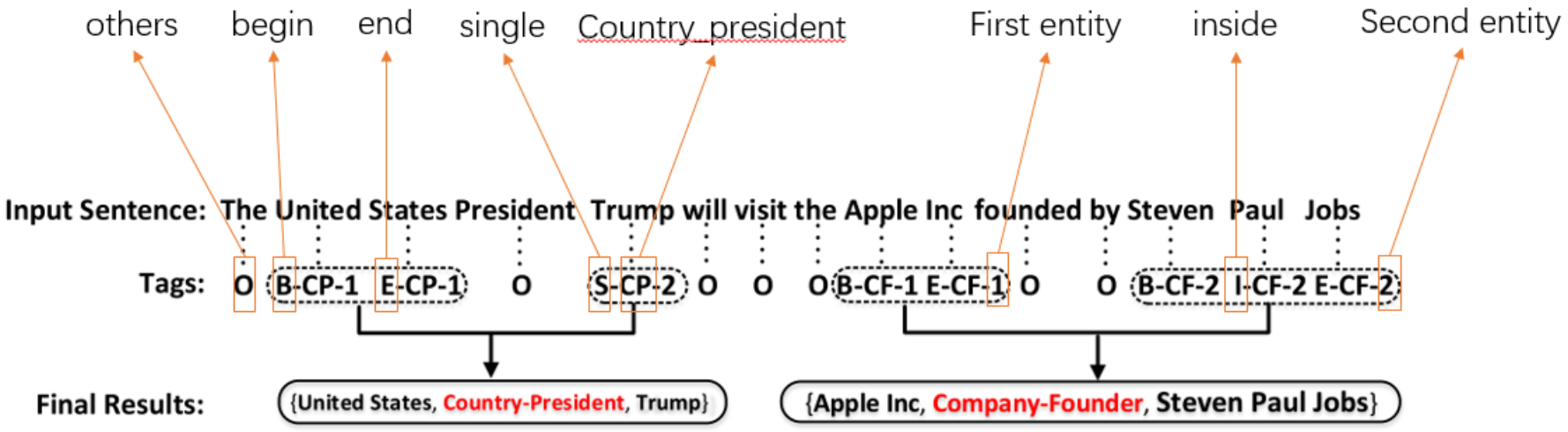
## 实验结果：

| Model | Feature Set | F1 |
|---|---|---|
| SVM<br>(Rink and Harabagiu, 2010) | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FramNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| CNN<br>(Zeng et al., 2014) | WV (Turian et al., 2010) (dim=50)<br>+ PF + WordNet | 69.7<br>82.7 |
| RNN<br>(Zhang and Wang, 2015) | WV (Turian et al., 2010) (dim=50) + PI<br>WV (Mikolov et al., 2013) (dim=300) + PI | 80.0<br>82.5 |
| SDP-LSTM<br>(Yan et al., 2015) | WV (pretrained by word2vec) (dim=200), syntactic parse<br>+ POS + WordNet + grammar relation embeddings | 82.4<br>83.7 |
| BLSTM<br>(Zhang et al., 2015) | WV (Pennington et al., 2014) (dim=100)<br>+ PF + POS + NER + WNSYN + DEP | 82.7<br>84.3 |
| BLSTM<br>Att-BLSTM<br>BLSTM<br>Att-BLSTM | WV (Turian et al., 2010) (dim=50) + PI<br>WV (Turian et al., 2010) (dim=50) + PI<br>WV (Pennington et al., 2014) (dim=100) + PI<br>WV (Pennington et al., 2014) (dim=100) + PI | 80.7<br>82.5<br>82.7<br>84.0 |

Table 1: Comparison with previous results. WV, PF, PI stand for word vectors, position features and position indicators respectively.

# 实体和关系联合抽取（Zheng ACL2017）

- 动机：
  - 真实世界中很多情况下要实体和关系联合抽取
  - 传统的方法大多数都假定给出了待抽取的两个实体



# 实体和关系联合抽取（Zheng ACL2017）

- 方法：
  - 对每个关系，将其与（Begin，Inside，End，Single）以及表示第几个实体的序号（1，2）组合，一共构成八个标签
  - 额外考虑一个Other标签，表示不属于任何一个关系
  - 如果总共有$|R|$个关系，那么一共有$2*4*|R|+1$个标签
  - 根据标注结果得到三元组
  - 标注方法尝试了
    - LSTM+CRF
    - LSTM+LSTM
    - LSTM+LSTM+bias

# 实体和关系联合抽取（ Zheng ACL2017 ）

- 实验：
  - 数据集：使用弱监督的NYT数据集，并且将其视为监督数据。训练集包括353k个三元组，测试集包括3880个三元组，一个24个关系。
  - 实验结果

| Elements | E1 | | | E2 | | | (E1,E2) | | |
|---|---|---|---|---|---|---|---|---|---|
| PRF | *Prec.* | *Rec.* | *F1* | *Prec.* | *Rec.* | *F1* | *Prec.* | *Rec.* | *F1* |
| LSTM-CRF | **0.596** | 0.325 | 0.420 | 0.605 | 0.325 | 0.423 | **0.724** | 0.341 | 0.465 |
| LSTM-LSTM | 0.593 | 0.342 | 0.434 | **0.619** | 0.334 | 0.434 | 0.705 | 0.340 | 0.458 |
| LSTM-LSTM-Bias | 0.590 | **0.479** | **0.529** | 0.597 | **0.451** | **0.514** | 0.645 | **0.437** | **0.520** |

# 传统关系抽取方法

- 基于特征向量方法 vs. 基于核函数方法 vs.基于神经网络方法

| | 基于特征向量方法 | 基于核函数方法 | 基于神经网络方法 |
|---|---|---|---|
| 优点 | 简单实用<br>计算速度较快 | 能够有效挖掘结构化信息 | 人工干预小，可扩性好，适用于大规模数据环境 |
| 缺点 | 难以进一步挖掘有效的平面特征，性能很难进一步提高 | 句法分析的错误引入了噪声，同时由于树核的计算速度非常慢，很难开发实用系统 | 可解释性差<br>训练时间长 |

# 小结

- 传统的**人工标注语料+机器学习算法**模式无法满足开放域开放语料下的信息抽取
  - 语料构建成本过高
  - 跨领域跨文本类别时抽取性能严重下降
  - 需要抽取的信息类别通常未预先指定

- 需要开放域关系抽取
  - 关系类型更多
  - 语料规模不受限
  - 抽取实体类型也更丰富

# 概述

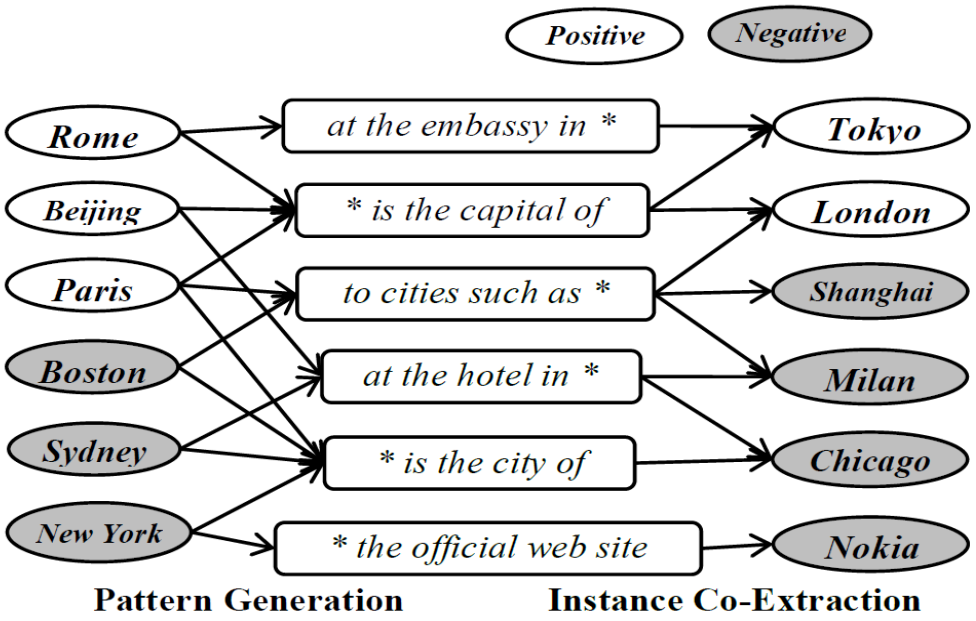
- 引言
- 实体识别与抽取
- 实体消歧
- 关系抽取和发现
  - 关系抽取任务定义
  - 传统关系抽取
  - 开放域关系抽取
  - 开放域关系发现
- 事件抽取
- 问题与挑战

# 开放域关系抽取特点

- 不限定关系类别
- 不限定目标文本
  - Web Page
  - Wikipedia
  - Query Log
- 难点问题
  - 如何获取训练语料
  - 如何获取实体关系类别
  - 如何针对不同类型目标文本抽取关系
- 需要研究新的抽取方法
  - 按需抽取—Bootstrapping，模板
  - 开放抽取—Open IE
  - 知识监督抽取—Distant Supervision

# 按需抽取：Bootstrapping

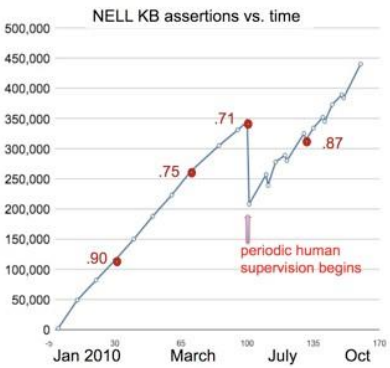
- Bootstrapping：模板生成->实例抽取->迭代直至收敛
- 语义漂移问题：迭代会引入噪音实例和噪音模板
  - 首都：*Rome* → 城市模板 *" * is a city of "*
    - Paris is a city of France.
    - Paris is a city of Romance.
- COLING 14：引入负实例来限制语义漂移



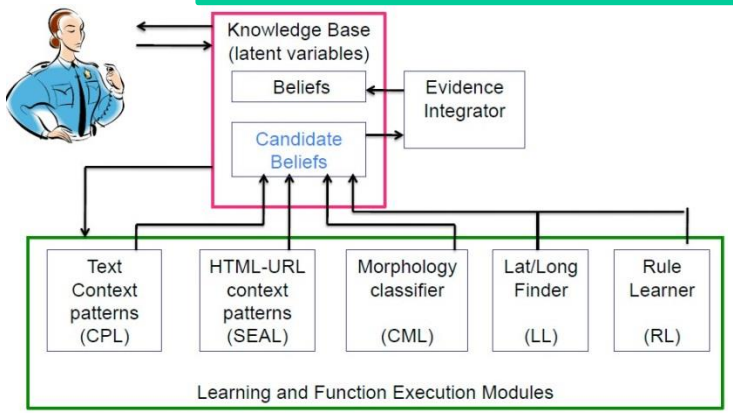
# Read the Web : NELL(Never-Ending Language Learning) ( Tom Mitchell AAAI2009 )

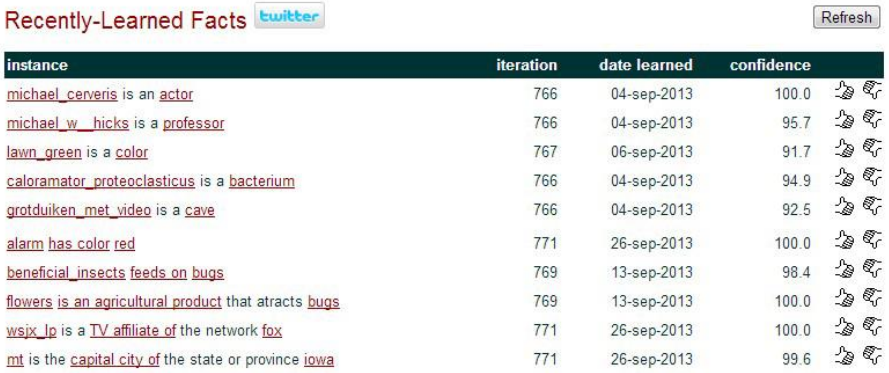
Research Project at Carnegie Mellon University

- 输入
  - Initial ontology
  - 500 million web pages
- 目标
  - Extract new instances of categories and relations
  - Learn to read better than yesterday




1,994,282 Instances of 874 different categories and relations



Recently-Learned Facts twitter Refresh

| instance | iteration | date learned | confidence |
|---|---|---|---|
| michael_cerveris is an actor | 766 | 04-sep-2013 | 100.0 |
| michael_w__hicks is a professor | 766 | 04-sep-2013 | 95.7 |
| lawn_green is a color | 767 | 06-sep-2013 | 91.7 |
| caloramator_proteoclasticus is a bacterium | 766 | 04-sep-2013 | 94.9 |
| grotduiken_met_video is a cave | 766 | 04-sep-2013 | 92.5 |
| alarm has color red | 771 | 26-sep-2013 | 100.0 |
| beneficial_insects feeds on bugs | 769 | 13-sep-2013 | 98.4 |
| flowers is an agricultural product that atracts bugs | 769 | 13-sep-2013 | 100.0 |
| wsjx_lp is a TV affiliate of the network fox | 771 | 26-sep-2013 | 100.0 |
| mt is the capital city of the state or province iowa | 771 | 26-sep-2013 | 99.6 |

# 开放抽取

- 通过识别表达语义关系的短语来抽取实体之间的关系
  - *(华为，总部位于, 深圳），(华为，总部设置于，深圳)，(华为，将其总部建于，深圳)*
- 同时使用句法和统计数据来过滤抽取出来的三元组
  - 关系短语应当是一个以动词为核心的短语
  - 关系短语应当匹配多个不同实体对
- 优点：无需预先定义关系类别
- 缺点：语义没有归一化，同一关系有不同表示

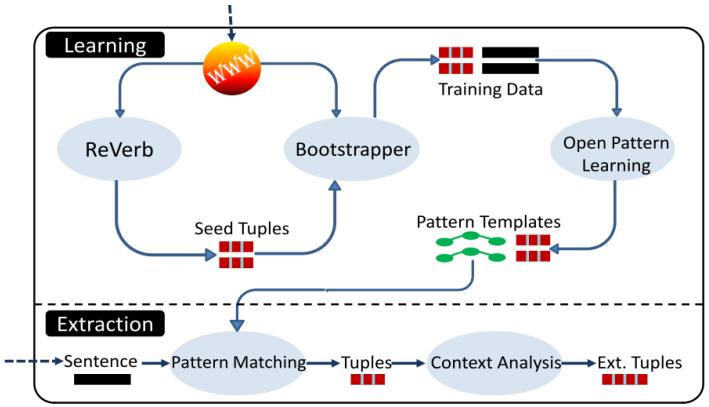
$$V \mid VP \mid VW^*P$$

$V$ = verb particle? adv?

$W$ = (noun | adj | adv | pron | det)

$P$ = (prep | particle | inf. marker)

**关系短语的句法结构约束**

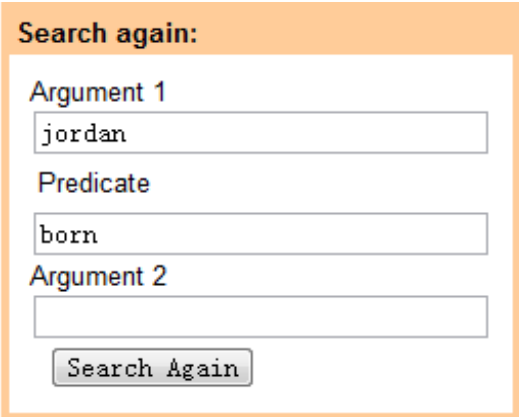


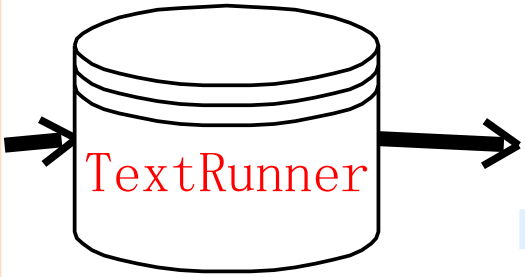
# 开放域关系抽取: Web Page

(Banko IJCAI 2007) (1/2)

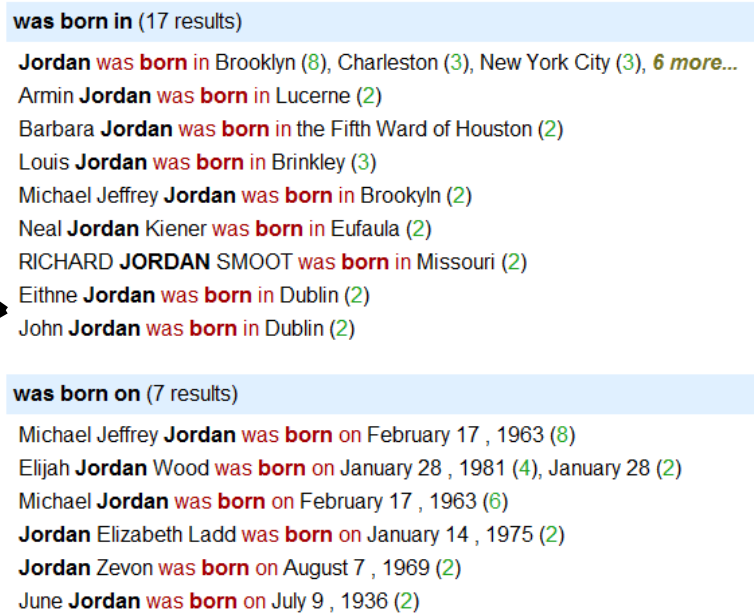
- TextRunner
  - 用户输入特定的谓词和论元
  - 利用搜索引擎返回与用户输入相关的句子
  - 用TextRunner抽取出谓词论元三元组的三元组

Input



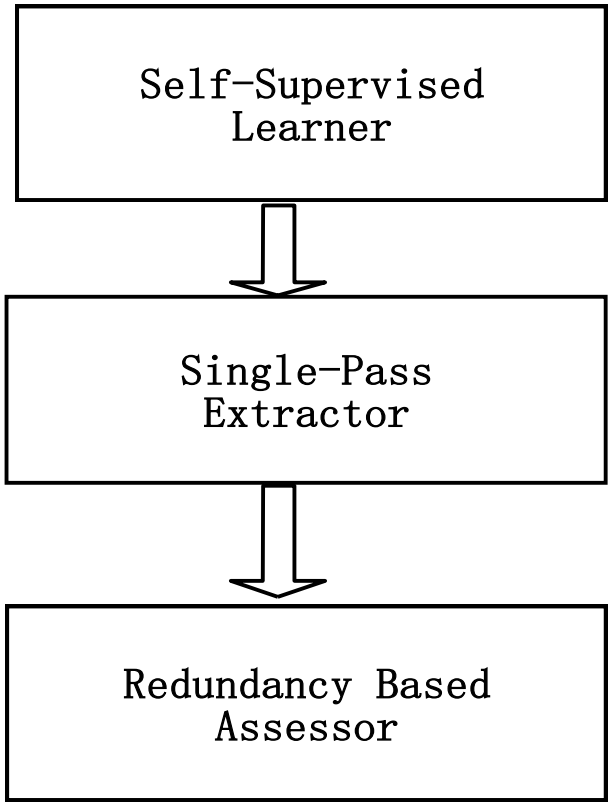

Output



# 开放域关系抽取: Web Page

(Banko IJCAI 2007) (1/2)

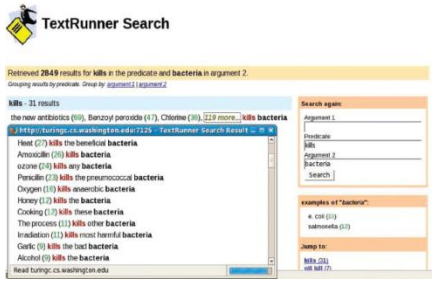
- 步骤：
  - 离线的训练集产生：利用简单的启发式规则，在宾州树库上产生训练语料
  - 离线的分类器训练：提取一些浅层句法特征，训练分类器，用来判断一个元组是否构成关系
  - 在线关系抽取：在网络语料上，找到候选句子，提取浅层句法特征，利用分类器，判断抽取的关系对是否“可信”
  - 在线的关系可信度评估：利用网络海量语料的冗余信息，对可信的关系对，进行评估
- 出发点：
  - 关系类别的产生：动词作为关系类别
  - 训练语料的产生：通过句法关系引出语义关系

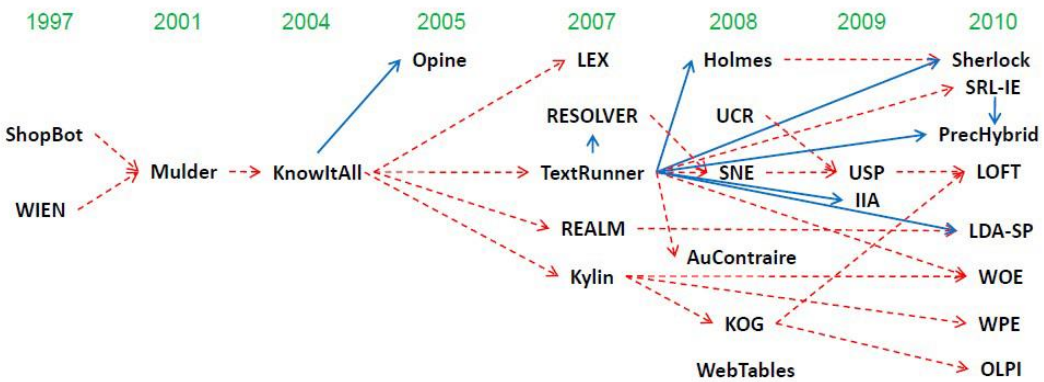


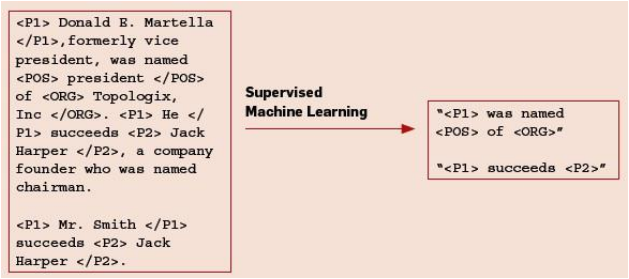
Turing Center
Investigating problems at the cross

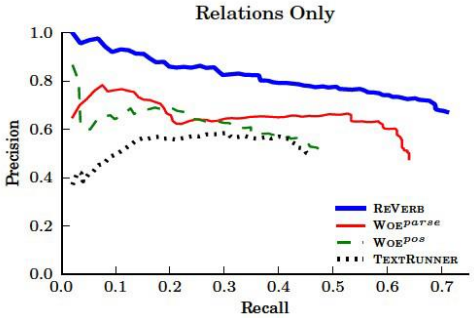
# Machine Reading

- TextRunner、ReVerb、WOE、OLLIE
  - 从Wikipedia Infobox获得关系名
  - 通过在句法树上回标获得句法关系模板



# 开放域关系抽取: Wikipedia (Wu CIKM 2007 )

- 任务：在Wikipedia文本中抽取关系（属性）信息
- 难点
  - 无法确定关系类别
  - 无法获取训练语料
- 方法
  - 在Infobox抽取关系信息
  - 在Wikipedia条目文本中进行回标，产生训练语料

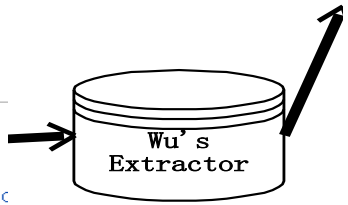
Input

Tsinghua University

From Wikipedia, the free encyclopedia
(Redirected from Qinghua)

*For the university in Taiwan, see National Tsing Hua University.*

**Tsinghua University** (**THU**; simplified Chinese: 清华大学; traditional C
Chinese as **Qinghua**, is a university in Beijing, China. The school is o
1911 under the name "Tsinghua Xuetang" or "Tsinghua College" and was re
was founded in 1925 and the name "National Tsinghua University" was ado
*Commitment*, Tsinghua University describes itself as being dedicated to
global development.[1] Tsinghua is often ranked as the first or second t
rankings.[2][3][4]




Output

Tsinghua University
清华大学

| | |
|---|---|
| Motto | 自强不息，厚德载物 |
| Motto in English | Self-discipline and Social Commitment |
| Established | 1911 |
| Type | Public |
| President | Gu Binglin |
| Academic staff | 2,857 |
| Undergraduates | 13,915 |
| Postgraduates | 12,831 |
| Location | Beijing, People's Republic of China |
| Campus | Urban, 395 ha (3.95 km²) |
| Flower | Redbud and Lilac |
| Colors | Purple and White |
| Affiliations | AEARU, APRU, C9 |
| Website | www.tsinghua.edu.cn |

# 开放域关系抽取: Wikipedia (Wu CIKM 2007)

| Clearfield County, Pennsylvania | |
|---|---|
| **Statistics** | |
| Founded | March 26, 1804 |
| Seat | Clearfield |
| Area | |
| - Total | 2,988 km² (1,154 mi²) |
| - Land | sq mi ( km²) |
| - Water | 17 km² (6 mi²), 0.56% |
| Population | |
| - (2000) | 83,382 |
| - Density | 28/km² |

Clearfield County was created on 1804 from parts of Huntingdon and Lycoming Counties but was administered as part of Centre County until 1812.
Its county seat is Clearfield.

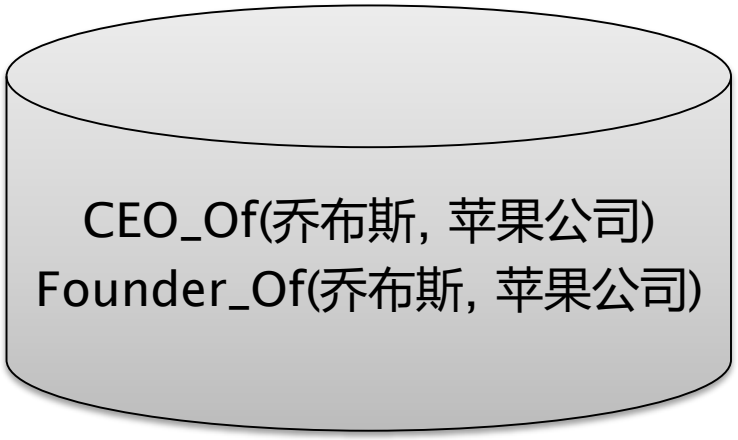
2,972 km² (1,147 mi²) of it is land and 17 km² (7 mi²) of it (0.56%) is water.

As of 2005, the population density was 28.2/km².

# 知识监督开放抽取：Distant Supervision

- 开放域信息抽取的一个主要问题是缺乏标注语料
- **Distant Supervision**: 使用知识库中的关系启发式的标注训练语料

知识库




标注训练语料

| Relation Instance | Label |
|---|---|
| ***S1:*** *乔布斯是苹果公司的创始人之一* | Founder-of, CEO-of |
| ***S2:*** *乔布斯回到了苹果公司* | Founder-of, CEO-of |

# 简单远距离监督方法

(Mintz et al.，ACL09)

- **DS假设**: 每一个同时包含两个实体的句子都会表述这两个实体在知识库中的对应关系
- 基于上述假设标注所有句子作为训练语料
- 使用最大熵分类器来构建IE系统
- 最大的问题：噪音训练实例

噪音训练实例

| Relation Instance | Label | |
|---|---|---|
| *S1:乔布斯是苹果公司的创始人之一* | Founder-of | √ |
| *S1:乔布斯是苹果公司的创始人之一* | CEO-of | x |
| *S2:乔布斯回到了苹果公司* | Founder-of | x |
| *S2:乔布斯回到了苹果公司* | CEO-of | x |

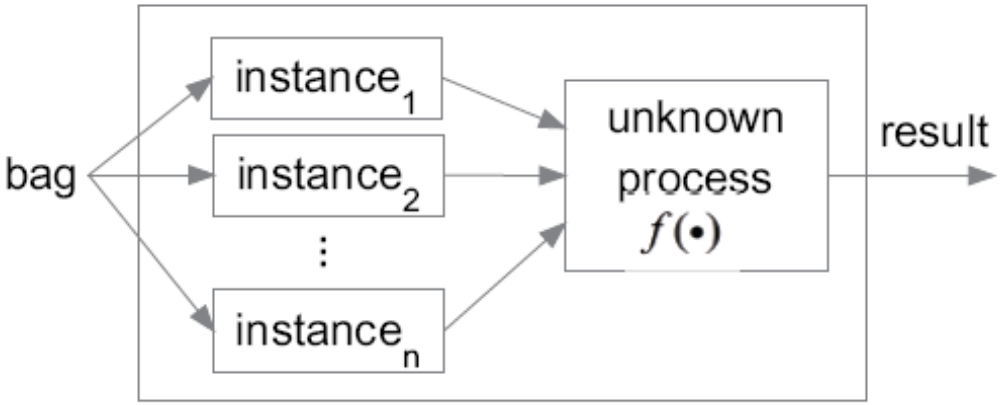
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Zeng EMNLP 2015)

- 实体类别和关系类别巨大
  - Freebase：4000多万实体，上万个属性关系，24多亿个事实三元组
  - DBpeida：400多万实体，48,293种属性关系，10亿个事实三元组
  - NELL： 519万实体，306种关系， 5亿候选三元组
  - Knowledge Vault: 4500万实体，4469种关系，2.7亿三元组
- 人工标注训练集不可行，需要寻求无指导或弱指导的关系抽取方法

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

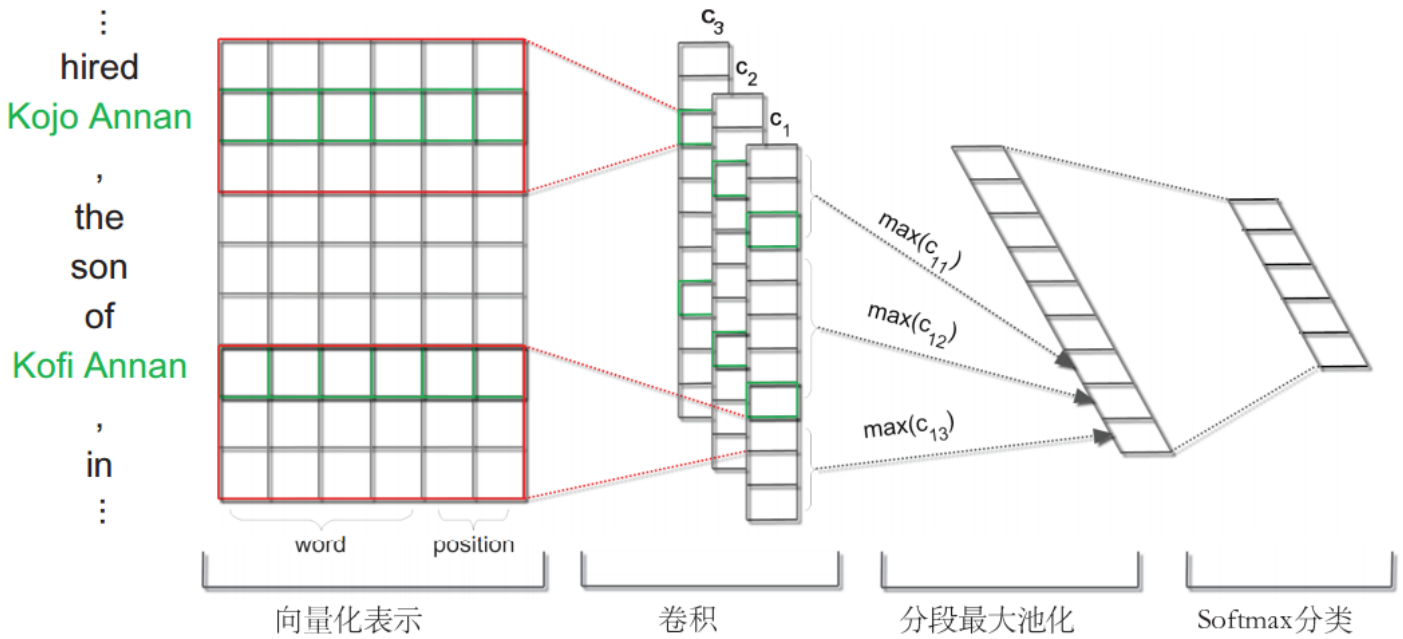
- 将弱监督关系抽取看作是多示例问题，每次选择回标的包中分类概率最大的那个示例更新参数
  - Steve Jobs was the co-founder and CEO of Apple and formerly Pixar.
  - Steve Jobs passed away the day before Apple unveiled iPhone 4S in late 2011.
  - The history of Microsoft began on April 4, 1975, when it was founded by Bill Gates and Paul Allen in Albuquerque.



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

- 利用分段卷积网络自动学习特征
  - 设计分段最大池化层，根据两个实体把句子分成三段，在每段里利用最大池化技术，更好地保留句子的结构化信息



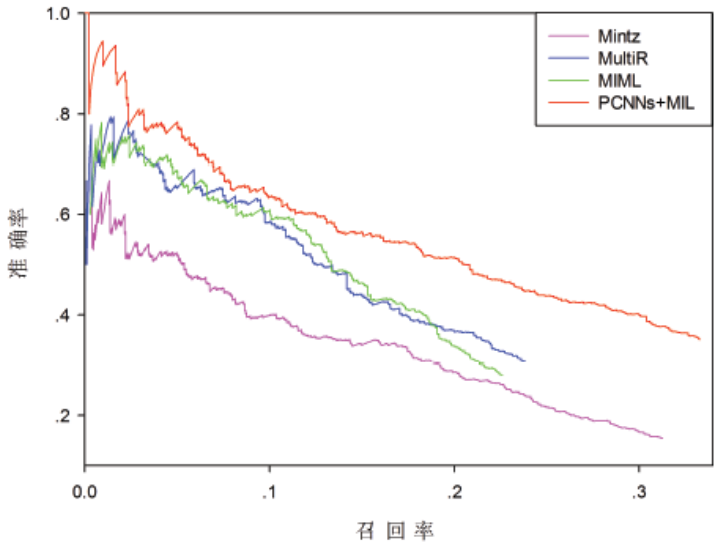
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

- 数据集：Riedel等开发
  - 设计分段最大池化层，保留句子中结构化信息
  - 知识库：Freebase
  - 文本：纽约时报（NYT）
- 评价方法
  - Held-out 评价：以Freebase 中存在关系的三元组作为标准
  - 人工评价：去掉已经在Freebase 中存在的实体对，人工标注top N 结果

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Zeng EMNLP 2015)

## 实验结果




Held out实验结果

| Top N | Mintz | MultiR | MIML | PCNNs+MIL |
|---|---|---|---|---|
| Top 100 | 0.77 | 0.83 | 0.85 | **0.86** |
| Top 200 | 0.71 | 0.74 | 0.75 | **0.80** |
| Top 500 | 0.55 | 0.59 | 0.61 | **0.69** |
| Average | 0.676 | 0.720 | 0.737 | **0.783** |

人工评价实验结果

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 动机：

**Freebase** /location/location/contains (Nevada, Las Vegas)

S1. **[Nevada] then sanctioned the sport , and the U.F.C. held its first show in [Las Vegas] in September 2001.**

S2. Pinnacle owns casinos in [Nevada], Louisiana , Indiana , Argentina and the Bahamas , but not in the top two American casino cities , Atlantic City and [Las Vegas].

S3. **He has retained two of [Nevada] 's most prominent criminal defense lawyers , Scott Freeman of Reno and David Chesnoff of [Las Vegas].**

S4. The state 's population is growing , but not skyrocketing the way it is in Arizona and [Nevada] , and with no city larger than 100,000 residents , Montana essentially does not have suburbs or exurbs like those spreading around Phoenix, [Las Vegas] and Denver.

**Descriptions**

[Nevada]: Nevada is a state in the Western, Mountain West, and Southwestern regions of the United States.

[Las Vegas]: officially the City of Las Vegas and often known as simply Vegas, is a city in the United States, the most populous city in the state of Nevada, the county seat of Clark County, and the city proper of the Las Vegas Valley.

- **传统方法**：NLP工具抽取句子特征，概率图模型选择多个有效的句子，然后分类。
- **神经网络方法**：卷积神经网络抽取特征，效果最好，但是Zeng的方法只能选择一个有效句子，不能充分利用监督信息。
- **动机一**：两种方法优点的结合，既能选择多个有效的句子，又使用卷积神经网络抽取特征。

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 动机：

**Freebase** /location/location/contains (Nevada, Las Vegas)

S1. **[Nevada] then sanctioned the sport , and the U.F.C. held its first show in [Las Vegas] in September 2001.**

S2. Pinnacle owns casinos in [Nevada], Louisiana , Indiana , Argentina and the Bahamas , but not in the top two American casino cities , Atlantic City and [Las Vegas].

S3. **He has retained two of [Nevada] 's most prominent criminal defense lawyers , Scott Freeman of Reno and David Chesnoff of [Las Vegas].**

S4. The state 's population is growing , but not skyrocketing the way it is in Arizona and [Nevada] , and with no city larger than 100,000 residents , Montana essentially does not have suburbs or exurbs like those spreading around Phoenix, [Las Vegas] and Denver.

**Descriptions**

[Nevada]: Nevada is a state in the Western, Mountain West, and Southwestern regions of the United States.

[Las Vegas]: officially the City of Las Vegas and often known as simply Vegas, is a city in the United States, the most populous city in the state of Nevada, the county seat of Clark County, and the city proper of the Las Vegas Valley.

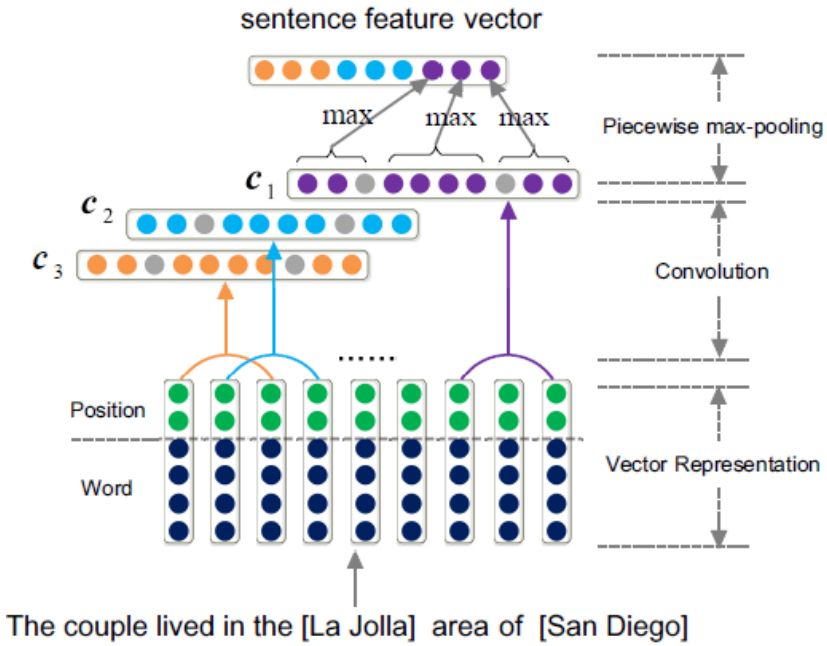
- 句子S1需要知道Nevada是个地名，不然根据其中单词sanctioned(批准)容易判断Nevada是个官员的名字，那么就没有location/location/contains的关系。
- **动机二**：利用实体描述补充背景知识

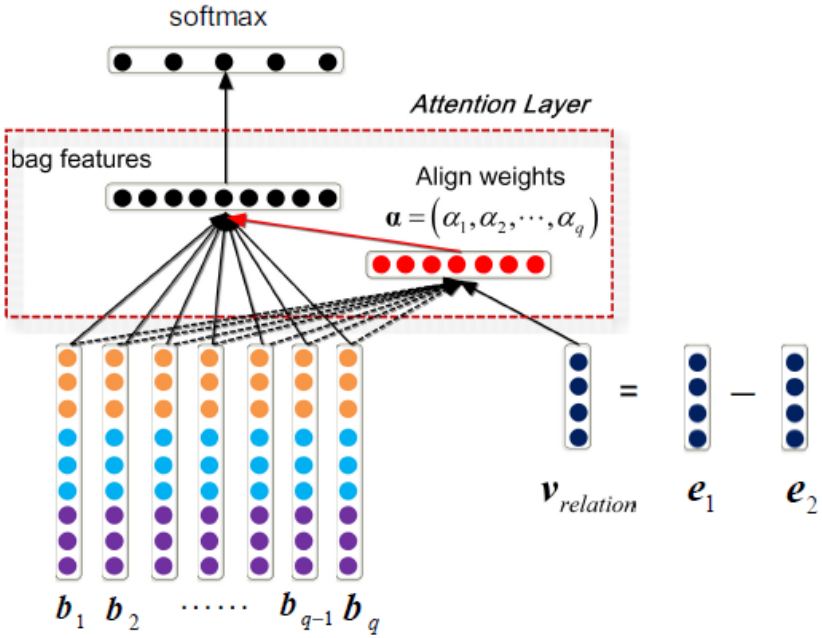
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

## ■基于句子级关注机制和实体描述的弱监督关系模型

- 利用背景知识库信息得到两个实体的关系，再利用这个关系向量和每个句子计算相似度，归一化后作为句子的关注度



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 数据集：Riedel等开发
  - 由(Riedel, Yao, and McCallum 2010) 发布的数据集，通过对齐Freebase和NTY的语料形成，共包含52种关系。数据统计如下：

| | 训练集 | 测试集 |
|---|---|---|
| 包的个数 | 67,946 | 96,676 |

  - 训练和预测的对象均是多示例包，包中的句子均没有标签。新版本的数据统计：

| | 训练集 | 测试集 |
|---|---|---|
| 包的个数 | 522,611 | 96,676 |

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

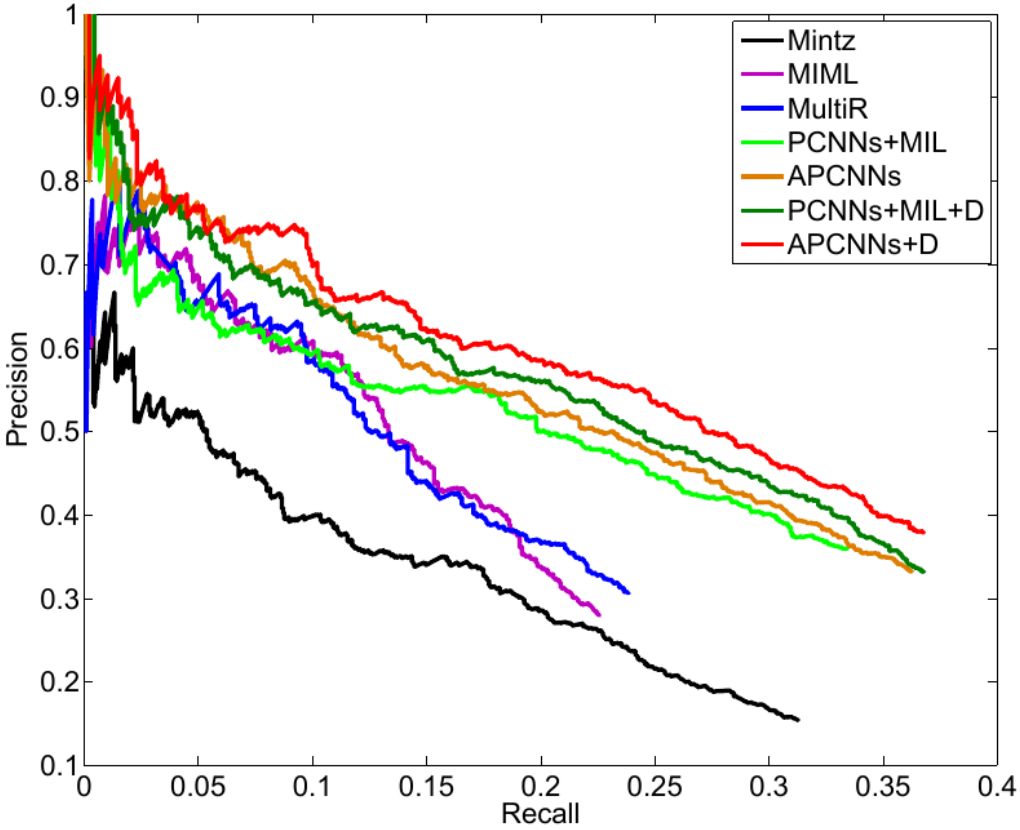
(Ji AAAI 2017)

- 评价指标
  - **自动评价**：测试集中的标签以知识谱图的标注为准，如果预测的标签与标注相同，则预测正确；否则预测错误。结果以精度-召回率曲线给出。
  - **人工评价**：由于知识图谱不完善，测试集中标注为”NA”的样本也可能真实地表达了某种关系，只是知识图谱标注错误了。此时选取被知识图谱标注为”NA”,但是模型预测出其他关系的样本，按照置信度的排列，选取top-k个样本进行人工评价，只报告正确率。

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

## 实验结果：自动评价



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

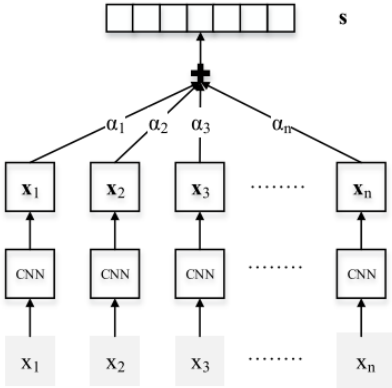
## 实验结果：人工评价

| Accuracy (%) | Top 100 | Top 200 | Top 500 | Average |
|---|---|---|---|---|
| Mintz | 0.77 | 0.71 | 0.55 | 0.676 |
| MultiR | 0.83 | 0.74 | 0.59 | 0.720 |
| MIML | 0.85 | 0.75 | 0.61 | 0.737 |
| PCNNs+MIL | 0.86 | 0.80 | 0.69 | 0.783 |
| APCNNs | **0.87** | 0.82 | 0.72 | 0.803 |
| PCNNs+MIL+D | 0.86 | 0.82 | 0.71 | 0.797 |
| APCNNs+D | **0.87** | **0.83** | **0.74** | **0.813** |

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Lin ACL 2016)

- 动机：弱监督数据中包含了误标记的数据，为了减轻误标记句子对分类结果的影响，引入了关注机制（Attention），每个包中可以选多个示例。




- 方法：
  - 给定一个句子集合 $S = \{x_1, x_2, \dots, x_n\}$
  - 将每个句子通过CNN表示成一个向量
    - $X_i = CNN(x_i)$
  - 句子集合表示为句子表示的加权和 $S = \sum_i \alpha_i X_i$
  - 最后通过 softmax 分类器对句子集合进行分类
  - 关注机制 $\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$
    - $e_i = x_i A r$，$A$ 是权重对角矩阵 $r$ 是关系r的向量表示

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Lin ACL 2016)

## 实验：

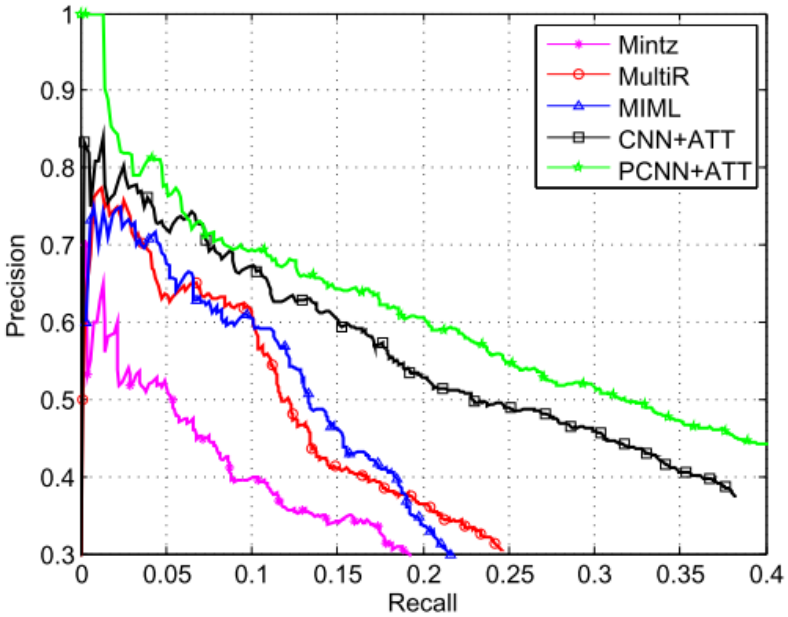


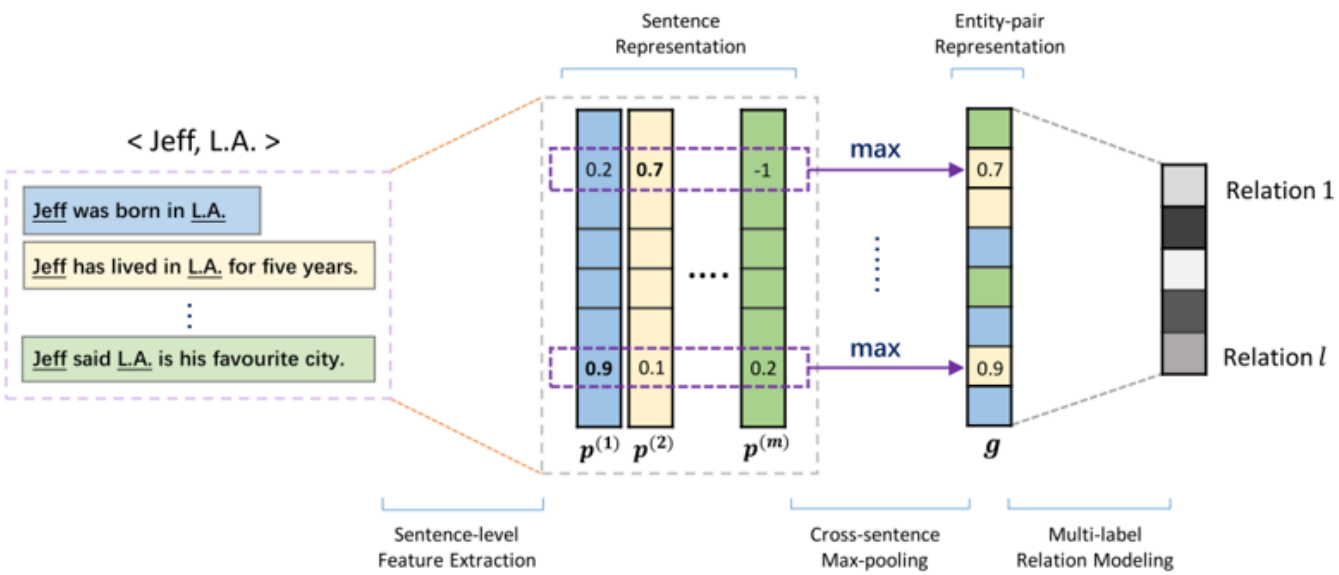

Figure 4: Performance comparison of proposed model and traditional methods

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Jiang COLING 2016)

- 动机：
  - 有些隐含的关系需要从多句话中抽取证据才能确定
  - 同一个实体对可以对应多个关系

- 方法：




1.通过跨句子的pooling实现多句话语义的联合考虑

2.通过多目标的损失函数实现一个实体对多关系的发现

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Jiang COLING 2016)

## 实验结果

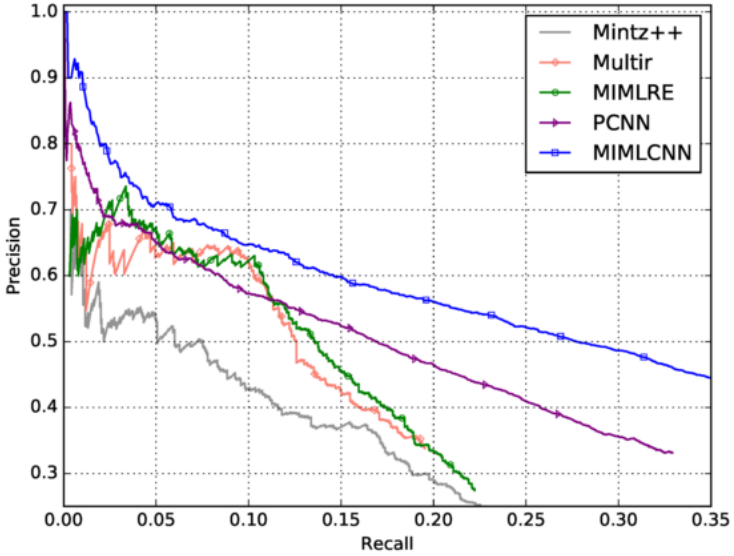
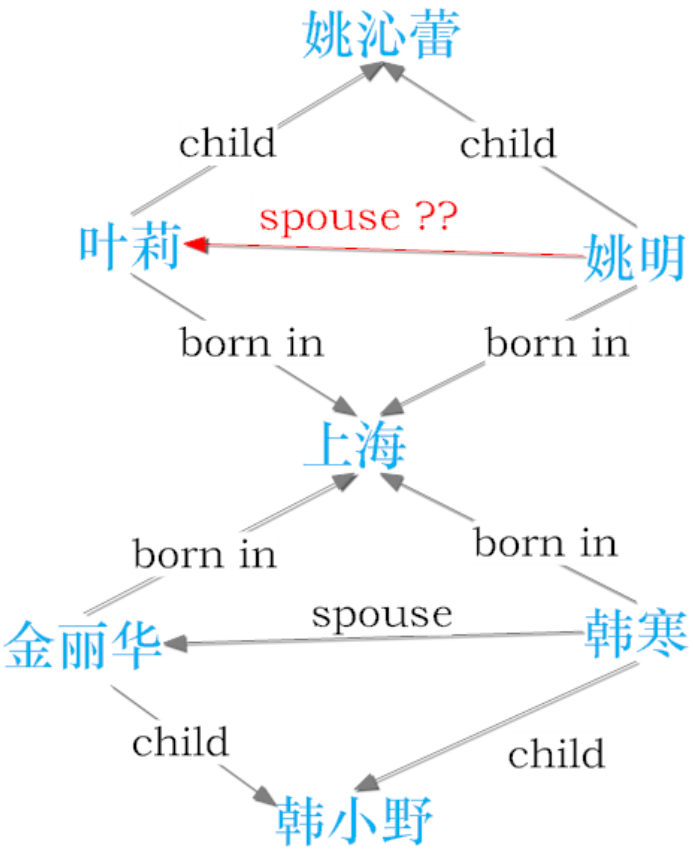
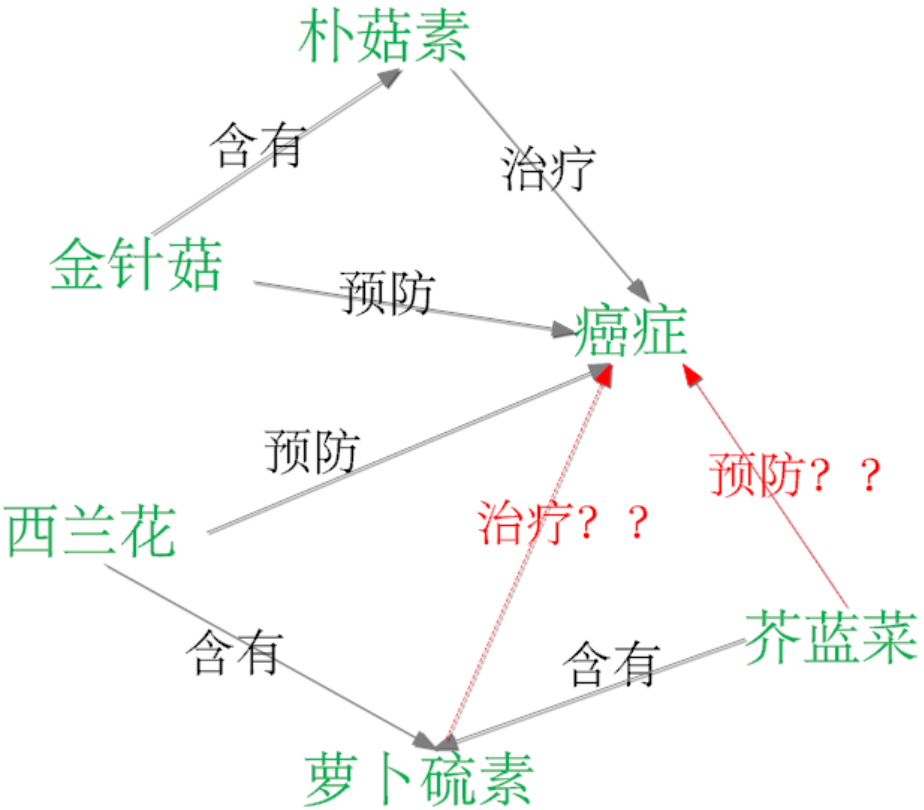



Figure 4: Precision-recall curves of the proposed method and four baselines.

# 关系发现：链接预测（知识库补全）

- 利用知识图谱中现有的知识推断未知知识（未知的链接）
- 知识图谱本身就蕴涵着推理模式（第10课时 王泉老师）



# 小结

- 由识别到抽取
  - 规范文本→有噪音、有冗余的海量网络数据
- 限定类别→开放类别
- 难点：关系类别缺乏体系结构

Q&A

# Thanks

# 实体消歧

赵 军（jzhao@nlpr.ia.ac.cn）

中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

# 实体消歧定义

- 命名实体的歧义指的是一个实体指称项可对应到多个真实世界实体，例如，给定如下的四个实体指称项“*Michael Jordan*”

  *MJ1: Michael Jordan is a researcher in machine learning.*

  *MJ2: Learning in Graphical Models: Michael Jordan*

  *MJ3: M. Jordan wins NBA MVP.*

  *MJ4 : Michael Jordan plays basketball in Chicago Bulls.*

- 确定一个实体指称项所指向的真实世界实体，这就是命名实体消歧

# 网络媒体中的实体歧义

Chen Guangcheng

| Morph | Target | Motivation |
|---|---|---|
| Blind Man (瞎子) | Chen Guangcheng(陈光诚) | Sensitive |
| Kimchi Country (泡菜国) | Korea (韩国) | Vivid |
| Rice Country (米国) | United States (美国) | Pronunciation |
| Miracle Brother (奇迹哥) | Wang Yongping (王勇平) | Irony |

# 实体消歧分类

- 基于聚类的实体消歧
  - 把所有实体指称项按其指向的目标实体进行聚类
  - 每一个实体指称项对应到一个单独的类别

*MJ1: Michael Jordan is a researcher in machine learning.*

*MJ2: Research in Graphical Models: Michael Jordan*

*MJ3: M. Jordan wins NBA MVP.*

*MJ4 : Michael Jordan plays basketball in Chicago Bulls*

- 基于实体链接的实体消歧
  - 将实体指称项与目标实体列表中的对应实体进行链接实现消歧

*MJ4 : Michael Jordan plays basketball in Chicago Bulls*

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

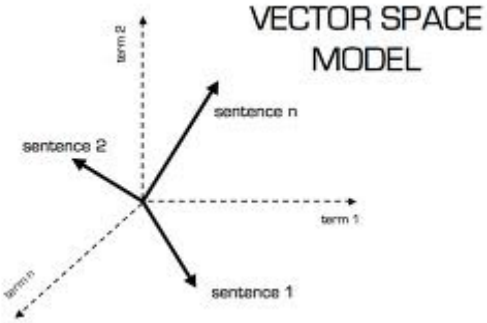
# 基于聚类的实体消歧

- 基本思路
  - 同一指称项具有近似的上下文
  - 利用聚类算法进行消歧
  - 核心问题：选取何种特征对于指称项进行表示
    - 词袋模型(Bagga et al., COLING, 1998)
    - 语义特征(Pederson et al., CLITP, 2005)
    - 社会化网络(Bekkerman et al., WWW, 2005)
    - 维基百科的知识(Han and Zhao, CIKM, 2009)
    - 多源异构语义知识融合(Han and Zhao, ACL, 2010)

# 基于聚类的实体消歧:词袋模型
## (Bagga et al. COLING 1998)

- 利用待消歧实体周边的词来构造向量

- 利用向量空间模型来计算两个实体指称项的相似度，进行聚类




MJ1: Michael Jordan is a researcher in machine learning.

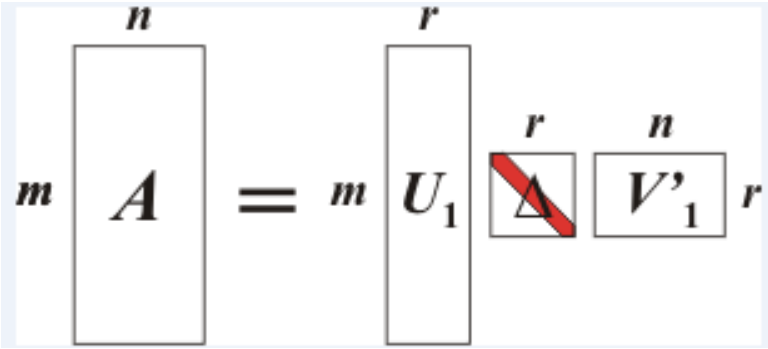
MJ1 → researcher → machine → learning

MJ1: Michael Jordan plays basketball in Chicago Bulls

MJ4 → plays → basketball → Chicago → Bulls

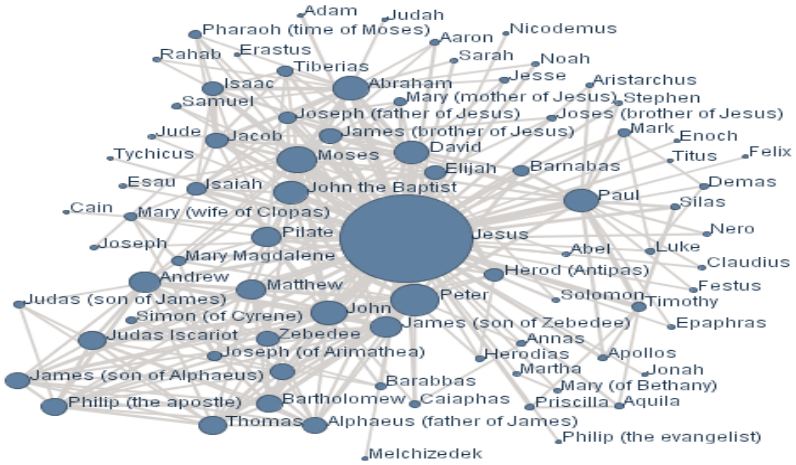
# 基于聚类的实体消歧: 语义特征
(Pederson et al. CLITP 2005)

- 词袋模型，没有考虑词的语义信息
- 利用SVD分解挖掘词的语义信息
- 利用词袋和浅层语义特征，共同表示指称项，利用余弦相似度来计算两个指称项的相似度



# 基于聚类的实体消歧：社会化网络
## (Bekkerman et al. WWW 2005)

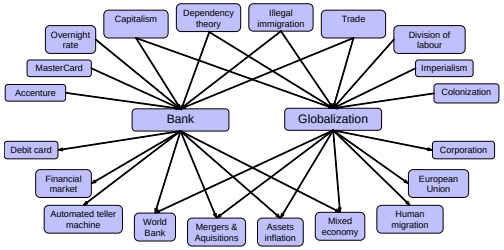
- 不同的人具有不同的社会关系
  - MJ（BasketBall）: Pippen, Buckley, Ewing, Kobe…
  - MJ（Machine Learning）: Liang, Mackey, Wauthier…
- MJ, Pippen, Buckley, Ewing, Kobe等的社会化关联信息所表现出来的网页链接特征，对网页进行聚类，从而实现网页内的人名聚类消歧。



# 基于聚类的实体消歧: Wikipedia
## (Han CIKM 2009) (1/3)

- Wikipedia中相关实体具有链接关系

- 这种链接关系反映条目之间的语义相关度
  - D. Milne and Ian H. Witten 2008: The higher semantic related Wikipedia concepts will share more semantic related concepts.




The concepts has links with a and b

$$sr(a,b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))}$$

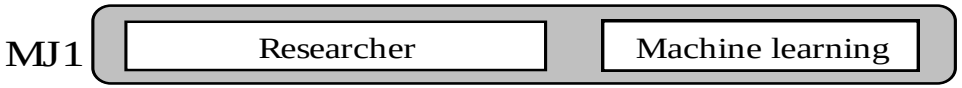
The Whole Wikipedia concepts

The concepts has links with a and b

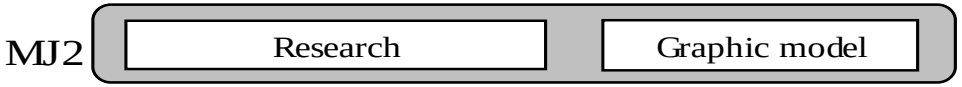
# 基于聚类的实体消歧: Wikipedia
## (Han CIKM 2009) (2/3)

- 用实体上下文的维基条目对于实体进行向量表示

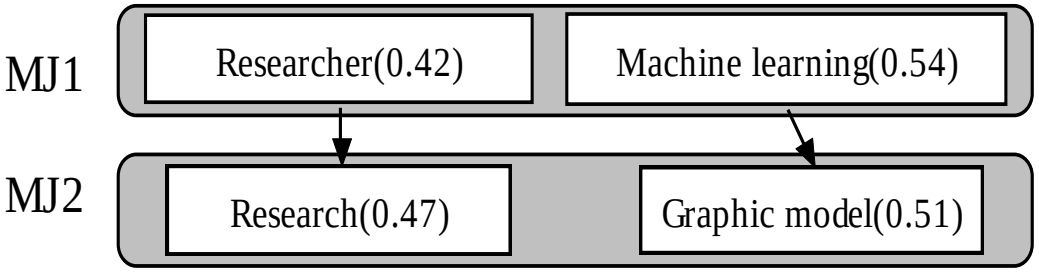
*MJ1: Michael Jordan is a Researcher in machine learning.*




*MJ2: Research in Graphical Models: Michael Jordan*




- 利用维基条目之间的相关度计算指称项之间的相似度（解决数据稀疏问题）



# 实验比较
## (Han CIKM 2009) (3/3)

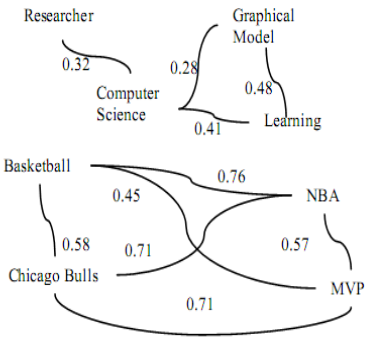
- 使用WePS数据集测试
- 使用结构化关联语义核的实体相似度能够提升10.7%的消歧性能

| Method | WePS1_training | | |
|---|---|---|---|
| | Pur | Inv_Pur | F |
| *BOW* | 0.71 | 0.88 | **0.78** |
| *SocialNetwork* | 0.66 | 0.98 | **0.76** |
| *WikipediaConcept* | 0.80 | 0.88 | **0.82** |
| ***WS-SameWeight*** | 0.84 | 0.89 | **0.85** |
| ***WS*** | 0.88 | 0.89 | **0.87** |

# 基于聚类的实体消歧: 多源异构知识
## (Han ACL 2010) (1/3)

- 仅仅考虑Wikipedia一种知识源，覆盖度有限
- 多源异构知识的挖掘与集成
  - 知识源中存在大量的多源异构知识
  - 挖掘和集成多源异构知识可以提高实体消歧的性能
  - Wikipedia
    - 用于捕捉概念之间的语义关联
  - WordNet
    - 用于捕捉词语之间的语言学关联
  - Web网页库
    - 用于捕捉命名实体之间的社会化关联

# 基于聚类的实体消歧：多源异构知识
## (Han ACL 2010) (2/3)




- 多源异构知识的表示框架：语义图
  - 等同概念识别
  - 概念连接
    - 同时捕捉显式语义知识和结构化语义知识
      - **语义图的边（显式语义知识）**——建模了所有从知识源中直接抽取出的概念之间的显式语义关联
      - **语义图的结构（结构化语义知识）**——建模了概念之间的隐藏语义关联
- 语义图中语义知识的挖掘和融合算法
  - 计算原则：“如果一个概念的邻居概念与另一个概念存在语义关联，则这个概念也与另一个概念存在语义关联”
  - 语义关联在图中的传递性

$$S_{ij} = \boxed{\lambda \sum_{l \in N_i} \frac{A_{il}}{d_i} S_{lj}} + \boxed{\mu A_{ij}}$$

邻居节点传递　　显式语义关联

# 基于聚类的实体消歧:多源异构知识
## (Han ACL 2010) (3/3)

- 实验比较
  - 使用WePS数据集测试
  - 使用多源知识能够有效提高消歧的准确度

| | WePS2_test | | |
|---|---|---|---|
| | **Pur** | **Inv_Pur** | **F** |
| *BOW* | 0.80 | 0.80 | **0.77** |
| *SocialNetwork* | 0.62 | 0.93 | **0.70** |
| *SSR-NoKnowledge* | 0.84 | 0.80 | **0.80** |
| *SSR-NoStructure* | 0.84 | 0.83 | **0.81** |
| ***SSR-NE*** | 0.78 | 0.88 | **0.80** |
| ***SSR-WordNet*** | 0.85 | 0.82 | **0.83** |
| ***SSR-Wikipedia*** | 0.84 | 0.81 | **0.82** |
| ***SSR*** | 0.90 | 0.86 | **0.88** |

# 基于聚类的实体消歧：评测(1/2)

- WePS： Web People Search Evaluation
  - WePS1是SEMEVAL2007的子任务
  - WePS2是WWW的一个workshop
  - 任务：Web环境中的人名消歧，即给定一个包含某个歧义人名的网页集合，按照网页中人名指称项所指向的人物概念来对网页进行聚类，以及抽取一个网页中关于某个人的特定属性来辅助进行人名消歧
  - 评测方法

$$\text{Purity} = \sum_i \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j)$$

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)$$

$$\text{F} = \frac{1}{\alpha \frac{1}{\text{Purity}} + (1-\alpha) \frac{1}{\text{Inverse Purity}}}$$

# 基于聚类的实体消歧：评测(2/2)

| | | Macro-averaged Scores | | | |
|---|---|---|---|---|---|
| | | F-measures | | | |
| rank | team-id | $\alpha = ,5$ | $\alpha = ,2$ | Pur | Inv_Pur |
| 1 | CU_COMSEM | ,78 | ,83 | ,72 | ,88 |
| 2 | IRST-BP | ,75 | ,77 | ,75 | ,80 |
| 3 | PSNUS | ,75 | ,78 | ,73 | ,82 |
| 4 | UVA | ,67 | ,62 | ,81 | ,60 |
| 5 | SHEF | ,66 | ,73 | ,60 | ,82 |
| 6 | FICO | ,64 | ,76 | ,53 | ,90 |
| 7 | UNN | ,62 | ,67 | ,60 | ,73 |
| *8* | *ONE-IN-ONE* | *,61* | *,52* | *1,00* | *,47* |
| 9 | AUG | ,60 | ,73 | ,50 | ,88 |
| 10 | SWAT-IV | ,58 | ,64 | ,55 | ,71 |
| 11 | UA-ZSA | ,58 | ,60 | ,58 | ,64 |
| 12 | TITPI | ,57 | ,71 | ,45 | ,89 |
| 13 | JHU1-13 | ,53 | ,65 | ,45 | ,82 |
| 14 | DFKI2 | ,50 | ,63 | ,39 | ,83 |
| 15 | WIT | ,49 | ,66 | ,36 | ,93 |
| 16 | UC3M_13 | ,48 | ,66 | ,35 | ,95 |
| 17 | UBC-AS | ,40 | ,55 | ,30 | ,91 |
| *18* | *ALL-IN-ONE* | *,40* | *,58* | *,29* | *1,00* |

WePS 1

| | | Macro-averaged Scores | | | |
|---|---|---|---|---|---|
| | | F-measures | | | |
| rank | run | $\alpha = ,5$ | $\alpha = ,2$ | Pur | Inv_Pur |
| | *BEST-HAC-TOKENS* | ,90 | ,89 | ,93 | ,88 |
| | *BEST-HAC-BIGRAMS* | ,90 | ,87 | ,94 | ,86 |
| 1 | PolyUHK | ,88 | ,87 | ,91 | ,86 |
| 2 | UVA_1 | ,87 | ,87 | ,89 | ,87 |
| 3 | ITC-UT_1 | ,87 | ,83 | ,95 | ,81 |
| | *CHEAT_SYS* | ,87 | ,94 | ,78 | 1,00 |
| 4 | UMD_4 | ,81 | ,76 | ,95 | ,72 |
| 5 | XMEDIA_3 | ,80 | ,76 | ,91 | ,73 |
| 6 | UCI_2 | ,80 | ,84 | ,75 | ,89 |
| 7 | LANZHOU_1 | ,80 | ,78 | ,85 | ,77 |
| 8 | FICO_3 | ,80 | ,76 | ,90 | ,73 |
| | *HAC-BIGRAMS* | ,78 | ,64 | ,96 | ,67 |
| 9 | UGUELPH_1 | ,74 | ,84 | ,64 | ,95 |
| 10 | CASIANED_4 | ,73 | ,77 | ,72 | ,83 |
| | *HAC-TOKENS* | ,71 | ,64 | ,96 | ,60 |
| 11 | AUG_4 | ,69 | ,68 | ,79 | ,68 |
| 12 | UPM-SINT_4 | ,67 | ,70 | ,69 | ,74 |
| | *ALL_IN_ONE* | ,67 | ,79 | ,56 | 1,00 |
| 13 | UNN_2 | ,64 | ,59 | ,80 | ,57 |
| 14 | ECNU_1 | ,53 | ,56 | ,60 | ,63 |
| 15 | PRIYAVEN | ,53 | ,49 | ,71 | ,48 |
| 16 | UNED_3 | ,51 | ,48 | ,71 | ,48 |
| 17 | BUAP_1 | ,37 | ,30 | ,89 | ,27 |
| | *ONE_IN_ONE* | ,34 | ,27 | 1,00 | ,24 |

WePS 2

# 小结

- 主要研究集中在实体指称项的语义表示
- 已有工作大多是通过扩展特征，增加更多的知识来提高消歧精度
- 挑战
  - 消歧目标难以确定
  - 缺乏实体的显式表示

# 概述

- 实体消歧概述
- 基于无监督的实体消歧
- 基于知识库链接的实体消歧

# 实体链接的任务

- 任务
  - 给定实体指称项和它所在的文本，将其链接到给定知识库中的相应实体上

*Michael Jordan*
*is an NBA player.*

*Michael Jordan*
*is Computer Science professor in UC Berkeley.*

| 实体 ID | 实体 |
|---|---|
| … | … |
| E00010 | Michael Jordan(NBA player) |
| E00011 | Michael Jordan(Berkeley Professor) |
| … | … |
| E10000 | Michael Jordan(football player) |
| E100001 | Michael Jackson |
| … | … |

目标实体列表

# 实体链接主要步骤

- 主要步骤
  - 候选实体的发现
    - 给定实体指称项，链接系统根据知识、规则等信息找到实体指称项的候选实体
  - 候选实体的链接
    - 系统根据指称项和候选实体之间的相似度等特征，选择实体指称项的目标实体

实体指称项文本：Michael Jordan is a former NBA player, active businessman and majority owner of the Charlotte Bobcats.

候选实体：
Michael Jordan (basketball player)
Michael Jordan (mycologist )
Michael Jordan (footballer)
Michael B. Jordan
Michael H. Jordan
Michael-Hakim Jordan
Michael Jordan (Irish polotician)
…

# 候选实体发现

- 如何根据实体指称项找出候选实体
  - 利用Wikipedia的信息
  - 利用上下文信息

| 实体指称项 | 候选实体 |
| --- | --- |
| Michael Jordan | Michael Jordan (basketball)<br>Michael Jordan (mycologist)<br>Michael Jordan (football)<br>Michael B. Jordan (American actor)<br>… |
| AI | Artificial intelligence<br>Ai (singer)<br>… |
| … | … |

# 利用Wikipedia信息获取候选实体

- 利用Wikipedia中锚文本的超级链接关系
  - Michal Jordan is a former NBA player

- 利用Wikipedia中的消歧页面

- 利用Wikipedia中的重定向页面

# 利用上下文获取缩略语候选实体
## (Zhang IJCAI 2011)

- 问题
  - 缩略语在实体指称项中十分常见，据统计，在KBP2009的测试数据，在3904个实体指称项中有827个为缩略语
- 动机
  - 缩略语指称项具有很强的歧义性，但它的全称往往是没有歧义的
  - ABC和American Broadcasting Company, AI和Artificial Intelligence等
  - 在实体指称项文本中，缩略语的全称出现过
- 解决方法
  - 利用人工规则抽取实体候选

# 候选实体链接

- 如何进行实体链接
  - 基本方法：计算实体指称项和候选实体的相似度，选择相似度最大的候选实体
  - 单一实体链接
    - BOW模型 (Honnibal TAC 2009, Bikel TAC 2009)
    - 加入候选实体的类别特征（Bunescu et al., EACL 2006）
    - 加入候选实体的流行度等特征（Han et al., ACL 2011）
  - 协同实体链接
    - 利用实体之间类别的共现特征（Cucerzan, EMNLP 2007）
    - 利用实体之间链接关系（Kulkarni et al., KDD 2009）
    - 利用同一篇文档中不同实体之间存在的语义关联特征（Han et al., SIGIR 2011）

# 实体链接的基本方法
## (Honnibal TAC 2009, Bikel TAC 2009)

- 基于词袋子模型计算相似度
  - 将实体指称项上下文文本与候选实体上下文文本表示成词袋子向量形式，通过计算向量间的夹角确定指称项与候选实体相似度，系统选择相似度最大的候选实体进行链接

$$score(q, e_k) = cos(q.T, e_k.T) = \frac{q.T}{\|q.T\|} \frac{e_k.T}{\|e_k.T\|}$$

$$\hat{e} = arg \max_{e_k} score(q, e_k)$$

# 类别特征
## (Bunescu EACL 2006)

- 动机
  - 候选实体的文本内容可能太短，会导致相似度计算的不准确
  - 加入指称项文本中的词与候选实体类别的共现特征
    - 例：除了计算待消歧文本和实体Wikipedia文本John Williams (composer)的相似度外，还考虑当前文本中的词语与**Music, Art** 等类别的共现信息
- 方法
  - 训练SVM分类器对候选实体进行选择
  - 训练数据由Wikipedia中的超级链接获得
  - 所采用的特征
    - 文本相似度
    - 指称项文本中词与候选实体类别的共现信息

John Williams (composer): Category={**Music, Art**…}
John Williams (wrestler): Category={Sport,…}
John Williams (VC): Category={Bank,…}

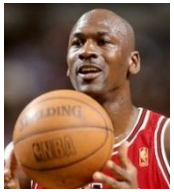
Williams has also composed numerous classical concerti, and he served as the principal conductor of the Boston Pops Orchestra from 1980 to 1993

类别：music

# 实体流行度等特征
## （Han ACL 2011）

- 动机
  - 传统的方法仅仅是计算实体指称项与候选实体的相似度，忽略了候选实体的背景知识与先验信息，如实体本身的流行度、实体与指称项的关系等
- 方法
  - 考虑实体的背景知识，将实体的背景知识融入到实体链接的过程，实体的背景知识和先验信息主要有
    - 实体流行度：实体e在知识库中的概率P(e)
    - 名称的知识：指称项s指向实体e的概率P(s|e)
    - 上下文知识：实体e出现在特定上下文环境c的概率P(c|e)

# 协同实体链接



During his standout career at **Bulls**,
**Jordan** also acts in the movie Space Jam .

- 实体指称项与目标实体的语义相似度
- 目标实体之间的语义相似度

# 协同学习策略

- 动机
  - 同一篇文档中实体之间具有语义相关性
  - 利用Pairwise优化策略

$$\frac{1}{\binom{|S_0|}{2}} \sum_{s \neq s' \in S_0} r(y_s, y_{s'}) + \frac{1}{|S_0|} \sum_{s \in S_0} w^\top f_s(y_s).$$

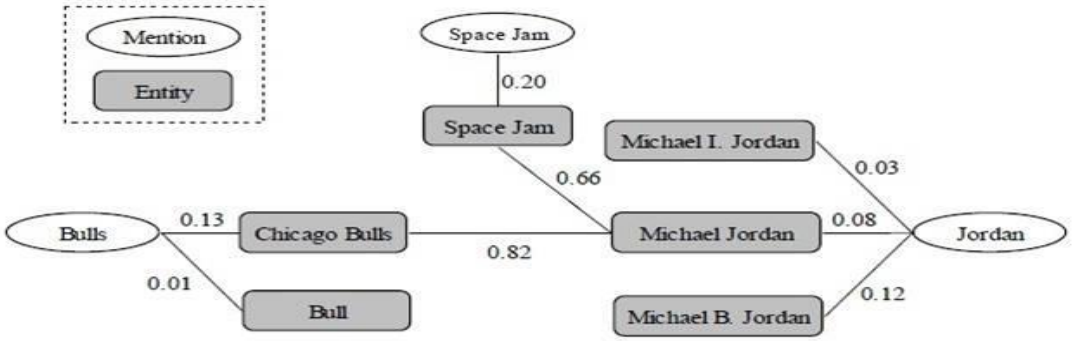
任意两个目标实体之间的语义相关度

实体指称项到目标实体的语义相似度

- 目标实体的语义相关度计算方法：
  - 利用实体类别重合度计算目标实体语义相似度（Cucerzan, EMNLP 2007）
  - 利用实体之间链接关系计算目标实体语义相似度（Kulkarni , KDD 2009 ）

# 基于图的协同链接
## （Han SIGIR 2011）

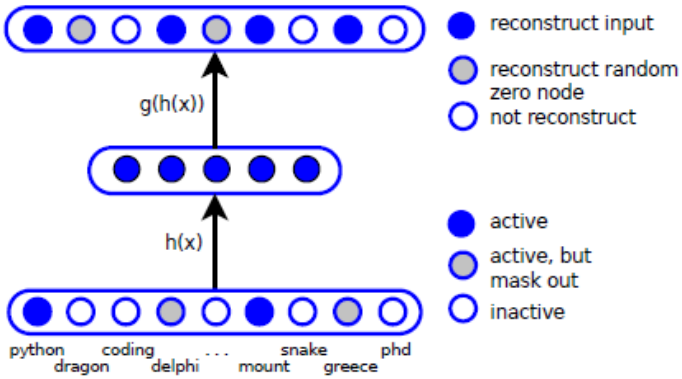
- 动机
  - Pairwise策略只考虑两两实体关系，结果不是全局最优的
  - 采用图方法，全局考虑目标实体之间的语义关联
- 方法：Referent Graph，两种关系构成
  - 指称项与实体之间的关系：该指称项文本与实体文本的相似度，由传统的VSM模型得到
  - 实体之间的语义关系：利用目标实体之间的链接关系计算实体之间的语义相关度

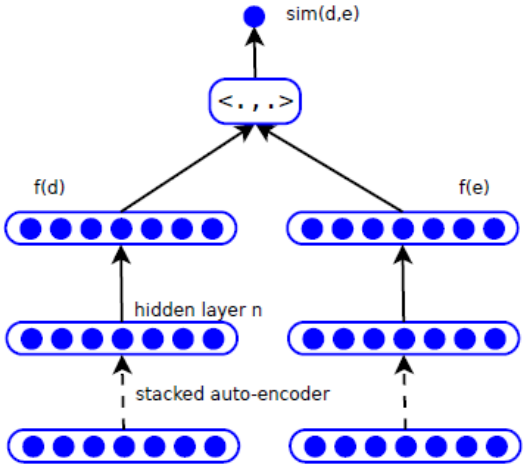


# 基于深度学习的方法
## （He ACL 2013）

- 动机
  - 传统的方法中，计算待消歧实体上下文和目标实体语义相似度的方法（点乘，余弦相似度，KL距离等）可扩展性差，没有考虑各个概念间的内在联系
  - 在协同过滤的方法中，计算待消歧实体上下文和目标实体语义相似度也是基础工作。
- 方法

提出利用深度学习的方法自动联合学习实体和文档的表示，进而完成实体链接任务（数据集：TAC-KBP 2010 和 AIDA）

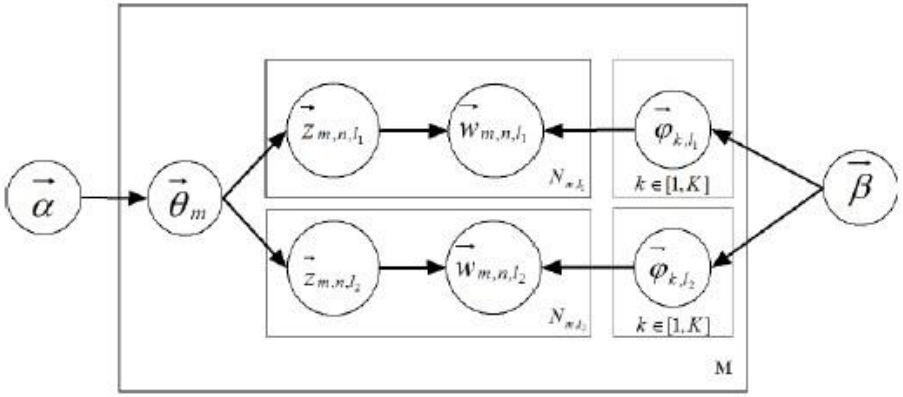


# 跨语言实体链接
## （Zhang IJCAI 2013）

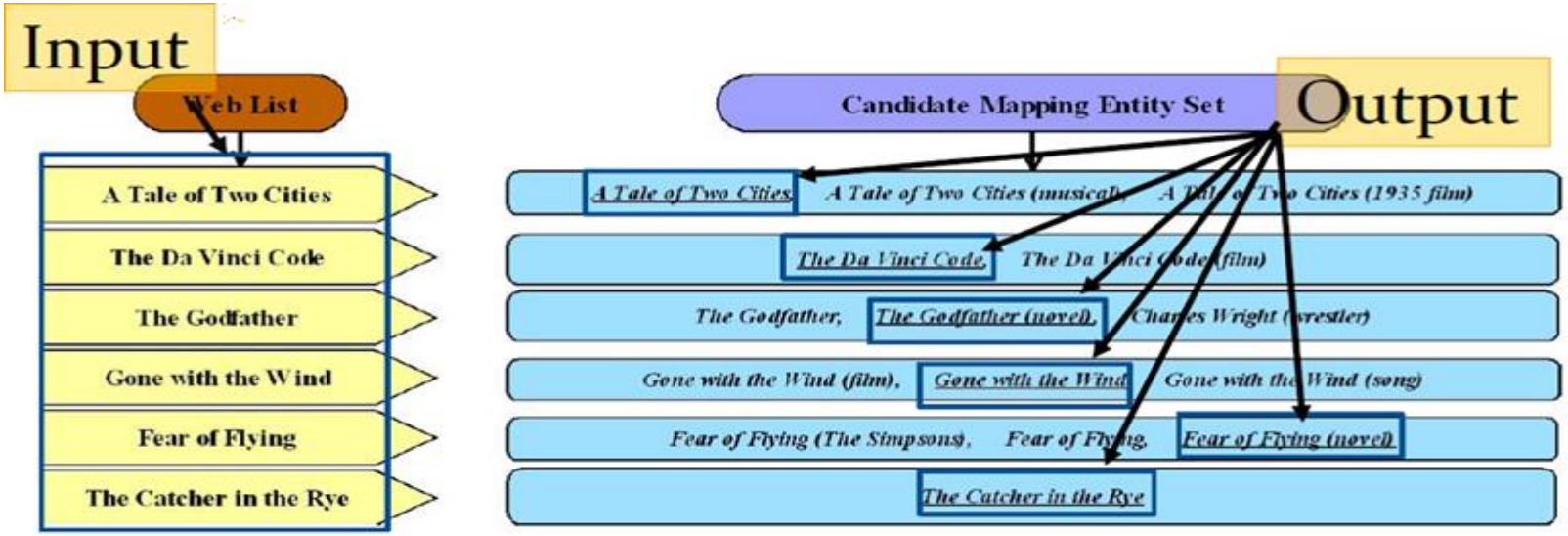
- 动机
  - 给定一种语言的实体指称项和其所在的上下文，将其链接到另外一种语言的知识库中
  - 传统方法要先翻译成目标语言，可能产生错误传递，需要大量的句子级平行的双语训练语料

- 方法：利用双语隐含主题模型将实体指称项与候选实体映射到同一个主题空间中（数据集：TAC-KBP 2011）
  - 每一个隐含主题有两种不同的分布，分别对应两种不同语言
  - 处于同一个主题分布下的两种不同语言的词的分布具有一些共性



# 结构化数据中的实体链接
## ( Shen SIGKDD 2012 )

- 动机
  - 没有上下文
  - 任务与传统的实体链接不同

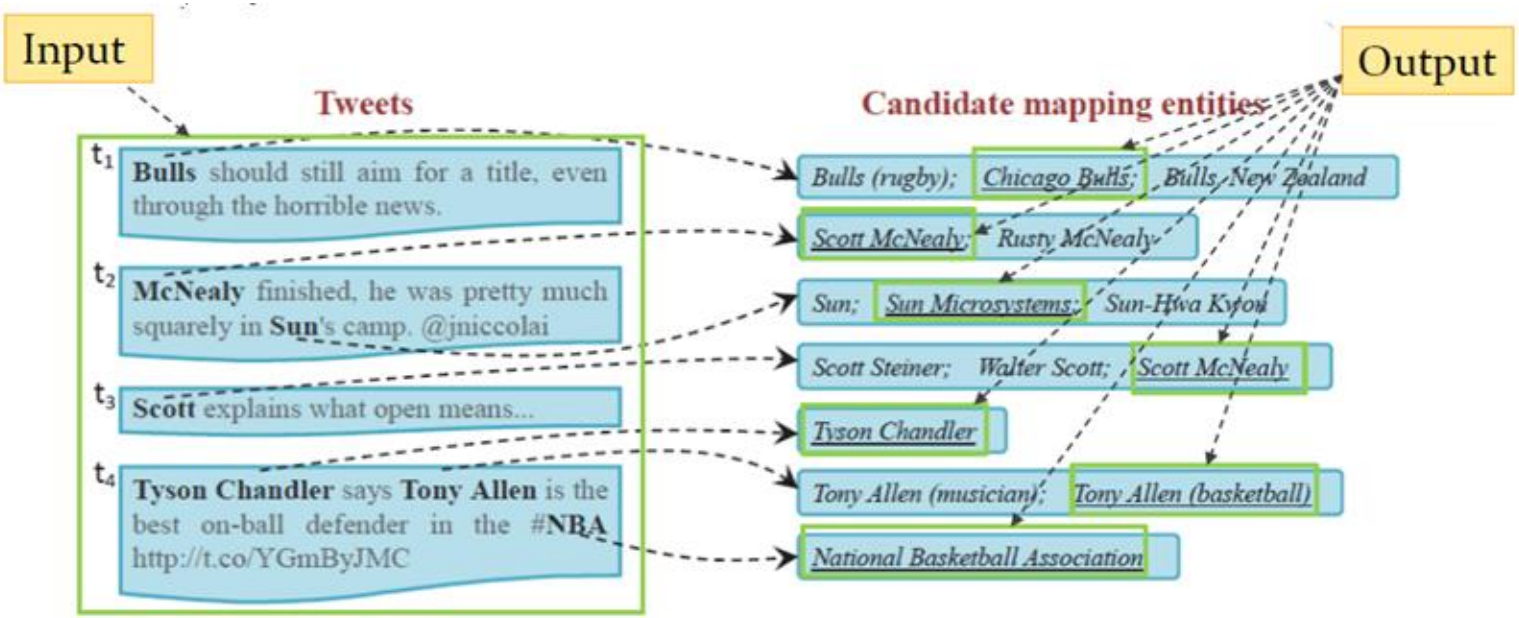



- 方法
  - 主要利用实体的流行度和实体共现类型去消歧

取自王建勇老师报告（第三届中文知识图谱研讨会）

# 社交数据中的实体链接
## （Shen SIGKDD 2013）

- 动机
  - 社交媒体（Twitter）是一种重要的信息来源
  - 社交媒体的上下文较短，语言表述不规范




- 方法
  - 利用tweet的用户信息和tweet的交互信息

取自王建勇老师报告（第三届中文知识图谱研讨会）

# 实体链接评测 (1/2)

- TAC-KBP (2009-Now)：Entity Linking
  - 任务：将文本中的目标实体链接到Wikipedia中的真实概念，达到消歧的目的
  - 评测方法：

以指称项为单位计算的准确率

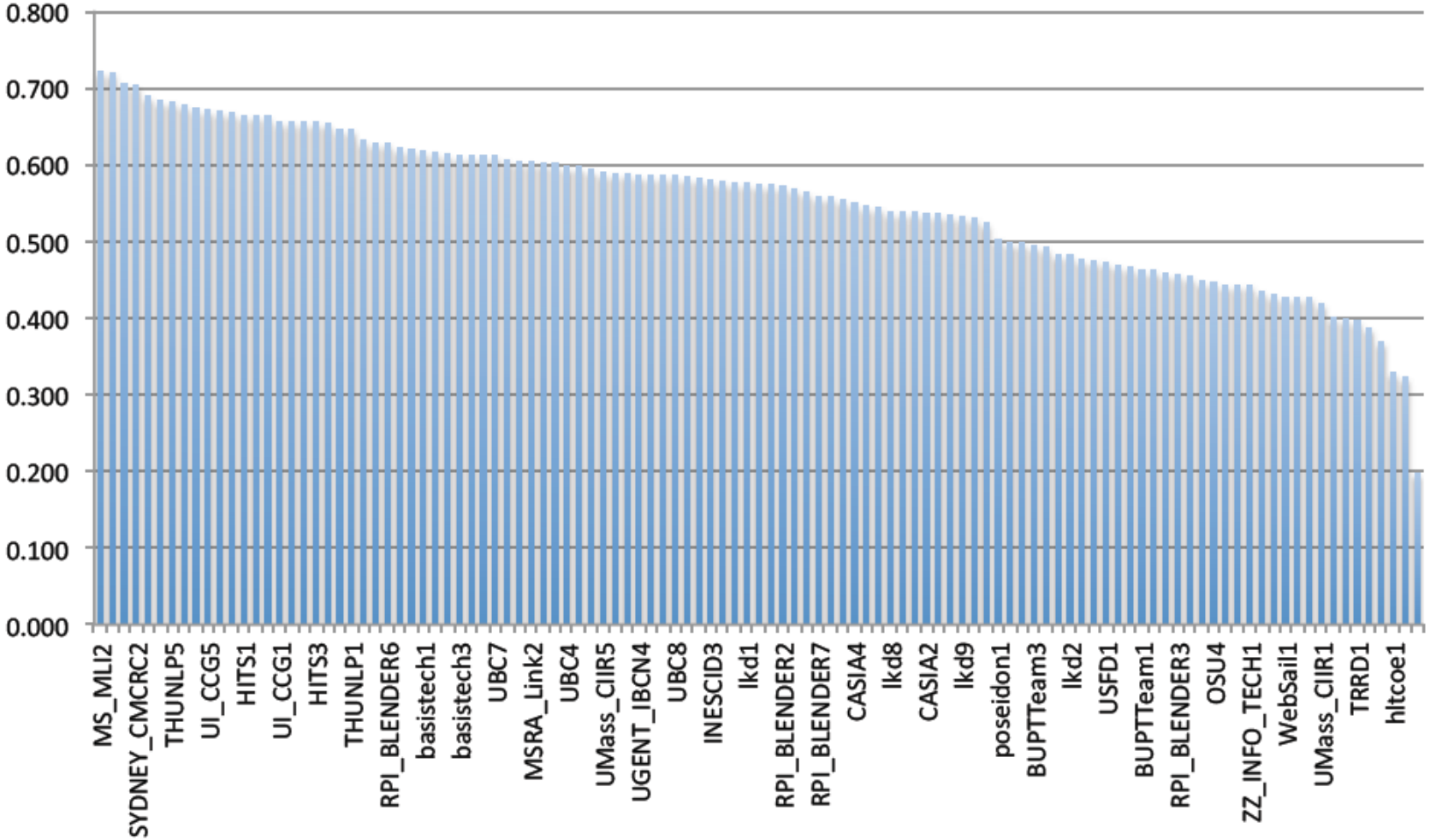
$$Accuracy_{micro} = \frac{NumCorrect}{NumQueries}$$

$$Accuracy_{macro} = \frac{\sum_{i}^{NumEntities} \frac{NumCorrect(E_i)}{NumQueries(E_i)}}{NumEntities}$$

以实体为单位计算的准确率

# 实体链接评测 (2/2)

2013评测结果(Micro Accuracy)



# 小结

- 目前实体链接方法主要是如何更有效挖掘实体指称项信息，如何更准确地计算实体指称项和实体概念之间的相似度
- 由单一实体链接向协同实体链接发展
- 难点：未登录实体的处理

# 关系抽取

# 概述

- 关系抽取任务定义
- 传统关系抽取
- 开放域关系抽取

# 实体间关系抽取的定义

- Alexander Schutz等人认为关系抽取是自动识别由一对概念和联系这对概念的关系构成的相关三元组
  - Example1: 比尔盖茨是微软的CEO
    - CEO(比尔盖茨, 微软)
  - Example2: CMU坐落于匹兹堡
    - Located-in(CMU, 匹兹堡)
  - 高阶关系：Michael Jordan获得1997/98赛季的MVP
    - Award(Michael Jordan, 1997/98赛季, MVP)

# 网络文本信息结构

- 结构化数据（Infobox）
  - 置信度高
  - 规模小
  - 缺乏个性化的属性信息
- 半结构化数据
  - 置信度较高
  - 规模较大
  - 个性化的信息
  - 形式多样
  - 含有噪声
- 纯文本
  - 置信度低
  - 复杂多样
  - 规模大

| | | | |
|---|---|---|---|
| 中文名： | 姚明 | 专业特点： | 20英尺外精确跳投 |
| 外文名： | Yao Ming | 主要奖项： | NBA全明星赛(7次) |
| 别名： | 小巨人 移动长城 | | ESPN全球最有潜力运动员奖(2000) |
| 国籍： | 中国 | | 劳伦斯世界最佳新秀奖(2003) |
| 民族： | 汉族 | | 中国篮球杰出贡献奖 |
| 出生地： | 上海 | 重要事件： | 专题影片《姚明年》发行 |
| 出生日期： | 1980年9月12日 | 祖籍： | 江苏苏州吴江震泽 |
| 身高： | 2.23米（7.32英尺） | 位置： | 中锋 |
| 体重： | 140.6kg | 鞋码： | 18码 |
| 运动项目： | 篮球 | 自传： | 《我的世界我的梦》 |
| 所属运动队： | NBA火箭队 | 生涯最高分： | 41分 |

1984：《一个和八个》摄影

1984：《黄土地》摄影

1986：《大阅兵》摄影

1987：《老井》主演

1987：《红高粱》导演

1989：《古今大战秦俑

1989：《代号美洲豹》

1990：《菊豆》导演

1991：《大红灯笼高高

**栖息环境**

沙狐栖息在广袤荒原及半沙中，行动快而敏捷。狐臭不显。

**饮食习惯**

食物包括各种鼠类、鼠兔、

**生长繁殖**

常在每年1-3月发情，妊娠期常每胎3～6仔。寿命约6年。

**个人简介**

姓名：姚明

**祖籍：**江苏苏州吴江震泽

**出生地：**上海市徐汇区

**出生医院：**上海交通大学附属第六人民医院

**小学：**上海市高安路第一小学

**初中：**上海市第二中学

**大学：**上海交通大学

**星座：**处女座

**曾效力球队：**CBA上海东方大鲨鱼、NBA休斯敦火箭

**前任主教练：**里克·阿德尔曼

**休斯敦火箭队球衣号码：**11

**国家队球衣号码：**13

**2008奥运会号码：**13

**手掌：**19厘米

**净身高：**2.26米

**臂展：**7英尺4.75英寸（2.25米）

**站立摸高：**9英尺7英寸（2.91米）

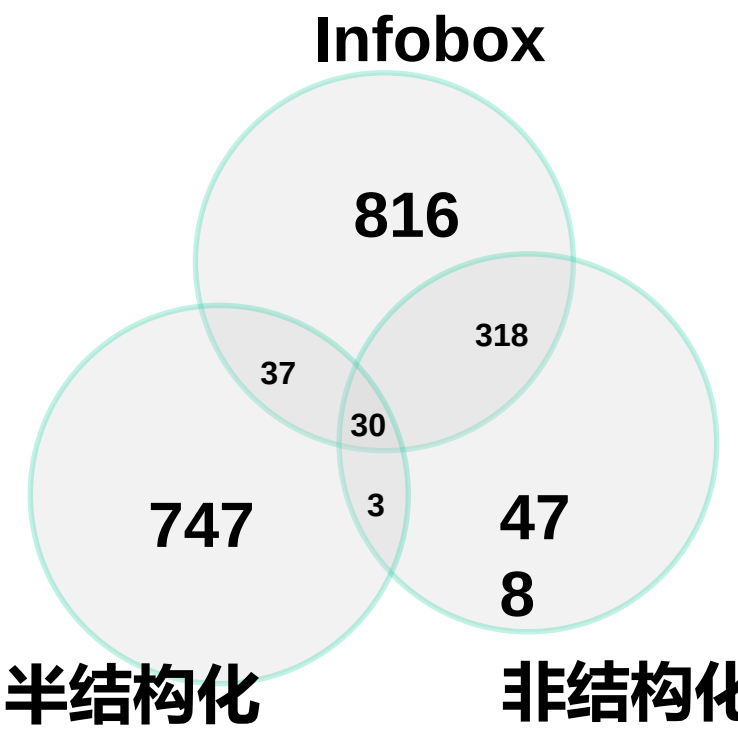
**原地弹跳：**19英寸（约48厘米）

张艺谋，陕西省西安人，中国电影导演，北京奥运会开幕式总导演。在电影学院学的是摄影专业。张艺谋是中国在国际影坛最具影响力的导演，曾多次荣获国际电影节大奖并成功执导北京奥运会开幕式。其代表作《红高粱》被认为是中国电影走向世界的新开始，《英雄》则是开启了中国电影的大片时代。

# 结构化 vs. 半结构化 vs. 非结构化

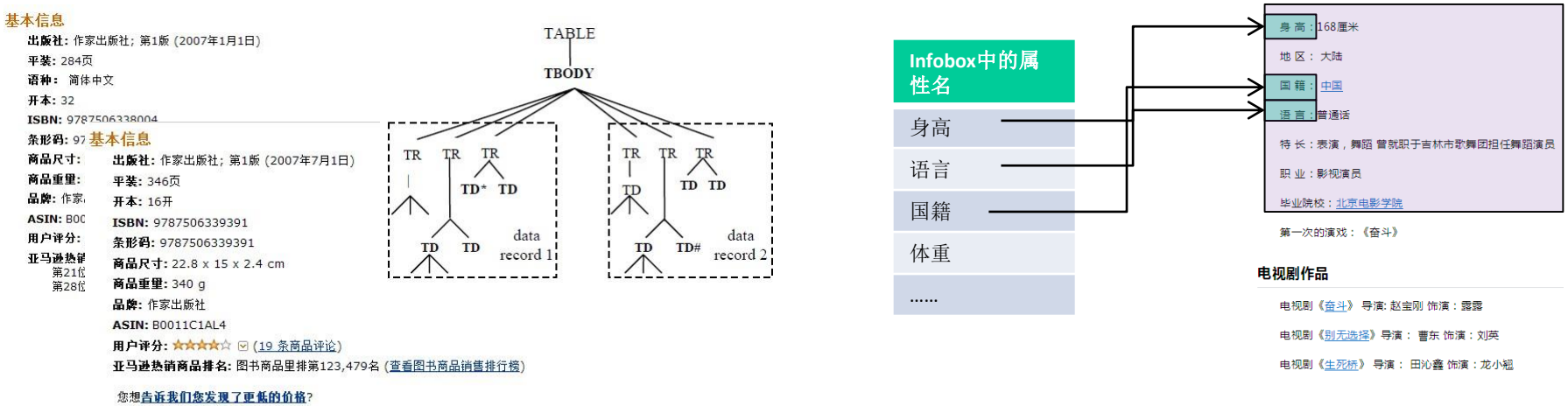
- 随机抽取100篇百科文档（共5类）
  - 对于其中三部分都包含的网页进行了统计

| | 人物 | 地理 | 电影 | 动物 | 图书 |
|---|---|---|---|---|---|
| InfoBox | 87 | 182 | 260 | 183 | 104 |
| 非结构化 | 79 | 147 | 109 | 107 | 36 |
| 半结构化 | 119 | 96 | 327 | 129 | 76 |
| OverLap Info vs. 非 | 62 | 101 | 87 | 72 | 26 |
| OverLap Info vs. 半 | 11 | 11 | 21 | 7 | 17 |
| OverLap 非 vs. 半 | 8 | 7 | 4 | 4 | 10 |
| 三方OverLap | 7 | 7 | 3 | 4 | 9 |



# 结构化与半结构化的关系抽取方法

- 结构化与半结构化文本信息（利用网页结构）
  - 信息块的识别
  - 模板的学习
  - 属性值的抽取




Bootstrapping

# 非结构化文本的关系抽取分类

- 传统关系抽取
  - 评测语料（MUC，ACE, …）
  - 专家制订类别，人工标注语料
- 开放域关系抽取
  - 类别自动获取
  - 语料自动生成
  - 语言证据清晰表达的关系

# 概述

- 关系抽取任务定义
- 传统关系抽取
- 开放域关系抽取

# 传统关系抽取

- 任务
  - 给定实体关系类别，给定语料，抽取目标关系对
- 评测语料（MUC, ACE, KBP, SemEval）
  - 专家标注语料，语料质量高
  - 抽取的目标类别已经定义好
  - 有公认的评价方式

# ACE抽取的目标关系列表

| | PER | ORG | GPE | LOC | FAC | WEA | VEH |
|---|---|---|---|---|---|---|---|
| PER | Per_Social.Bus Per_Social.Family, Per_Social.Lasting, Gen_Aff.Ideology, Gen_Aff.CRRE | Org_Aff.Employment, Org_Aff.Ownership, Org_Aff.Student/Alum, Org_Aff.Sports_Affiliation, Org_Aff.Investor/Shareholder, Org_Aff.Membership, Org_Aff.Founder, Gen_Aff.CRRE | Physical.Located, Physical.Near, Org_Aff.Employment, Org_Aff.Investor/Shareholder, Org_Aff.Founder, Gen_Aff.CRRE | Physical.Located, Physical.Near, Gen_Aff.CRRE | Physical.Located Physical.Near, Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| ORG | | Part_Whole.Subsidiary, Org_Aff.Investor/Shareholder, Org_Aff.Membership | Part_Whole.Subsidiary, Org_Aff.Investor/Shareholder, Gen_Aff.Loc/Origin | Gen_Aff.Loc/Origin | Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| GPE | | Org_Aff.Investor/Shareholder,Org_Aff.Membership, | Physical.Near, Part_Whole.Geographical Org_Aff.Investor/Shareholder | Physical.Near, Part_Whole.Geographical | Agent/Artifact.UOIM | Agent/Artifact.UOIM | Agent/Artifact.UOIM |
| LOC | | | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | | |
| FAC | | | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | Physical.Near, Part_Whole.Geographical | | |
| WEA | | | | | | Part_Whole.Artifact | |
| VEH | | | | | | | Part_Whole.Artifact |

# TAC-KBP抽取的目标关系列表

| Person | Organization | Geo-Political Entity |
|---|---|---|
| alternate names | alternate names | alternate names |
| age | political/religious affiliation | capital |
| birth: date, place | top members/employees | subsidiary orgs |
| death: date, place, cause | number of employees | top employees |
| national origin | members | political parties |
| residences | member of | established |
| spouse | subsidiaries | population |
| children | parents | currency |
| parents | founded by | |
| siblings | founded | |
| other family | dissolved | |
| schools attended | headquarters | |
| job title | shareholders | |
| employee-of | website | |
| member-of | | |
| religion | | |
| criminal charges | | |

# SemEval-2010 Task 8

- 目标关系列表：19种

| 关　系 | 比　例 |
|---|---|
| Other | 280 (31.43%) |
| Cause-Effect | 156 (17.51%) |
| Product-Producer | 114 (12.79%) |
| Entity-Origin | 102 (11.45%) |
| Instrument-Agency | 85 ( 9.54%) |
| Component-Whole | 71 ( 7.97%) |
| Content-Container | 57 ( 6.40%) |
| Entity-Destination | 13 ( 1.46%) |
| Member-Collection | 8 ( 0.90%) |
| Message-Topic | 5 ( 0.56%) |

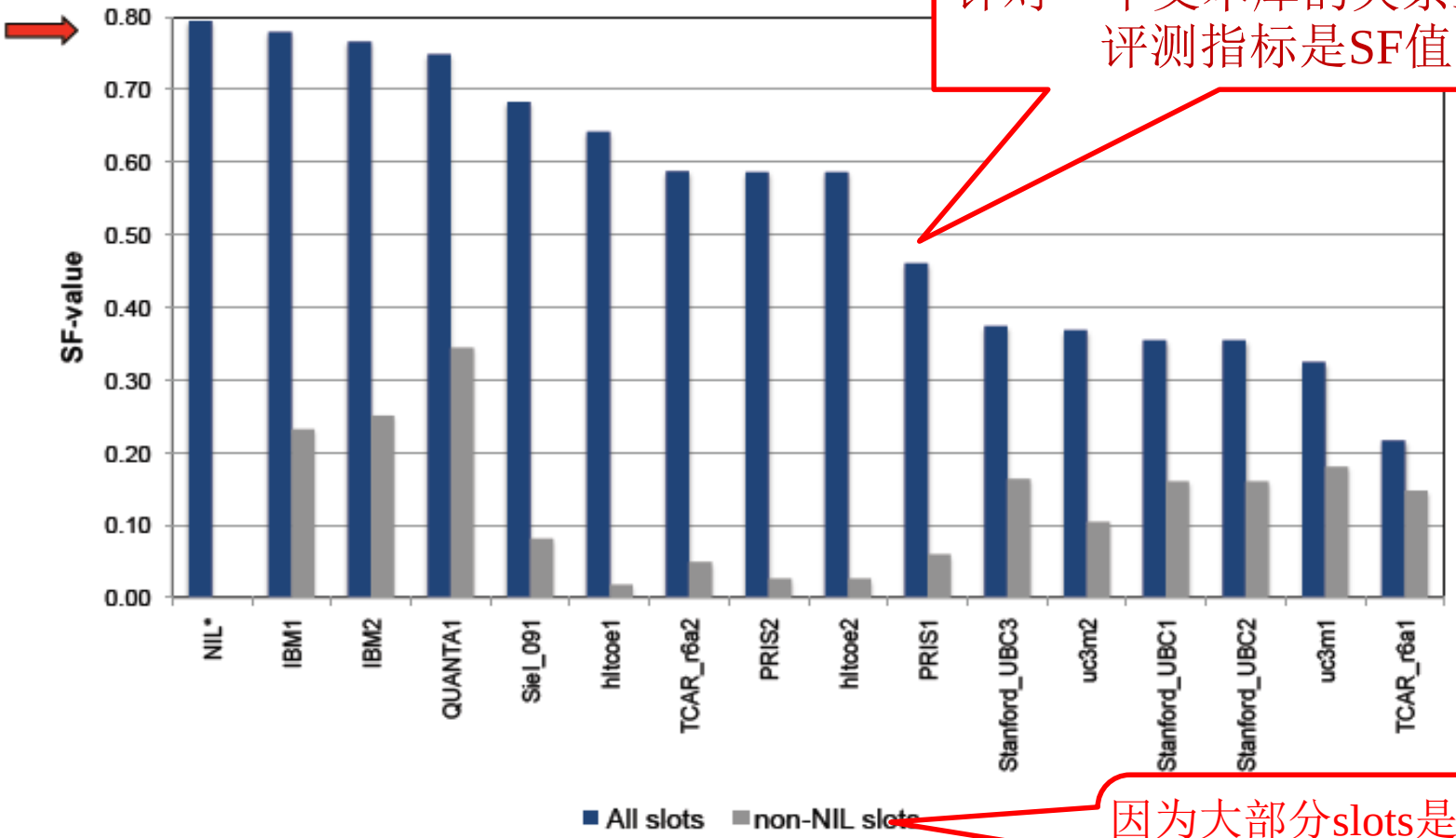
# ACE(2005-2007)评测结果

| 语种 | 语料来源类型 | ACE 2005 | | | | | ACE 2007 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EDR | RDR | TERN | VAL | VDR | EDR | RDR | TERN | VAL | VDR |
| Chinese | Broadcast News | 70.5 | 24.4 | 81.8 | 48.7 | 11.2 | 49.7 | 16.8 | | | |
| | Newswire | 70.1 | 28.6 | 84.3 | 42.1 | 10.8 | 46.9 | 18.7 | 9.9 | | |
| | Weblogs | 67.1 | 26.6 | 86.2 | 71.2 | 2 | 35 | 15.7 | 24.8 | | |
| | 总分 | 69.2 | 26.8 | 83.7 | 49.7 | 10.2 | 45 | 17.6 | 14.8 | | |
| English | Broadcast News | 77.3 | 33.1 | 65.6 | 12.6 | 12.3 | 65.9 | 24.7 | 68.6 | | 12.9 |
| | Newswire | 73.1 | 27.3 | 72.6 | 43.4 | 17.8 | 58.1 | 21.2 | 67.4 | | 15.9 |
| | Weblogs | 71.6 | 20.5 | 62.8 | 38.5 | 15.4 | 52.7 | 18.2 | 54.8 | | 15 |
| | Broadcast Conversation | 72.7 | 16 | 48 | 30 | 6.2 | 50.5 | 11 | [illegible] | | 7.4 |
| | Telephone | 67.7 | 35.6 | 58.6 | 25 | 15.2 | 49.2 | 32.4 | [illegible] | | 11.3 |
| | Usenet | 61.5 | 19.6 | 63.4 | 49.7 | 13.1 | 44 | 19.6 | | | |
| | 总分 | 71.9 | 25.2 | 63.7 | 34.8 | 14.4 | 56.3 | 21.6 | | | |
| Arabic | Broadcast News | | | | | | 51.9 | | | | |
| | Newswire | | | | | | 49.4 | | | | |
| | Weblogs | | | | | | 42.1 | | | | |
| | 总分 | | | | | | 48.8 | | | | |
| Spanish | Newswire | | | | | | 51 | | 46.5 | | |
| | 总分 | | | | | | | | 46.5 | | |

针对每个句子的关系识别，评测指标是F值

(http://www.nist.gov/speech/tests/ace/ace05/doc/acee05eval_official_results_20060110.htm
http://www.nist.gov/speech/tests/ace/ace05/doc/acee05eval_official_results_20070402.htm)

# TAC-KBP 2009 Slot Filling Track评测结果




针对一个文本库的关系抽取，评测指标是SF值

因为大部分slots是空的，所以non-NIL指标反映了系统的性能

(Paul McNamee, 2009)

# KBP 2013 Slot Filling Track评测结果

| | Diagnostic Scores | | | Official Scores | | |
|---|---|---|---|---|---|---|
| | Recall | Precision | F1 | Recall | Precision | F1 |
| lsv | **32.93** | 38.50 | 35.50 | **33.17** | 42.53 | **37.28** |
| ARPANI* | 29.10 | 47.83 | **36.18** | 27.45 | 50.38 | 35.54 |
| RPI-BLENDER | 30.62 | 38.19 | 33.98 | 29.02 | 40.73 | 33.89 |
| PRIS2013 | 27.82 | 35.33 | 31.13 | 27.59 | 38.87 | 32.27 |
| BIT | 22.06 | 57.86 | 31.94 | 21.73 | 61.35 | 32.09 |
| Stanford | 28.46 | 32.30 | 30.26 | 28.41 | 35.86 | 31.70 |
| NYU | 17.35 | 50.70 | 25.85 | 16.76 | 53.83 | 25.56 |
| UWashington | 10.31 | **59.72** | 17.59 | 10.29 | **63.45** | 17.70 |
| CMUML | 10.63 | 28.79 | 15.53 | 10.69 | 32.30 | 16.07 |
| SAFT_KRes | 13.43 | 12.43 | 12.91 | 14.99 | 15.67 | 15.32 |
| UMass_IESL | 18.47 | 9.43 | 12.48 | 18.46 | 10.88 | 13.69 |
| utaustin | 7.91 | 21.85 | 11.62 | 8.11 | 25.16 | 12.26 |
| UNED | 9.11 | 15.08 | 11.36 | 9.33 | 17.59 | 12.19 |
| Compreno | 13.19 | 8.69 | 10.48 | 12.74 | 9.74 | 11.04 |
| TALP_UPC | 9.67 | 6.54 | 7.81 | 9.81 | 7.69 | 8.62 |
| IIRG | 3.20 | 7.38 | 4.46 | 2.86 | 7.72 | 4.17 |
| SINDI | 2.80 | 7.26 | 4.04 | 2.59 | 7.84 | 3.89 |
| CohenCMU | 3.68 | 1.69 | 2.32 | 3.68 | 1.98 | 2.57 |
| LDC | 58.35 | 83.81 | 68.80 | 57.08 | 85.60 | 68.49 |

# SemEval-2010 Task 8 评测结果

| System | TD1 | TD2 | TD3 | TD4 | Acc TD4 | Rank | Best Cat | Worst Cat-9 |
|---|---|---|---|---|---|---|---|---|
| Baseline | 33.04 | 42.41 | 50.89 | 57.52 | 50.0 | - | MC (75.1) | IA (28.0) |
| ECNU-SR-1 | 52.13 | 56.58 | 58.16 | 60.08 | 57.1 | 4 | CE (79.7) | IA (32.2) |
| ECNU-SR-2 | 46.24 | 47.99 | 69.83 | 72.59 | 67.1 | | CE (84.4) | IA (52.2) |
| ECNU-SR-3 | 39.89 | 42.29 | 65.47 | 68.50 | 62.0 | | CE (83.4) | IA (46.5) |
| ECNU-SR-4 | 67.95 | 70.58 | 72.99 | 74.82 | 70.5 | | CE (84.6) | IA (61.4) |
| *ECNU-SR-5* | 49.32 | 50.70 | 72.63 | 75.43 | 70.2 | | CE (85.1) | IA (60.7) |
| ECNU-SR-6 | 42.88 | 45.54 | 68.87 | 72.19 | 65.8 | | CE (85.2) | IA (56.7) |
| ECNU-SR-7 | 58.67 | 58.87 | 72.79 | 75.21 | 70.2 | | CE (86.1) | IA (61.8) |
| FBK_IRST-6C32 | 60.19 | 67.31 | 71.78 | 76.81 | 72.4 | 2 | ED (82.6) | IA (69.4) |
| FBK_IRST-12C32 | 60.66 | 67.91 | 72.04 | 76.91 | 72.4 | | MC (84.2) | IA (68.8) |
| FBK_IRST-12VBC32 | 62.64 | 69.86 | 73.19 | 77.11 | 72.3 | | ED (85.9) | PP (68.1) |
| FBK_IRST-6CA | 60.58 | 67.14 | 71.63 | 76.28 | 71.4 | | CE (82.3) | IA (67.7) |
| FBK_IRST-12CA | 61.33 | 67.80 | 71.65 | 76.39 | 71.4 | | ED (81.8) | IA (67.5) |
| *FBK_IRST-12VBCA* | 63.61 | 70.20 | 73.40 | 77.62 | 72.8 | | ED (86.5) | IA (67.3) |
| *FBK_NK-RES1* | 55.71* | 64.06* | 67.80* | 68.02 | 62.1 | 7 | ED (77.6) | IA (52.9) |
| FBK_NK-RES2 | 54.27* | 63.68* | 67.08* | 67.48 | 61.4 | | ED (77.4) | PP (55.2) |
| FBK_NK-RES3 | 54.25* | 62.73* | 66.11* | 66.90 | 60.5 | | MC (76.7) | IA (56.3) |
| FBK_NK-RES4 | 44.11* | 58.85* | 63.06* | 65.84 | 59.4 | | MC (76.1) | IA/PP (58.0) |
| *ISI* | 66.68 | 71.01 | 75.51 | 77.57 | 72.7 | 3 | CE (87.6) | IA (61.5) |
| *ISTI-1* | 50.49* | 55.80* | 61.14* | 68.42 | 63.2 | 6 | ED (80.7) | PP (53.8) |
| ISTI-2 | 50.69* | 54.29* | 59.77* | 66.65 | 61.5 | | ED (80.2) | IA (48.9) |
| *JU* | 41.62* | 44.98* | 47.81* | 52.16 | 50.2 | 9 | CE (75.6) | IA (27.8) |
| *SEKA* | 51.81 | 56.34 | 61.10 | 66.33 | 61.9 | 8 | CE (84.0) | PP (43.7) |
| TUD-base | 50.81 | 54.61 | 56.98 | 60.50 | 56.1 | 5 | CE (80.7) | IA (31.1) |
| TUD-wp | 55.34 | 60.90 | 63.78 | 68.00 | 63.5 | | ED (82.9) | IA (44.1) |
| TUD-comb | 57.84 | 62.52 | 66.41 | 68.88 | 64.6 | | CE (83.8) | IA (46.8) |
| *TUD-comb-θ* | 58.35 | 62.45 | 66.86 | 69.23 | 65.4 | | CE (83.4) | IA (46.9) |
| *UNITN* | 16.57* | 18.56* | 22.45* | 26.67 | 27.4 | 10 | ED (46.4) | PP (0) |
| *UTD* | **73.08** | **77.02** | **79.93** | **82.19** | 77.9 | 1 | CE (89.6) | IA (68.5) |

# 传统关系抽取方法

- 目前主要采用统计机器学习的方法，将关系实例转换成高维空间中的特征向量或直接用离散结构来表示，在标注语料库上训练生成分类模型，然后再识别实体间关系
  - 基于特征向量方法：最大熵模型(Kambhatla 2004)和支持向量机(Zhao et al., 2005；Zhou et al., 2005; Jiang et al., 2007)等
  - 基于核函数的方法：浅层树核（Zelenko et al., 2003)、依存树核（Culotta et al., 2004)、最短依存树核（Bunescu et al., 2005)、卷积树核（Zhang et al., 2006；Zhou et al., 2007）
  - 基于神经网络的方法：递归神经网络（ Socher et al., 2012)、基于矩阵空间的递归神经网络（ Socher et al., 2012)、卷积神经网络（Zeng et al., 2014）

# 传统关系抽取方法

- 基于特征向量方法:
  - 主要问题：如何获取各种有效的词法、句法、语义等特征，并把它们有效地集成起来，从而产生描述实体语义关系的各种局部特征和简单的全局特征
  - 特征选取：从自由文本及其句法结构中抽取出各种词汇特征以及结构化特征
    - 实体词汇及其上下文特征
    - 实体类型及其组合特征（PER，LOC等）
    - 交叠特征（两个实体或词组块是否在同一个名词短语、动词短语或者介词短语之中、两个实体或者词组块之间单词的个数等）
    - 句法树特征（连接两个实体的语法路径）

# 传统关系抽取方法

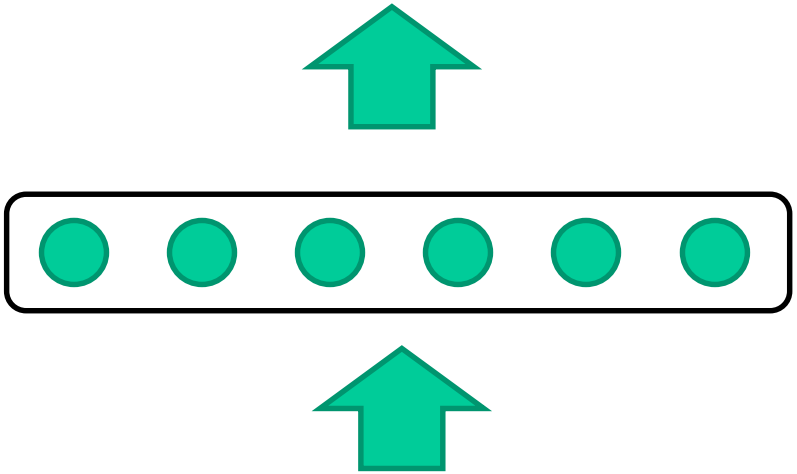
- 基于核函数方法:
  - 主要问题：如何有效挖掘反映语义关系的结构化信息及如何有效计算结构化信息之间的相似度
  - 卷积树核：用两个句法树之间的公共子树的数目来衡量它们之间的相似度
    - 标准的卷积树核(CTK)
      - 在计算两棵子树的相似度时，只考虑子树本身，不考虑子树的上下文信息
    - 上下文相关卷积树核函数(CS-CTK)
      - 在计算子树相似度时，同时考虑子树的祖先信息，如子树根结点的父结点、祖父结点信息，并对不同祖先的子树相似度加权平均

# 传统关系抽取方法

- 基于神经网络的方法:
  - 主要问题：如何设计合理的网络结构，从而捕捉更多的信息，进而更准确地完成关系的抽取
  - 网络结构：不同的网络结构捕捉文本中不同的信息
    - 递归神经网络（RNN）
      - 网络的构建过程更多的考虑到句子的句法结构，但是需要依赖复杂的句法分析工具
    - 卷积神经网络（CNN）
      - 通过卷积操作完成句子级信息的捕获，不需要复杂的NLP工具
    - 循环神经网络（RNN）
      - 通过循环神经网络建模词语之间的依赖关系，自动捕获句子级信息

# 自然语言处理过程

分类, 聚类, （结构）预测, 排序...



伦齐在宪改公投惨败后，宣布有意辞职。

特征
- 符号特征：特征抽取
- 数值特征：表示学习

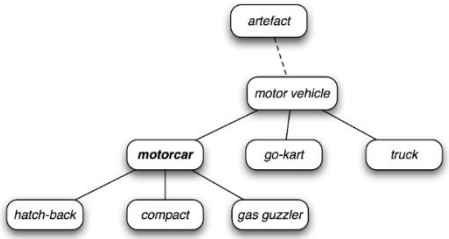
# 自然语言的表示学习

- 基本语言单元的语义表示
  - 词的表示
  - 分布式原则
- 更大语言单元的语义表示
  - 句子的表示
  - 语义组合原则

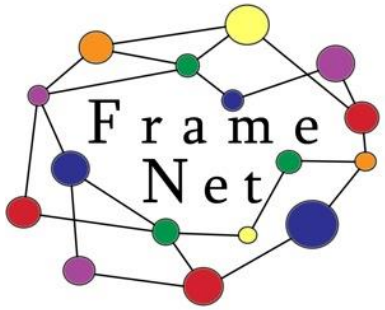
深度学习的本质：用数值（向量，矩阵等）表示各个语言单元的语义（特征）

# 词表示

- 一般认为，词是语言处理中最基本的语言单元
- 词以及词间关系的表示和建模是NLP任务中重要基础工作




WordNet [Miller 1995]



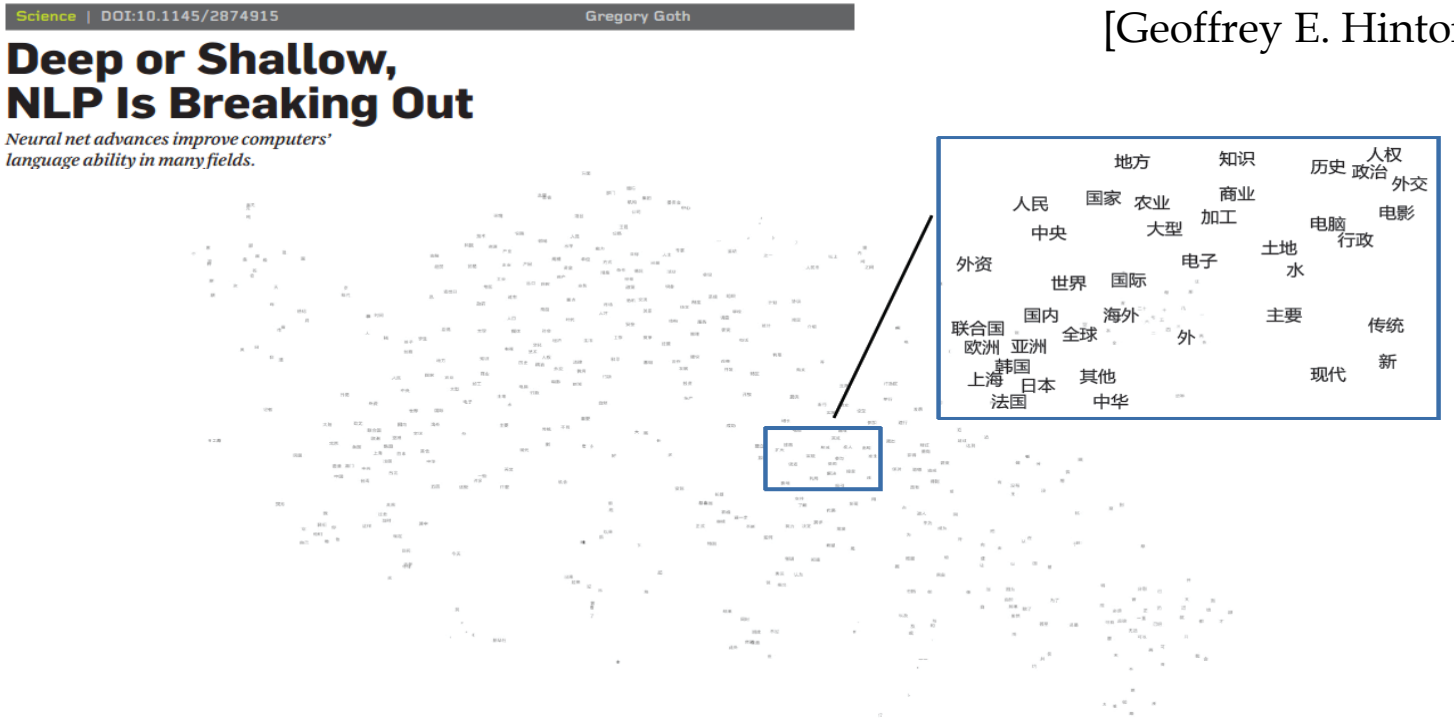

FrameNet [Baker 1998]

人工方法：刻画词之间多维度关系，例如：聚合关系，组合关系
深度学习：用数值化的方法表示词的语义信息

# 词的分布表示

- 深度学习框架下，词的语义表示为向量的形式，也就是分布表示。

The first thing you do with a word symbol is you convert it to a word vector. And you learn to do that, you learn for each word how to turn a symbol into a vector, say, 300 components, and after you’ ve done learning, you’ ll discover the vector for Tuesday is very similar to the vector for Wednesday.




[Geoffrey E. Hinton. 2015]

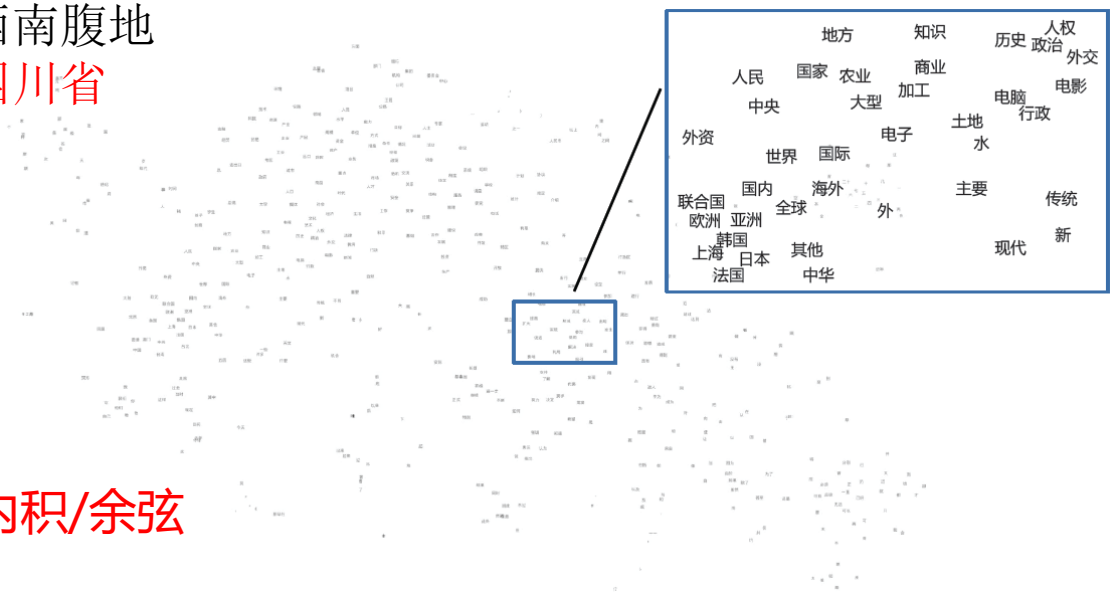
# 词的分布表示

- Harris和firth的分布假说：上下文相似的词 → 词义相似
  - 词的语义可以通过大规模语料库统计获得
  - 词语的语义相似度可以度量

[Harris 1954]

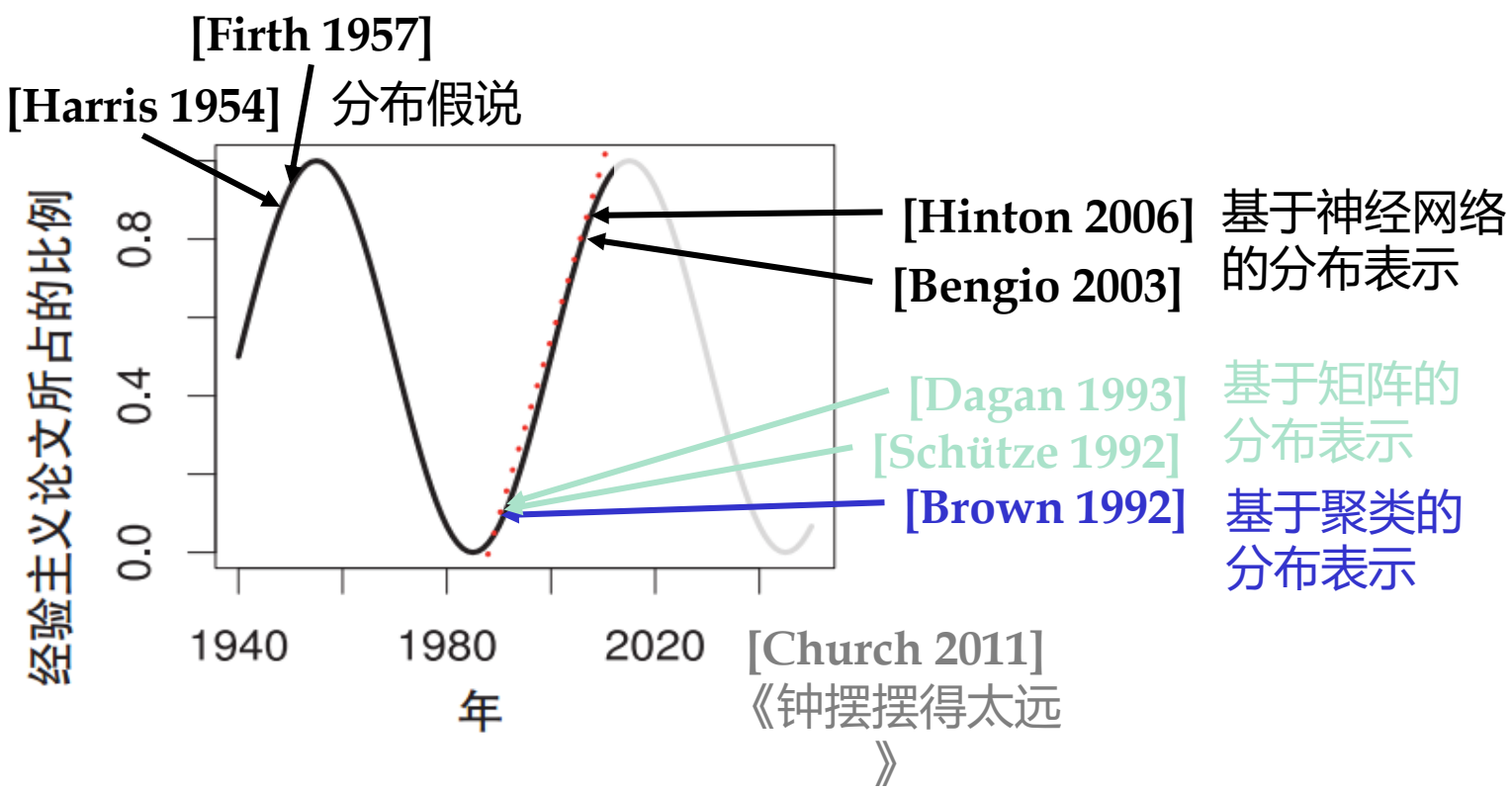
江西省地处中国东南部
携程带你玩转江西省

四川省地处中国西南腹地
携程带你玩转四川省




向量 & 内积/余弦

# 词的分布表示：历史的角度



# 词表示学习

- 上下文的表示：文档、词、n元词组
- 相似度的衡量：向量的内积（余弦）
- 词间关系一般分为聚合关系和组合关系，分布式表示学习常常从聚合关系的角度去建模

Syntagmatic Relation

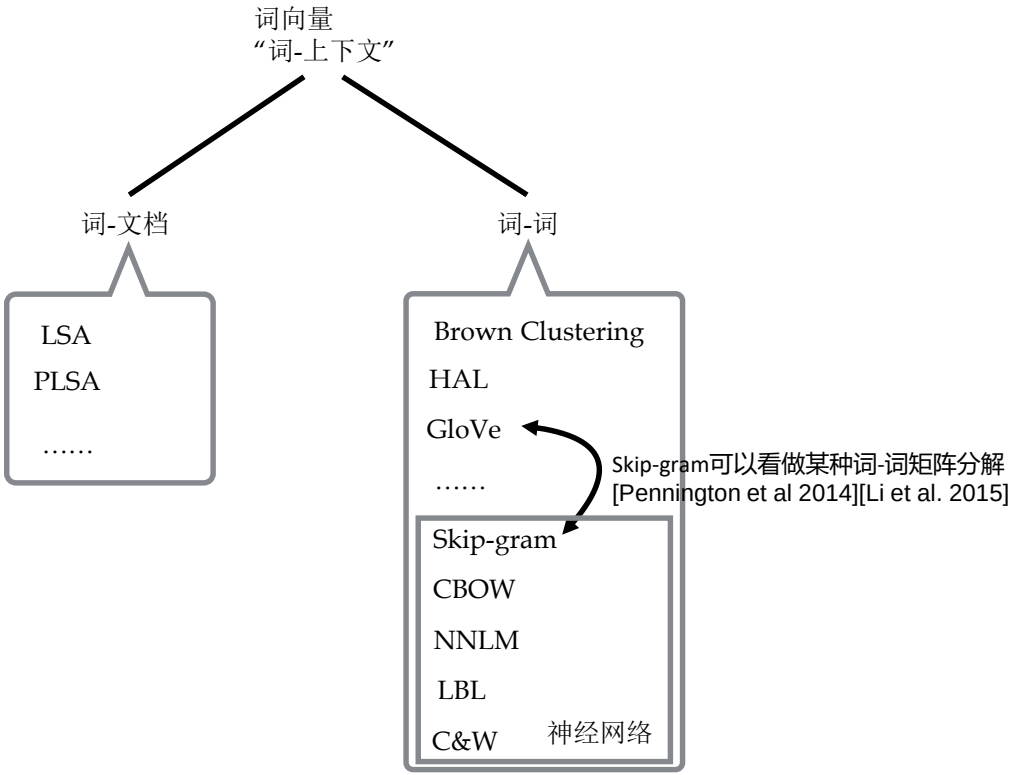
清华 校园 景色 很好

中科院 几乎 没有 校园

Paradigmatic Relation

# 词表示学习方法

- 传统方法：也是从分布的角度建模，但是没有用到神经网络的方法
  - ✓ LSA：矩阵分解
  - ✓ Brown Clustering：聚类
- 神经网络：NNLM, Skip-gram, CBOW, GloVe，LBL，C&W



# NNLM: Neural Network Language Model

- 三层前馈神经网络，以前n-1个词作为输入，第n个词作为输出，进而完成网络参数的优化，实现词向量的表示学习
  - 拼接输入词词向量

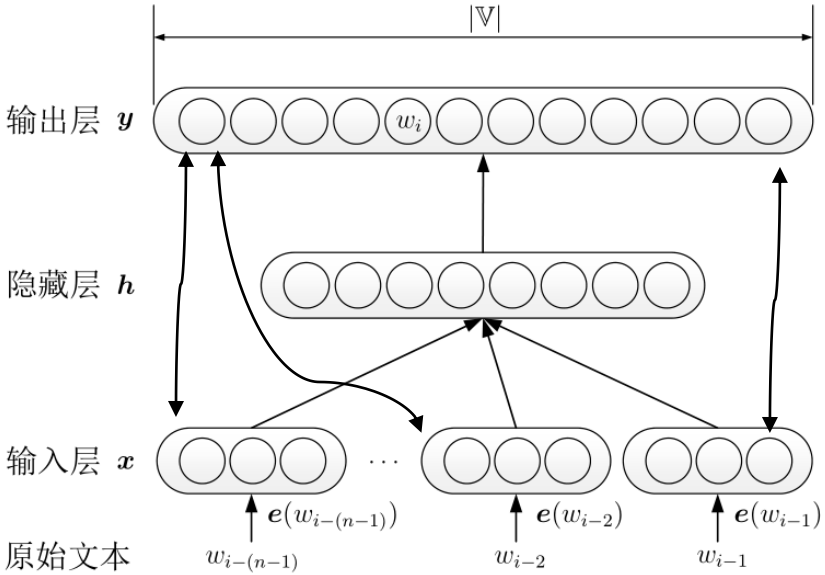
$$\boldsymbol{x} = [\boldsymbol{e}(w_{i-(n-1)}); \ldots; \boldsymbol{e}(w_{i-2}); \boldsymbol{e}(w_{i-1})]$$

  - 隐藏层

$$\boldsymbol{h} = \tanh(\boldsymbol{b}^{(1)} + H\boldsymbol{x})$$

  - 输出层

$$\boldsymbol{y} = \boldsymbol{b}^{(2)} + W\boldsymbol{x} + U\boldsymbol{h}$$



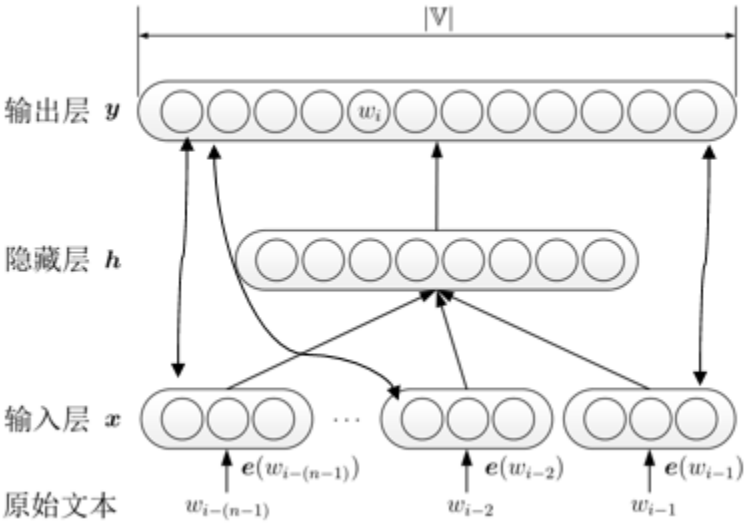

$$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{\exp(\boldsymbol{y}(w_i))}{\sum_{k=1}^{|\mathbb{V}|} \exp(\boldsymbol{y}(v_k))}$$

通过调节输入词向量、输出词向量以及网络的参数，使得网络每次都能正确地预测出第N个词

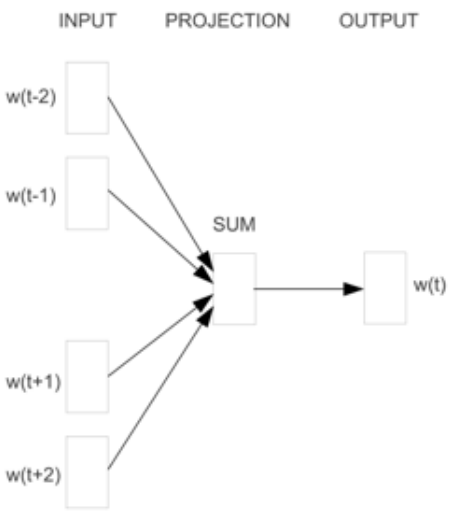
# CBOW & Skip-gram

- Continuous Bag of Word
- 特点：
  - 去除隐藏层
  - 去除词序

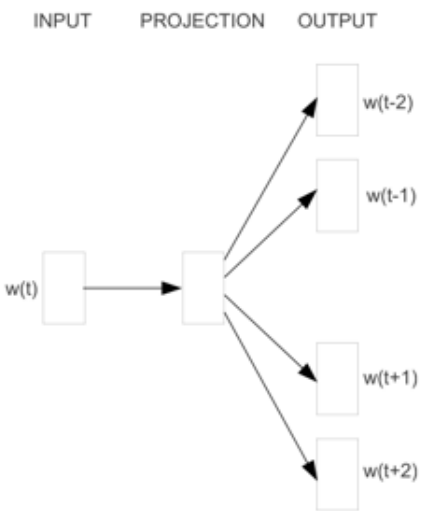
研表究明，汉字序顺并不定一影阅响读！事证实明了也许当你看这完句话之后才发字现都乱是的。




NNLM




CBOW
用周边词预测当前词




Skip-gram
用当前词预测周边的词

# 工具示例

- Word2vec：Google在2013年年中开源的一款将词表征为实数值向量的高效工具，采用的模型有CBOW和Skip-Gram两种。
- 代码链接：https://code.google.com/p/word2vec/
- 输入文本：英文（token后的文本）中文（分词后的文本）
- 输出：指定维度的词向量

```
,
the
.
of
a
and
to
in
''
that
```

```
0.556314,0.024821,-0.486807,1.029960,0.431357,0.831494,-0.327807,0.066419,-
0.120521,2.321336,-0.976254,2.738440,0.458854,-0.919605,-0.237615,1.961676,
0.389097,0.664733,-1.738388,1.373171,0.987761,0.429780,-0.178071,0.069103,-
0.204649,1.058529,-1.495981,1.302744,-0.883764,-0.671500,1.039746,1.182477,
0.631253,-0.997995,-0.646097,2.619554,-1.440681,-0.128711,-0.336226,2.46306
1.517981,1.651188,-0.554875,0.146960,0.805013,-0.004309,0.383949,1.272646,-
1.078778,1.145040,-2.936579,2.731101,1.569286,0.121215,-1.080063,0.769050,(
0.051939,0.881635,-1.327522,1.980237,0.499068,0.391517,1.666283,2.997930,1.
1.111387,-0.773637,-1.408212,2.719081,0.558309,0.661429,-2.004113,-2.10238
0.892893,1.864152,-2.325571,3.441925,0.562785,0.590744,-1.772657,0.423680,-
```

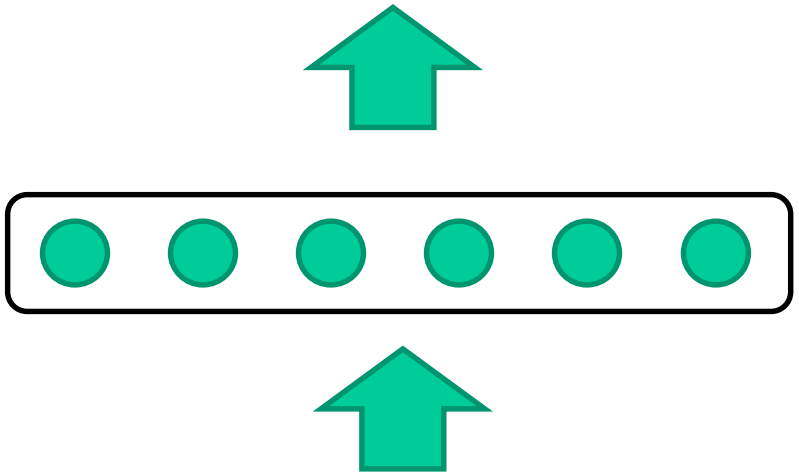
# 自然语言的表示学习

- 基本语言单元的语义表示
  - 词的表示
  - 分布式原则

- 更大语言单元的语义表示
  - 句子的表示
  - 语义组合原则

# 组合语义

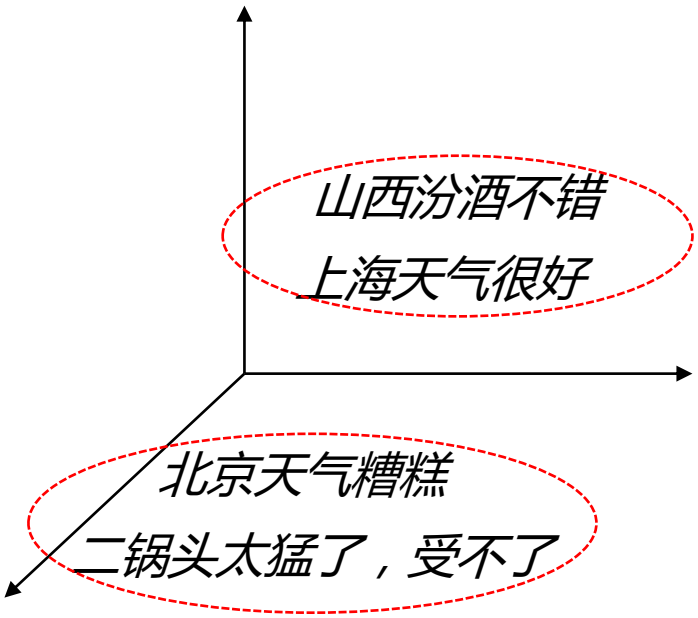
分类, 聚类, 预测, 排序…



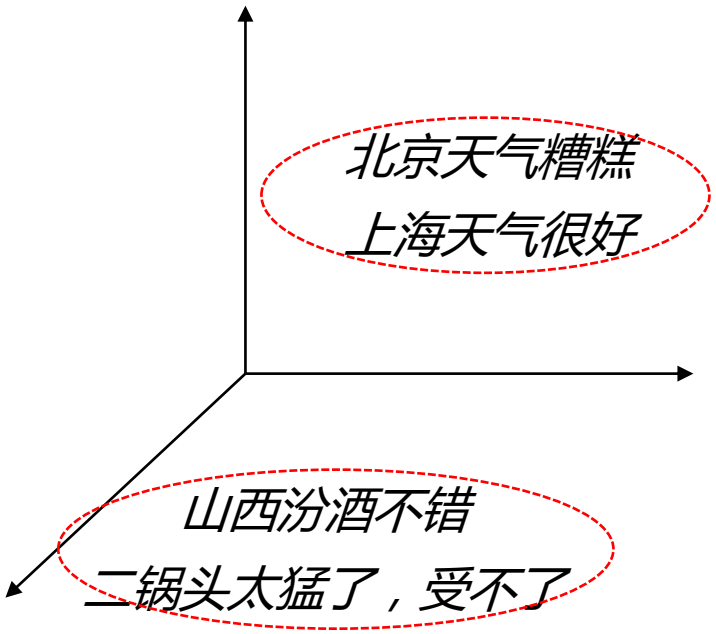
伦齐在宪改公投惨败后，宣布有意辞职。

- 一个复杂对象的意义是由其各组成部分的意义以及它们的组合规则来决定
- The meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.

# 意义的组合和具体任务相关




情感分类

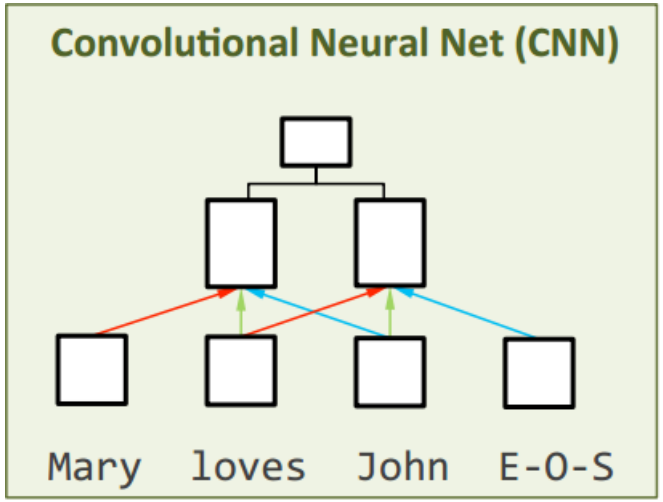



主题分类

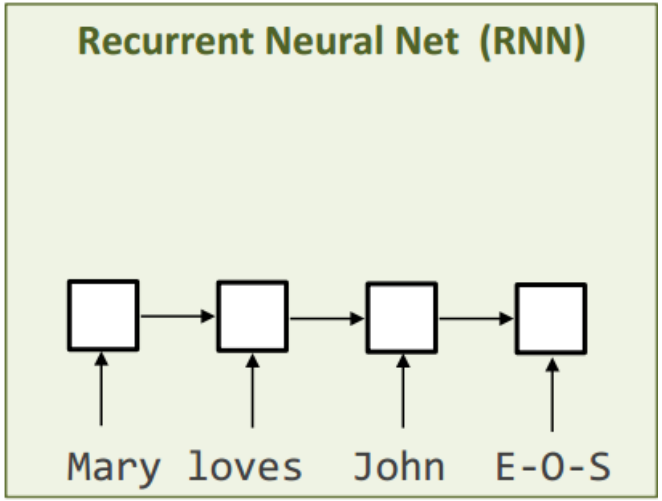
# 语义组合模型

- 基于词语等基本语义单元表示句子等更大语义单元的过程
- 语义组合模型就是要建模句子的句法/语义结构

两类最基本语义组合模型



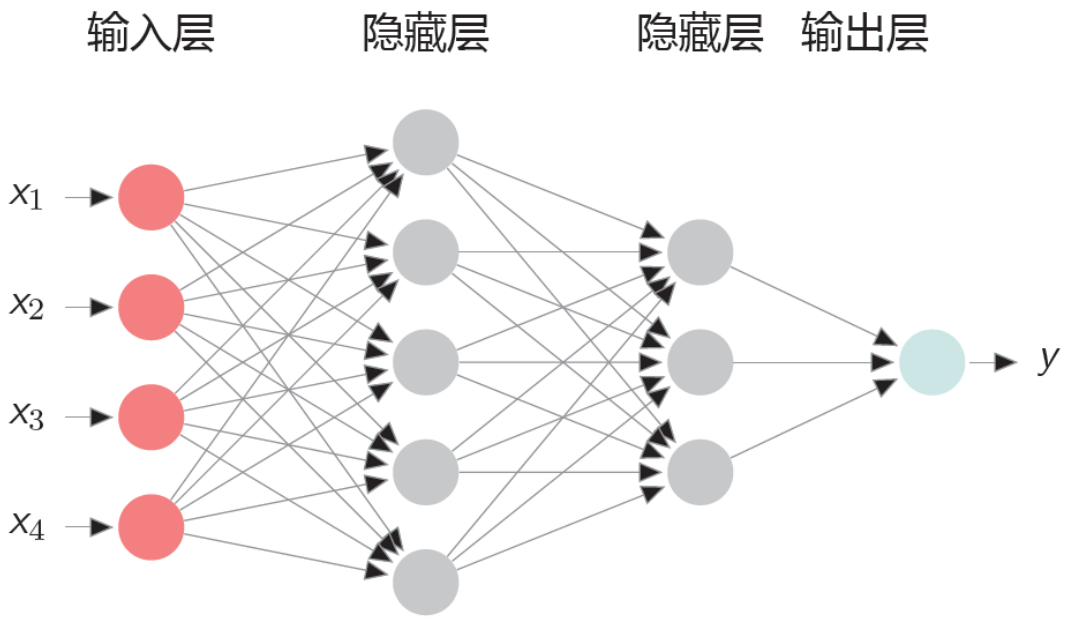

局部空间的特征提取




词语之间的时序关系

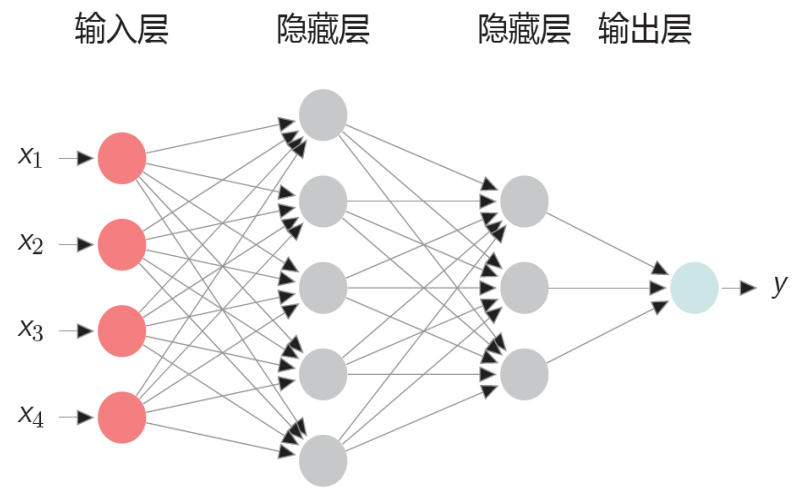
# 基础：多层前馈神经网络

- 在前馈神经网络中，各神经元分别属于不同的层。整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。

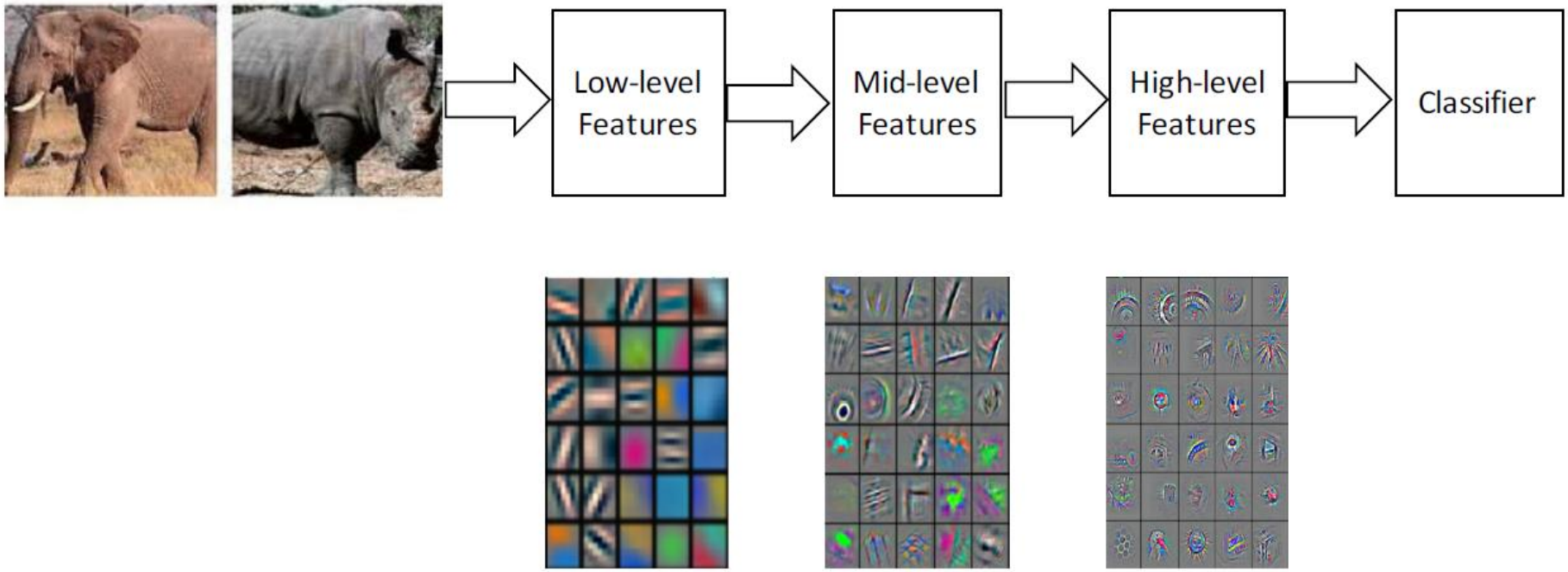



多层前馈神经网络

# 基础：多层前馈神经网络

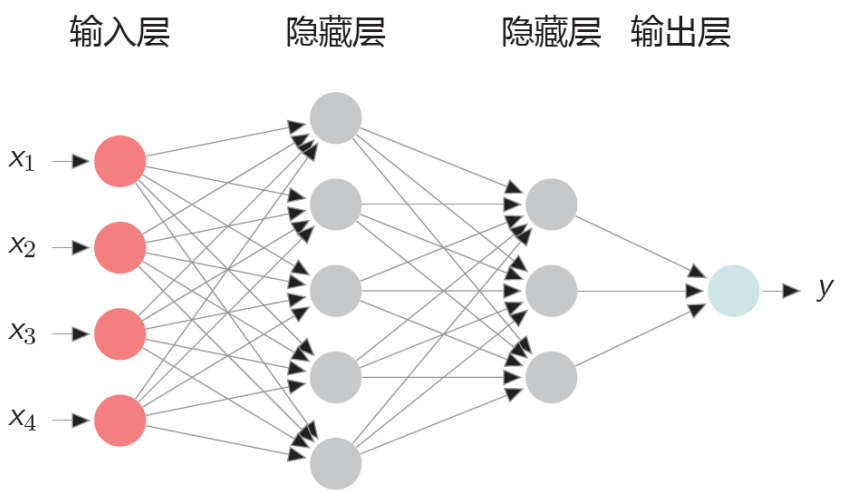



## 示例



# 基础：多层前馈神经网络

- 全连接层：相邻两层的任何两个不同层神经元之间都存在连接。在全连接前馈神经网络中
- 如果第l 层有$n^l$个神经元，*L*表示神经网络的层数
- 则参数个数：$\prod_{l=1}^{L} n^l$
- 权重矩阵参数非常多，训练效率低下
- 数据不足时，欠学习



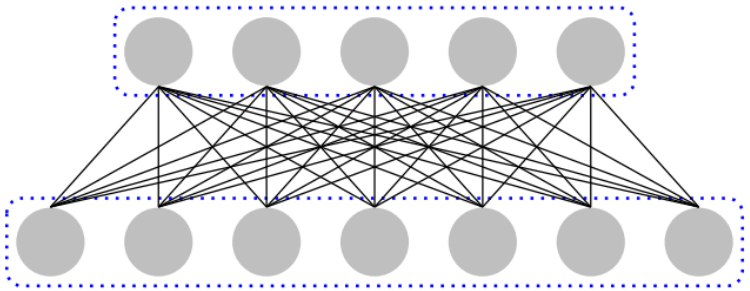

多层前馈神经网络

# 卷积神经网络

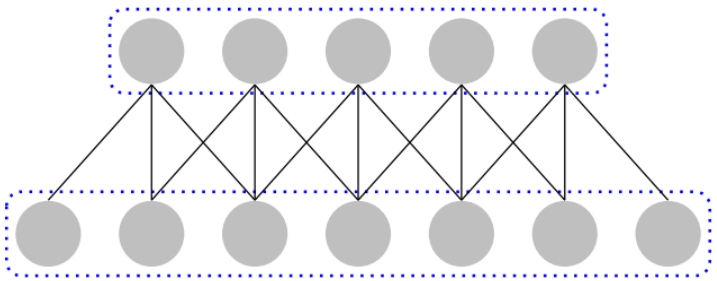
- Convolutional Neural Networks，CNN
- 是一种前馈神经网络
- 是受生物学上感受野（Receptive Field）的机制而提出的。
  - 感受野主要是指听觉系统、感觉系统和视觉系统中神经元的一些性质。
    - 例如：在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。
- 特性：卷积神经网络有三个结构上的特性
  - 局部连接
  - 权重共享
  - 采样

这些特性使得卷积神经网络具有一定程度上的平移、缩放和扭曲不变性

# 全连接 vs 卷积



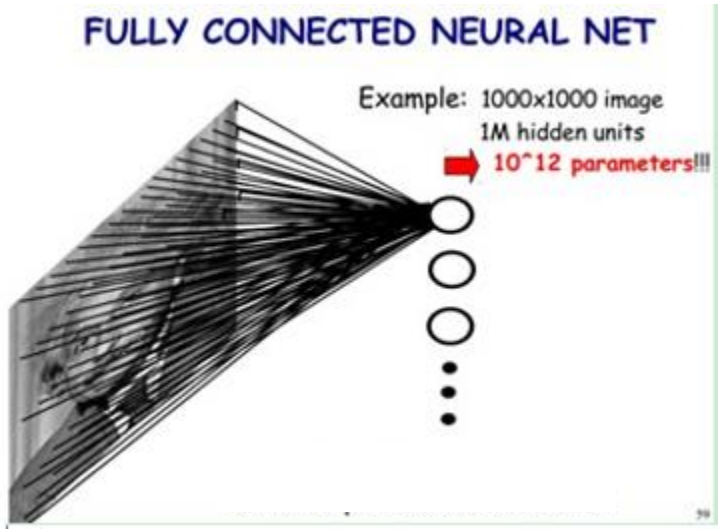
全连接

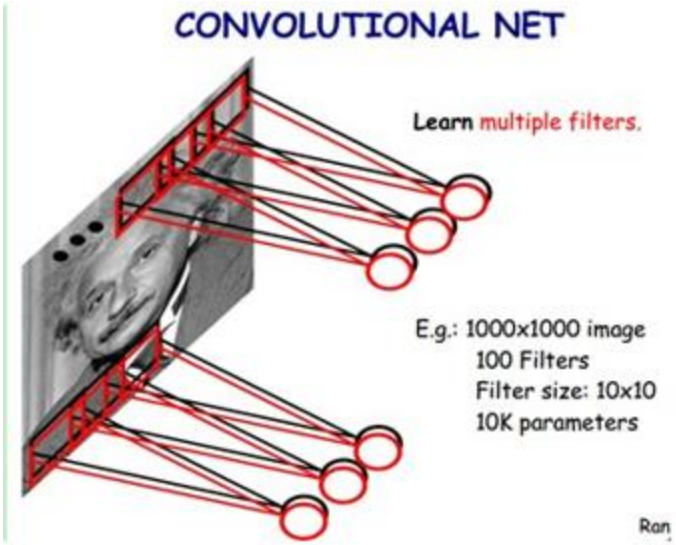


CNN

- 不同
  - 全连接 vs 局部连接
  - 权重独立 vs 权重共享
  - 一般不采样 vs 一般采样

# 例子：参数规模（全连接 vs 卷积）




$1000*1000*1000000=10^{12}$

$10*10*100=10^{4}$

- 左边的图表示全连接，输入为1000*1000，假设下一层神经元的数目为1M，参数规模为$10^{12}$。
- 右边的图是卷积网络示意，当一个滤波器的大小为10*10时，一个滤波器处理图片不同区域时权重共享，所以卷积结果规模为991*991，已经接近1M；即使使用100个滤波器，参数规模也为$10*10*100=10^{4}$，参数个数远小于全连接的情况。

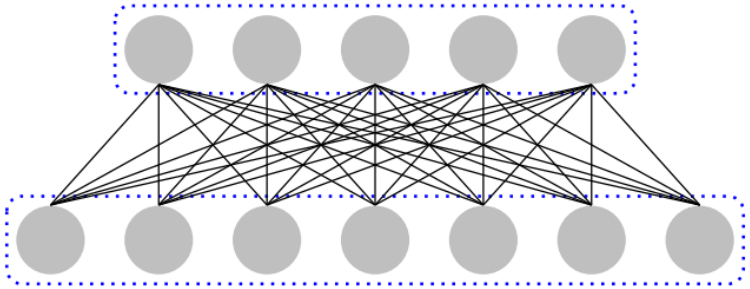
# 一维卷积

- 信号X，信号长度n，X可以表示为（$x_1$，$x_2$，…，$x_n$）
- $x_{i:i+j}$表示（$x_i$，$x_{i+1}$，…，$x_{i+j}$）
- 滤波器（filter）W，滤波器长度为m，W可以表示为（$w_1, w_2, \ldots, w_m$）
- 则卷积后的输出值Y中的每个位置可以如下表示

$$y_i = f(Wx_{i:i+m-1} + b) = f(\sum_{k=1}^{m} w_k x_{\mathrm{i}+k-1} + \mathrm{b})$$

偏置

# 全连接 vs 卷积

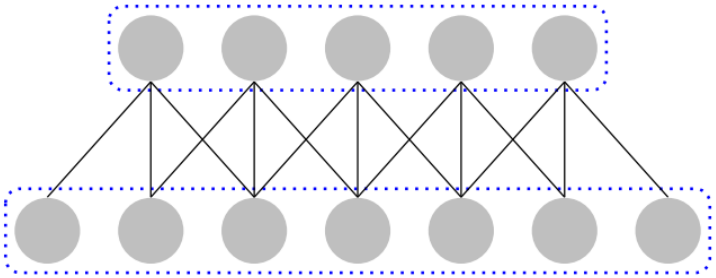


全连接

$$y_i = f(W_i x_{1:n} + b) = f(\sum_{k=1}^{n} w_{ik} x_k + b)$$

n是输入层神经元个数

m是滤波器窗口大小



CNN

局部连接

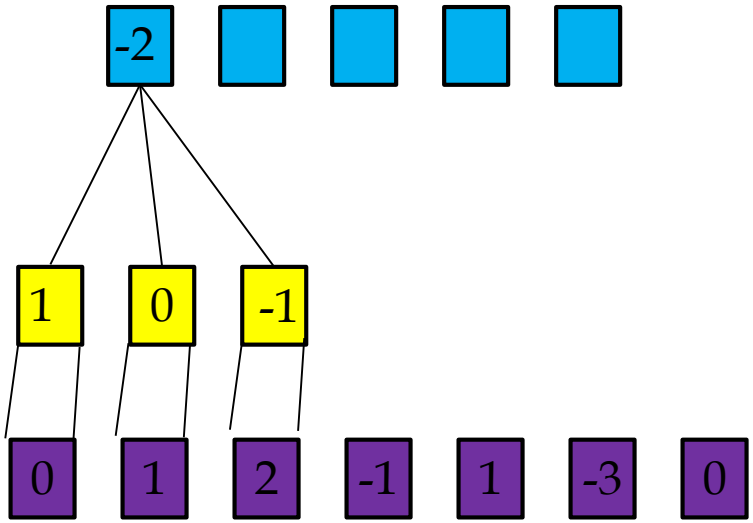
权重共享

$$y_i = f(W x_{i:i+m-1} + b) = f(\sum_{k=1}^{m} w_k x_{i+k-1} + b)$$

f是神经元的激活函数，一般取非线性变换函数

# 一维卷积示例

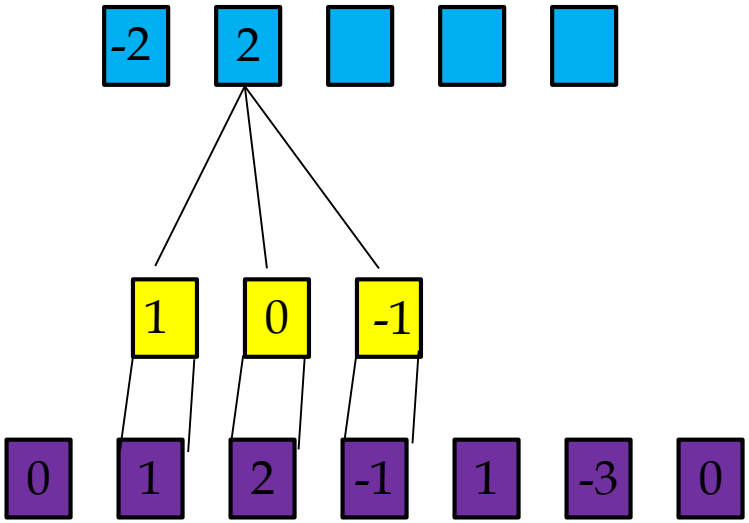
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

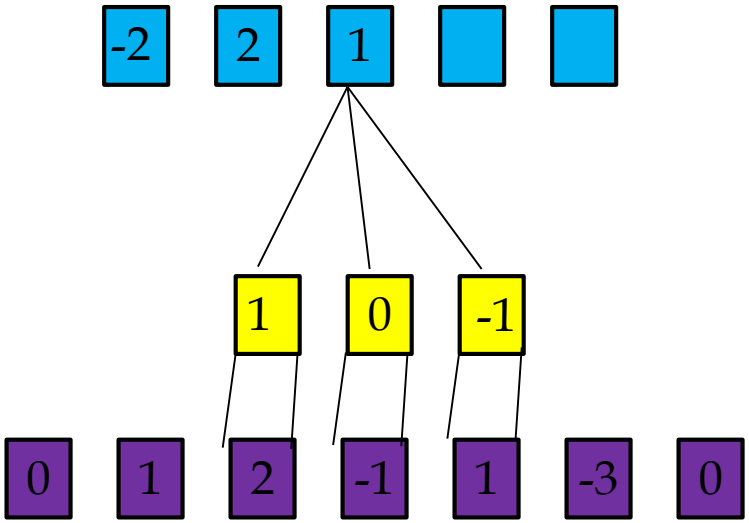
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

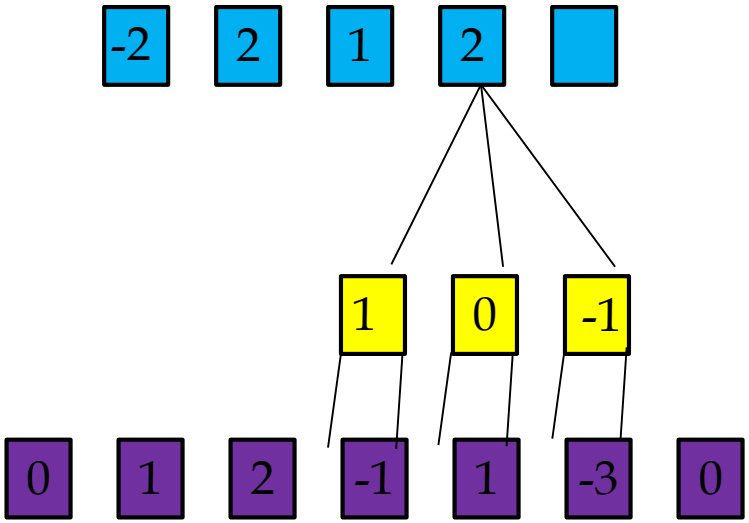
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

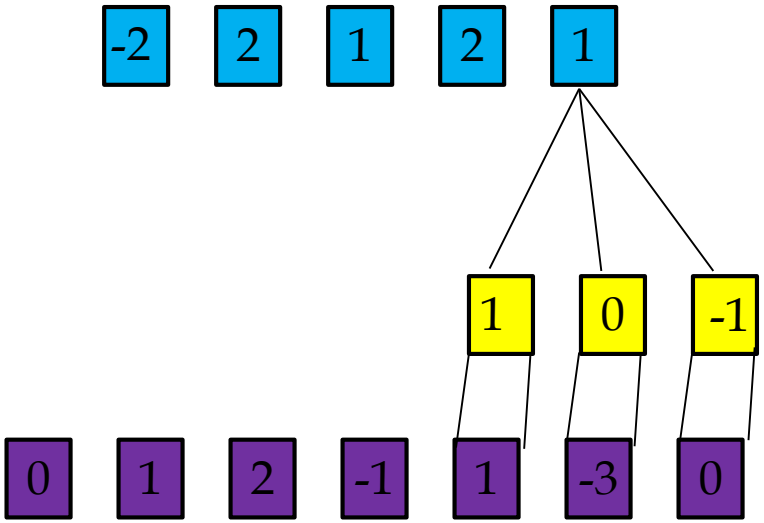
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

# 一维卷积示例

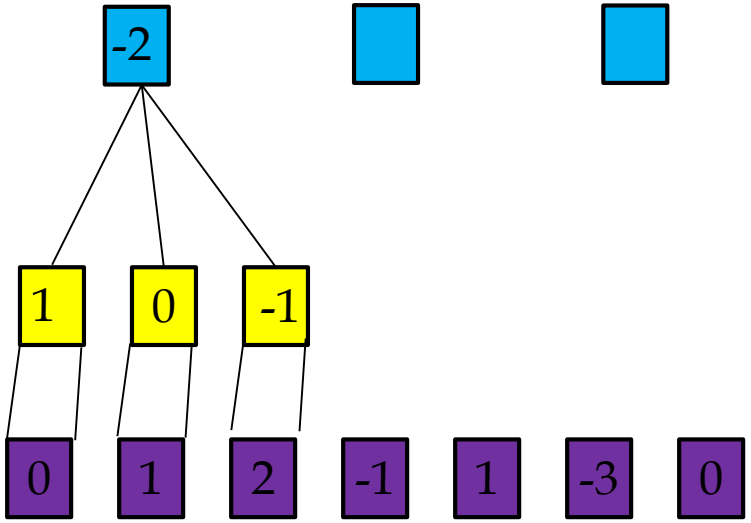
- 输入：X（0，1，2，-1，1，-3，0）
- 滤波器f（1,0,-1）




步长为1的卷积过程

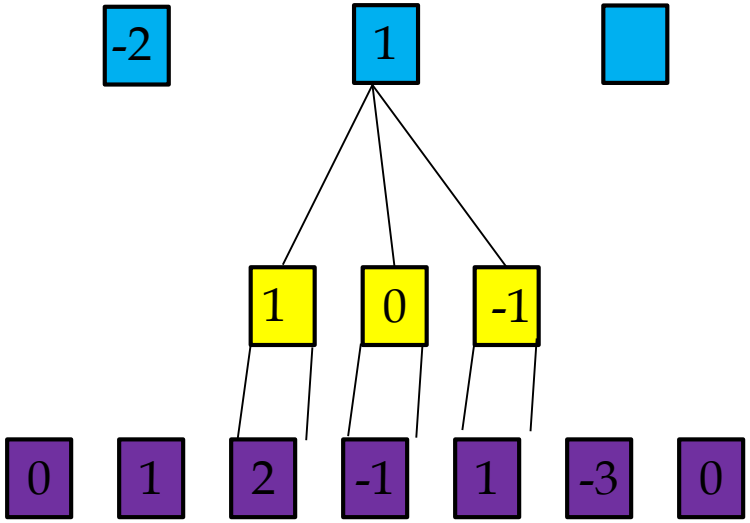
# 卷积的参数

- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2



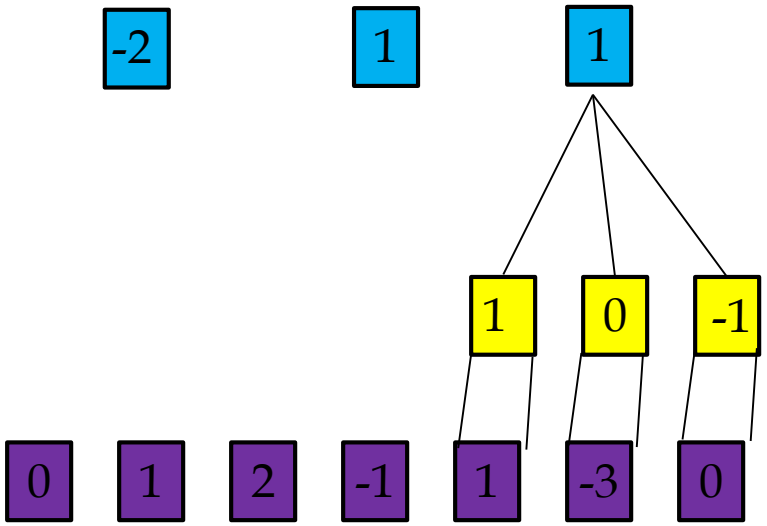
# 卷积步长

- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2



# 卷积步长

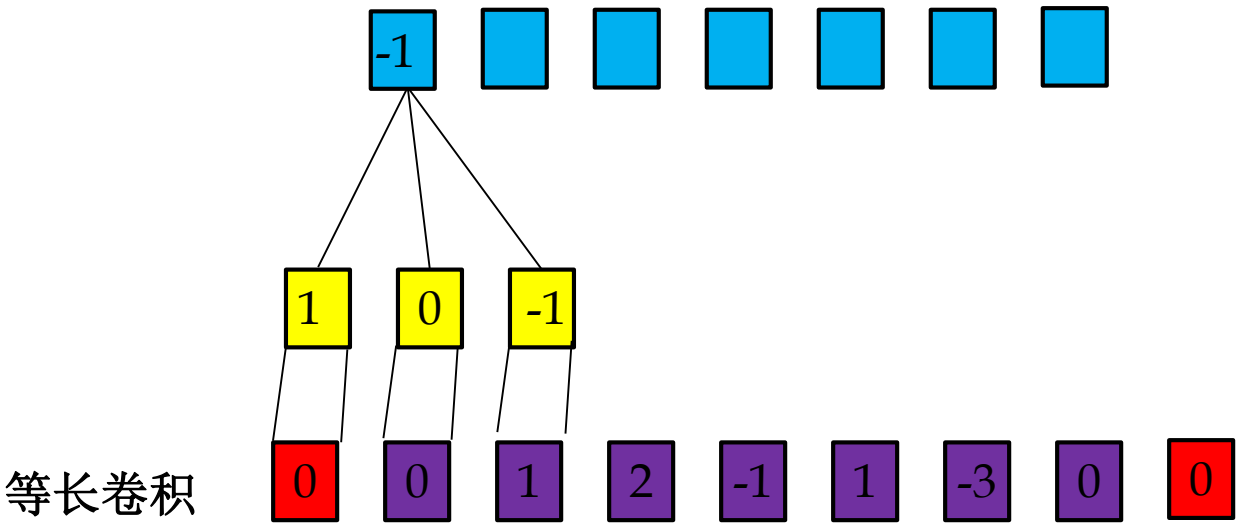
- **卷积窗口**：滤波器的参数数目，表征了每次编码的信息单元的大小
- **卷积步长**：滤波器编码信息时每次移动的距离
- **例子**：窗口为3，步长为2




以下如无特殊说明，步长取1

# 窄卷积、宽卷积、等长卷积

- **窄卷积**：输出长度n-m+1，不补零。
- **宽卷积**：输出长度n + m -1，对于不在[1,n] 范围之外的$x_t$ 用零补齐（zero-padding）。（Padding=m-1）
- **等长卷积**：输出长度n，对于不在[1,n]范围之外的$x_t$ 用零补齐（zero-padding）。(Padding=(m-1)/2)




- 在这里除了特别声明，我们一般说的卷积默认为**窄卷积**。

# 二维卷积

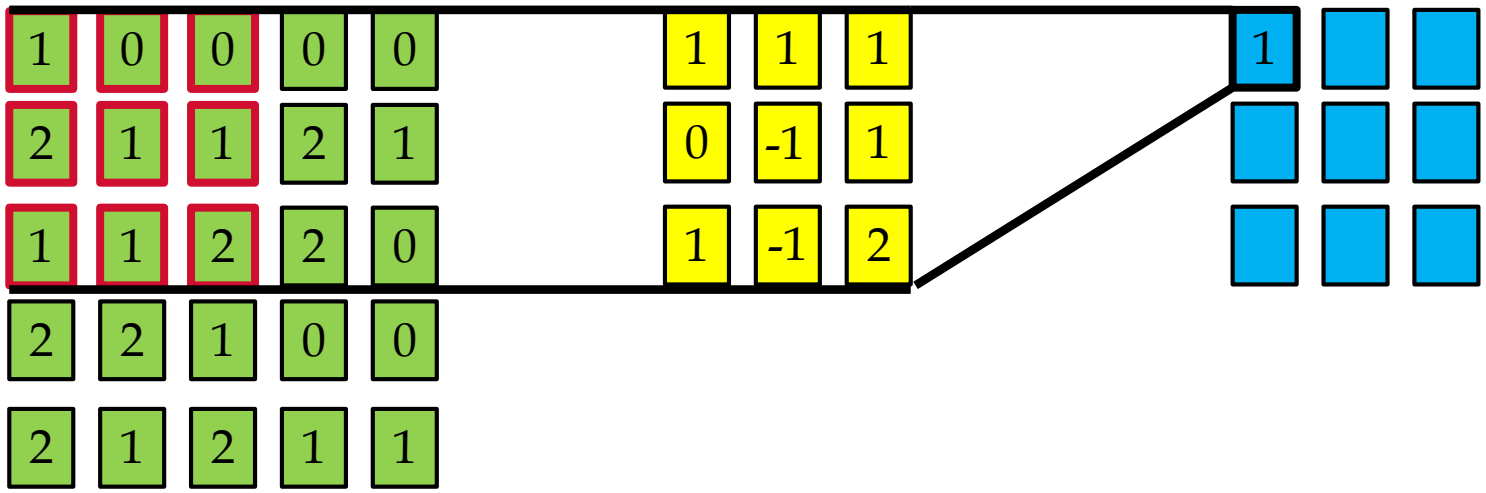
- 两维卷积经常用在图像和自然语言处理中
- 给定一个图像或句子X=[$x_{ij}$], i∈ [1, M], j ∈ [1, $N$]
- 滤波器W=[$w_{ij}$], i∈ [1, m], j ∈ [1, n] 一般$m << M; n << N$
- 则卷积后的输出值Y中的每个位置如下表示

$$y_{\mathrm{ij}} = f\left(W x_{i:i+m-1,j:i+n-1} + b\right)$$

$$= f\left(\sum_{u=1}^{m}\sum_{v=1}^{n} w_{uv} x_{\mathrm{i}+u-1,j+v-1} + \mathrm{b}\right)$$

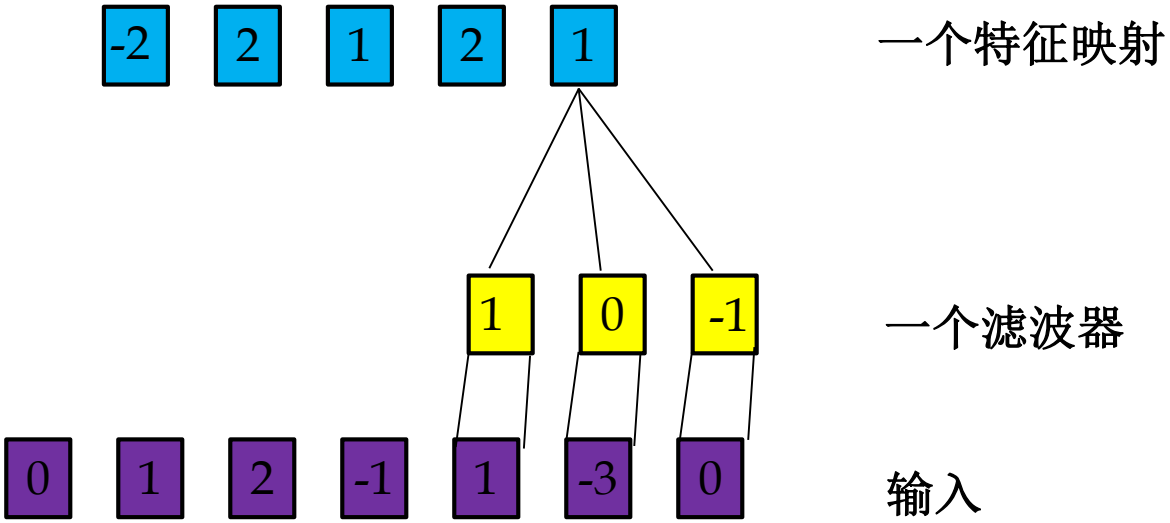
# 二维卷积示例

- 输入X(5，5)
- 滤波器（3，3）



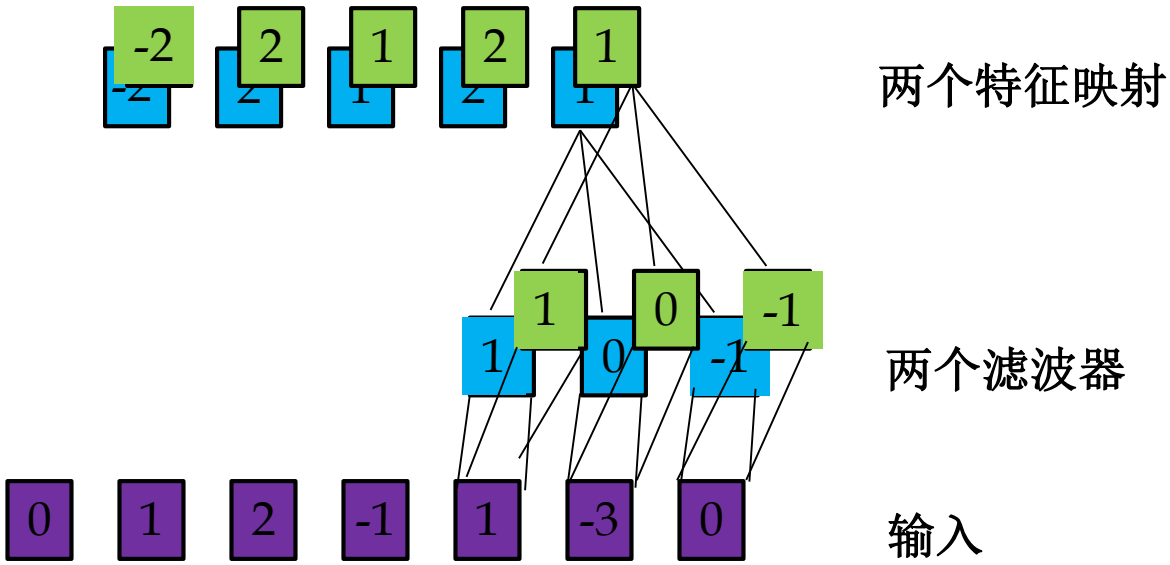
# 滤波器：特征提取器

- 如果把滤波器看成一个特征提取器，每一组输出都可以看成是输入经过一个特征抽取后得到的特征。因此，在卷积神经网络中每一组输出也叫作一个**特征映射**（Feature Map）。



# 多个滤波器

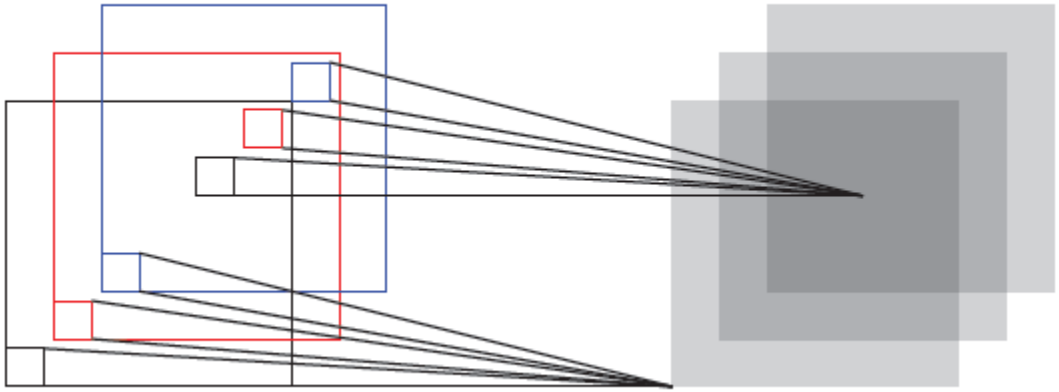
- 为了增强卷积层的表示能力，可以使用*K*个不同的滤波器来得到*K*组输出，即K个特征映射。




每个滤波器的参数随机初始化，滤波器个数是网络中重要的参数

# 卷积层作用

- 提取一个局部区域的特征，每一个滤波器相当于一个特征提取器。



# 子采样层：池化操作

- 虽然卷积层通过局部连接和权重共享显著减少了连接的个数，但是每一个特征映射的维度并没有显著减少。
- 这样，如果后面接一个分类器，分类器的输入维数依然很高，很容易出现过拟合。
- 为了解决这个问题，卷积神经网络一般会在卷积层之后再加一个池化（Pooling）操作，即子采样（Subsampling），构成一个子采样层。通过子采样显著降低特征维数，避免过拟合。

# 子采样层

- 对于卷积层得到的一个特征映射$X^{(l)}$，则采样层输出有：

$$X^{(l+1)} = f(w^{(l+1)} \cdot down(X^{(l)}) + b^{(l+1)})$$

- 其中，$w^{(l+1)}$和 $b^{(l+1)}$分别是可训练的权重和偏置参数
  $down(X^{(l)})$是指子采样后的特征映射

# 子采样层

- 最大值采样（Maximum Pooling）

$$pool_{max}(X^{(l)}) = \max_{i \in X^{(l)}} X_i^{(l)}$$

- 最小值采样（Minimum Pooling）

$$pool_{min}(X^{(l)}) = \min_{i \in X^{(l)}} X_i^{(l)}$$

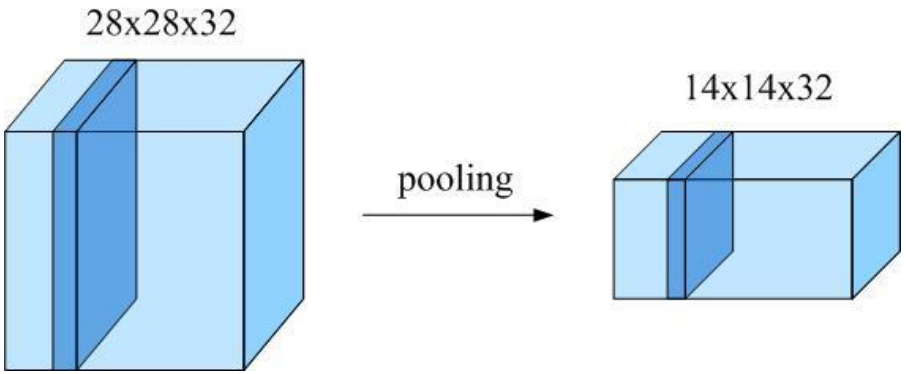
- 平均值（Average Pooling）

$$pool_{avg}(X^{(l)}) = \frac{1}{|X^{(l)}|} \sum_{i \in X^{(l)}}^{|R_k|} X_i^{(l)}$$

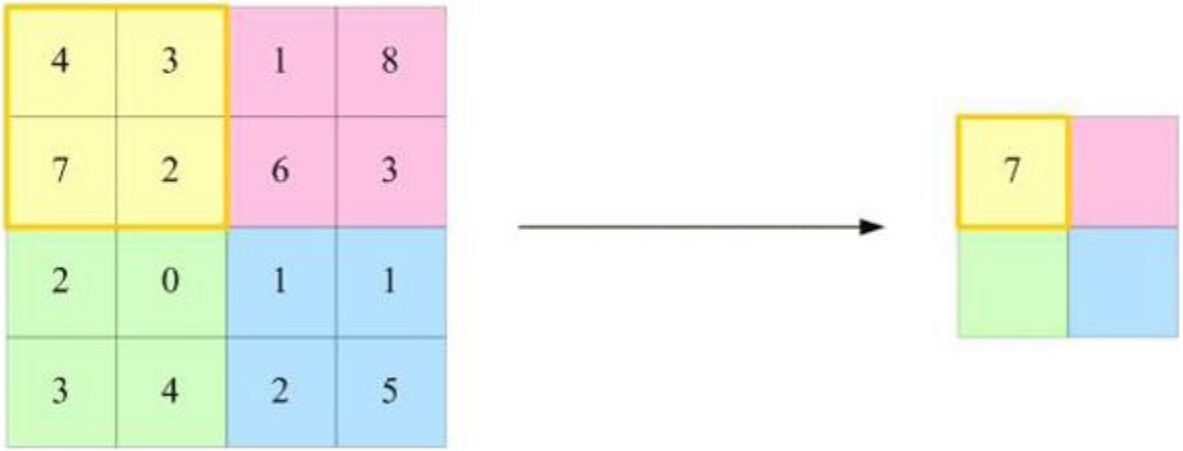
- TopK采样（Average Pooling）

$$pool_k(X^{(l)}) = \operatorname*{top}_{i \in X^{(l)}} X_i^{(l)}$$

# 子采样层示例




采样区域
自行设定



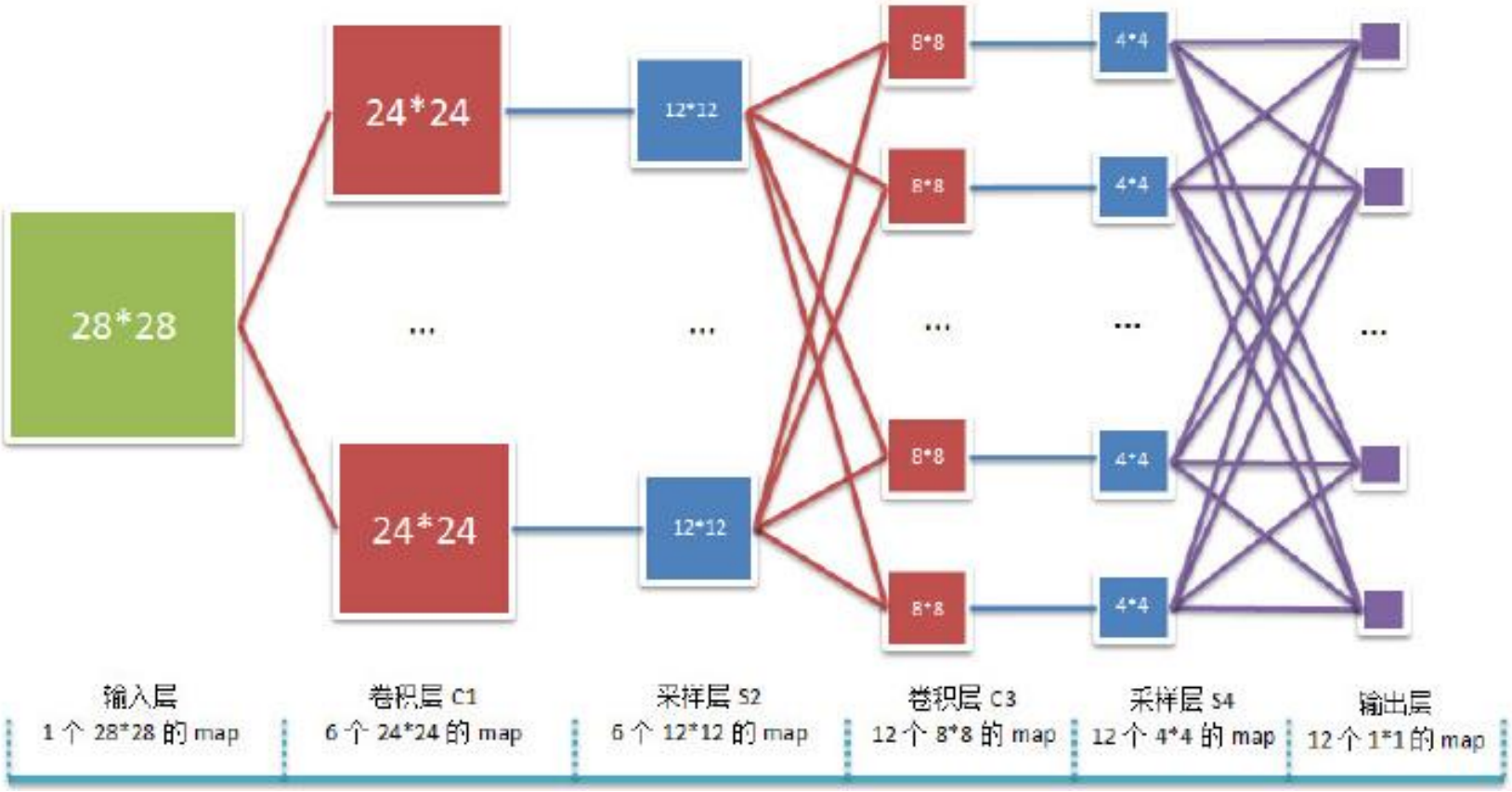
# 参数训练

- 任务：一个滤波器，就是训练W和b

$$y_i = f(Wx_{i:i+m-1} + b) = f(\sum_{k=1}^{m} \boxed{w_k} x_{i+k-1} + \boxed{b})$$

- 采用反向传播算法
  - 选择一个样本 x，信息从输入层经过逐级的变换，传送到输出层
  - 计算该样本的实际输出 o 与相应的理想输出 t 的差
  - 按极小化误差反向传播方法调整权矩阵，也可以调整输入词向量

# 示例


28*28
24*24
24*24
12*12
12*12
8*8
8*8
8*8
8*8
4*4
4*4
4*4
4*4
输入层
1 个 28*28 的 map
卷积层 C1
6 个 24*24 的 map
采样层 S2
6 个 12*12 的 map
卷积层 C3
12 个 8*8 的 map
采样层 S4
12 个 4*4 的 map
输出层
12 个 1*1 的 map

**Softmax层**

# 基于卷积网络的关系抽取
(Zeng et al., Coling 2014)

## 任务：判别句子中实体之间的语义关系

S: 2013年4月20日8时02分四川省雅安市[芦山县]$_{e1}$ 发生了7.0级[地震]$_{e2}$

震中 (e1,e2)

- 语义关系分类

# 基于卷积网络的关系抽取
(Zeng et al., Coling 2014)

## 传统特征提取需要NLP预处理+人工设计的特征

2013年4月20日8时02分四川省雅安市[芦山县]$_{e1}$ 发生了7.0级[地震]$_{e2}$

震中 (e1,e2)

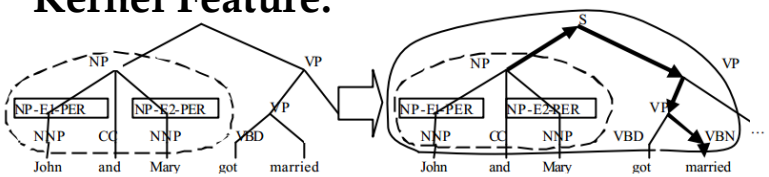
**传统特征：**

**Words：** *芦山县*$_{m11}$, *地震*$_{b1}$, *发生*$_{b2}$, *在*$_{21}$

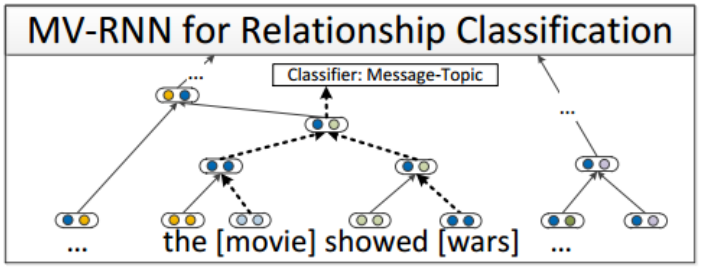
**Entity Type：** *Noun*$_{m1}$ , *Location*$_{m2}$

**Parse Tree：** *Location-VP-PP-Noun*

**Kernel Feature:**




**基于神经网络的特征抽取：**



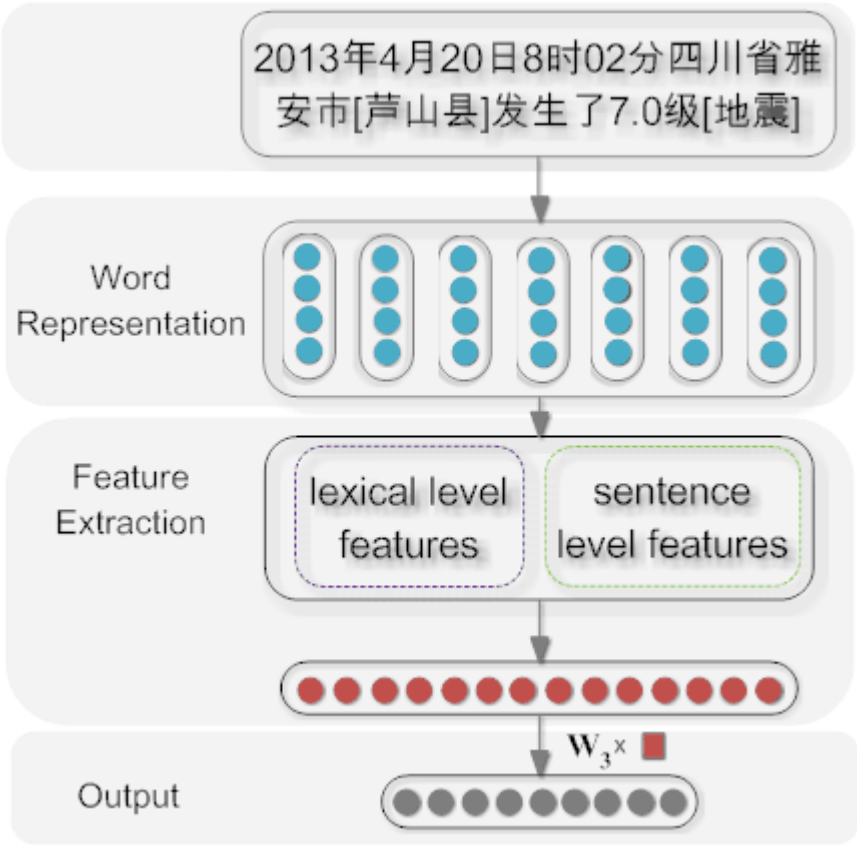

- 问题1：对于缺少NLP处理工具和资源的语言，无法提取文本特征
- 问题2：NLP处理工具引入的“错误累积”
- 问题3：人工设计的特征不一定适合当前任务

# 解决方法

- 基于卷积神经网络的实体关系分类方法
  - 通过CNN学习文本语义特征
  - 不需要人工设计特征

通过Word Embeddings挖掘词汇的语义表示

Lexical Level Features: 实体本身的语义特征
Sentence Level Features: 通过CNN网络挖掘句子级别的文本特征



# 词汇级特征

- 利用词向量（Word Embedding）信息作为词汇级特征

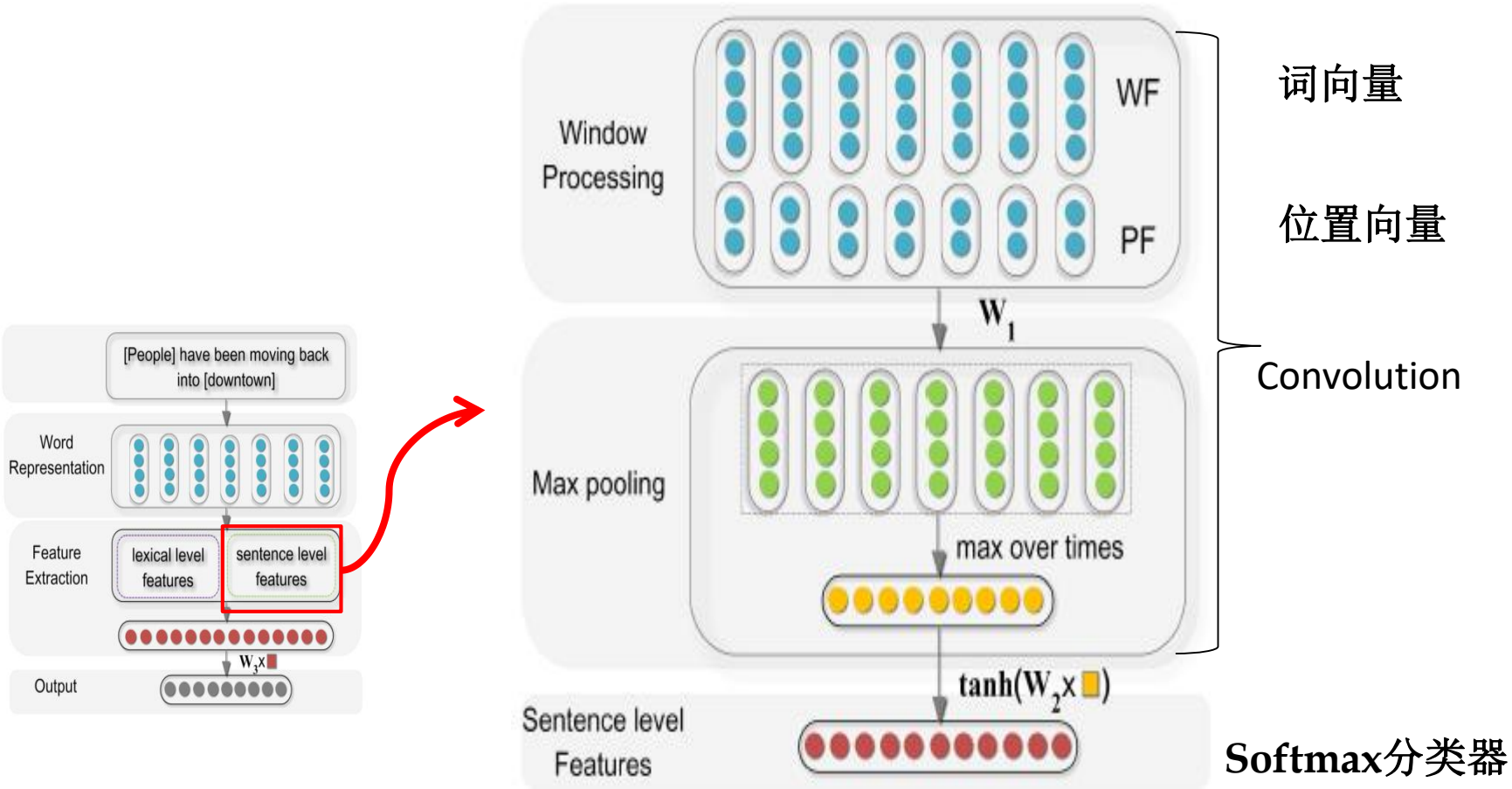
| Features | Remark |
|---|---|
| L1 | Noun 1 |
| L2 | Noun 2 |
| L3 | Left and right tokens of noun 1 |
| L4 | Left and right tokens of noun 2 |
| L5 | WordNet hypernyms of nouns |

The [haft]$_{e1}$ of the [axe]$_{e2}$ is made of yew wood.

(0.5 , 0.2, -0.1, 0.4) , 0.3, -0.1, 0.4) 0.3, 0.1, 0.4)

# 句子级特征

- 利用CNN捕获句子级特征



# 实验结果

| # of training instance | # of test instance | # of relationships |
|---|---|---|
| 8,000 | 2,717 | 19 |

| Classifier | Feature Sets | F1 |
|---|---|---|
| SVM | POS, stemming, syntactic patterns | 60.1 |
| SVM | word pair, words in between | 72.5 |
| SVM | POS, stemming, syntactic patterns, WordNet | 74.8 |
| MaxEnt | POS, morphological, noun compound, thesauri, Google n-grams, WordNet | 77.6 |
| SVM | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | -<br>POS, NER, WordNet, syntactic tree | 74.8<br>77.6 |
| MVRNN | -<br>POS, NER, WordNet, syntactic tree | 79.1<br>82.4 |
| **Proposed** | word pair, WordNet | **82.7** |

# 实验结果

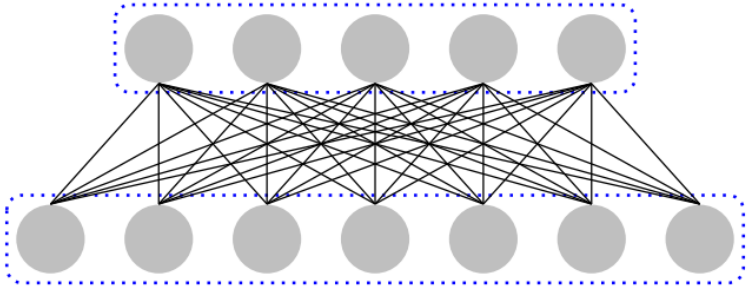
| # of training instance | # of test instance | # of relationships |
|---|---|---|
| 8,000 | 2,717 | 19 |

| | Feature Sets | F1 |
|---|---|---|
| Lexical | L1 | 34.7 |
| | +L2 | 53.1 |
| | +L3 | 59.4 |
| | +L4 | 65.9 |
| | +L5 | 73.3 |
| Sentence | WF | 69.7 |
| | +PF | 78.9 |
| Combination | all | 82.7 |

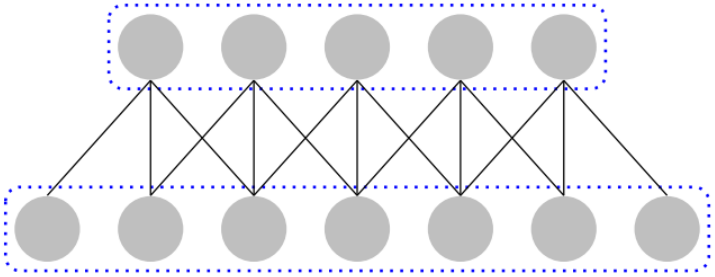
| Classifier | Feature Sets | F1 |
|---|---|---|
| SVM | POS, stemming, syntactic patterns | 60.1 |
| SVM | word pair, words in between | 72.5 |
| SVM | POS, stemming, syntactic patterns, WordNet | 74.8 |
| MaxEnt | POS, morphological, noun compound, thesauri, Google n-grams, WordNet | 77.6 |
| SVM | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | -<br>POS, NER, WordNet, syntactic tree | 74.8<br>77.6 |
| MVRNN | -<br>POS, NER, WordNet, syntactic tree | 79.1<br>82.4 |
| **Proposed** | word pair, WordNet | **82.7** |

# 前馈神经网络的缺点

- 输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据
  - 句子是有长有短，图像大小也不固定
- 连接存在于层与层之间，每层的节点之间是无连接的，假设每次输入都是独立的，无法建模时序或者位置信息



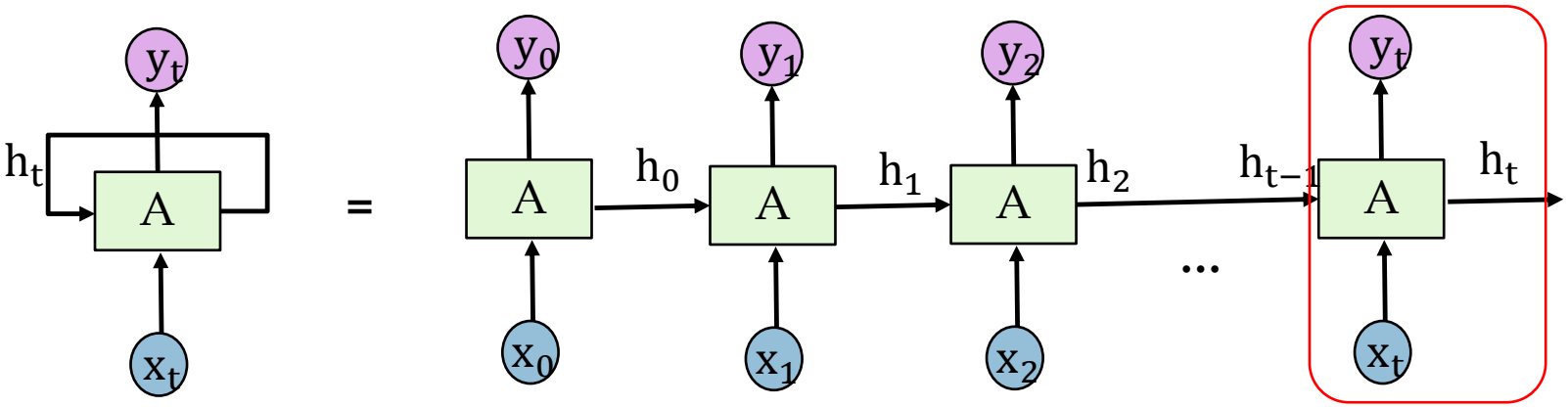
全连接层



卷积层

# 循环神经网络

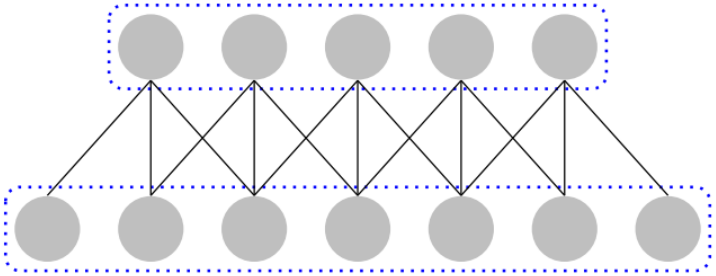
- 是一种反馈神经网络
  - 全部或者部分神经元可以接受来自其它神经元或自身的神经元信号（cf. 前馈神经网络只接受前一层的神经元信号）
  - 可以建立同层神经元的关联，从而每层节点之间有连接，因此可以建模时序或者位置信息，处理变长的序列数据
- 循环神经网络：至少包含一个反馈连接的神经网络结构。因此，网络的激励可以沿着一个loop 进行流动。这种网络结构特别适合于处理时序数据。

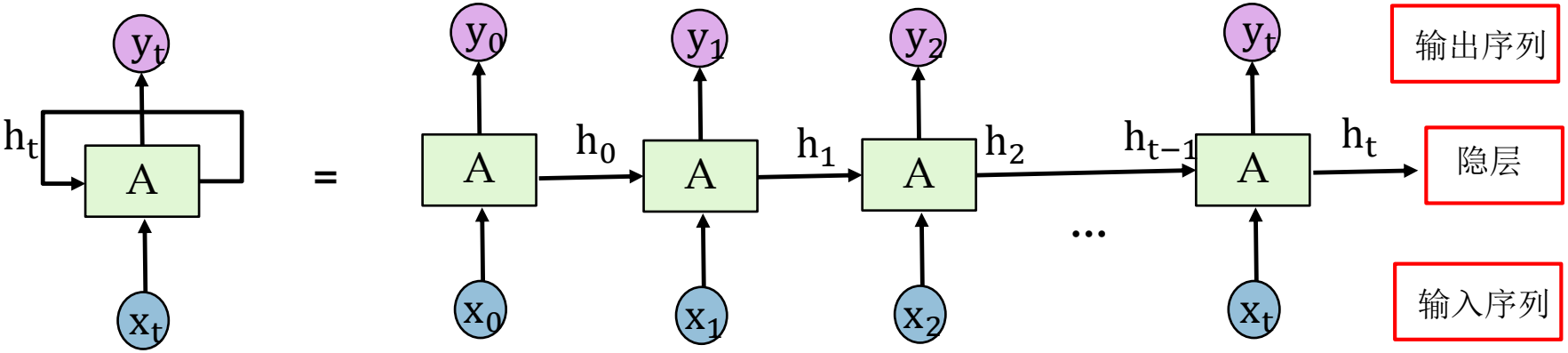





A是RNN单元（RNN的参数）；t时刻A的输入为t-1时刻的隐层$h_{t-1}$和t时刻的输入xt；$h_t$为t时刻的隐层，编码了$x_0$-$x_t$的信息；$y_t$为t时刻的输出；最后根据$y_0$-$y_t$完成A的参数更新

# 循环神经网络






- 假设时刻$t$时，输入为$x_t$，隐层状态为$h_t$。 $h_t$不仅和当前时刻的输入相关，也和上一个时刻的隐层状态$h_{t-1}$相关。
- A代表一个RNN单元，由一套参数组成
- 一般使用如下函数：

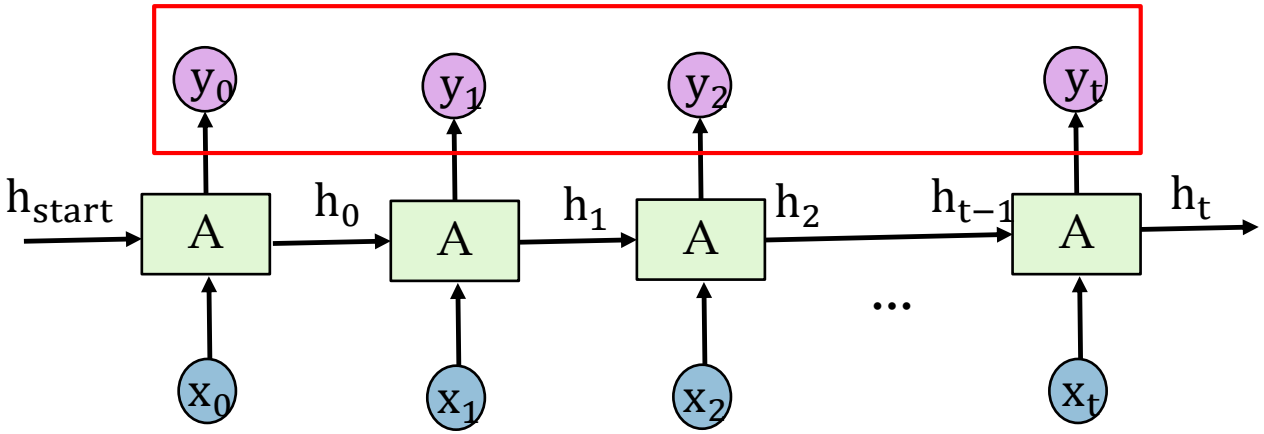
$$\boldsymbol{h}_t = \varphi(\boxed{\boldsymbol{W}}\boldsymbol{x}_t + \boxed{\boldsymbol{U}}\boldsymbol{h}_{t-1} + \boxed{b}) \qquad \boldsymbol{y}_t = softmax(\boxed{V}\boldsymbol{h}_t)$$

其中，$\varphi$是非线性函数，一般为tanh 函数。

◆ht编码了从0到t时刻的所有信息；ht经过softmax分类得到分类结果

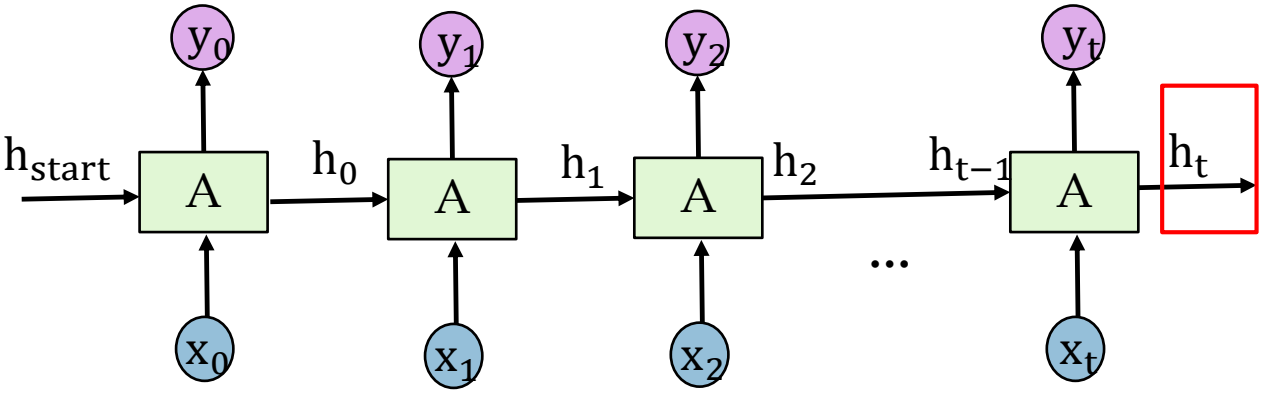
# 循环神经网络：应用场景（一）

- 输入：任意长度的序列
- 输出：同样长度的序列
- 适合：序列标注任务等
- 例如：命名实体识别



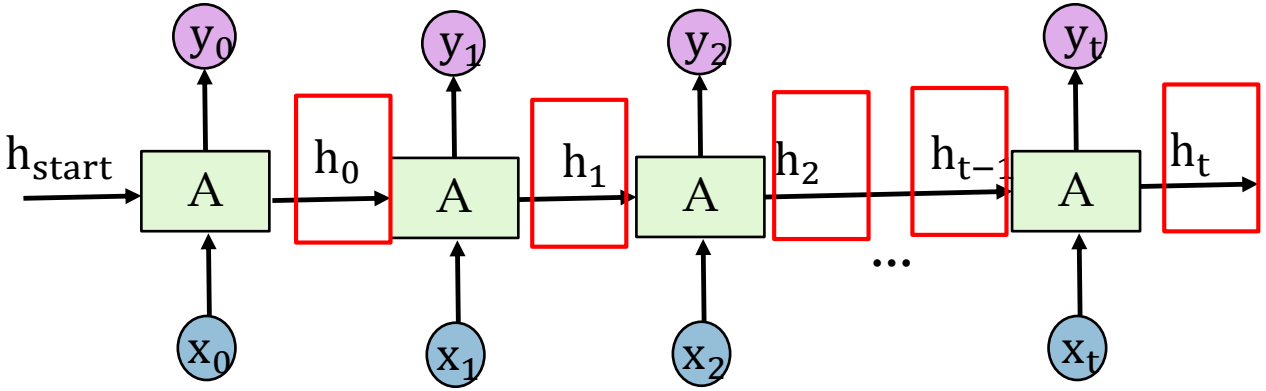
# 循环神经网络：应用场景（二）

- 输入：任意长度的序列
- 输出：关于该序列的一个向量表示
- 适合：对输入序列进行统一表示
- 应用：文本分类、关系分类（$h_t$是文本的表示，再加一个分类层）



# 循环神经网络：应用场景（三）

- 输入：任意长度的序列
- 输出：关于该序列的多个向量表示
- 适合：输入序列中每个时刻的表示
- 应用：关系分类
  - $h_k$和$h_j$是待分类的两个实体的表示
  - 再加一个分类器完成关系分类



# 参数学习

- 循环神经网络（4个参数），其中U,W,V是矩阵，b是向量

$$\boldsymbol{h}_t = \varphi(\boxed{\boldsymbol{W}}\boldsymbol{x}_t + \boxed{\boldsymbol{U}}\boldsymbol{h}_{t-1} + \boxed{b}) \qquad \boldsymbol{y}_t = softmax(\boxed{V}\boldsymbol{h}_t)$$

- 参数训练可以利用反向传播算法（Backpropagation Through Time，BPTT）
  - 详细参考P. J. Werbos. “Backpropagation through time: what it does and how to do it”. In: Proceedings of the IEEE 10 (1990).
- 问题：序列长度过长时会导致梯度爆炸或者梯度消失问题
- 解决办法：RNN的典型变形-LSTM和GRU
  - LSTM：Long Short-Term Memory
  - GRU: Gated Recurrent Unit

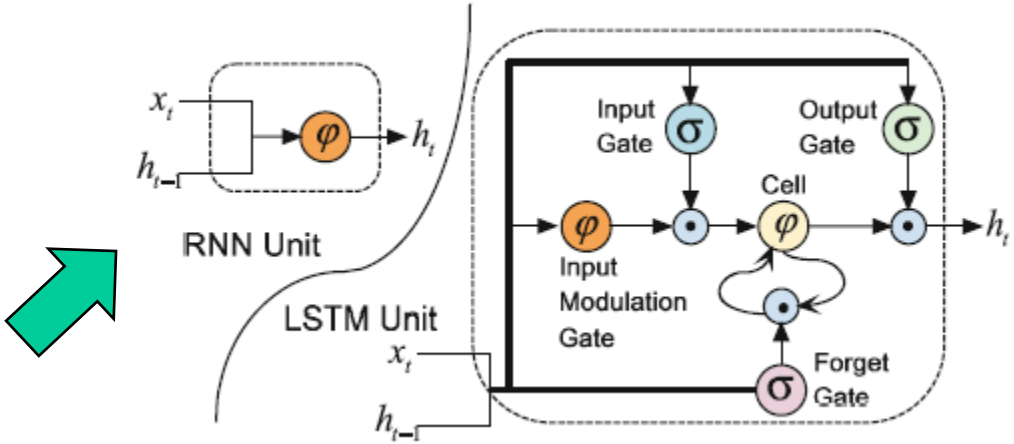
# 长短时记忆神经网络：LSTM

- 长短时记忆神经网络
- RNN的一个变体，可以有效地解决简单循环神经网络的梯度爆炸或消失问题
- 关键：引入了一组记忆单元（Memory Units），允许网络可以学习何时遗忘历史信息，何时用新信息更新记忆单元。具体地，
  - 在时刻 t ，记忆单元 $\mathbf{c}_t$ 记录了到当前时刻为止的所有历史信息，并受三个“门”控制：输入门 $\mathbf{i}_t$，遗忘门$\mathbf{f}_t$和输出门$\mathbf{o}_t$。三个门的元素的值在 [0, 1] 之间。

# 长短时记忆神经网络

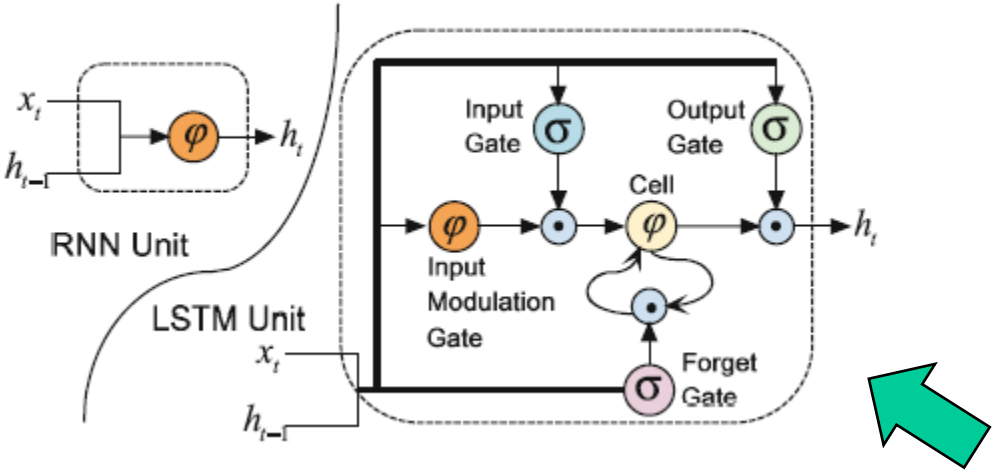
## RNN vs LSTM

RNN：$h_t = \varphi(W_h x_t + U_h h_{t-1} + b_h)$



# 长短时记忆神经网络

## RNN vs LSTM



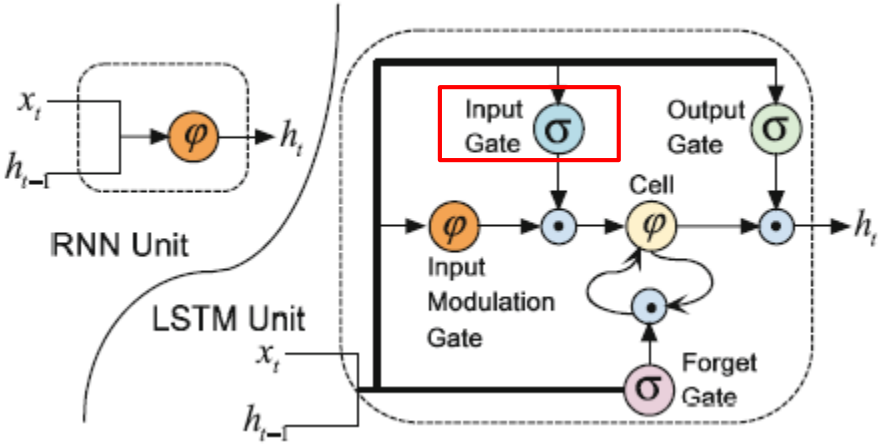

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

# 长短时记忆神经网络：结构

## LSTM：输入门

$$\boldsymbol{h}_t = \varphi(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1} + b)$$



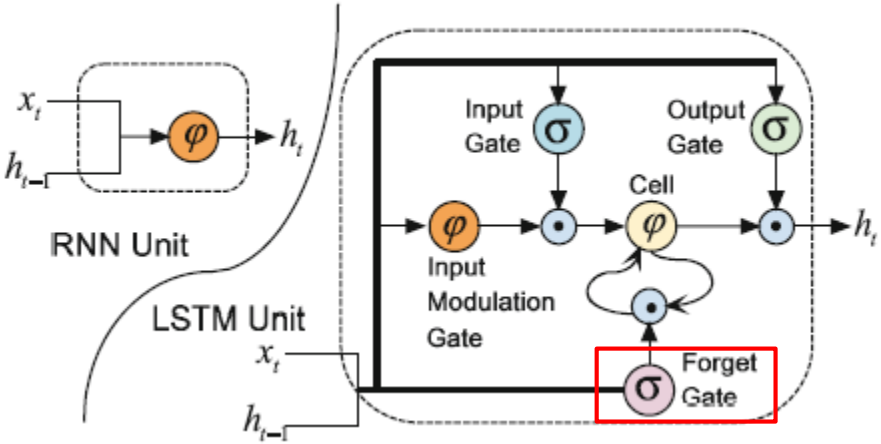

LSTM：

$$\boxed{i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_t)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$u_t = \varphi(W_u x_t + U_u h_{t-1} + b_u)$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot \varphi(c_t)$$

输入门 $\mathbf{i}_t$是一个0-1向量，控制每一个记忆单元加入多少新的信息

# 长短时记忆神经网络：结构

## LSTM：遗忘门



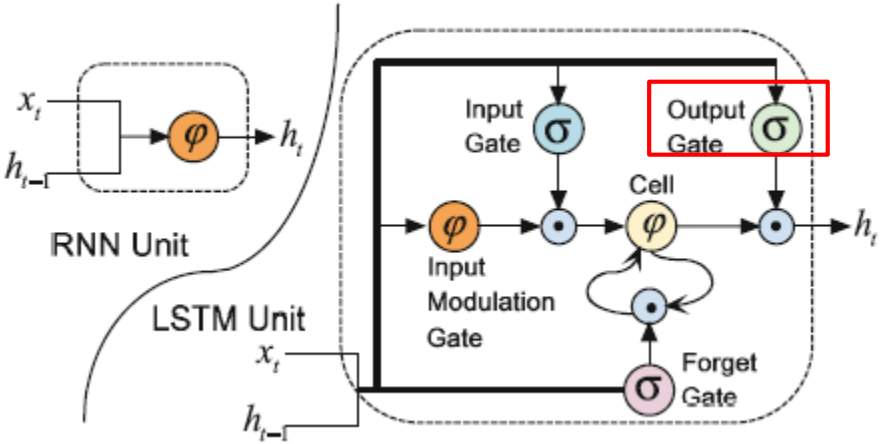

LSTM：

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$\boxed{f_t = \sigma(W_f x_t + U_f h_{t-1} + b_t)}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$u_t = \varphi(W_u x_t + U_u h_{t-1} + b_u)$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot \varphi(c_t)$$

遗忘门 $\mathbf{f}_t$也是一个0-1向量，控制每一个记忆单元需要遗忘多少信息

# 长短时记忆神经网络：结构

## LSTM：输出门



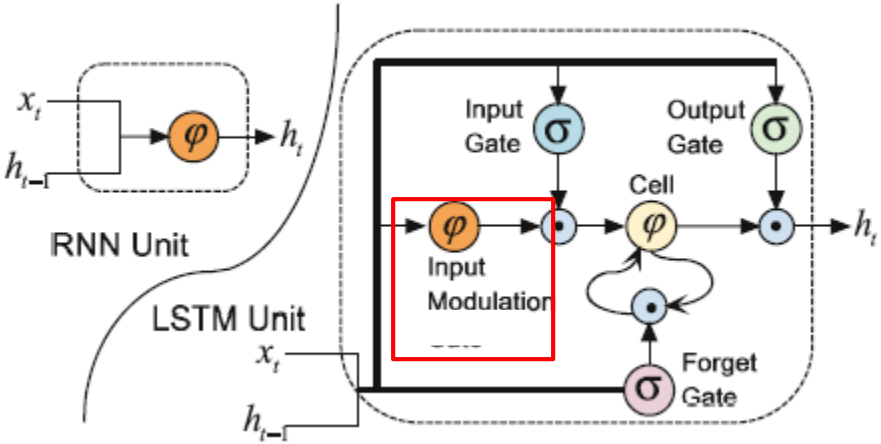

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

输出门$\mathbf{o}_t$也是一个0-1向量，控制每一个记忆单元输出多少信息。

# 长短时记忆神经网络：结构

## LSTM：输入控制



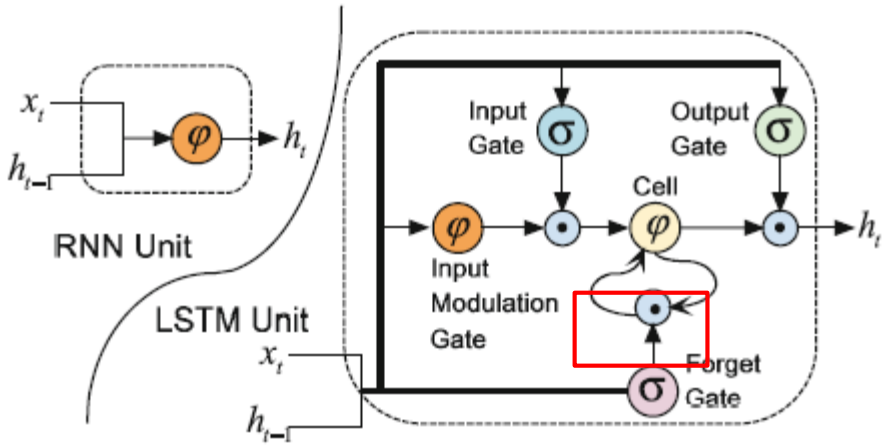

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

输入控制$\mathbf{u}_t$（不是0-1向量）是t时刻当前信息的编码，和输入门$\mathbf{i}_t$共同决策当前输入对记忆单元 $\mathbf{c}_t$ 的贡献

# 长短时记忆神经网络：结构

## LSTM：记忆单元



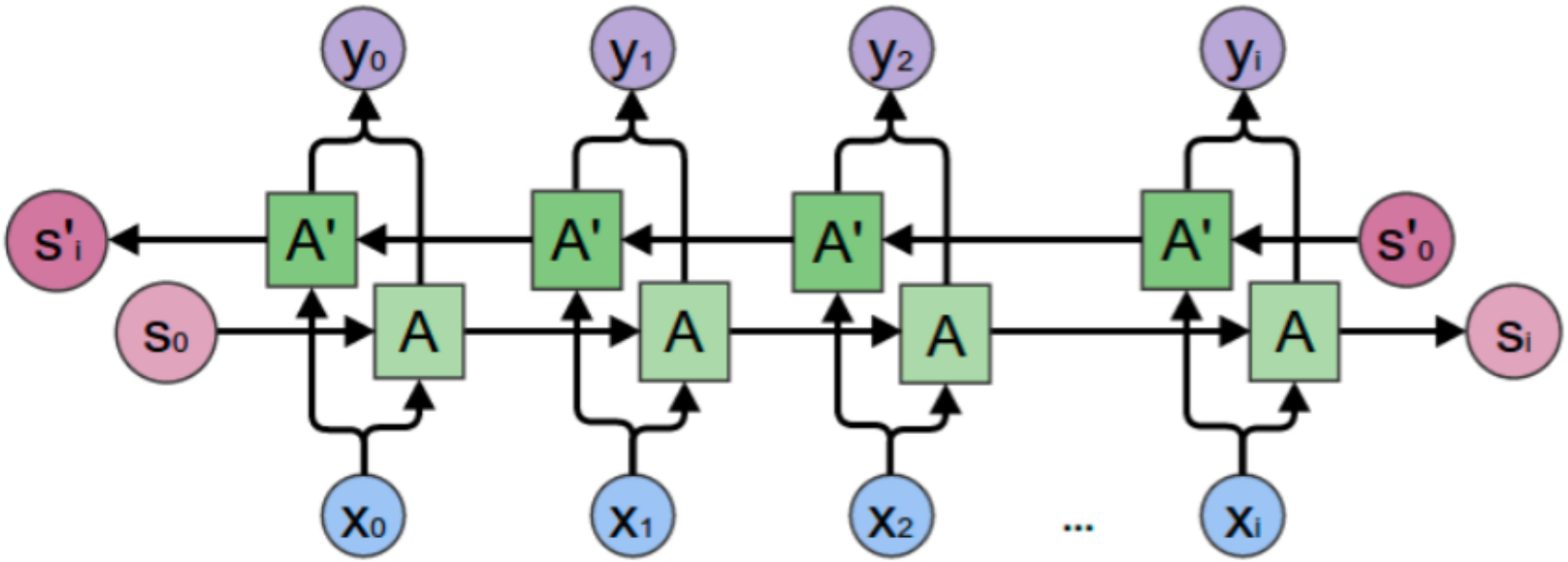

LSTM：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_t) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
u_t &= \varphi(W_u x_t + U_u h_{t-1} + b_u) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \varphi(c_t)
\end{aligned}
$$

记忆单元 $\mathbf{c}_t$ 记录了到当前时刻为止的所有历史信息

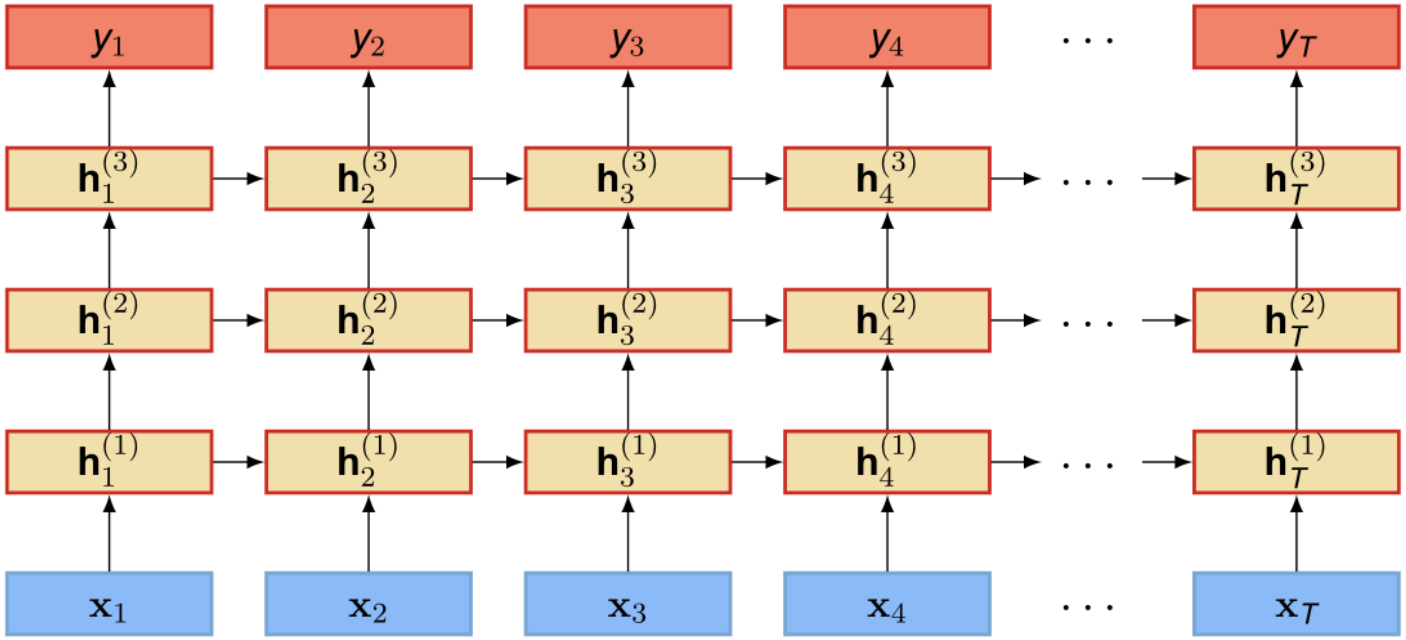
# 双向循环神经网络

- 单向的循环神经网络只能建模单一方向的信息，忽略了另一方向的信息




A和A′可以为任意具有循环结构的网络，例如：RNN,LSTM,GRU(只有更新门和重置门)

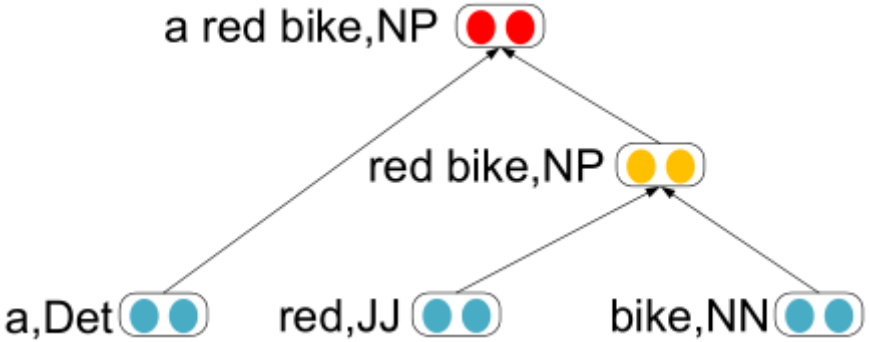
# 层叠(Stack)循环神经网络




本质就是多个循环网络逐层使用，用更多的层建模更高层的特征

# 递归神经网络

- 递归神经网络：Recursive Neural Network (RecNN)



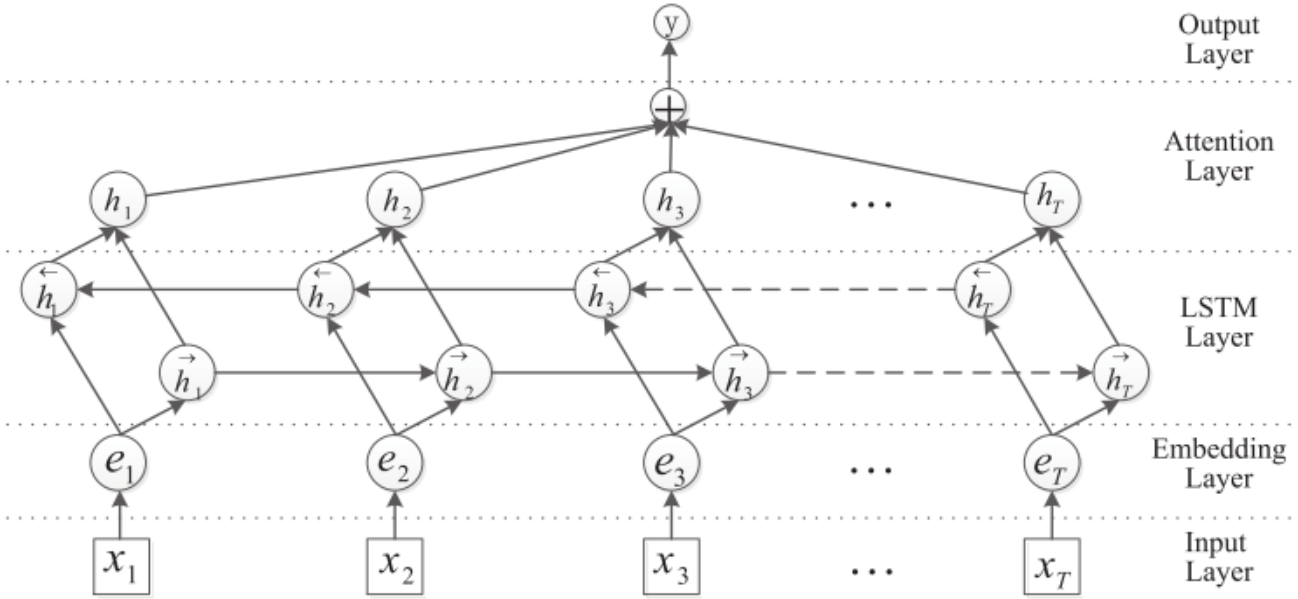

根据一个给定的拓扑结构，
进行词语的语义组合。

# 利用LSTM做关系抽取（Zhou ACL2016）

- 动机：
  - 句子级语义信息对关系抽取任务至关重要
  - 句子中的不同位置的词有不同程度的重要性
  - 传统方法依赖NLP工具，人工提取特征，造成错误传递
  - CNN的方法难以建模句子中长距离的依赖关系

# 利用LSTM做关系抽取（Zhou ACL2016）

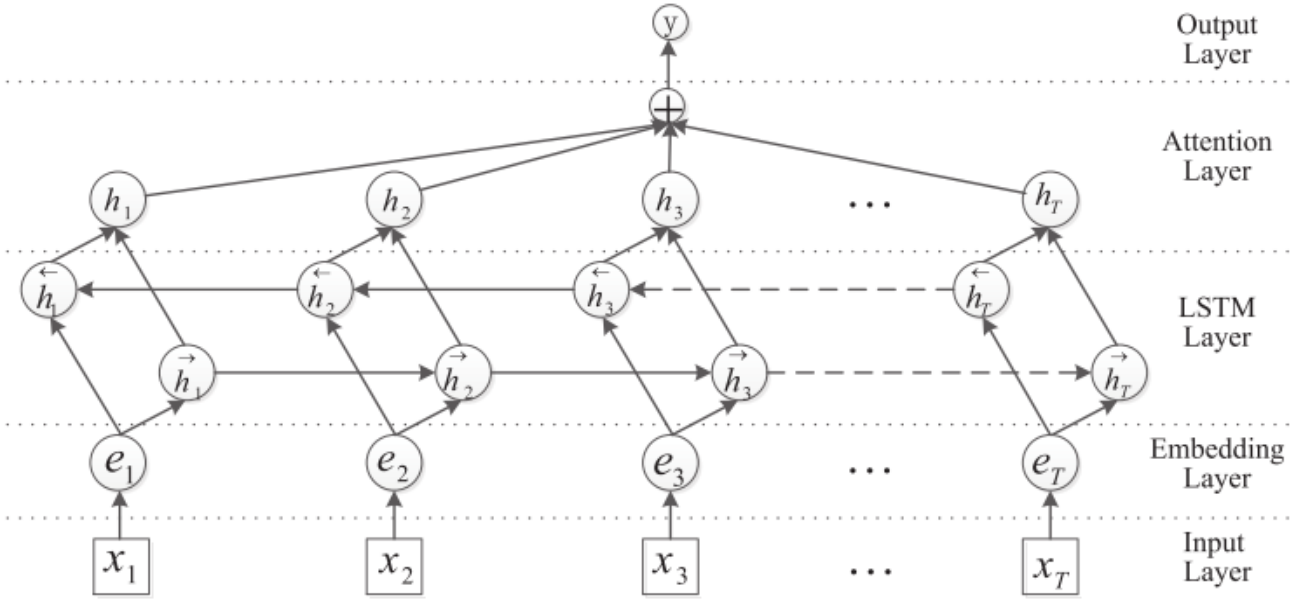
## 方法：




- 利用双向LSTM编码每个词的表示
- $\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_T} = \overleftarrow{LSTM(x_1, \ldots, x_T)}$
- $\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_T} = \overrightarrow{LSTM(x_1, \ldots, x_T)}$
- $h_i = [\overleftarrow{h_2} \otimes \overrightarrow{h_2}]$（$\otimes$ 是对应位置求和）

# 利用LSTM做关系抽取（Zhou ACL2016）

- 方法：




- 关注机制（通过对隐层的softmax得到关注的权重）
- $H = [h_1, h_2, \ldots, h_T]$ , $M = \tanh(H)$
- $\alpha = softmax(\omega^T M)$ , $r = H\alpha^T$
- $h^* = \tanh(r)$

# 利用LSTM做关系抽取（ Zhou ACL2016 ）

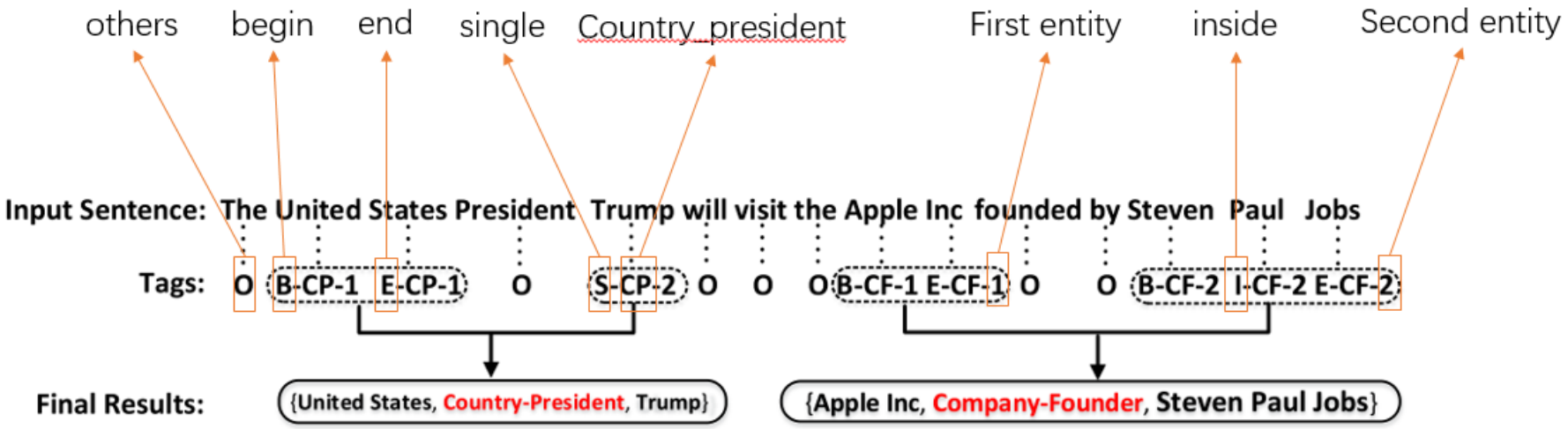
## 实验结果：

| Model | Feature Set | F1 |
|---|---|---|
| SVM<br>(Rink and Harabagiu, 2010) | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FramNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| CNN<br>(Zeng et al., 2014) | WV (Turian et al., 2010) (dim=50)<br>+ PF + WordNet | 69.7<br>82.7 |
| RNN<br>(Zhang and Wang, 2015) | WV (Turian et al., 2010) (dim=50) + PI<br>WV (Mikolov et al., 2013) (dim=300) + PI | 80.0<br>82.5 |
| SDP-LSTM<br>(Yan et al., 2015) | WV (pretrained by word2vec) (dim=200), syntactic parse<br>+ POS + WordNet + grammar relation embeddings | 82.4<br>83.7 |
| BLSTM<br>(Zhang et al., 2015) | WV (Pennington et al., 2014) (dim=100)<br>+ PF + POS + NER + WNSYN + DEP | 82.7<br>84.3 |
| BLSTM<br>Att-BLSTM<br>BLSTM<br>Att-BLSTM | WV (Turian et al., 2010) (dim=50) + PI<br>WV (Turian et al., 2010) (dim=50) + PI<br>WV (Pennington et al., 2014) (dim=100) + PI<br>WV (Pennington et al., 2014) (dim=100) + PI | 80.7<br>82.5<br>82.7<br>84.0 |

Table 1: Comparison with previous results. WV, PF, PI stand for word vectors, position features and position indicators respectively.

# 实体和关系联合抽取（Zheng ACL2017）

- 动机：
  - 真实世界中很多情况下要实体和关系联合抽取
  - 传统的方法大多数都假定给出了待抽取的两个实体



# 实体和关系联合抽取（Zheng ACL2017）

- 方法：
  - 对每个关系，将其与（Begin，Inside，End，Single）以及表示第几个实体的序号（1，2）组合，一共构成八个标签
  - 额外考虑一个Other标签，表示不属于任何一个关系
  - 如果总共有$|R|$个关系，那么一共有$2*4*|R|+1$个标签
  - 根据标注结果得到三元组
  - 标注方法尝试了
    - LSTM+CRF
    - LSTM+LSTM
    - LSTM+LSTM+bias

# 实体和关系联合抽取（ Zheng ACL2017 ）

- 实验：
  - 数据集：使用弱监督的NYT数据集，并且将其视为监督数据。训练集包括353k个三元组，测试集包括3880个三元组，一个24个关系。
  - 实验结果

| Elements | E1 | | | E2 | | | (E1,E2) | | |
|---|---|---|---|---|---|---|---|---|---|
| PRF | *Prec.* | *Rec.* | *F1* | *Prec.* | *Rec.* | *F1* | *Prec.* | *Rec.* | *F1* |
| LSTM-CRF | **0.596** | 0.325 | 0.420 | 0.605 | 0.325 | 0.423 | **0.724** | 0.341 | 0.465 |
| LSTM-LSTM | 0.593 | 0.342 | 0.434 | **0.619** | 0.334 | 0.434 | 0.705 | 0.340 | 0.458 |
| LSTM-LSTM-Bias | 0.590 | **0.479** | **0.529** | 0.597 | **0.451** | **0.514** | 0.645 | **0.437** | **0.520** |

# 传统关系抽取方法

- 基于特征向量方法 vs. 基于核函数方法 vs.基于神经网络方法

| | 基于特征向量方法 | 基于核函数方法 | 基于神经网络方法 |
|---|---|---|---|
| 优点 | 简单实用<br>计算速度较快 | 能够有效挖掘结构化信息 | 人工干预小，可扩性好，适用于大规模数据环境 |
| 缺点 | 难以进一步挖掘有效的平面特征，性能很难进一步提高 | 句法分析的错误引入了噪声，同时由于树核的计算速度非常慢，很难开发实用系统 | 可解释性差<br>训练时间长 |

# 小结

- 传统的**人工标注语料+机器学习算法**模式无法满足开放域开放语料下的信息抽取
  - 语料构建成本过高
  - 跨领域跨文本类别时抽取性能严重下降
  - 需要抽取的信息类别通常未预先指定

- 需要开放域关系抽取
  - 关系类型更多
  - 语料规模不受限
  - 抽取实体类型也更丰富

# 概述

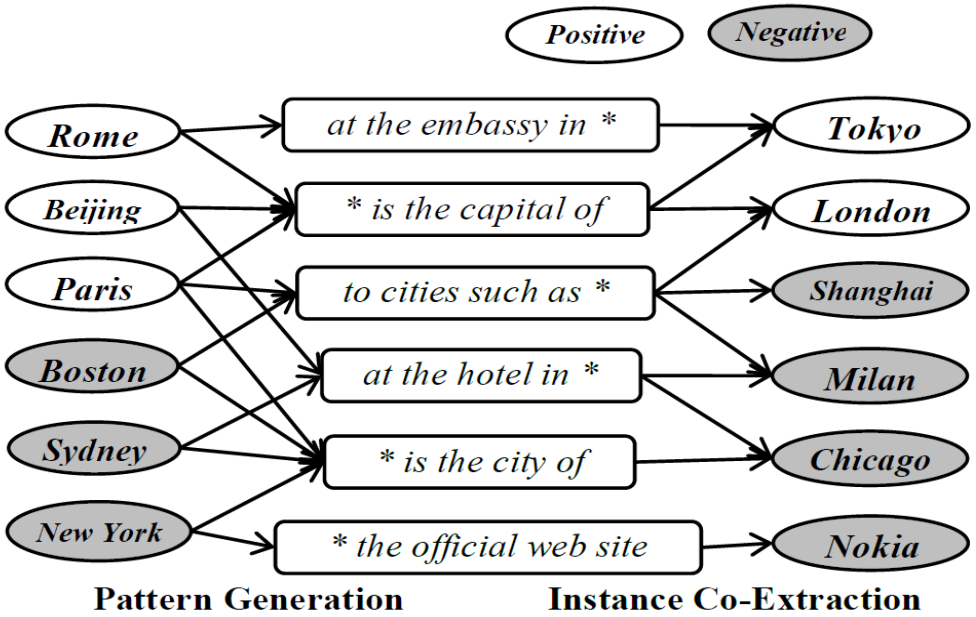
- 引言
- 实体识别与抽取
- 实体消歧
- 关系抽取和发现
  - 关系抽取任务定义
  - 传统关系抽取
  - 开放域关系抽取
  - 开放域关系发现
- 事件抽取
- 问题与挑战

# 开放域关系抽取特点

- 不限定关系类别
- 不限定目标文本
  - Web Page
  - Wikipedia
  - Query Log
- 难点问题
  - 如何获取训练语料
  - 如何获取实体关系类别
  - 如何针对不同类型目标文本抽取关系
- 需要研究新的抽取方法
  - 按需抽取—Bootstrapping，模板
  - 开放抽取—Open IE
  - 知识监督抽取—Distant Supervision

# 按需抽取：Bootstrapping

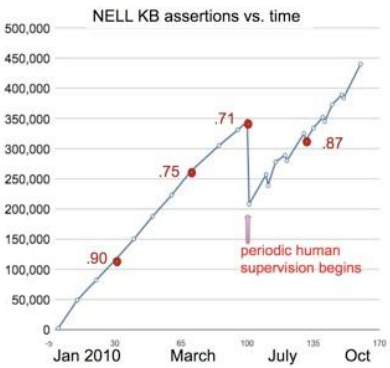
- Bootstrapping：模板生成->实例抽取->迭代直至收敛
- 语义漂移问题：迭代会引入噪音实例和噪音模板
  - 首都：*Rome* → 城市模板 *" * is a city of "*
    - Paris is a city of France.
    - Paris is a city of Romance.
- COLING 14：引入负实例来限制语义漂移



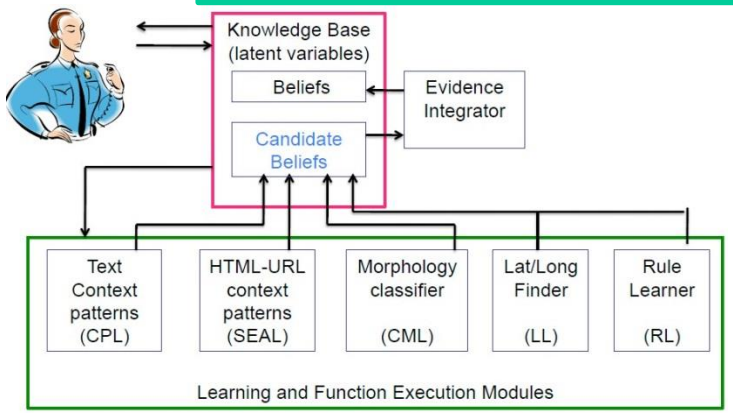
# Read the Web : NELL(Never-Ending Language Learning) ( Tom Mitchell AAAI2009 )

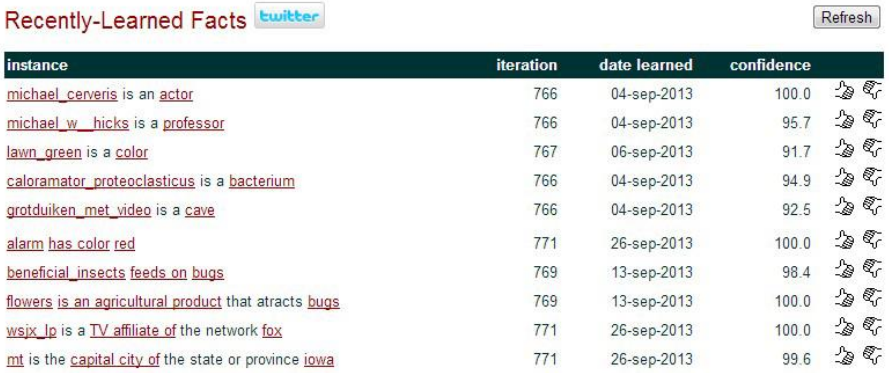
Research Project at Carnegie Mellon University

- 输入
  - Initial ontology
  - 500 million web pages
- 目标
  - Extract new instances of categories and relations
  - Learn to read better than yesterday




1,994,282 Instances of 874 different categories and relations



Recently-Learned Facts twitter Refresh

| instance | iteration | date learned | confidence |
|---|---|---|---|
| michael_cerveris is an actor | 766 | 04-sep-2013 | 100.0 |
| michael_w__hicks is a professor | 766 | 04-sep-2013 | 95.7 |
| lawn_green is a color | 767 | 06-sep-2013 | 91.7 |
| caloramator_proteoclasticus is a bacterium | 766 | 04-sep-2013 | 94.9 |
| grotduiken_met_video is a cave | 766 | 04-sep-2013 | 92.5 |
| alarm has color red | 771 | 26-sep-2013 | 100.0 |
| beneficial_insects feeds on bugs | 769 | 13-sep-2013 | 98.4 |
| flowers is an agricultural product that atracts bugs | 769 | 13-sep-2013 | 100.0 |
| wsjx_lp is a TV affiliate of the network fox | 771 | 26-sep-2013 | 100.0 |
| mt is the capital city of the state or province iowa | 771 | 26-sep-2013 | 99.6 |

# 开放抽取

- 通过识别表达语义关系的短语来抽取实体之间的关系
  - *(华为，总部位于, 深圳），(华为，总部设置于，深圳)，(华为，将其总部建于，深圳)*
- 同时使用句法和统计数据来过滤抽取出来的三元组
  - 关系短语应当是一个以动词为核心的短语
  - 关系短语应当匹配多个不同实体对
- 优点：无需预先定义关系类别
- 缺点：语义没有归一化，同一关系有不同表示

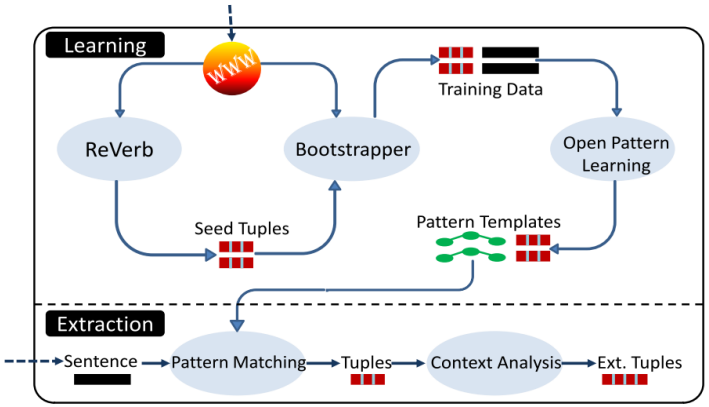
$$V \mid VP \mid VW^*P$$

$V$ = verb particle? adv?

$W$ = (noun | adj | adv | pron | det)

$P$ = (prep | particle | inf. marker)

**关系短语的句法结构约束**

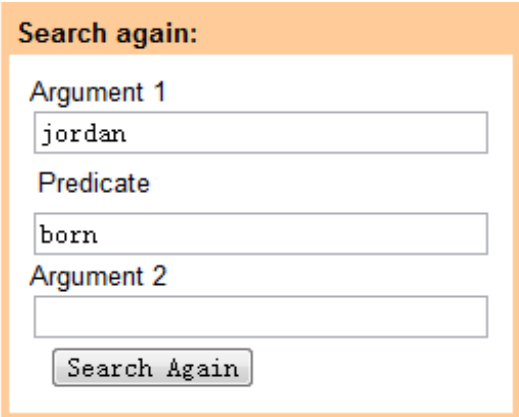


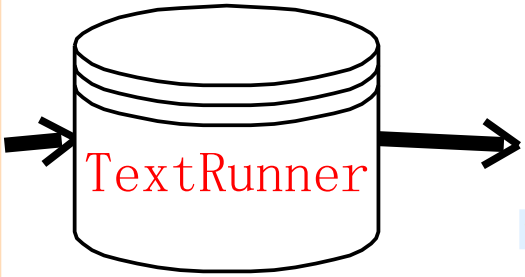
# 开放域关系抽取: Web Page

(Banko IJCAI 2007) (1/2)

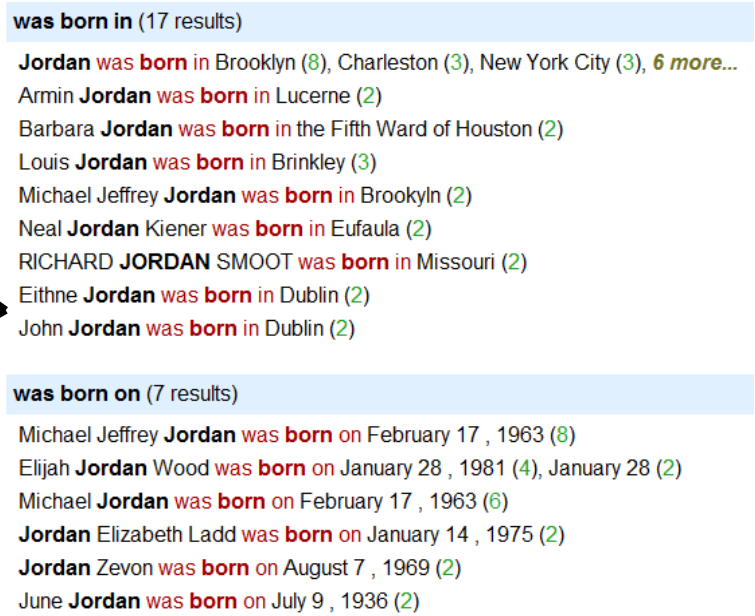
- TextRunner
  - 用户输入特定的谓词和论元
  - 利用搜索引擎返回与用户输入相关的句子
  - 用TextRunner抽取出谓词论元三元组的三元组

Input



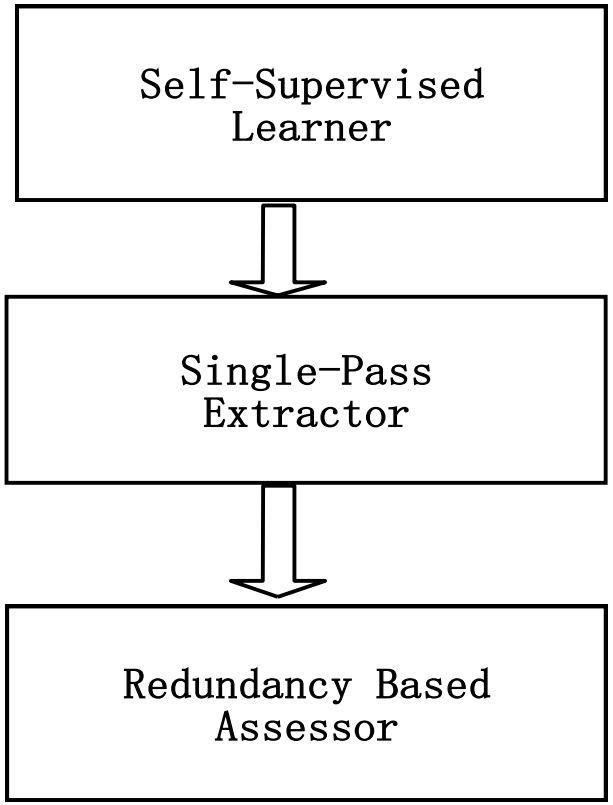

Output



# 开放域关系抽取: Web Page

(Banko IJCAI 2007) (1/2)

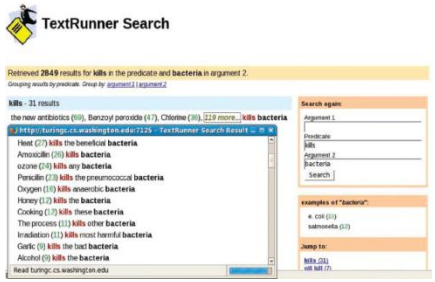
- 步骤：
  - 离线的训练集产生：利用简单的启发式规则，在宾州树库上产生训练语料
  - 离线的分类器训练：提取一些浅层句法特征，训练分类器，用来判断一个元组是否构成关系
  - 在线关系抽取：在网络语料上，找到候选句子，提取浅层句法特征，利用分类器，判断抽取的关系对是否“可信”
  - 在线的关系可信度评估：利用网络海量语料的冗余信息，对可信的关系对，进行评估
- 出发点：
  - 关系类别的产生：动词作为关系类别
  - 训练语料的产生：通过句法关系引出语义关系

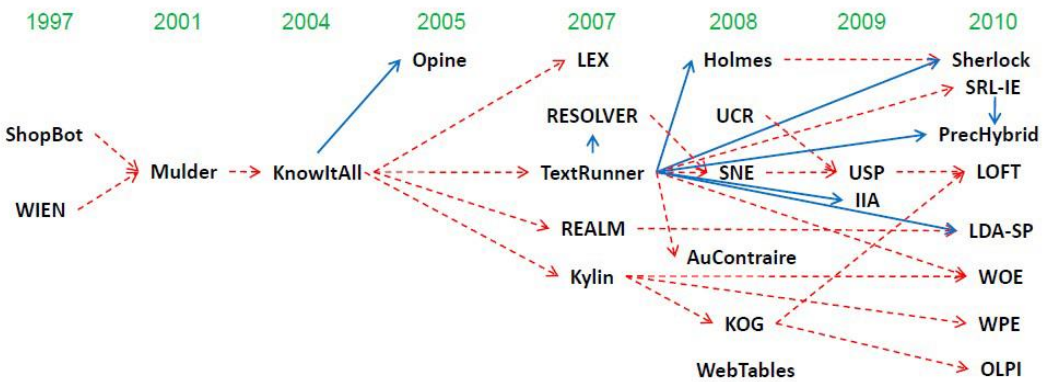


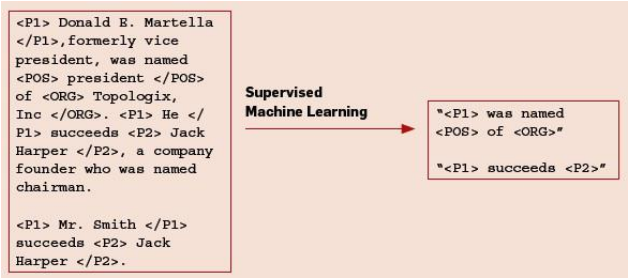
Turing Center
Investigating problems at the cross

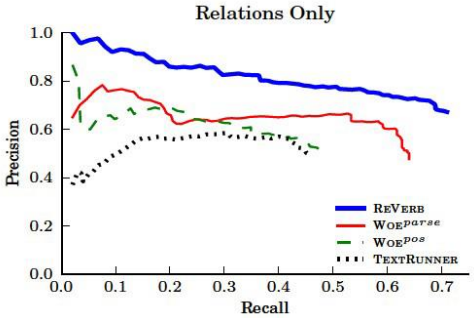
# Machine Reading

- TextRunner、ReVerb、WOE、OLLIE
  - 从Wikipedia Infobox获得关系名
  - 通过在句法树上回标获得句法关系模板



# 开放域关系抽取: Wikipedia (Wu CIKM 2007 )

- 任务：在Wikipedia文本中抽取关系（属性）信息
- 难点
  - 无法确定关系类别
  - 无法获取训练语料
- 方法
  - 在**Infobox**抽取关系信息
  - 在Wikipedia条目文本中进行回标，产生训练语料

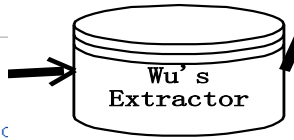
Input

Tsinghua University

From Wikipedia, the free encyclopedia
(Redirected from Qinghua)

*For the university in Taiwan, see National Tsing Hua University.*

**Tsinghua University** (**THU**; simplified Chinese: 清华大学; traditional C
Chinese as **Qinghua**, is a university in Beijing, China. The school is o
1911 under the name "Tsinghua Xuetang" or "Tsinghua College" and was re
was founded in 1925 and the name "National Tsinghua University" was ado
*Commitment*, Tsinghua University describes itself as being dedicated to
global development.[1] Tsinghua is often ranked as the first or second t
rankings.[2][3][4]




Output

Tsinghua University
清华大学

| | |
|---|---|
| Motto | 自强不息，厚德载物 |
| Motto in English | Self-discipline and Social Commitment |
| Established | 1911 |
| Type | Public |
| President | Gu Binglin |
| Academic staff | 2,857 |
| Undergraduates | 13,915 |
| Postgraduates | 12,831 |
| Location | Beijing, People's Republic of China |
| Campus | Urban, 395 ha (3.95 km²) |
| Flower | Redbud and Lilac |
| Colors | Purple and White |
| Affiliations | AEARU, APRU, C9 |
| Website | www.tsinghua.edu.cn |

# 开放域关系抽取: Wikipedia (Wu CIKM 2007)

| Clearfield County, Pennsylvania | |
|---|---|
| **Statistics** | |
| Founded | March 26, 1804 |
| Seat | Clearfield |
| Area | |
| - Total | 2,988 km² (1,154 mi²) |
| - Land | sq mi ( km²) |
| - Water | 17 km² (6 mi²), 0.56% |
| Population | |
| - (2000) | 83,382 |
| - Density | 28/km² |

Clearfield County was created on 1804 from parts of Huntingdon and Lycoming Counties but was administered as part of Centre County until 1812.
Its county seat is Clearfield.

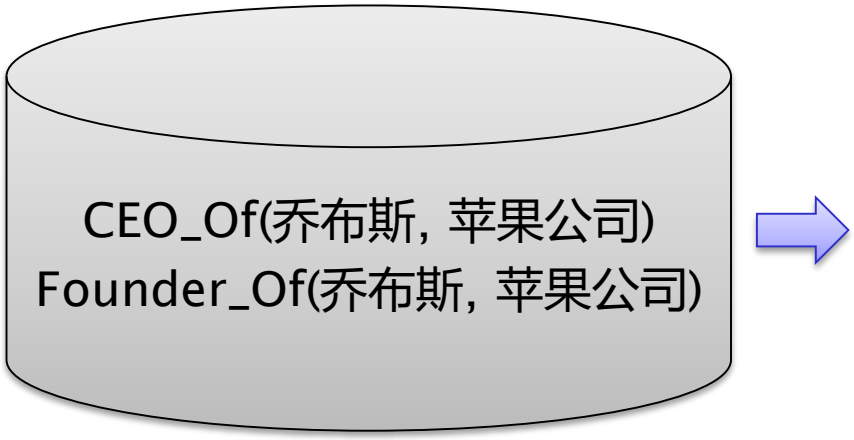
2,972 km² (1,147 mi²) of it is land and 17 km² (7 mi²) of it (0.56%) is water.

As of 2005, the population density was 28.2/km².

# 知识监督开放抽取：Distant Supervision

- 开放域信息抽取的一个主要问题是缺乏标注语料
- **Distant Supervision**: 使用知识库中的关系启发式的标注训练语料

知识库




标注训练语料

| Relation Instance | Label |
|---|---|
| ***S1:*** *乔布斯是苹果公司的创始人之一* | Founder-of, CEO-of |
| ***S2:*** *乔布斯回到了苹果公司* | Founder-of, CEO-of |

# 简单远距离监督方法

(Mintz et al.，ACL09)

- **DS假设**: 每一个同时包含两个实体的句子都会表述这两个实体在知识库中的对应关系
- 基于上述假设标注所有句子作为训练语料
- 使用最大熵分类器来构建IE系统
- 最大的问题：噪音训练实例

噪音训练实例

| Relation Instance | Label | |
|---|---|---|
| *S1:乔布斯是苹果公司的创始人之一* | Founder-of | √ |
| *S1:乔布斯是苹果公司的创始人之一* | CEO-of | x |
| *S2:乔布斯回到了苹果公司* | Founder-of | x |
| *S2:乔布斯回到了苹果公司* | CEO-of | x |

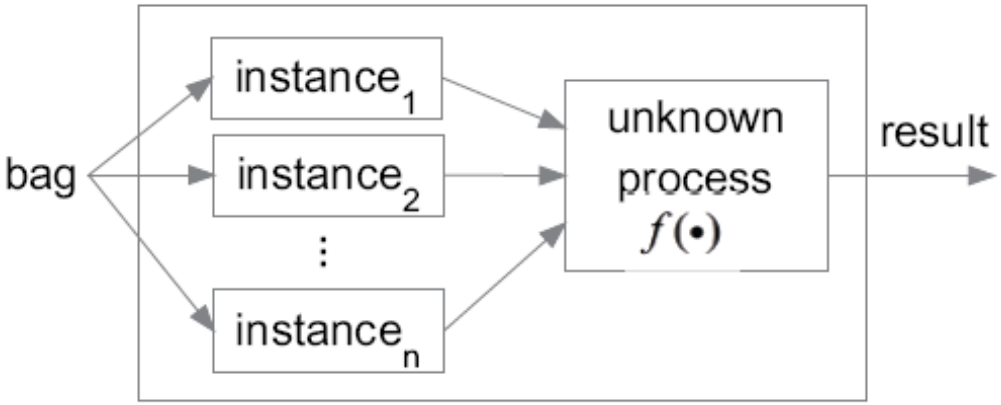
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Zeng EMNLP 2015)

- 实体类别和关系类别巨大
  - Freebase：4000多万实体，上万个属性关系，24多亿个事实三元组
  - DBpeida：400多万实体，48,293种属性关系，10亿个事实三元组
  - NELL： 519万实体，306种关系， 5亿候选三元组
  - Knowledge Vault: 4500万实体，4469种关系，2.7亿三元组
- 人工标注训练集不可行，需要寻求无指导或弱指导的关系抽取方法

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

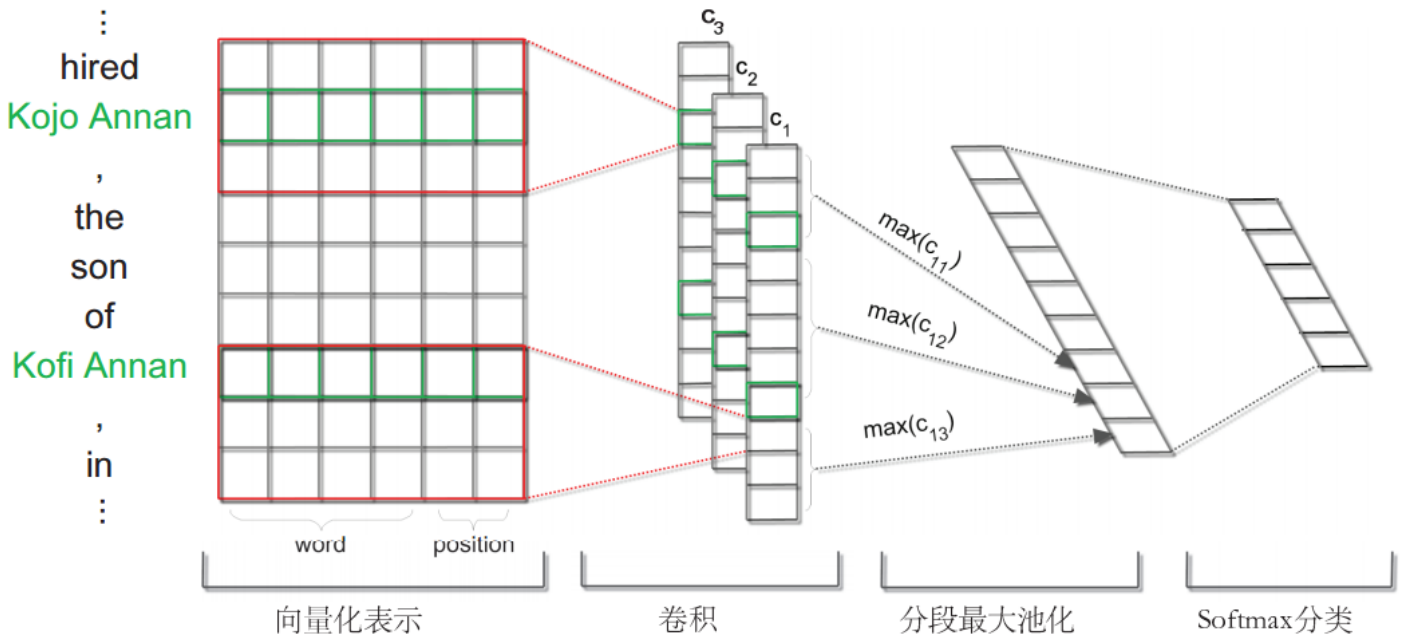
- 将弱监督关系抽取看作是多示例问题，每次选择回标的包中分类概率最大的那个示例更新参数
  - Steve Jobs was the co-founder and CEO of Apple and formerly Pixar.
  - Steve Jobs passed away the day before Apple unveiled iPhone 4S in late 2011.
  - The history of Microsoft began on April 4, 1975, when it was founded by Bill Gates and Paul Allen in Albuquerque.



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

- 利用分段卷积网络自动学习特征
  - 设计分段最大池化层，根据两个实体把句子分成三段，在每段里利用最大池化技术，更好地保留句子的结构化信息



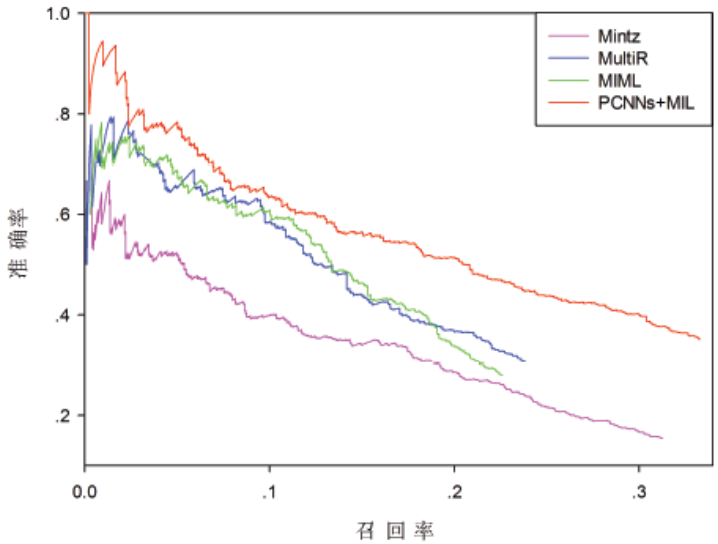
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Zeng EMNLP 2015)

- 数据集：Riedel等开发
  - 设计分段最大池化层，保留句子中结构化信息
  - 知识库：Freebase
  - 文本：纽约时报（NYT）
- 评价方法
  - Held-out 评价：以Freebase 中存在关系的三元组作为标准
  - 人工评价：去掉已经在Freebase 中存在的实体对，人工标注top N 结果

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Zeng EMNLP 2015)

## 实验结果




Held out实验结果

| Top N | Mintz | MultiR | MIML | PCNNs+MIL |
|---|---|---|---|---|
| Top 100 | 0.77 | 0.83 | 0.85 | **0.86** |
| Top 200 | 0.71 | 0.74 | 0.75 | **0.80** |
| Top 500 | 0.55 | 0.59 | 0.61 | **0.69** |
| Average | 0.676 | 0.720 | 0.737 | **0.783** |

人工评价实验结果

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 动机：

**Freebase** /location/location/contains (Nevada, Las Vegas)

S1. **[Nevada] then sanctioned the sport , and the U.F.C. held its first show in [Las Vegas] in September 2001.**

S2. Pinnacle owns casinos in [Nevada], Louisiana , Indiana , Argentina and the Bahamas , but not in the top two American casino cities , Atlantic City and [Las Vegas].

S3. **He has retained two of [Nevada] 's most prominent criminal defense lawyers , Scott Freeman of Reno and David Chesnoff of [Las Vegas].**

S4. The state 's population is growing , but not skyrocketing the way it is in Arizona and [Nevada] , and with no city larger than 100,000 residents , Montana essentially does not have suburbs or exurbs like those spreading around Phoenix, [Las Vegas] and Denver.

**Descriptions**

[Nevada]: Nevada is a state in the Western, Mountain West, and Southwestern regions of the United States.

[Las Vegas]: officially the City of Las Vegas and often known as simply Vegas, is a city in the United States, the most populous city in the state of Nevada, the county seat of Clark County, and the city proper of the Las Vegas Valley.

- **传统方法**：NLP工具抽取句子特征，概率图模型选择多个有效的句子，然后分类。
- **神经网络方法**：卷积神经网络抽取特征，效果最好，但是Zeng的方法只能选择一个有效句子，不能充分利用监督信息。
- **动机一**：两种方法优点的结合，既能选择多个有效的句子，又使用卷积神经网络抽取特征。

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 动机：

**Freebase** /location/location/contains (Nevada, Las Vegas)

S1. **[Nevada] then sanctioned the sport , and the U.F.C. held its first show in [Las Vegas] in September 2001.**

S2. Pinnacle owns casinos in [Nevada], Louisiana , Indiana , Argentina and the Bahamas , but not in the top two American casino cities , Atlantic City and [Las Vegas].

S3. **He has retained two of [Nevada] 's most prominent criminal defense lawyers , Scott Freeman of Reno and David Chesnoff of [Las Vegas].**

S4. The state 's population is growing , but not skyrocketing the way it is in Arizona and [Nevada] , and with no city larger than 100,000 residents , Montana essentially does not have suburbs or exurbs like those spreading around Phoenix, [Las Vegas] and Denver.

**Descriptions**

[Nevada]: Nevada is a state in the Western, Mountain West, and Southwestern regions of the United States.

[Las Vegas]: officially the City of Las Vegas and often known as simply Vegas, is a city in the United States, the most populous city in the state of Nevada, the county seat of Clark County, and the city proper of the Las Vegas Valley.

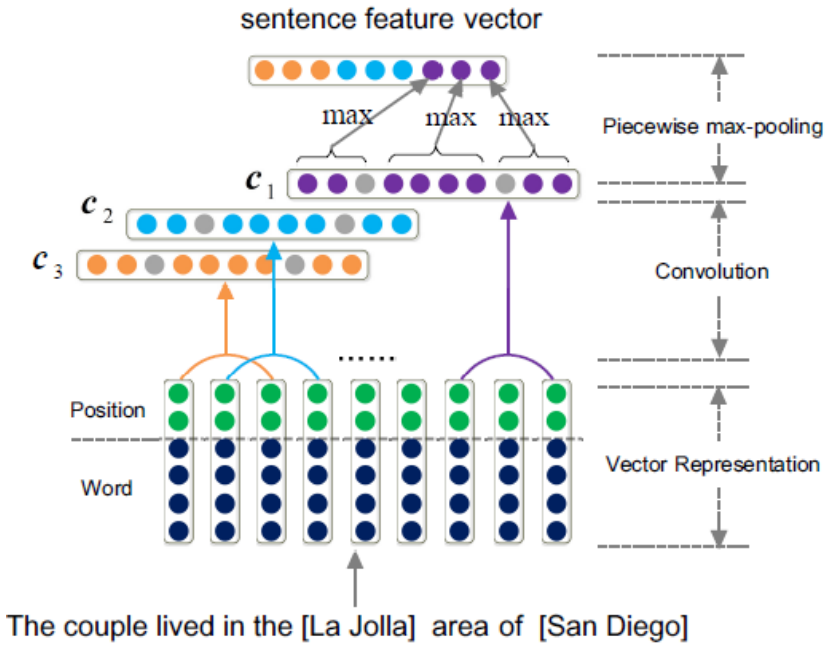
- 句子S1需要知道Nevada是个地名，不然根据其中单词sanctioned(批准)容易判断Nevada是个官员的名字，那么就没有location/location/contains的关系。
- **动机二**：利用实体描述补充背景知识

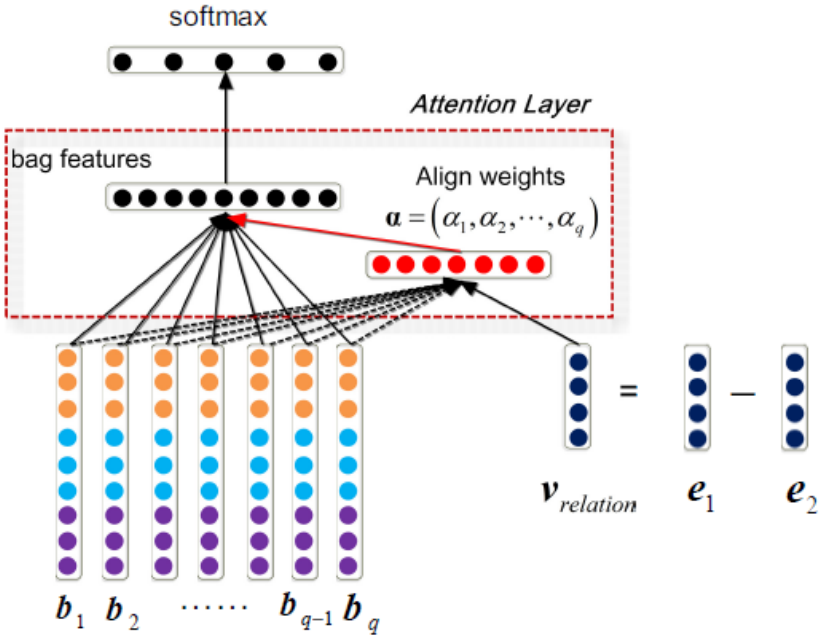
# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

## ■基于句子级关注机制和实体描述的弱监督关系模型

- 利用背景知识库信息得到两个实体的关系，再利用这个关系向量和每个句子计算相似度，归一化后作为句子的关注度



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

- 数据集：Riedel等开发
  - 由(Riedel, Yao, and McCallum 2010) 发布的数据集，通过对齐Freebase和NTY的语料形成，共包含52种关系。数据统计如下：

| | 训练集 | 测试集 |
|---|---|---|
| 包的个数 | 67,946 | 96,676 |

  - 训练和预测的对象均是多示例包，包中的句子均没有标签。新版本的数据统计：

| | 训练集 | 测试集 |
|---|---|---|
| 包的个数 | 522,611 | 96,676 |

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

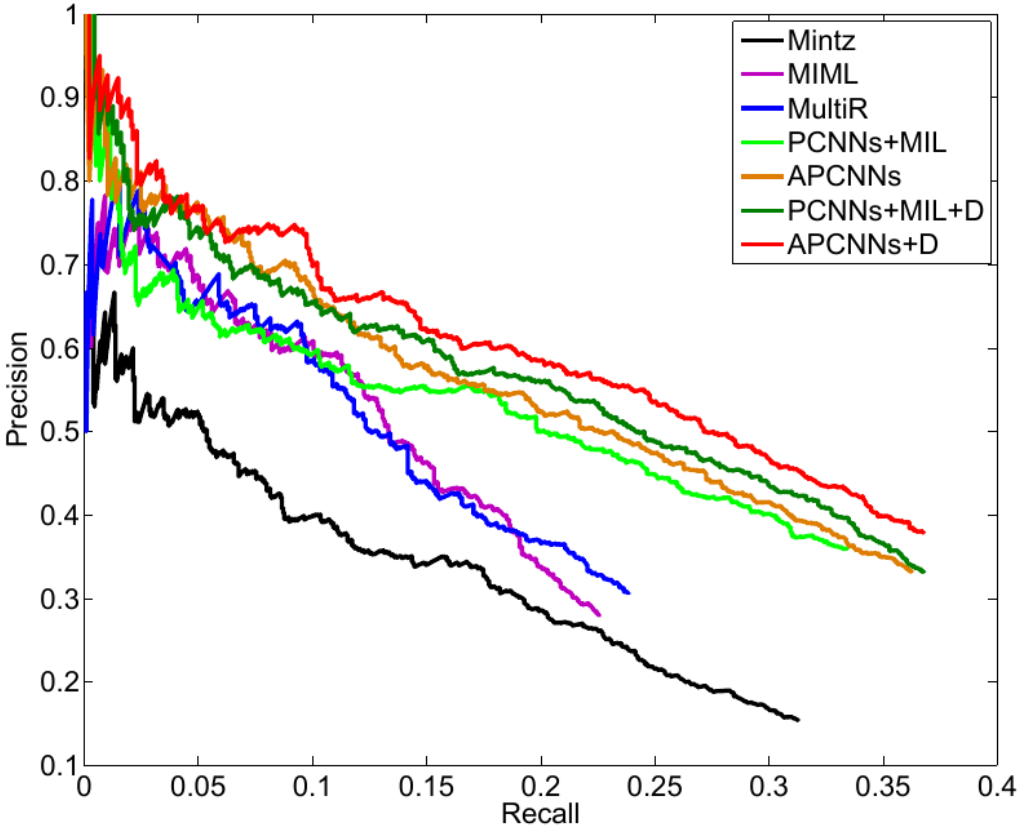
(Ji AAAI 2017)

- 评价指标
  - **自动评价**：测试集中的标签以知识谱图的标注为准，如果预测的标签与标注相同，则预测正确；否则预测错误。结果以精度-召回率曲线给出。
  - **人工评价**：由于知识图谱不完善，测试集中标注为”NA”的样本也可能真实地表达了某种关系，只是知识图谱标注错误了。此时选取被知识图谱标注为”NA”,但是模型预测出其他关系的样本，按照置信度的排列，选取top-k个样本进行人工评价，只报告正确率。

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

## 实验结果：自动评价



# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Ji AAAI 2017)

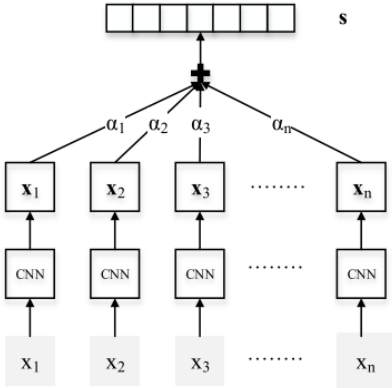
## 实验结果：人工评价

| Accuracy (%) | Top 100 | Top 200 | Top 500 | Average |
|---|---|---|---|---|
| Mintz | 0.77 | 0.71 | 0.55 | 0.676 |
| MultiR | 0.83 | 0.74 | 0.59 | 0.720 |
| MIML | 0.85 | 0.75 | 0.61 | 0.737 |
| PCNNs+MIL | 0.86 | 0.80 | 0.69 | 0.783 |
| APCNNs | **0.87** | 0.82 | 0.72 | 0.803 |
| PCNNs+MIL+D | 0.86 | 0.82 | 0.71 | 0.797 |
| APCNNs+D | **0.87** | **0.83** | **0.74** | **0.813** |

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别

(Lin ACL 2016)

- 动机：弱监督数据中包含了误标记的数据，为了减轻误标记句子对分类结果的影响，引入了关注机制（Attention），每个包中可以选多个示例。




- 方法：
  - 给定一个句子集合 $S = \{x_1, x_2, \dots, x_n\}$
  - 将每个句子通过CNN表示成一个向量
    - $X_i = CNN(x_i)$
  - 句子集合表示为句子表示的加权和 $S = \sum_i \alpha_i X_i$
  - 最后通过 softmax 分类器对句子集合进行分类
  - 关注机制 $\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$
    - $e_i = x_i A r$，$A$ 是权重对角矩阵 $r$ 是关系r的向量表示

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Lin ACL 2016)

## 实验：

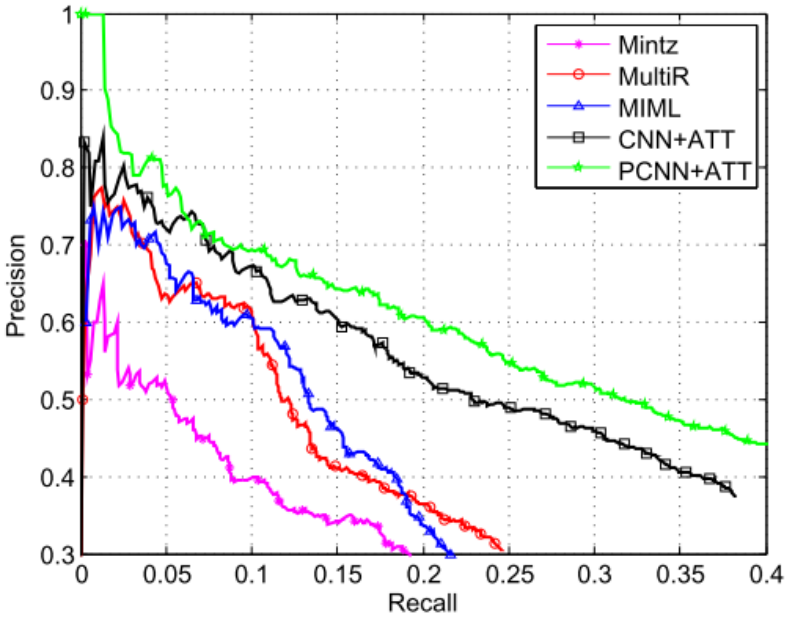
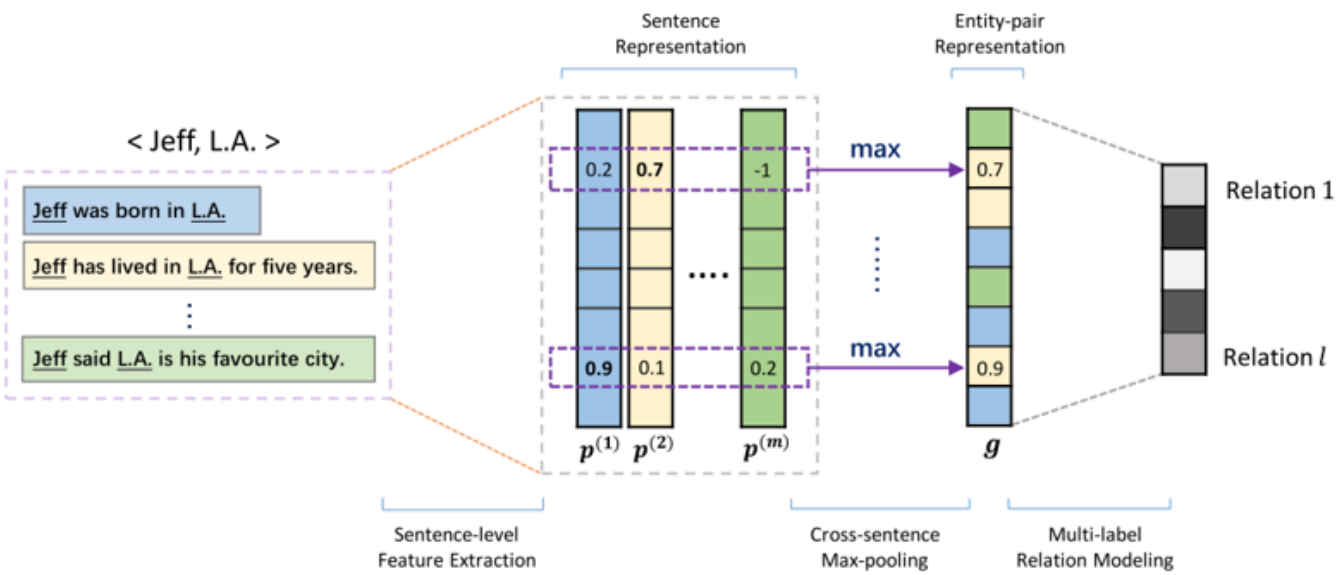



Figure 4: Performance comparison of proposed model and traditional methods

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Jiang COLING 2016)

- 动机：
  - 有些隐含的关系需要从多句话中抽取证据才能确定
  - 同一个实体对可以对应多个关系

- 方法：




1.通过跨句子的pooling实现多句话语义的联合考虑
2.通过多目标的损失函数实现一个实体对多关系的发现

# 开放域关系抽取: 从NYT中抽取Freebase的关系类别
(Jiang COLING 2016)

## 实验结果

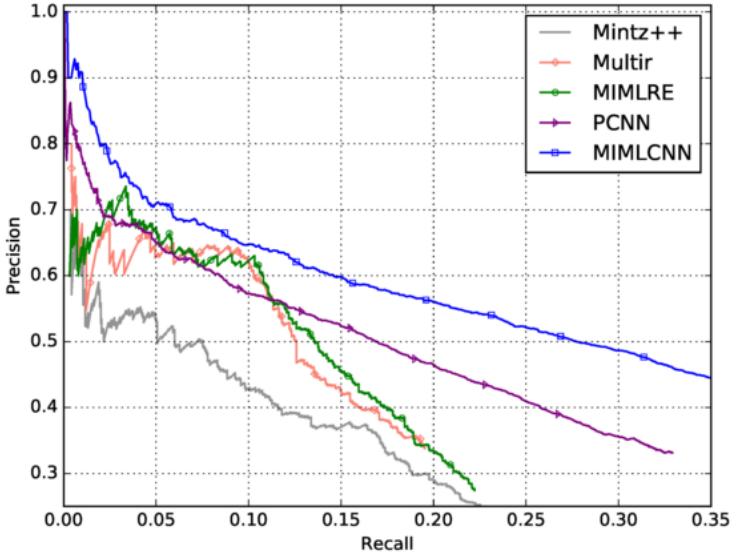
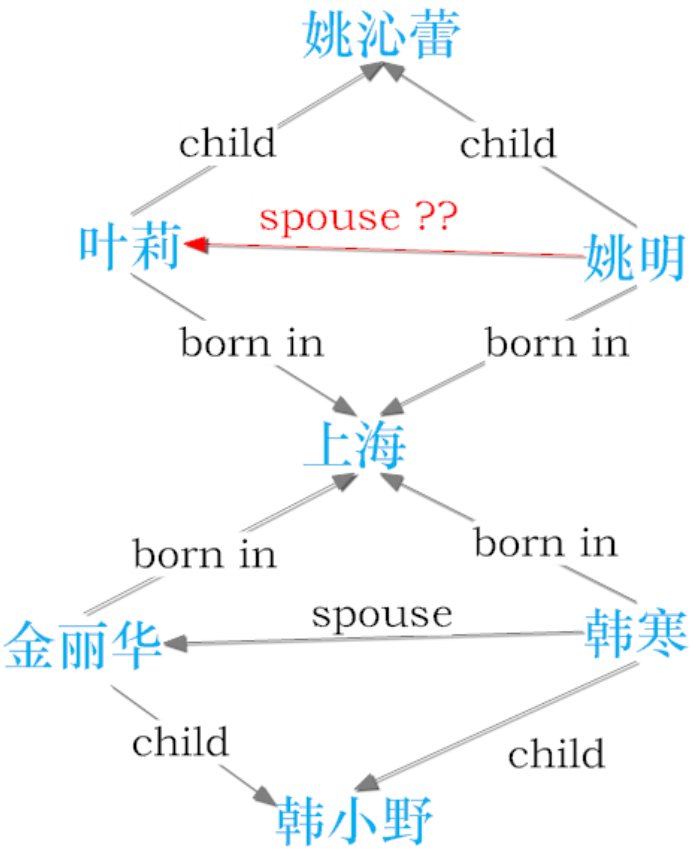
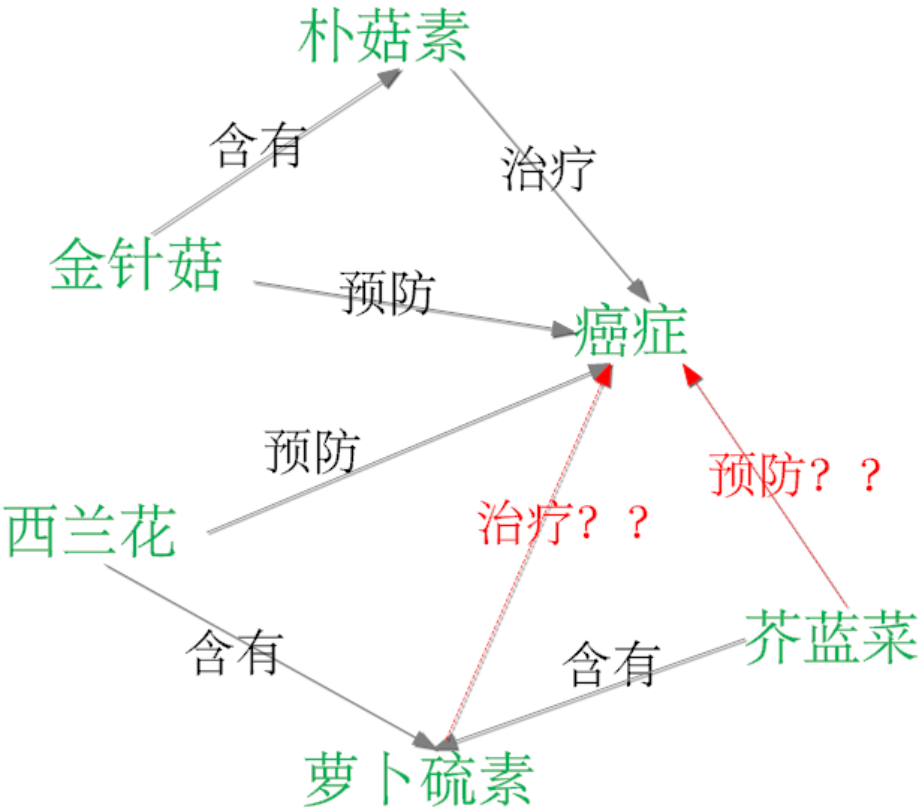



Figure 4: Precision-recall curves of the proposed method and four baselines.

# 关系发现：链接预测（知识库补全）

- 利用知识图谱中现有的知识推断未知知识（未知的链接）
- 知识图谱本身就蕴涵着推理模式（第10课时 王泉老师）



# 小结

- 由识别到抽取
  - 规范文本→有噪音、有冗余的海量网络数据
- 限定类别→开放类别
- 难点：关系类别缺乏体系结构

Q&A

# Thanks

# 事件抽取

赵 军（jzhao@nlpr.ia.ac.cn）

中国科学院自动化研究所
模式识别国家重点实验室

中国科学院自动化研究所
Institute of Automation, Chinese Academy of Sciences

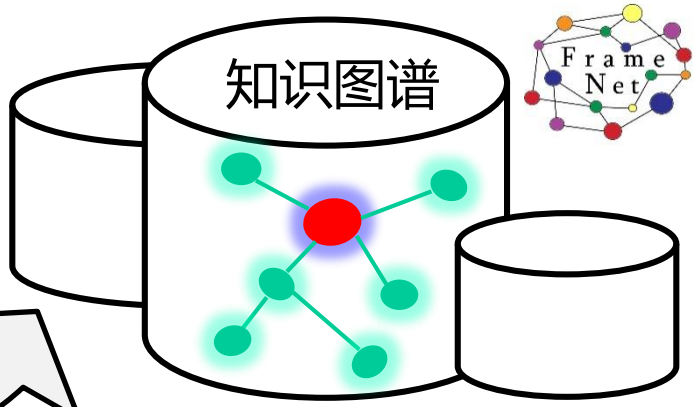
# 概述

- 事件抽取的任务定义
- 事件抽取评测及语料资源
- 事件抽取的典型方法
- 事件抽取的相关应用

# 背景

实体抽取
关系抽取
事件抽取

信息抽取技术



# 背景

- 事件抽取是信息抽取中的难点问题
  - 事件抽取依赖实体抽取和关系抽取
  - 相较于实体抽取和关系抽取，事件抽取难度更大

*例：1992 年10 月3 日，奥巴马与米歇尔在三一联合基督教堂结婚*

实体抽取：
1992年10月3日， 奥巴马，米歇尔，三一联合基督教堂

关系抽取：
夫妻关系：奥巴马，米歇尔

事件抽取：
事件类型：结婚事件，配偶：奥巴马，配偶：米歇尔
，时间： 1992年10月3日，地点：*三一联合基督教堂*

# 事件的定义

事件起源于认知科学，常常在哲学、语言学、计算机科学等领域被广泛讨论。但遗憾的是目前对事件还没有统一的定义，在不同领域，针对不同的应用，不同的人对事件有不同的描述。

# 事件的定义

**事件**是发生在某个特定的**时间点或时间段**、某个特定的**地域范围内**，由**一个或者多个角色**参与的**一个或者多个动作**组成的**事情**或者**状态的改变**

- 不同的动作或者状态的改变代表不同类型的事件

- 同一个类型的事件中不同的元素代表了不同的事件实例
- 同一个类型的事件中不同粒度的元素代表不同粒度的事件实例

第二次世界大战

抗日战争

淞沪会战

# 事件抽取的定义

## ■事件抽取的定义

从自然语言文本中抽取出用户感兴趣的事件信息并以结构化的形式呈现出来，如什么人，什么时间，在什么地方，做了什么事

## ■事件抽取相关的术语

事件描述（Event mention），事件触发词（Event Trigger），事件元素（Event argument），元素角色（Argument role）

*例：1992年10月3日，奥巴马与米歇尔在三一联合基督教堂结婚。*

| 事件触发词 | 结婚(a “Life/Marry”event) | |
|---|---|---|
| 事件元素 | 角色 = 配偶 | 奥巴马 |
| | 角色 = 配偶 | 米歇尔 |
| | 角色 = 时间 | 1992年10月3日 |
| | 角色=地点 | 三一联合基督教堂 |

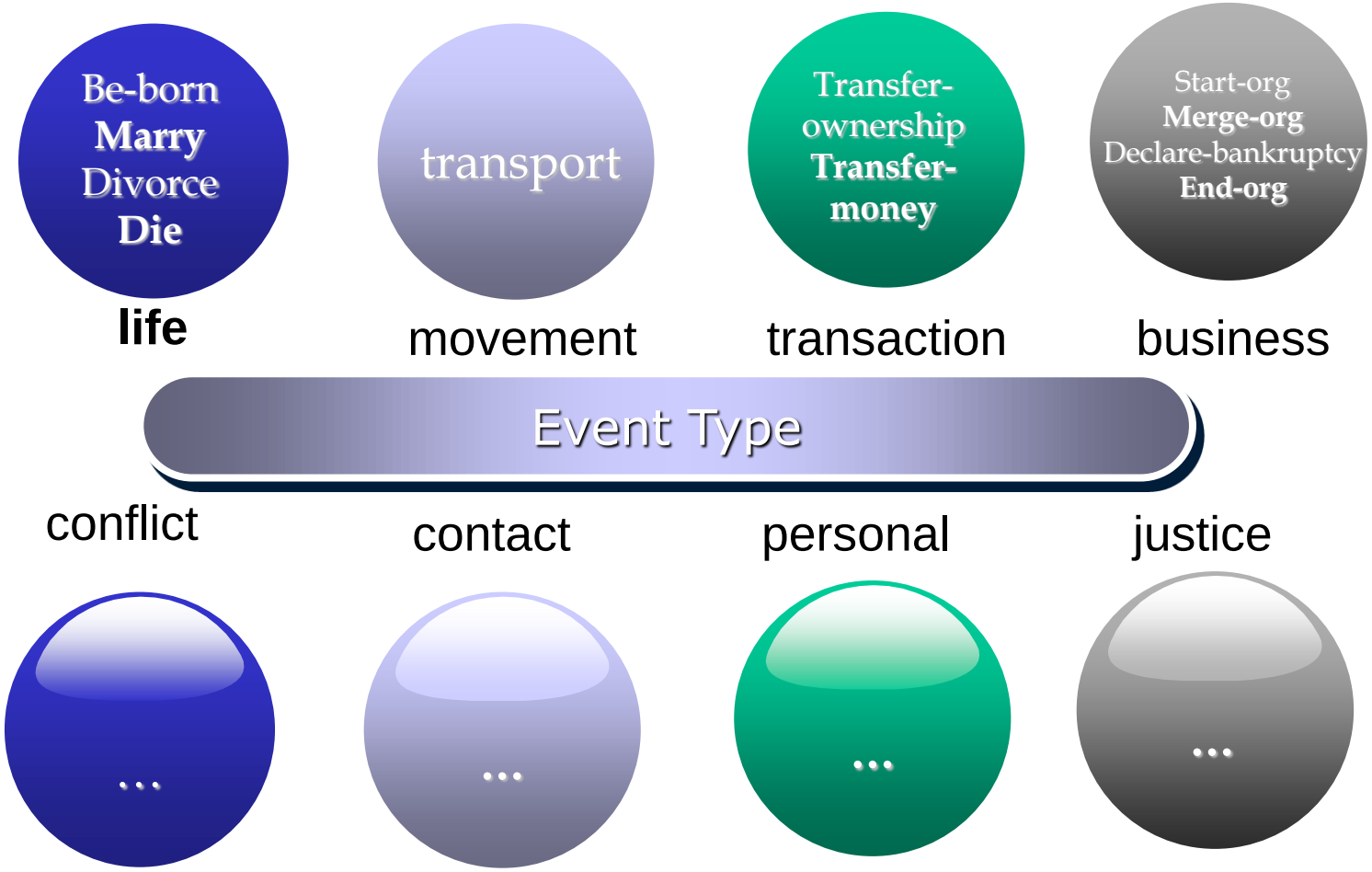
# 概述

- 事件抽取的任务定义
- 事件抽取评测及语料资源
- 事件抽取的典型方法
- 事件抽取的相关应用

# 事件抽取评测及语料

| | MUC | TDT | ACE(KBP) |
|---|---|---|---|
| 全 称 | Message Understanding Conference | Topic Detection and Tracking | Automatic Content Extraction |
| 举办方 | Defense Advanced Research Projects Agency（DARPA） | DARPA | National Institute of Standards and Technology(NIST) |
| 时 间 | 1987-1997 | 1998-2004 | ACE：2000-2008<br>KBP：2014-2017 |
| 评测内容 | 抽取指定的事件，包括参与这些事件的各个实体、属性和关系。例如：MUC-2是从海军军事情报中抽取事件填入预定义模板中，共10个槽 | 将文本切割为不同的新闻报道，监控其中新事件的报道，并且将同一话题下的分散的报道按照某种结构有效组织起来。TDT-3:240个topic | 指定的源语言数据中发现特定类型的事件，并且识别出与事件相关的信息填入预设的事件模板中。ACE中共计8大类33个小类的事件 |

# ACE事件类型示例

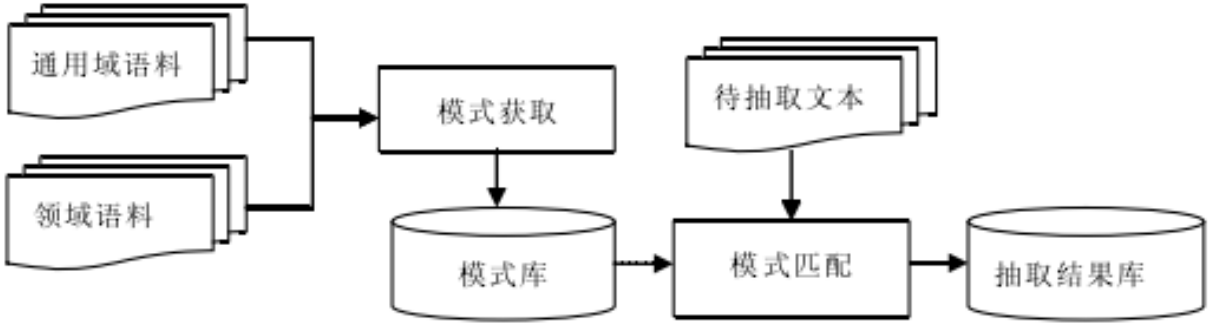

Be-born
Marry
Divorce
Die
life
transport
movement
Transfer-ownership
Transfer-money
transaction
Start-org
Merge-org
Declare-bankruptcy
End-org
business
Event Type
conflict
contact
personal
justice
…
…
…
…

# 概述

- 事件抽取的任务定义
- 事件抽取评测及语料资源
- 事件抽取的典型方法
- 事件抽取的相关应用

# 基于模式匹配的方法

- 平面模式主要基于词袋等字符串特征构成模式
- 结构模式更多地考虑了句子的结构信息，融入句法分析特征




模式匹配的信息抽取基本框架

- 缺点：
  - 领域相关，可扩展性差
  - 需要大量的人工标注，耗时耗力
- 相关工作：

  （Riloff 1993, Kim 1995, Yangarber 2001, Califf 2003等）

# 基于机器学习的方法

- 目前主要采用统计机器学习的方法，将事件实例转换成高维空间中的特征向量或直接用离散结构来表示，在标注语料库上训练生成模型，然后再识别事件及其元素
  - 基于特征向量方法：利用最大熵、朴素贝叶斯和支持向量机等（Ralph Grishman 2005, Ahn 2006, Ji 2008, Ji 2009, Liao 2010 Hong 2011）
  - 基于结构的方法：依存树结构、自定义联合结构等（ McClosky et al. 2011 ，Li et al.2013，Li et al. 2014 ）
  - 基于神经网络的方法：多层感知机、卷积神经网络、循环神经网络等（Chen et al. 2015，Feng et al. 2016， Liu et al. 2017，Chen et al. 2017，Liu et al. 2017 ）

# 基于机器学习的方法

- 基于特征向量的方法：
  - 主要问题：如何获取各种有效的词法、句法、语义等特征，并把它们有效地集成起来，从而产生描述单词或者词组触发事件和候选事件元素扮演事件角色的各种局部特征和简单的全局特征
  - 基于句子级信息： 从句子中提取特征，利用最大熵、朴素贝叶斯和支持向量机等模型去完成事件抽取 （ Ralph Grishman 2005, Ahn 2006, ）
  - 基于篇章和背景信息的： 在句子级信息基础上更多地考虑篇章级信息和丰富的背景知识（跨文档信息、跨语言信息、跨文本事件信息、跨实体信息）（Ji 2008, Ji 2009, Liao 2010, Hong 2011, Liu 2016等）

# 基于机器学习的方法

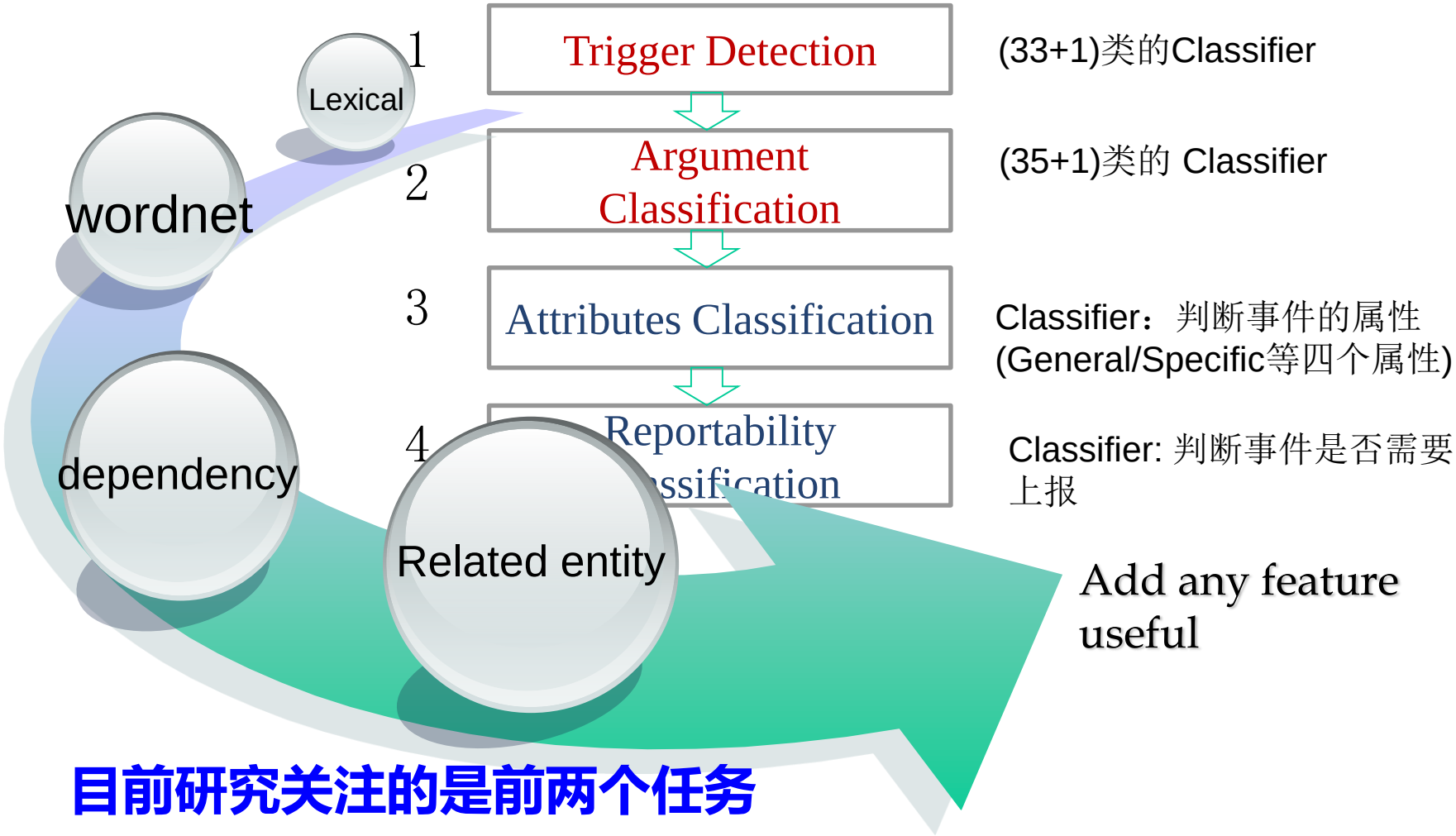
- 基于结构的方法:
  - 主要问题：将事件抽取看成一个结构预测的问题。如何建模事件的结构并有效地挖掘反映事件的结构化的信息及特征
  - 看做依存树结构预测问题：生物事件（McClosky et al. 2011）
  - 自定义联合结构的预测问题：触发词和事件元素联合预测（Li et al.2013，Li et al. 2014）

# 基于机器学习的方法

- 基于神经网络的方法:
  - 主要问题：如何设计合理的网络结构，从而捕捉更多的信息，进而更准确的完成关系的抽取
  - 网络结构：不同的网络结构捕捉文本中不同的信息
    - 卷积神经网络（CNN）（Chen et al. 2015, Nguyen et al. 2015, Chen et al.2017）
    - 循环神经网络（RNN）(Feng et al. 2016, Nguyen et al. 2016)
    - 多层感知机（MLN）(Liu et al. 2016, Liu et al.2017)

# 基于特征向量的方法：句子级特征（传统特征+多分类器）

The stages of event extraction （Ahn，ACL 2006）




**目前研究关注的是前两个任务**

# 基于特征向量的方法：句子级特征（传统特征+多分类器）

The stages of event extraction （Ahn，ACL 2006）

## 实验

- 数据集

  ACE 2005

- 结果

| | anchors | args | attrs | coref | ACE value |
|---|---|---|---|---|---|
| 1 | gold | gold | gold | none | 81.1% |
| 2 | gold | learned | gold | none | 45.5% |
| 3 | learned | learned | maj | none | 22.1% |
| 4 | gold | gold | maj | none | 75.8% |
| 5 | gold | gold | learned | none | 76.3% |
| 6 | gold | learned | learned | none | 43.9% |
| 7 | learned | learned | learned | none | 21.7% |
| 8 | gold | gold | gold | learned | 88.6% |
| 9 | gold | gold | learned | learned | 79.4% |
| 10 | gold | learned | gold | learned | 46.1% |
| 11 | gold | learned | learned | learned | 43.3% |
| 12 | learned | learned | learned | learned | 22.3% |

# 基于特征向量的方法：句子级特征+背景信息（cross-entity）

Using Cross-Entity Inference to Improve Event Extraction（Hong，ACL 2011）

## ■动机

- 实体的类别信息能辅助事件抽取
  - E.g. left (End-position or Transport)
  - 1)He left *the bathroom*

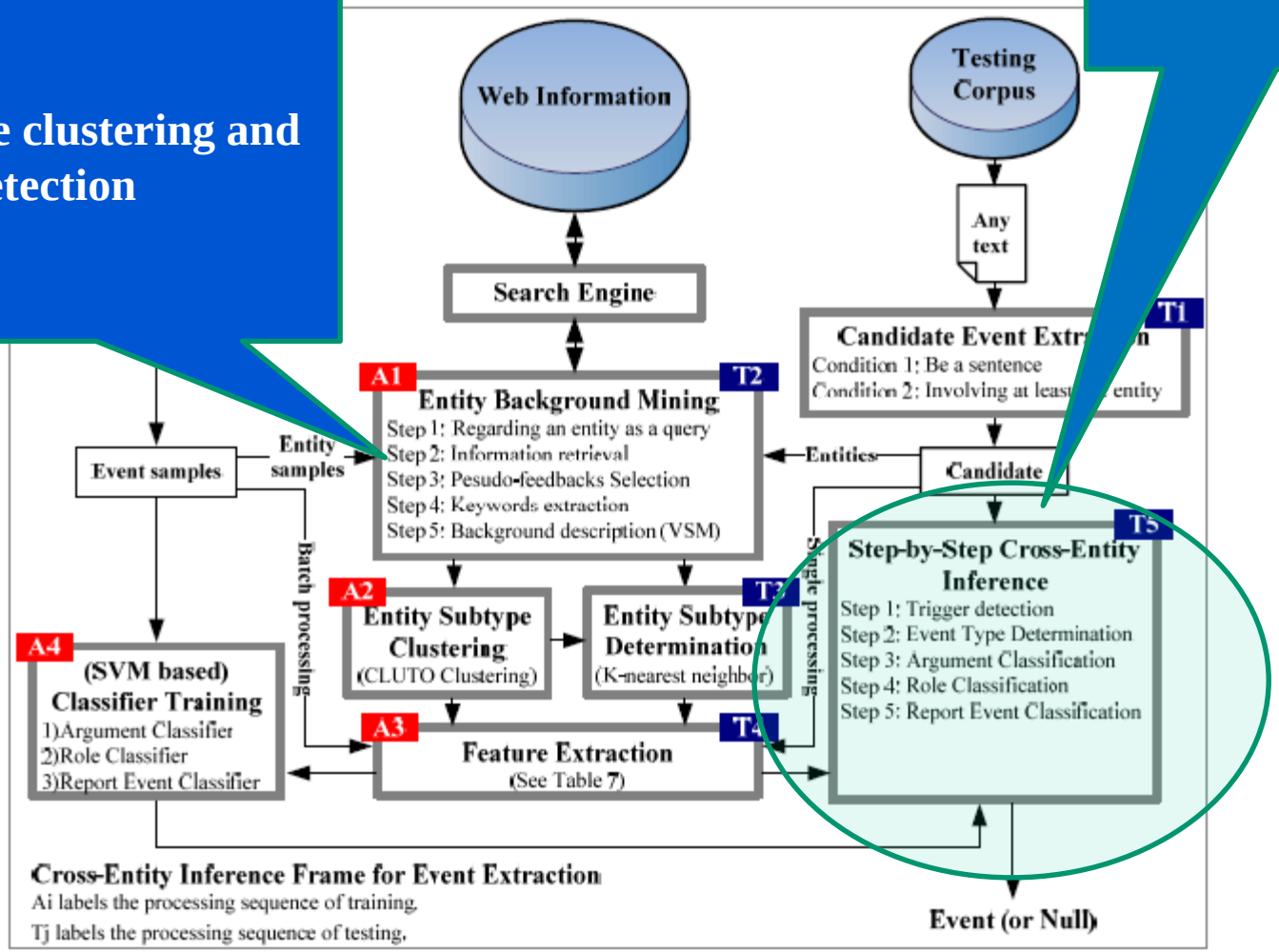
  - 2)He left *the Microsoft*
  - (what are their event types?)

# 基于特征向量的方法：句子级特征+背景信息（ cross-entity ）

Using Cross-Entity Inference to Improve Even
（Hong，ACL 2011）

**Entity type clustering and detection**

**Here is the step-by-step cross-entity inference**




**Cross-Entity Inference Frame for Event Extraction**
Ai labels the processing sequence of training.
Tj labels the processing sequence of testing.

# 基于特征向量的方法：句子级特征+背景信息（ cross-entity ）

Using Cross-Entity Inference to Improve Event Extraction（Hong，ACL 2011）

## 实验

- 数据集

  ACE 2005

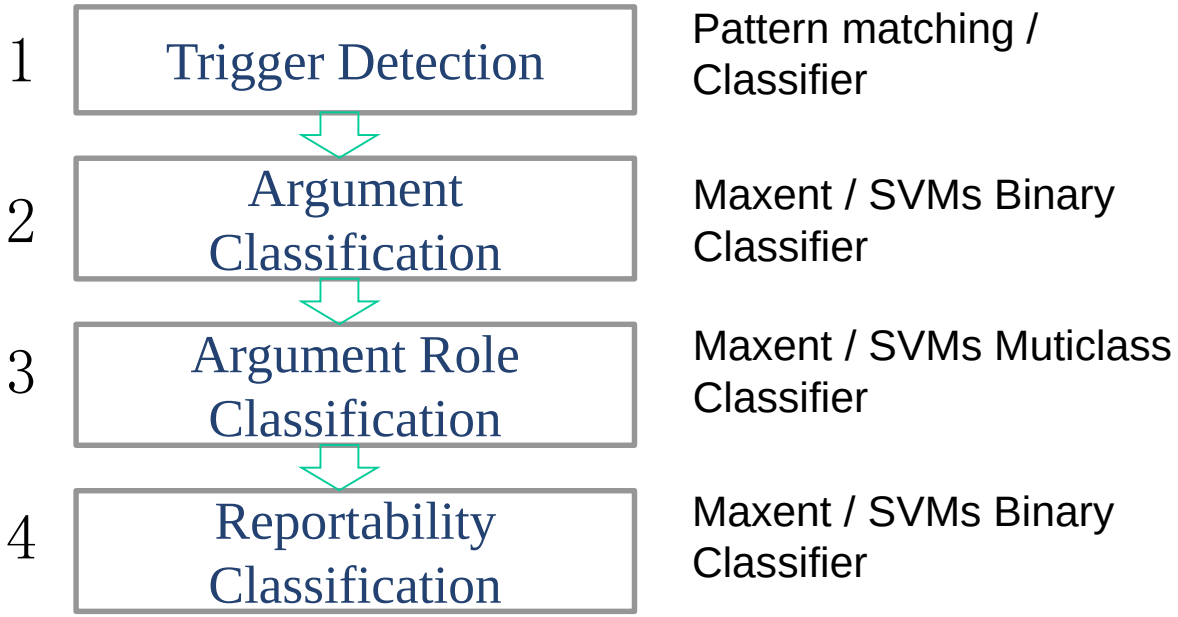
- 结果

| Performance / System/Human | Trigger (%) | | | Argument (%) | | | Role (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F |
| Sentence-level baseline | 67.56 | 53.54 | 59.74 | 46.45 | 37.15 | 41.29 | 41.02 | 32.81 | 36.46 |
| **Cross-event inference** | **68.71** | **68.87** | **68.79** | **50.85** | **49.72** | **50.28** | **45.06** | **44.05** | **44.55** |
| Cross-entity inference (optimum) | 73.4 | 66.2 | 69.61 | 56.96 | 55.1 | 56 | 49.3 | 46.59 | 47.9 |
| Cross-entity inference (worst) | 71.3 | 64.17 | 66.1 | 51.28 | 50.3 | 50.78 | 46.3 | 44.3 | 45.28 |
| **Cross-entity inference (mean)** | **72.9** | **64.3** | **68.33** | **53.4** | **52.9** | **53.15** | **51.6** | **45.5** | **48.36** |
| Human annotation 1 (blind) | 58.9 | 59.1 | 59.0 | 62.6 | 65.9 | 64.2 | 50.3 | 57.69 | 53.74 |
| Human annotation 2 (know rules) | 74.3 | 76.2 | 75.24 | 68.5 | 75.8 | 71.97 | 61.3 | 68.8 | 64.86 |

# 基于结构的方法

Joint Event Extraction via Structured Prediction with Global Features（Li et al., ACL 2013）

- 动机




- 传统方法
  - 将事件抽取分成多个步骤错误会由上层向下层传递
  - 没有考虑触发词和事件元素之间的相互影响

# 基于结构的方法

## 动机

1 ) In Baghdad, a *cameraman* died when an ***American tank*** *fired* on the Palestine Hotel.

2) He has *fired* his ***air defense chief***.

- "*fired*" 存在歧义：Attack, End-Position, NULL…
- 同时考虑事件元素和事件触发词的标注会提升最终效果

American tank:
**Instrument (Attack)**

Air defense chief:
**Position (End-Position)**

"You should check your e-mails more often. I fired you over three weeks ago."

# 基于结构的方法

## 方法

- 将事件抽出看成一个结构预测问题
- 利用beam search缩小搜索解空间

$x = \langle (x_1, x_2, ..., x_s), \mathcal{E} \rangle$ 代表句子的实例

$\mathcal{E} = \{e_k\}_{k=1}^{m}$ 代表这句话中的事件元素候选集

$y = (t_1, a_{1,1}, \ldots, a_{1,m}, \ldots, t_s, a_{s,1}, \ldots, a_{s,m})$ 代表预测的整个结构

## 实验对比

| Methods | Trigger Identification (%) | | | Trigger Identification + classification (%) | | | Argument Identification (%) | | | Argument Role (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| Sentence-level in Hong et al. (2011) | N/A | | | 67.6 | 53.5 | 59.7 | 46.5 | 37.15 | 41.3 | 41.0 | 32.8 | 36.5 |
| Staged MaxEnt classifiers | 76.2 | 60.5 | 67.4 | 74.5 | 59.1 | 65.9 | 74.1 | 37.4 | 49.7 | 65.4 | 33.1 | 43.9 |
| Joint w/ local features | 77.4 | 62.3 | 69.0 | 73.7 | 59.3 | 65.7 | 69.7 | 39.6 | 50.5 | 64.1 | 36.5 | 46.5 |
| Joint w/ local + global features | 76.9 | 65.0 | **70.4** | 73.7 | 62.3 | **67.5** | 69.8 | 47.9 | **56.8** | 64.7 | 44.4 | **52.7** |
| Cross-entity in Hong et al. (2011)† | N/A | | | 72.9 | 64.3 | 68.3 | 53.4 | 52.9 | 53.1 | 51.6 | 45.5 | 48.3 |

# 基于神经网络的方法：
# 基于动态最大池化技术的卷积神经网络

Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks（Chen et al., ACL 2015）

## 动机

- 传统方法主要提取两类特征：词汇特征和句子级特征

词汇特征：过于依赖人工；是一种独热表示，缺乏语义

S1: Obama **beats** McCain. (Elect Event)

S2: Tyson **beats** his opponent. (Attack Event)

- Beat有两种语义，触发不同的事件。如果知道Obama和mccain是总统，Tyson是拳击选手这些，能更好地判别beat的类型。
- 我们用词向量的方法捕捉这种语义信息。

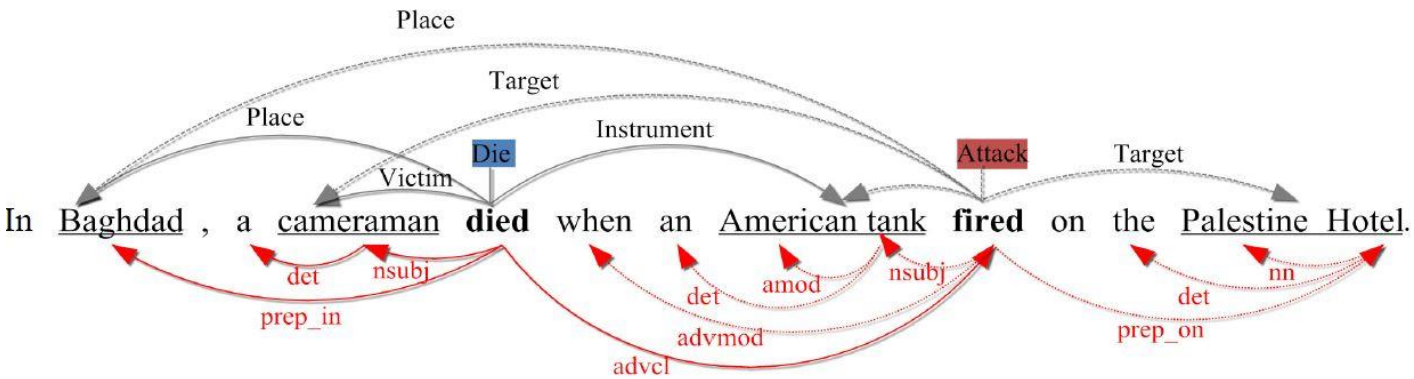
# 基于神经网络的方法：
# 基于动态最大池化技术的卷积神经网络

Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks（Chen et al., ACL 2015）

## 动机

- 传统方法主要提取两类特征：词汇特征和句子级特征

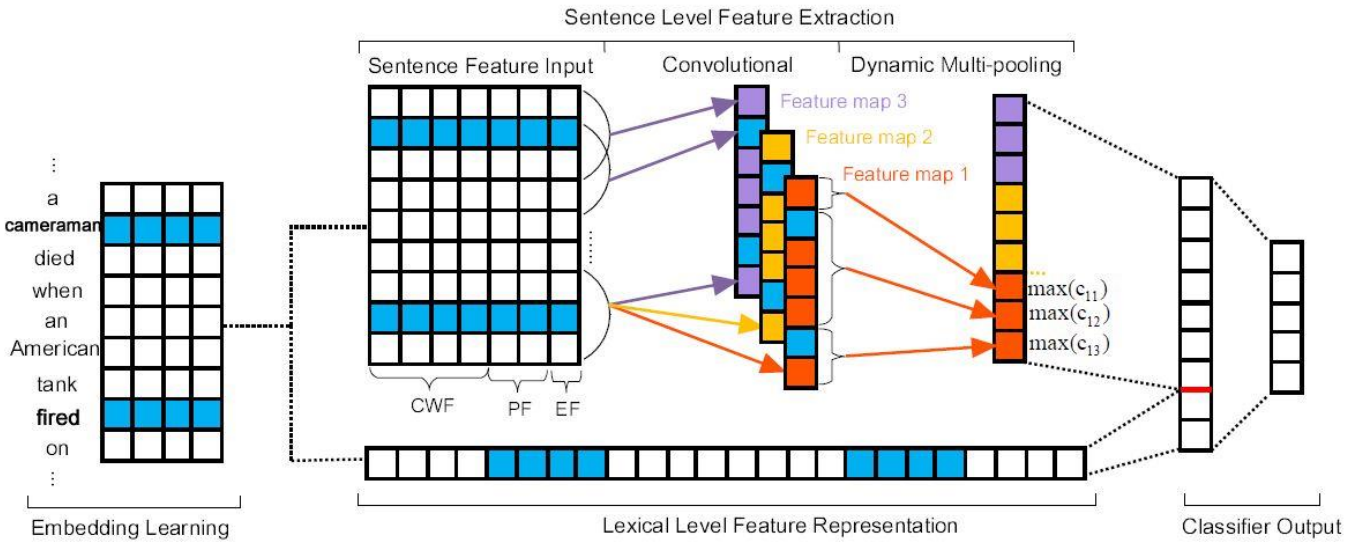
句子级特征：会用到复杂的NLP工具，引起错误传递




nsubj -> (cameraman plays the Victim role in die event)
????? -> (cameraman plays the Target role in Attack event)

# 基于神经网络的方法：
# 基于动态最大池化技术的卷积神经网络

## 方法-DMCNN




- 由4部分组成：word-embedding learning; lexical-level feature representation； sentence-level feature extraction；argument classifier output
- 传统的CCN利用最大池化技术，选取最大值代表每个feature map 但是事件抽取任务中一句话存在两个或者多个事件所以提出动态最大池化技术来根据trigger和argument候选的位置，在不损失最大值的前提下，获取更多的有用信息

# 基于神经网络的方法：
# 基于动态最大池化技术的卷积神经网络

## 实验比较

| Methods | Trigger Identification(%) | | | Trigger Identification + Classification(%) | | | Argument Identification(%) | | | Argument Role(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Li's baseline | 76.2 | 60.5 | 67.4 | 74.5 | 59.1 | 65.9 | 74.1 | 37.4 | 49.7 | 65.4 | 33.1 | 43.9 |
| Liao's cross-event | N/A | | | 68.7 | 68.9 | 68.8 | 50.9 | 49.7 | 50.3 | 45.1 | 44.1 | 44.6 |
| Hong's cross-entity | N/A | | | 72.9 | 64.3 | 68.3 | 53.4 | 52.9 | 53.1 | 51.6 | 45.5 | 48.3 |
| Li's structure | 76.9 | 65.0 | 70.4 | 73.7 | 62.3 | 67.5 | 69.8 | 47.9 | 56.8 | 64.7 | 44.4 | 52.7 |
| **DMCNN model** | 80.4 | 67.7 | **73.5** | 75.6 | 63.6 | **69.1** | 68.8 | 51.9 | **59.1** | 62.2 | 46.9 | **53.5** |

| Stage | 1/1 (%) | 1/N (%) |
|---|---|---|
| Trigger | 72.7 | **27.3** |
| Argument | 76.8 | **23.2** |

| Stage | Method | 1/1 | 1/N | all |
|---|---|---|---|---|
| | | $F_1$ | $F_1$ | $F_1$ |
| Trigger | Embedding+T | 68.1 | 25.5 | 59.8 |
| | CNN | 72.5 | 43.1 | 66.3 |
| | DMCNN | **74.3** | **50.9** | **69.1** |
| Argument | Embedding+T | 37.4 | 15.5 | 32.6 |
| | CNN | 51.6 | 36.6 | 48.9 |
| | DMCNN | **54.6** | **48.7** | **53.5** |

# 基于神经网络的方法：基于有监督关注机制的多层感知机模型

Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms（Liu et al., ACL 2017）

- 动机
  - 事件的主要元素：

    触发词和事件角色

  - 事件角色为事件识别任务提供重要线索

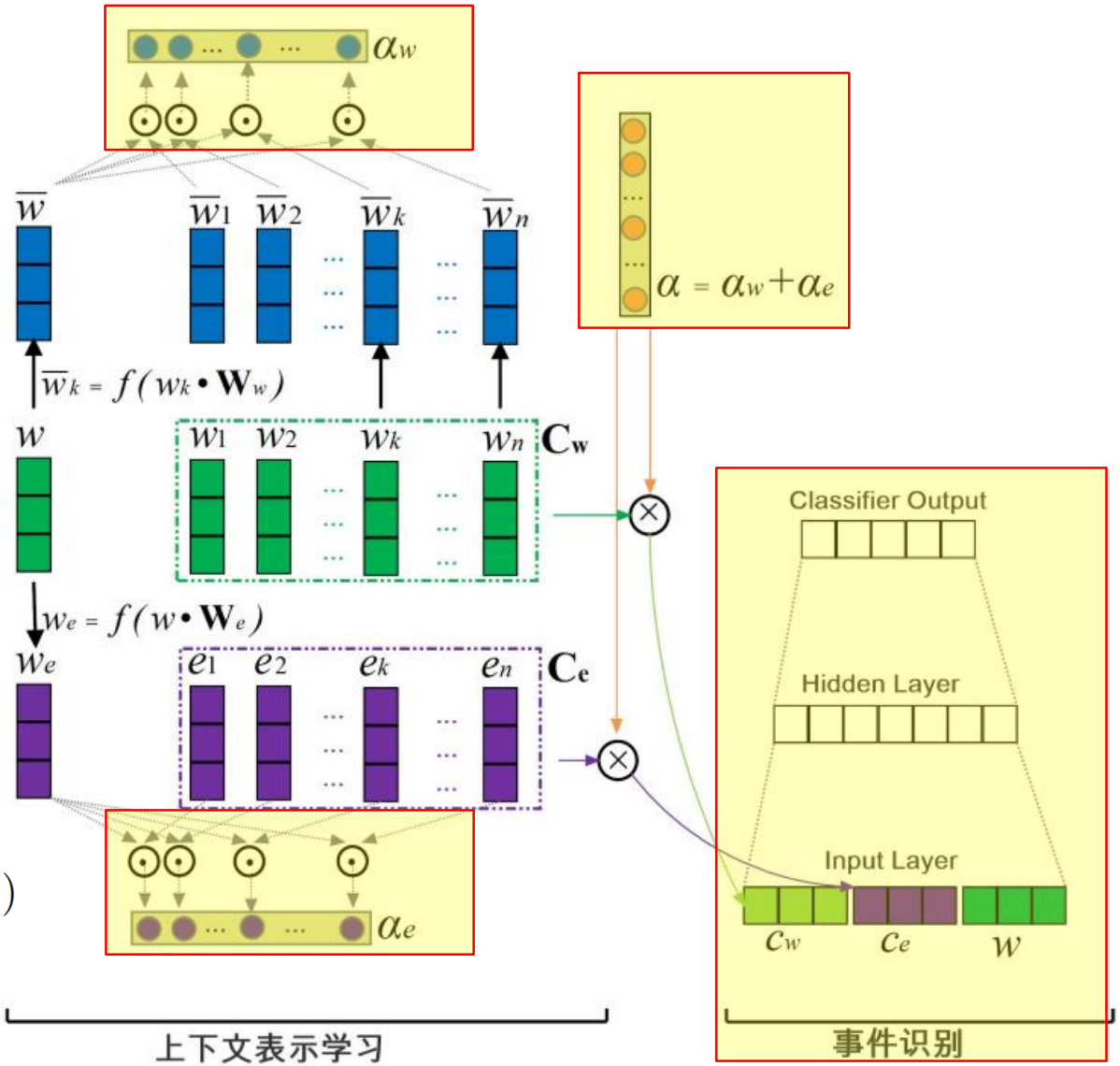
*前任门徒*

*Mohanmad* fired *Anwar*, his *former protege*, in *1998*.

*袭击事件* or *结束职位事件*?

# 基于神经网络的方法：
# 基于有监督关注机制的多层感知机模型

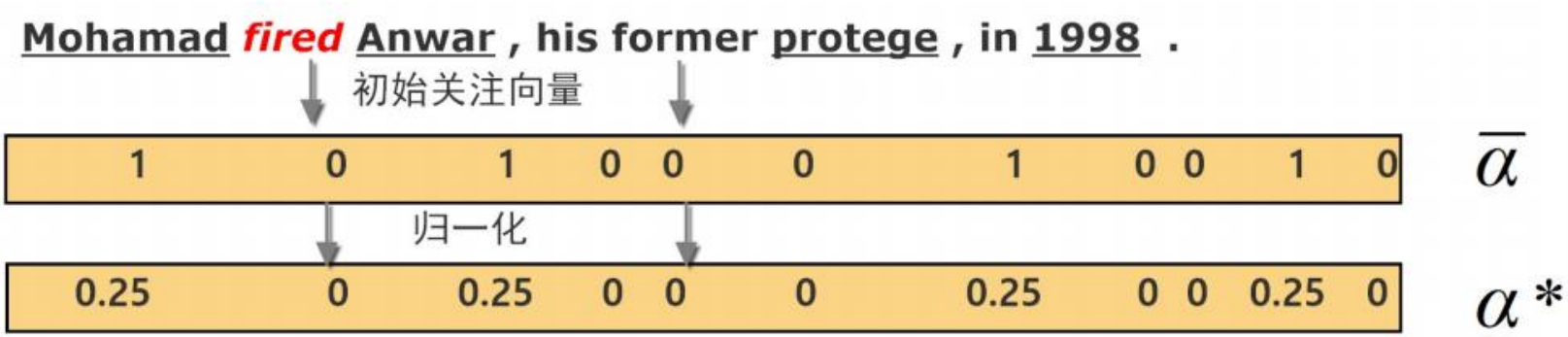
## 模型




$$J(\theta) = -\sum_{i=1}^{T} \log p(y^{(i)} | \mathbf{x}^{(\mathbf{i})}, \theta)$$

# 基于神经网络的方法：
# 基于有监督关注机制的多层感知机模型

- 构建标准关注向量
  - 策略1：只关注角色词




  - 策略2：同时关注角色词及其周围的词，距离角色词越近的词受到的关注度越高

# 基于神经网络的方法：
# 基于有监督关注机制的多层感知机模型

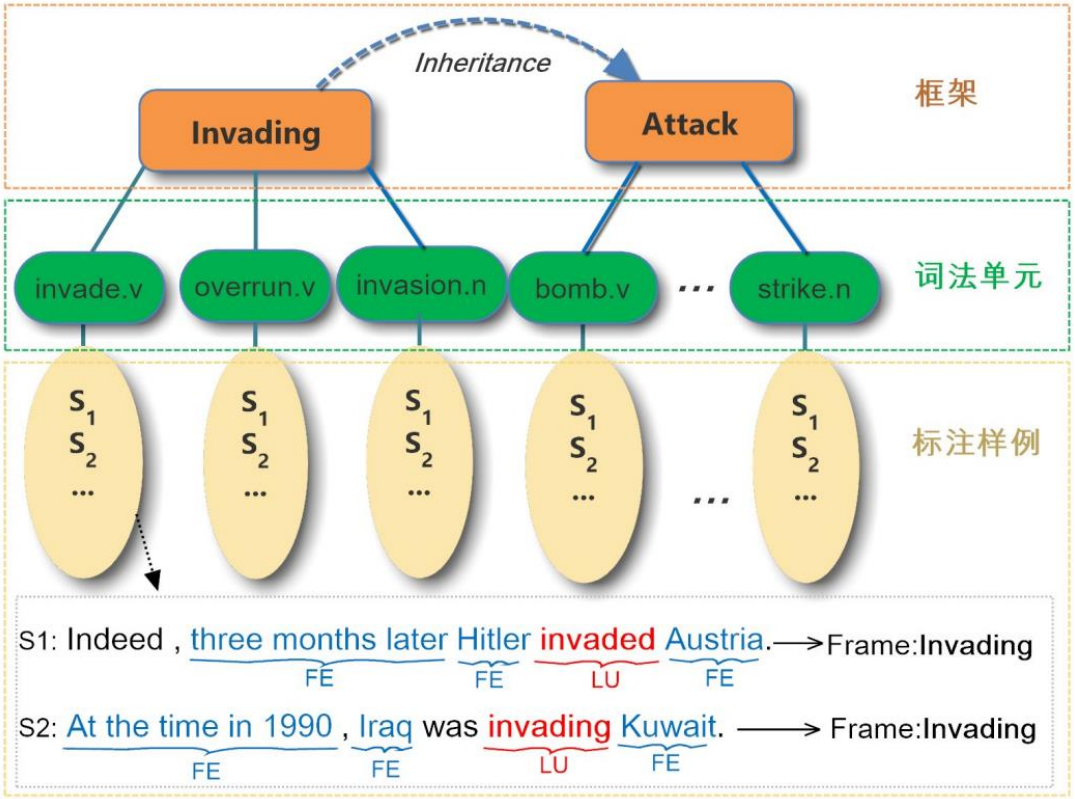
## 实验

| 方法 | 正确率（%） | 召回率（%） | $F_1$ 值（%） |
|---|---|---|---|
| Li's joint model (2013) | 73.7 | 62.3 | 67.5 |
| Liu's PSL (2016) | 75.3 | 64.4 | 69.4 |
| Liu's FN-Based (2016) | 77.6 | 65.2 | 70.7 |
| Ngyuen's joint (2016) | 66.0 | **73.0** | 69.3 |
| Nguyen's Skip-CNN (2016) | N/A | | 71.3 |
| ANN | 73.2 | 57.9 | 64.6 |
| ANN-S1† | **81.4** | 62.4 | 70.8 |
| ANN-S2† | 78.0 | 66.3 | **71.7** |

# 融合FrameNet的事件识别方法（Liu et al., ACL 2016）

- FrameNet
  - 语言学家定义及标注的语义框架资源
  - 层级的组织结构

- FrameNet规模
  - 1000+ 框架
  - 10000+ 词法单元
  - 150000+ 标注例句



# 融合FrameNet的事件识别方法（Liu et al., ACL 2016）

## ■结构相似性

- 框架：一个词法单元和若干框架元素
- 事件： 一个触发词和若干事件角色

## ■含义相似性

| 框架名称 | FrameNet 中的例句 | 事件类型 |
|---|---|---|
| Attack | *Aeroplane **bombed** London.* | Attack |
| Rape | *Girl, 16, **raped** by gang.* | Attack |
| Fining | *The court **fined** her $40.* | Fine |
| Execution | *He was **executed** yesterday.* | Execute |

# 融合FrameNet的事件识别方法
# （Liu et al., ACL 2016）

## ■系统框架图

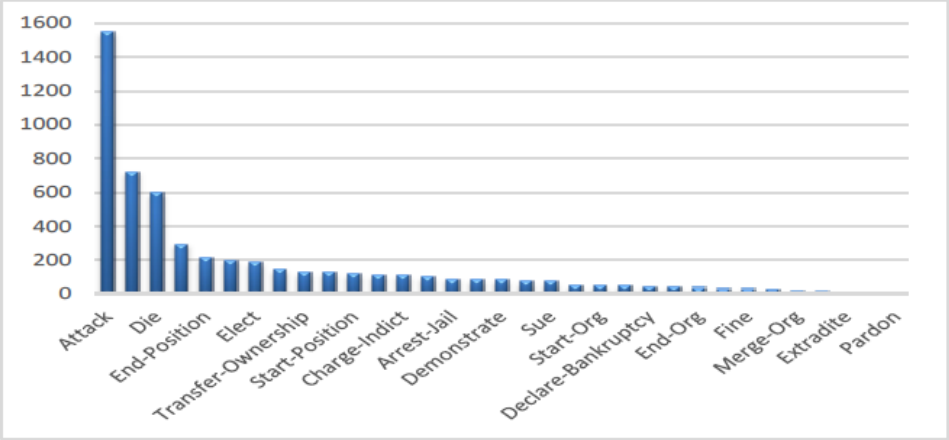
# 融合FrameNet的事件识别方法
# （Liu et al., ACL 2016）

- 实验

| 方法 | 正确率 (%) | 召回率 (%) | $F_1$ 值 (%) |
|---|---|---|---|
| Nguyen's CNN(2015) | 71.8 | **66.4** | 69.0 |
| Chen's DMCNN(2015) | 75.6 | 63.6 | 69.1 |
| Liu's Approach(2016) | 75.3 | 64.4 | 69.4 |
| ANN (Ours) | **79.5** | 60.7 | 68.8 |
| ANN-FN (Ours) | 77.6 | 65.2 | **70.7** |

# 基于世界知识和语言学知识的事件语料大规模自动生成方法（Chen et al.，ACL2017）

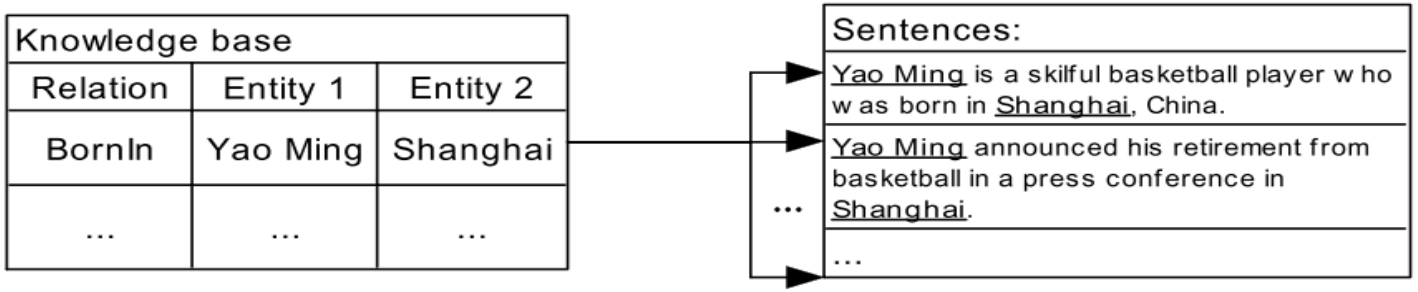
- 现有方法的问题
  - 有监督的事件抽取方法：
    过分依赖人工标注的数据
    耗时、费力、成本高昂，规模小，覆盖度低




  - 无监督的事件抽取方法：
    抽取的结果没有规范的语义标签
    很难直接映射到现有知识库

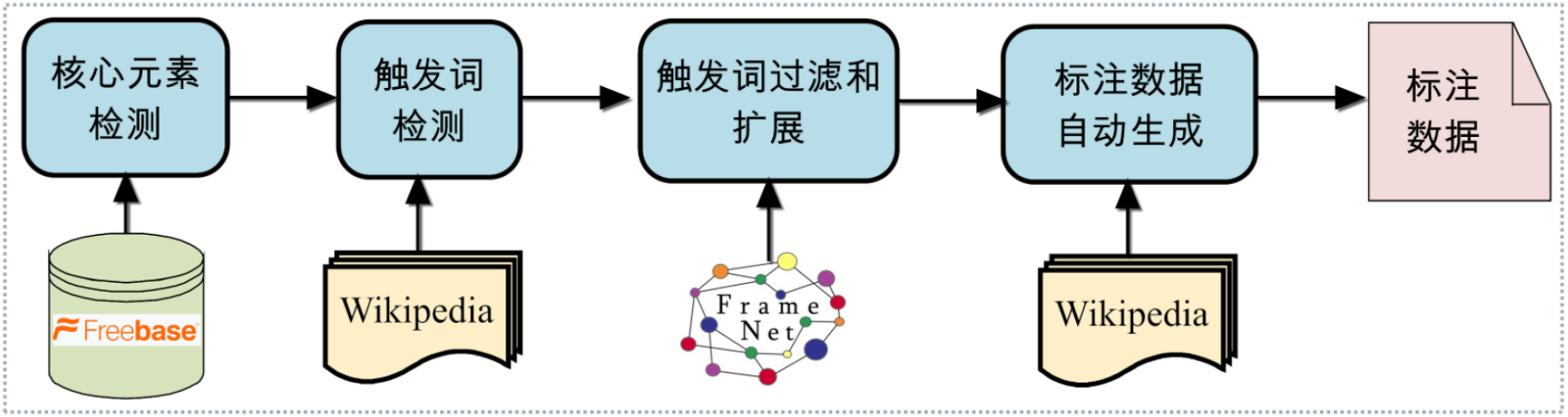
# 基于世界知识和语言学知识的事件语料大规模自动生成方法（Chen et al.，ACL2017）

- 动机
  - 自动生成标注语料
    开放域环境
    大规模自动抽取
  - 远距离监督的方法在关系抽取中取得成功



# 基于世界知识和语言学知识的事件语料大规模自动生成方法（Chen et al.，ACL2017）

- 方法
  - 利用世界知识和语言学知识
  - 自动生成大规模事件语料



# 基于世界知识和语言学知识的事件语料大规模自动生成方法（Chen et al.，ACL2017）

## 实验

### 自动生成的数据

| Event Type | Freebase Size | Sentences (KA) | Sentences (KA+T) | Examples of argument roles sorted by KR | Examples of triggers |
|---|---|---|---|---|---|
| people.marriage | 152,276 | 56,837 | 26,349 | spouse, spouse, from, to, location | marriage, marry, wed, wedding, couple,..., wife |
| music.group_membership | 239,813 | 90,617 | 20,742 | group, member, start, role, end | musician, singer, sing, sang, sung, concert,..., play |
| education.education | 530,538 | 26,966 | 11,849 | student, institution, degree,..., minor | educate, education, graduate, learn, study,..., student |
| organization.leadership | 43,610 | 5,429 | 3,416 | organization, person, title,..., to | CEO, charge, administer, govern, rule, boss,..., chair |
| olympics.olympic_medal_honor | 20,790 | 4,056 | 2,605 | medalist, olympics, event,..., country | win, winner, tie, victor, gold, silver,..., bronze |
| ... | ... | ... | ... | ... | ... |
| sum of 21 selected events | 3,870,492 | 421,602 | 72,611 | argument1, argument2 ,..., argumentN | trigger1, trigger2, trigger3, ... , triggerN |

-当仅利用两个核心元素回标时，生成421,602个标注数据，但是这个数据中没有标注触发词信息

-当同时利用核心元素和事件触发词回标时，生成72,611个标注数据

-与ACE人工标注的将近6,000个的标注数据相比，我们提出的方法能自动生成大规模训练数据

# 基于世界知识和语言学知识的事件语料大规模自动生成方法（Chen et al.，ACL2017）

## 实验

### 标注数据的人工评价

##001 He is the uncle of [**Amal Clooney**], [**wife**] of the actor [**George Clooney**].
**Trigger:** wife **Event Type:** Marriage **MannalAnotate[Y/N]:**
**Argument:** Amal Clooney **Role:**Spouse **MannalAnotate[Y/N]**:
**Argument:** George Clooney **Role:**Spouse **MannalAnotate[Y/N]**:

##002 She was [**married**] to the cinematographer [**Theo Nischwitz**] and was sometimes credited as [**Gertrud Hinz-Nischwitz**].
**Trigger:** married **Event Type:** Marriage **MannalAnotate[Y/N]**:
**Argument:** Theo Nischwitz **Role:**Spouse **MannalAnotate[Y/N]**:
**Argument:** Gertrud Hinz-Nischwitz **Role:**Spouse **MannalAnotate[Y/N]**:

| 阶段 | 平均正确率 (%) |
|---|---|
| 触发词标注 | 88.9 |
| 元素标注 | 85.4 |

### 标注数据的自动评价

| Methods | Trigger Identification(%) | | | Trigger Identification + Classification(%) | | | Argument Identification(%) | | | Argument Role(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Li's structure trained with ACE | 76.9 | 65.0 | 70.4 | 73.7 | 62.3 | 67.5 | 69.8 | 47.9 | 56.8 | 64.7 | 44.4 | 52.7 |
| Chen's DMCNN trained with ACE | 80.4 | 67.7 | 73.5 | 75.6 | 63.6 | 69.1 | 68.8 | 51.9 | 59.1 | 62.2 | 46.9 | 53.5 |
| Nguyen's JRNN trained with ACE | 68.5 | 75.7 | 71.9 | 66.0 | 73.0 | 69.3 | 61.4 | 64.2 | 62.8 | 54.2 | 56.7 | 55.4 |
| DMCNN trained with ED Only | 77.6 | 67.7 | 72.3 | 72.9 | 63.7 | 68.0 | 64.9 | 51.7 | 57.6 | 58.7 | 46.7 | 52.0 |
| DMCNN trained with ACE+ED | 79.7 | 69.6 | **74.3** | 75.7 | 66.0 | **70.5** | 71.4 | 56.9 | **63.3** | 62.8 | 50.1 | **55.7** |

# 事件关系：无监督学习叙事事件链

Unsupervised Learing of Narrative Event Chains（**Nathael Chambers ACL2008**）

■ 背景

尽管特定场景下发生的一系列事件有多个参与者，但是有一个是主导参与者。

叙事事件链(Narrative Event Chain)是指围绕主导参与者发生的部分有序的一系列事件。

✓ e1~e4构成了一个描述了一个围绕参与人X的叙事事件链：

e1:X进餐馆->e2:X点餐->e3:为X上菜->e4:X吃饭

# 事件关系：无监督学习叙事事件链

Unsupervised Learing of Narrative Event Chains（**Nathael Chambers ACL2008**）

■事件表示

利用主导参与者和事件指示词之间的句法角色表示事件，例如：

*John pulled Tom aside and told him he’d start. =>*

✓Tom是主导参与者

✓e1: (pulled, object) -> e2:(told, object) -> e3:(start, subject)}

✓有了事件表示，对候选事件进行聚类可得到叙事事件链。

# 事件关系：无监督学习叙事事件链

Unsupervised Learing of Narrative Event Chains（**Nathael Chambers ACL2008**）

■事件之间相关性

用互信息表示事件e(w, d)和e(v, g)之间的相关性：

$$pmi(e(w,d),e(v,g)) = log\frac{P(e(w,d),e(v,g))}{P(e(w,d))P(e(v,g))} \quad (1)$$

where $e(w,d)$ is the verb/dependency pair $w$ and $d$ (e.g. e(*push*,*subject*)). The numerator is defined by:

$$P(e(w,d),e(v,g)) = \frac{C(e(w,d),e(v,g))}{\sum_{x,y}\sum_{d,f}C(e(x,d),e(y,f))} \quad (2)$$

其中，C(a, b)指a和b同时出现的频次。

# 事件关系：无监督学习叙事事件链

Unsupervised Learing of Narrative Event Chains（**Nathael Chambers ACL2008**）

■应用

给定叙事序列的前n个事件，预测接下来要发生的事件。

■方法

将所有事件当做候选，选择候选中和给定的n个事件互信息之和最大的事件作为答案。

$$\max_{j:0<j<m} \sum_{i=0}^{n} pmi(e_i, f_j) \qquad (3)$$

**Known events**:
(pleaded subj), (admits subj), (convicted obj)

**Likely Events**:

| | | | |
|---|---|---|---|
| sentenced obj | 0.89 | indicted obj | 0.74 |
| paroled obj | 0.76 | fined obj | 0.73 |
| fired obj | 0.75 | denied subj | 0.73 |

Figure 1: Three narrative events and the six most likely events to include in the same chain.

# 事件关系：无监督学习叙事事件链

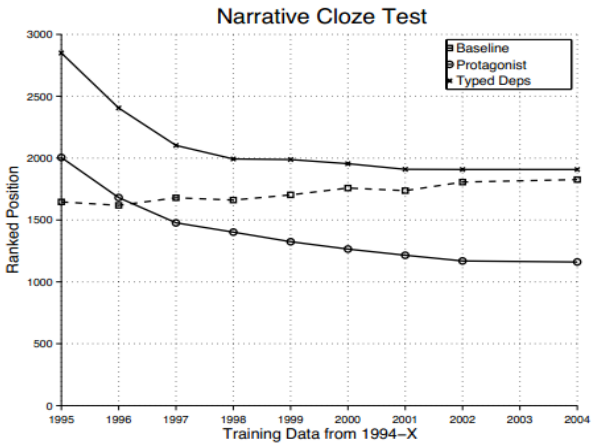
Unsupervised Learing of Narrative Event Chains（**Nathael Chambers ACL2008**）

■ 实验

BaseLine:只用动词表示事件，
同样利用互信息表示两个动词间的相关性

Protagonist：只用动词表示事件，但选择答案时加上共同参与者的限制。

Typed Deps:本文所提方法，用动词、及其与主导参与者之间的句法关系表示事件，选择答案时加上共同参与者的限制。




■ 评价标准：每个缺失的事件在预测事件列表中排序后的位置，位置越小，表示系统越好。

# 事件框架学习：基于模板的事件抽取系统

**Template-Based Information Extraction without Templates** （ ***Nathael Chambers*** ACL2011）

■任务

无监督地学习事件框架。

事件框架描述了一类事件的结构，例如下图描述了爆炸事件的框架：

**Bombing Template**

{detonate, blow up, plant, explode, defuse, destroy}

| | |
|---|---|
| *Perpetrator*: | Person who detonates, plants, blows up |
| *Instrument*: | Object that is planted, detonated, defused |
| *Target*: | Object that is destroyed, is blown up |

# 事件框架学习：基于模板的事件抽取系统

**Template-Based Information Extraction without Templates**（*Nathael Chambers* ACL2011）

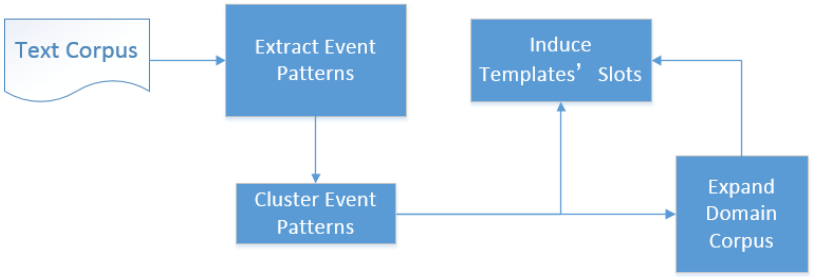
■ 方法

Step1：从文本语料中抽取事件元素（和上一工作类似，用参与者，动词以及参与者与动词之间的句法关系表示事件元素）。

Step2：对抽取到的事件元素进行初步聚类（相似度算法和上一工作类似，利用互信息衡量）。

Step3：对每个聚类结果，在搜索引擎中搜索类内的事件元素，将返回的文档作为扩充语料。

Step4：利用扩充语料对每个初步聚类结果内的元素进行再聚类，再聚类的结果作为事件框架（每个聚类结果是一个事件，结果内的元素是该事件的元素）。



# 事件框架学习：基于模板的事件抽取系统

**Template-Based Information Extraction without Templates** （ *Nathael Chambers* ACL2011）

■结果：两个聚类结果

✓蓝色的词表示事件类别，绿色的词表示事件的参与者。其中，第一个事件框架Kidnap（绑架）是MUC定义的事件；

✓第二个事件框架Weapons Smuggling是新的事件类别。

✓聚类结果中，一个聚类结果是一个事件类别，并没有事件的名称，这里的事件名称是人工标记的。事件角色同样如此。

**Kidnap Template** (MUC-4)

**Perpetrator** *Person/Org* who releases, abducts, kidnaps, ambushes, holds, forces, captures, is imprisoned, frees

**Target** *Person/Org* who is kidnapped, is released, is freed, escapes, disappears, travels, is harmed, is threatened

**Police** *Person/Org* who rules out, negotiates, condemns, is pressured, finds, arrests, combs

**Weapons Smuggling Template** (NEW)

**Perpetrator** *Person/Org* who smuggles, is seized from, is captured, is detained

**Police** *Person/Org* who raids, seizes, captures, confiscates, detains, investigates

**Instrument** A *physical object* that is smuggled, is seized, is confiscated, is transported

# 事件框架学习：大规模语料中生成事件框架

Generating Coherent Event Schemas at Scale（**Oren Etzioni EMNLP2013**）

■ 任务

和上一工作类似，无监督地在大规模语料中生成事件框架

■ 方法

Step1：利用关系抽取的方法（Reverb等）获得大量三元组<arg1, rel, arg2>

Step2：根据这些三元组在文档中的共现信息建立一个无向图，边权反应两个元组共现的可能性

Step3：选取图中边权最高的一些点作为种子，每个种子认为是一个事件的中心

Step4：针对每个种子，利用Personalized PageRank算法求解和其最相关的点集，每个点集表示一个事件

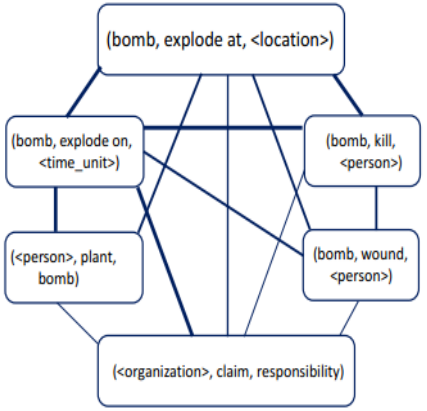



Figure 2: Part of a Rel-graph showing tuples strongly associated with *(bomb, explode at, <location>)*. Undirected edges are weighted by symmetric conditional probability with line thickness indicating weight.

# 概述

- 事件抽取的任务定义
- 事件抽取评测及语料资源
- 事件抽取的典型方法
- 事件抽取的相关应用
  - 生成故事大纲
  - 预测新事件

# 生成故事大纲

- 一篇文章中多个事件

  Task Joint Inference for Event Timeline Construction ( Quang Xuan Do et al., ) ACL2012

- 一个事件在多篇文章

  Building Event Threads out of Multiple News Articles ( Xavier Tannier et al., ) EMNLP2013

- 社交媒体中的故事大纲生成

  Generating Event Storylines from Microblogs ( li et al., ) CIKM2012

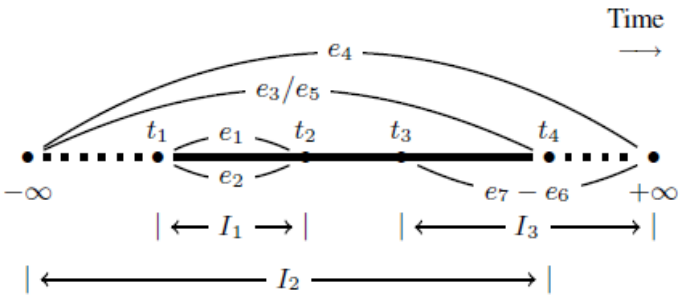
# 生成故事大纲

- 一篇文章中多个事件（Quang Xuan Do EMNLP2012）
  - 任务: 输入一篇文章，将文章中的事件按时序排序
  - 基本概念:

1.事件：同ACE中的事件

2.时序关系： before, after,

overlap and no relation




3.时间间隔： pairs of time endpoints

共有两类：Explicit intervals，Implicit intervals

The litigation covers buyers in auctions outside the United States between January 1, 1993 and February 7, 2000.

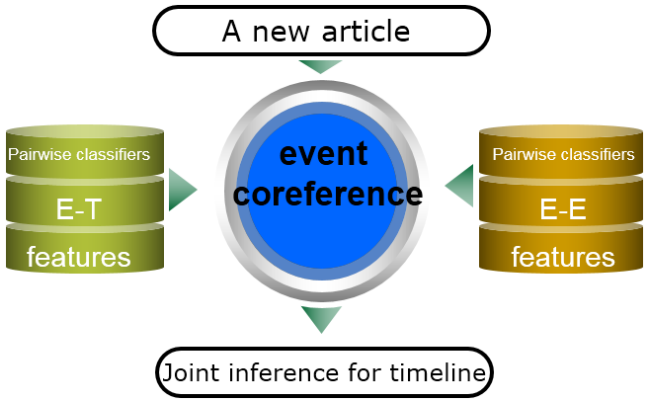
January 1, 1993->[1993-01-01 00:00:00, 1993-01-01 23:59:59]

February7, 2000 ->[2000-02-07 00:00:00, 2000-02-07 23:59:59]

January 1, 1993 and February 7, 2000 -> [1993-01-01 00:00:00, 2000-02-07 23:59:59].

# 生成故事大纲

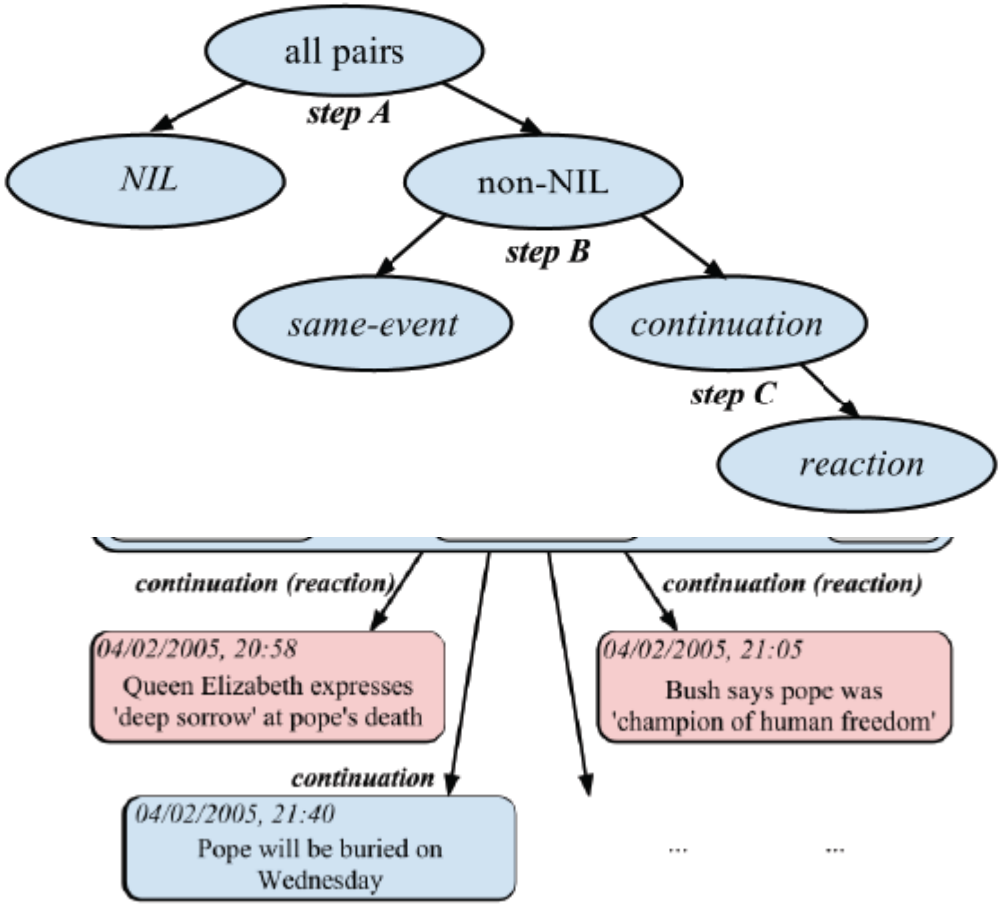
- 一篇文章中多个事件（Quang Xuan Do EMNLP2012）
  - 方法
    - 识别时间构建Timeline
    - 识别事件
    - 建立事件和时间的关系
    - 同一个时间段内建立事件间的时序关系
  - 实验
    - 选自ACE2005中的20篇newswire
    - 人工标注E-T 和E-E




| | Model | $C_{E-T}$ | | | $C_{E-E}$ | | | Overall | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | *Prec.* | *Rec.* | $F_1$ | *Prec.* | *Rec.* | $F_1$ | *Prec.* | *Rec.* | $F_1$ |
| 1 | **Baseline** | 33.29 | 33.29 | 33.29 | 20.86 | 32.81 | 25.03 | 27.06 | 33.05 | 29.16 |
| 2 | **No Event Coref.** | | | | | | | | | |
| | Local classifiers | 62.70 | 34.50 | 43.29 | 40.46 | 42.42 | 40.96 | 51.58 | 38.46 | 42.13 |
| | Global inference | 47.88 | 47.88 | 47.88 | 41.42 | 48.04 | 44.14 | 44.65 | 47.96 | 46.01 |
| 3 | **With Gold Event Coref.** | | | | | | | | | |
| | Local classifiers | 50.88 | 50.88 | 50.88 | 43.86 | 52.65 | 47.46 | 47.37 | 51.77 | 49.17 |
| | Global inference | 50.88 | 50.88 | 50.88 | 48.04 | 62.45 | 54.05 | 49.46 | 56.67 | 52.47 |
| 4 | **With Learned Event Coref.** | | | | | | | | | |
| | Local classifiers | 46.37 | 46.37 | 46.37 | 40.83 | 45.28 | 42.60 | 43.60 | 45.83 | 44.49 |
| | Global inference | 46.37 | 46.37 | 46.37 | 42.09 | 52.50 | 46.47 | 44.23 | 49.44 | 46.42 |

# 生成故事大纲

- 一个事件在多篇文档中（Xavier Tannier EMNLP 2013）
  - 任务: 从多篇文章中抽取同一事件的不同阶段
  - 方法：层级分类器
  - A:区分事件对之间有关系还是没有关系
  - B:区分相同事件或者顺承事件
  - C:从顺承事件中寻找反应事件



# 生成故事大纲

## 一个事件在多篇文档中（Xavier Tannier EMNLP 2013）

### Dataset

选自French news agency（AFP），定义了一系列复杂的规则来指导标注

| | Pair number | Learning | Evaluation |
|---|---|---|---|
| *Same event* | 762 | 458 | 304 |
| *Continuation* | 1134 | 748 | 386 |
| *Reaction* | 182 | 123 | 59 |
| *Nil* | 918 | 614 | 304 |
| TOTAL | **2996** | 1943 | 1053 |

Characteristics of the corpus.

### Result

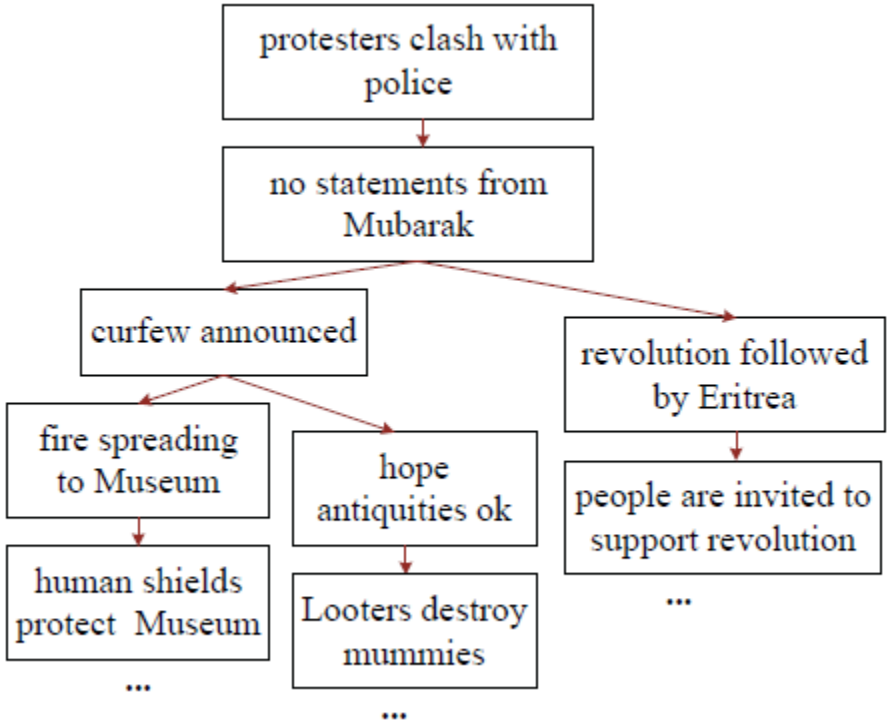
Results obtained by the baseline system.

| Relation | Precision | Recall | F1 |
|---|---|---|---|
| *NIL* | 0.754 | 0.821 | 0.786 |
| *same-event* | 0.832 | 0.812 | 0.822 |
| *continuation* | 0.736 | 0.696 | 0.715 |
| ↳ *reaction* | 0.273 | 0.077 | 0.120 |

| Step | Relation | Precision | Recall | F1 |
|---|---|---|---|---|
| **A.** *NIL* vs *non-NIL classifier* | | | | |
| A.1 | NIL | 0.764 | 0.815 | 0.788 |
| | non-NIL | 0.921 | 0.896 | 0.910 |
| A.2 *** | NIL | 0.907 | 0.811 | 0.857 |
| | non-NIL | 0.925 | 0.966 | 0.945 |
| **B.** *Same-event* vs *continuation classifier* | | | | |
| B.1 | NIL | 0.907 | 0.811 | 0.857 |
| | same-event | 0.870 | 0.553 | 0.676 |
| | continuation | 0.664 | 0.867 | 0.752 |
| B.2 *** | NIL | 0.947 | 0.831 | 0.885 |
| | same-event | 0.894 | 0.724 | 0.800 |
| | continuation | 0.744 | 0.911 | 0.819 |
| B.3 ** | NIL | 0.884 | 0.831 | 0.857 |
| | same-event | 0.943 | 0.819 | 0.877 |
| | continuation | 0.797 | 0.906 | 0.848 |
| B.4 * | NIL | 0.890 | 0.831 | 0.860 |
| | same-event | 0.943 | 0.819 | 0.877 |
| | continuation | 0.798 | 0.911 | 0.851 |
| **C.** *Reaction* vs *continuation* | | | | |
| C.1 | NIL | **0.890** | **0.831** | **0.860** |
| C.2 | same-event | **0.943** | **0.819** | **0.877** |
| | continuation | **0.798** | **0.911** | **0.851** |
| | ↳ reaction | **0.778** | **0.359** | **0.491** |

# 生成故事大纲

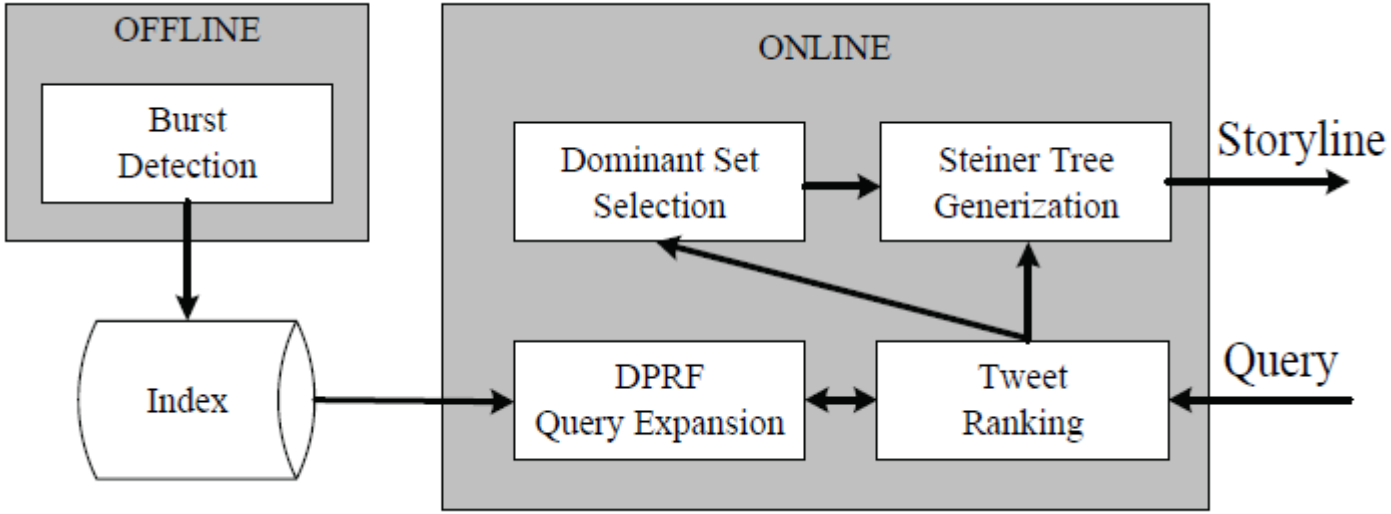
- 社交媒体中的故事大纲生成
  - 任务: 给定一个关键词的查询，根据tweet消息生成故事大纲




A sample storyline for event query "Egypt Revolution"

# 生成故事大纲

- 方法
  - 检索得到相关的Tweet
  - 选取每个阶段中代表性的Tweet
  - 由各个阶段中代表性的Tweet联合起来得到最终的故事大纲



# 生成故事大纲

## Experiment

- **Dataset**
  - Tweets2011 corpus for TREC 2011 microblog track

- Result

| Type | Model | P@30 | P@100 | MAP | R-PREC |
|---|---|---|---|---|---|
| Baseline | Lu | | | | 0.2257 |
| | P | | | | 0.2245 |
| | KI | | | | 0.2614 |
| | R | | | | 0.2297 |
| PRF | Ro | | | | 0.2182 |
| | E | | | | 0.2961 |
| | F | | | | 0.2781 |
| DPRF | M | | | | 0.3055(++) |
| | I | | | | **0.3206**(++) |
| | S | | | | 0.3038(++) |

| Methods | ROUGE2 | ROUGE-SU |
|---|---|---|
| Random | 0.0425 | 0.0903 |
| MostRelevant | 0.0526 | 0.1075 |
| LSA | 0.0403 | 0.0857 |
| K-means | 0.0489 | 0.1002 |
| NMF | 0.0534 | 0.1043 |
| SNMF | 0.0593 | 0.1203 |
| NCut | 0.0635 | 0.1156 |
| Qs-MRC | 0.0647 | 0.1255 |
| MSSF | 0.0639 | 0.1324 |
| DS Only | 0.0731 | 0.1280 |
| DS+ST | 0.0895(++) | 0.1363(+) |

Storyline 的结果

# 预测新事件

- 预测自然事件

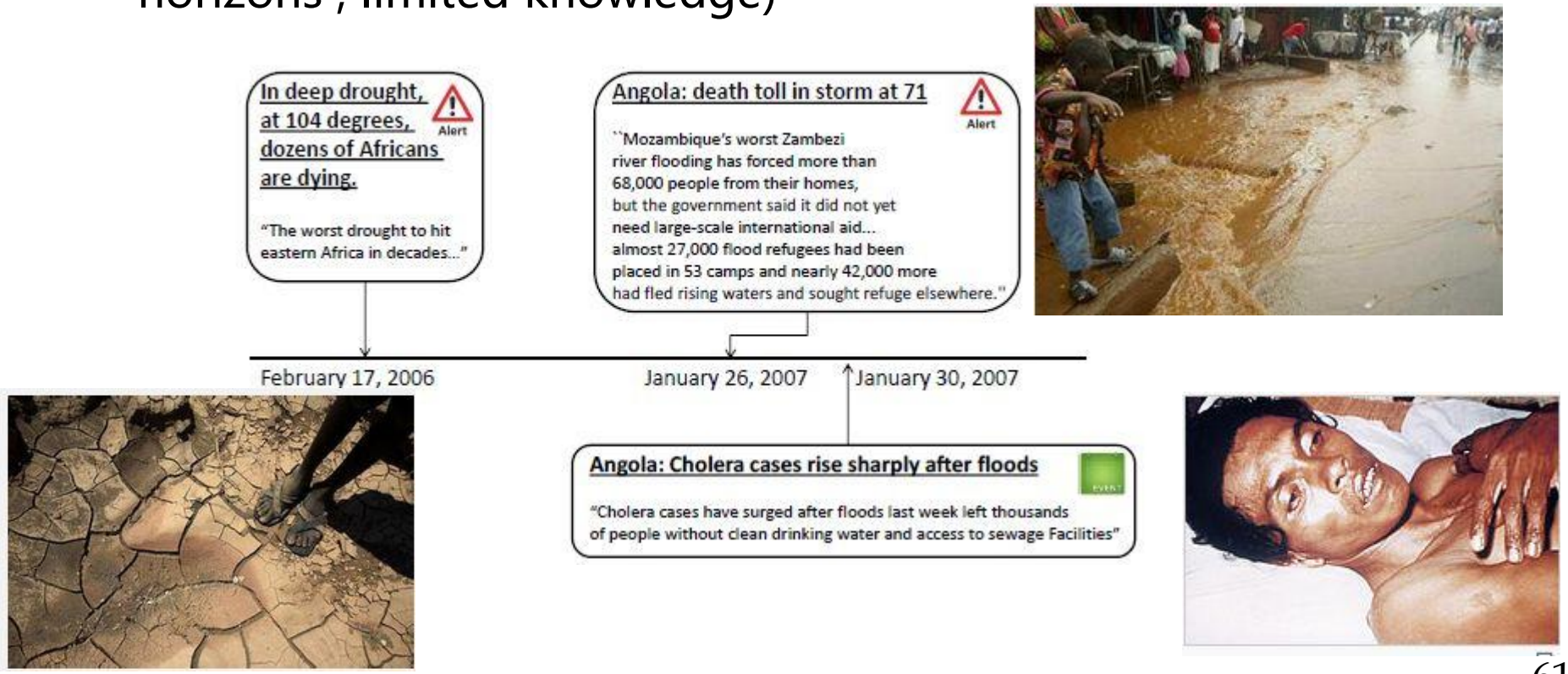
  Mining the Web to Predict Future Events（Kira Radinsky et al.,）WSDM 2013

- 预测股票事件

  Using Structured Events to Predict Stock Price Movement : An Empirical Investigation（ Ding et al.,）EMNLP2014

# 预测新事件

- 预测自然事件
  - Mark Twain famously said that “the past does not repeat itself, but it rhymes.”
  - Traditional method(heuristic assessments, shorter time horizons , limited knowledge)



# 预测新事件

## 预测自然事件

- 典例展示

| Date | Title |
|---|---|
| Jan 16, 1992 | Jury in Shooting by Officer Hears Conflicting Accounts |
| Feb 11, 1992 | Closing Arguments Conflict on Killing by Teaneck Officer |
| **Feb 12, 1992** | **[Past Event] Officer Acquitted in Teaneck Killing** |
| Feb 13, 1992 | Acquitted Officer Expresses Only Relief, Not Joy |
| **Feb 16, 1992** | **[Past Riot] 250 March in Rain to Protest Teaneck Verdict** |
| Feb 24, 2000 | Diallo Jurors Begin Deliberating In Murder Trial of Four Officers |
| **Feb 26, 2000** | **[Riot Alert] 4 officers in Diallo shooting are acquitted of all charges** |
| Feb 26, 2000 | Rage Boils Over, and Some Shout 'Murderers' at Police |
| Feb 26, 2000 | Civil Rights Prosecution Is Considered |
| **Feb 27, 2000** | **[Riot Event] Marchers Protest Diallo Verdict...** |
| Feb 27, 2000 | 2 jurors defend Diallo acquittal |

- Interesting results

| Cause | Effect | Probability |
|---|---|---|
| Drought | Flood | 18% |
| Flood | cholera | 1% |
| Flood in Rwanda | cholera in Rwanda | 67% |
| Flood in Lima | cholera in Lima | 33% |
| Flood in Country with water coverage > 5% | cholera in Country | 14% |
| Flood in Country with water coverage > 5%, population density > 100 | cholera in Country | 16% |

# 预测新事件

## 预测股票事件

- 新闻事件会影响股民的决定，股民的交易会影响股票价格
- 所以新闻事件会影响股票价格

## 事件定义

一个事件被定义为 E = (O1, P, O2, T),

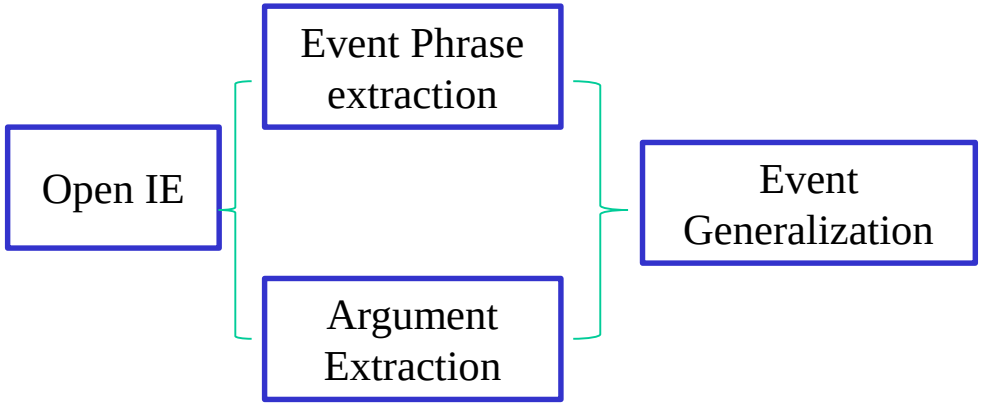
The event “Sep 3, 2013- Microsoft agrees to buy Nokia’s mobile phone business for $7.2 billion. ”

(Actor =Microsoft, Action = buy, Object = Nokia’s mobile phone business, Time = Sep 3, 2013).

# 预测新事件

## 方法

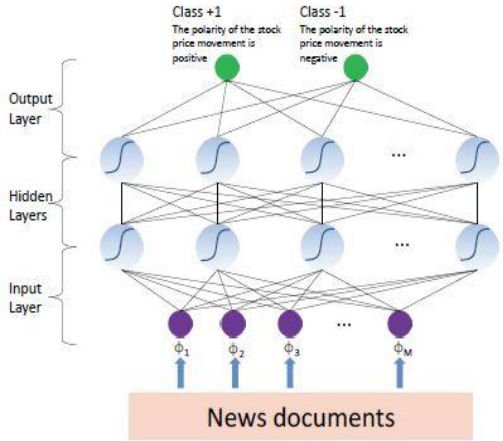
- 从海量新闻数据中获取结构化的事件信息

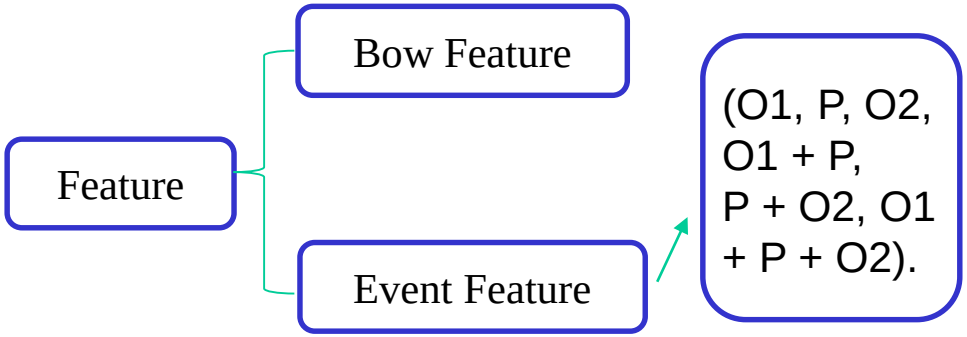



Private sector adds 114,000 jobs in July

(private sector , multiply class, 114,000 job)

2. How to predict stock price movement using structured events accurately ?



# 小结

- 由识别到抽取
  - 规范文本→有噪音、有冗余的海量网络数据
  - 独立抽取 →联合抽取、更多的考虑事件各部分之间的影响
- 限定类别→开放类别
- 难点
  - 事件的定义：没有统一的标准，基于任务的定义
  - 事件的体系结构：现有的事件体系结构库都是人工构建，规模较小
  - 事件间的关系：事件之间并非相互独立，探索并建模事件之间的关系

Q&A

# Thanks

# 第9章 知识图谱的存储与检索

王 泉

中国科学院信息工程研究所
中国科学院大学网络空间安全学院

# 提纲

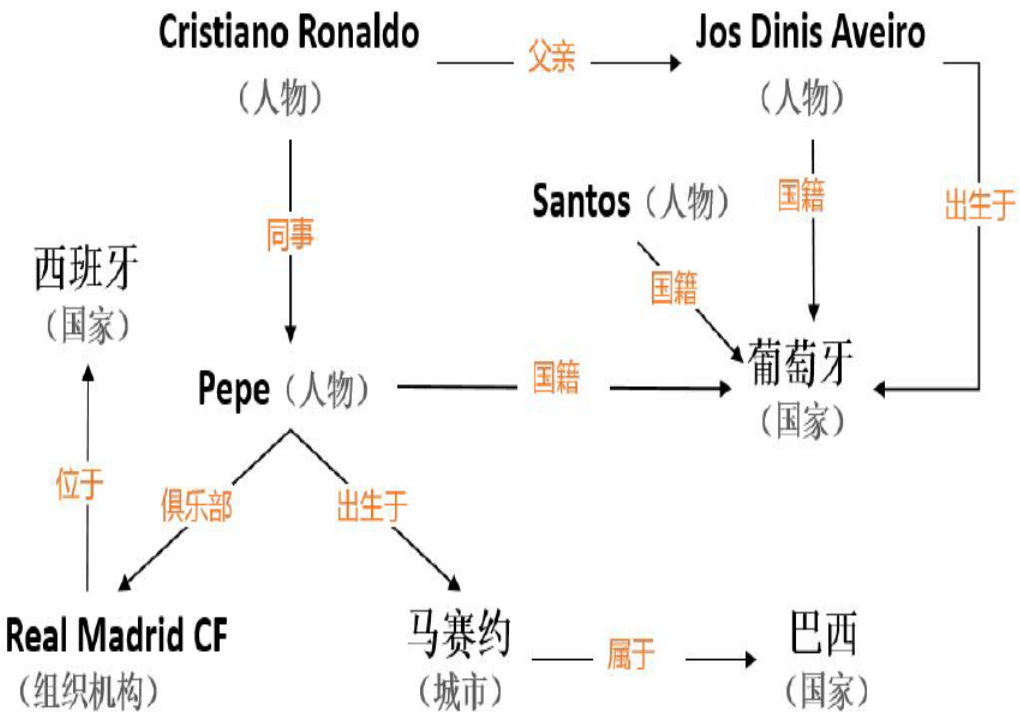
- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 知识图谱

- 知识图谱是一种有向图结构，描述了现实世界中存在的实体、事件或者概念以及它们之间的相关关系。



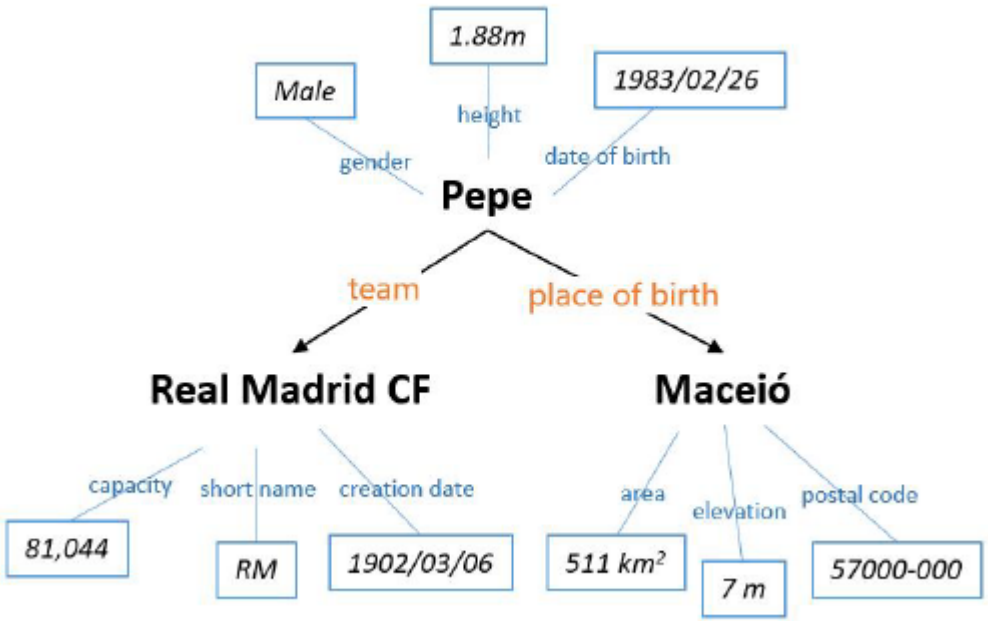

- **实体：**Cristiano Ronaldo, Jos Dinis Aveiro, Real Madrid CF、马赛约等
- **实体类型：**人物、国家、城市、组织机构等
- **属性：**人物有姓名、性别、出生日期、兴趣爱好、职业等属性；国家有国庆日、国家代码、货币、时区、等属性
- **关系：**人物和人物间的同事关系、人物和国家间的国籍关系、城市和国家间的属于关系等

# 知识图谱中的知识表示

- 知识图谱中的知识是通过RDF的结构进行表示的，其基本构成单元是事实，每个事实被表示为一个形如<subject, predicate, object>的三元组。
  - subject: 主体（也称主语），其取值通常是实体、事件或者概念中的任何一个
  - predicate：谓词（也称谓语），其取值通常是关系或者属性
  - object：客体（也称宾语），其取值既可以是实体、事件、概念，也可以是普通的值（如数字、字符串等）

# 知识图谱中的知识表示

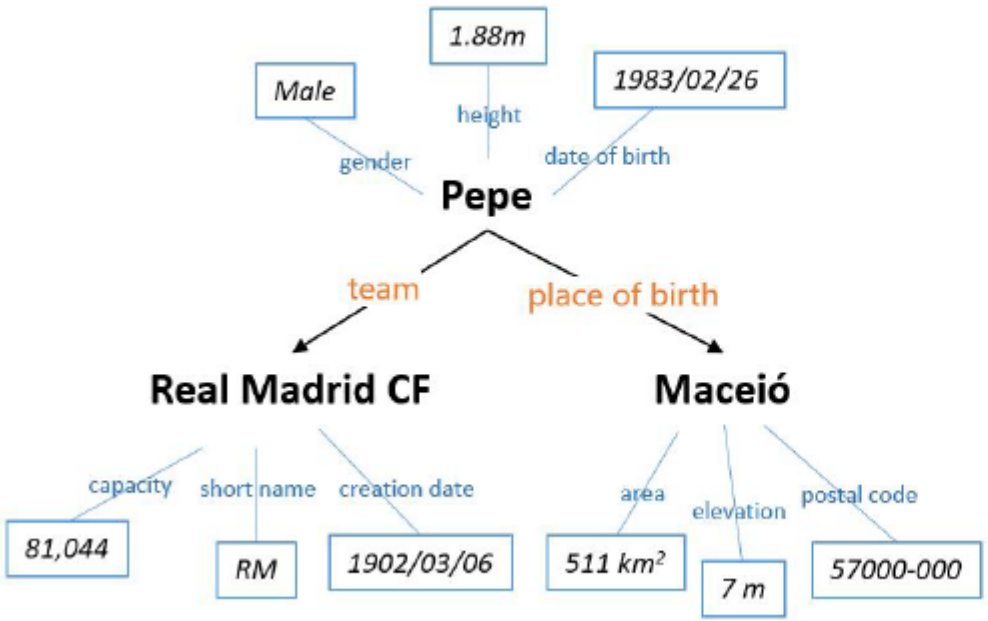
- 知识图谱中的知识是通过RDF的结构进行表示的，其基本构成单元是事实，每个事实被表示为一个形如<subject, predicate, object>的三元组。




- **实体：**Pepe, Real Madrid CF、Maceio
- **属性：**gender, height, date of birth, capacity, short name, creation date, area, elevation, postal code
- **属性值：**male, 1.88m, 1983/02/26, 81,044, RM, 1902/03/06, 511km², 7m, 57000-000
- **关系：**team, place of birth

# 知识图谱中的知识表示

- 知识图谱中的知识是通过RDF的结构进行表示的，其基本构成单元是事实，每个事实被表示为一个形如<subject, predicate, object>的三元组。




**RDF Data**

| <S, | P, | O> |
|---|---|---|
| <Pepe, | gender, | male> |
| <Pepe, | height, | 1.88m> |
| <Pepe, | date of birth, | 1983/02/06> |
| <Pepe, | team, | Ream Madrid CF> |
| <Pepe, | place of birth, | Maceió> |
| <Maceió, | area, | 511km²> |
| <Maceió, | elevation, | 7m> |
| <Maceió, | postal code, | 57000-000> |

. . .

. . .

知识图谱的目标是构建一个能够刻画现实世界的知识库，为自动问答、信息检索等应用提供支撑。因此，对知识的持久化存储并提供对目标知识的高效检索是合格的知识图谱必须具备的基本功能。

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 知识图谱的存储

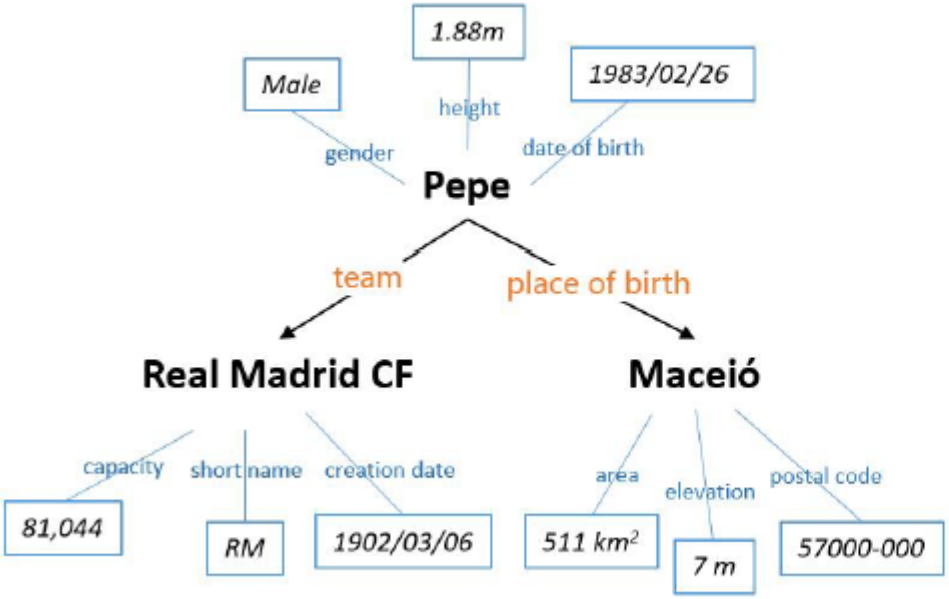
- 按照存储方式的不同，知识图谱的存储可以分为基于表结构的存储和基于图结构的存储。
  - 基于表结构的存储：利用二维的数据表对知识图谱中的数据进行存储
    - 三元组表、类型表、关系数据库
  - 基于图结构的存储：利用图的方式对知识图谱中的数据进行存储
    - 图数据库

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 三元组表

- 知识图谱中的事实是一个个的三元组，一种最简单直接的存储方式是设计一张三元组表用于存储知识图谱中所有的事实。




**三元组表**

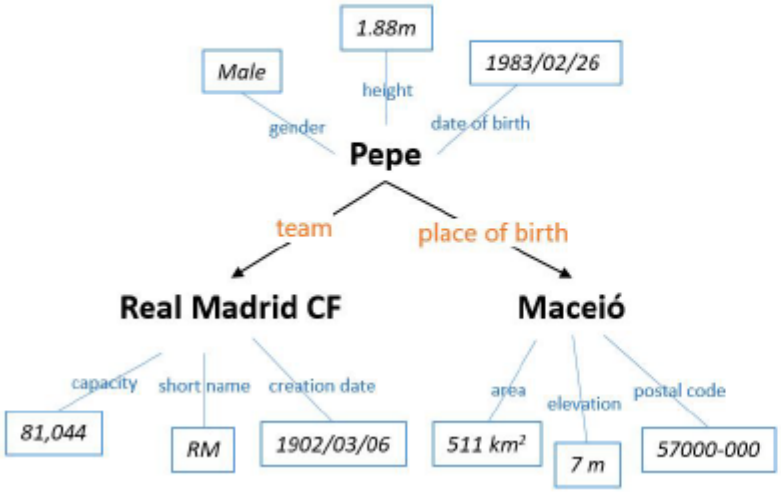
| S. | P. | O. |
|---|---|---|
| Pepe | gender | male |
| Pepe | height | 1.88m |
| Pepe | date of birth | 1983/02/06 |
| Pepe | team | Ream Madrid CF |
| Pepe | place of birth | Maceió |
| Maceió | area | 511km$^2$ |
| Maceió | elevation | 7m |
| Maceió | postal code | 57000-000 |
| ... | ... | ... |

# 三元组表特性

- 基于三元组表的存储方式的优点是简单直接，易于理解；然而缺点也非常明显，主要有以下两点：
  - 整个知识图谱都存储在一张表中，导致单表的规模太大。对大表进行查询、插入、删除、修改等操作的开销很大，这将导致知识图谱的实用性大打折扣。
  - 复杂查询在这种存储结构上的开销巨大。由于数据表只包括三个字段，因此复杂的查询只能拆分成若干简单查询的复合操作，大大降低了查询的效率。例如，查询“佩佩的身高和性别是什么？”需要拆分为“佩佩的身高是多少？”和“佩佩的性别是什么？”

# 类型表

- 为每种类型构建一张表，同一类型的实例存放在相同的表中。表的每一列表示该类实体的一个属性，每一行存储该类实体的一个实例。




**类型表**

City Table

| subject | area | elevation | postal code |
|---|---|---|---|
| Maceió | 511km² | 7m | 57000-000 |

Person Table

| subject | gender | height | date of birth | team | place of birth |
|---|---|---|---|---|---|
| Pepe | male | 1.88m | 1983/02/06 | Ream Madrid CF | Maceió |

# 类型表的不足

- 大量数据字段的冗余存储。假设知识图谱中既有“演员”也有“歌手”，那么同属于这两个类别的实例将会同时被存储在这两个表中，其中它们共有的属性会被重复存储。

Actor Table

| subject | 性别 | 出生日期 | 出生地 | 艺名 | 经纪公司 |
|---|---|---|---|---|---|
| 刘德华 | 男 | 1961/09/27 | 香港 | 刘德华 | 映艺娱乐 |

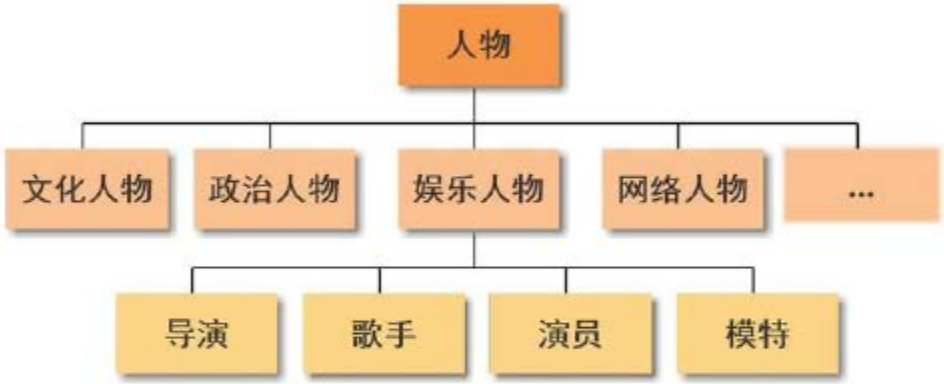
Singer Table

| subject | 性别 | 出生日期 | 出生地 | 艺名 | 经纪公司 | 唱片公司 |
|---|---|---|---|---|---|---|
| 刘德华 | 男 | 1961/09/27 | 香港 | 刘德华 | 映艺娱乐 | 东亚唱片 |

- 大量的数据列为空值。通常知识图谱中并非每个实体在所有属性或关系上都有值，这种存储方式会导致表中存在大量的空值。

# 考虑层级关系的类型表

- 构建数据表时，将知识图谱的类别体系考虑进来。具体来说，每个类型的数据表只记录属于该类型的特有属性，不同类别的公共属性保存在上一级类型对应的数据表中，下级表继承上级表的所有属性。




Person (人物) Table

| subject | 性别 | 出生日期 | 出生地 |
|---|---|---|---|
| 刘德华 | 男 | 1961/09/27 | 香港 |

Entertainer (娱乐人物) Table

| subject | 艺名 | 经纪公司 |
|---|---|---|
| 刘德华 | 刘德华 | 映艺娱乐 |

Actor (演员) Table

| subject |
|---|
| 刘德华 |

Singer (歌手) Table

| subject | 唱片公司 |
|---|---|
| 刘德华 | 东亚唱片 |

# 类型表特性

- 类型表克服了三元组表面临的单表过大和结构简单的问题，但是也有明显的不足之处：
  - 由于数据表是和具体类型对应的，不同的数据表具有不同的结构，因此在查询之前必须知道目标对象的类型才能确定查找的数据表。
  - 当查询涉及到不同类型的实体时，需要进行多表的连接，这一操作开销巨大，限制了知识图谱对复杂查询的处理能力。
  - 知识图谱通常包含丰富的实体类型，因此需要创建大量的数据表，并且这些数据表之间又具有复杂的关系，这为数据的管理增加了很大的难度。

# 关系数据库基本概念

- 关系数据库以二维表结构对数据进行组织和存储。
  - 属性 (attribute)：表中的每一列称为一个属性（也称字段），用来描述实体集的某个特征。每个属性都有自己的取值范围，称为域。
  - 元组 (tuple)：表中的每一行由一个实体的相关属性取值构成，称为元组（也称记录），它相对完整地描述了一个实体。元组中的一个属性值称为分量。

| 学号 | 姓名 | 性别 | 出生日期 | 身份证号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 310103199509222317 |
| 1202 | 张红 | 女 | 1995-11-07 | 410104199511072437 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 510105199604082223 |

# 关系数据库基本概念

- 上述二维表格有以下限制：
  - 每一属性必须是基本的、不能再拆分的数据类型，如整型、实型、字符型等。结构或数组不能作为属性的类型。

# 关系数据库基本概念

- 上述二维表格有以下限制：
  - 每一属性必须是基本的、不能再拆分的数据类型，如整型、实型、字符型等。结构或数组不能作为属性的类型。

| 姓名 | 籍贯 | |
|---|---|---|
| | 省 | 市 |
| 李强 | 黑龙江 | 齐齐哈尔 |
| 张红 | 浙江 | 杭州 |
| 王鹏 | 江西 | 南昌 |

| 姓名 | 省 | 市 |
|---|---|---|
| 李强 | 黑龙江 | 齐齐哈尔 |
| 张红 | 浙江 | 杭州 |
| 王鹏 | 江西 | 南昌 |

# 关系数据库基本概念

- 上述二维表格有以下限制：
  - 每一属性必须是基本的、不能再拆分的数据类型，如整型、实型、字符型等。结构或数组不能作为属性的类型。
  - 属性的所有值必须是同类型、同语义的。如果某一列包含学生的学号，则该表中所有行的此列都必须是学生的学号。
  - 属性的值只能是域中的值。例如学生的性别只能在“男”或“女”中取值。
  - 属性必须有唯一的名字，但不同的属性可以出自相同的域。列在表中的顺序可以任意交换。

# 关系数据库基本概念

- 上述二维表格有以下限制：
  - 每一属性必须是基本的、不能再拆分的数据类型，如整型、实型、字符型等。结构或数组不能作为属性的类型。
  - 属性的所有值必须是同类型、同语义的。如果某一列包含学生的学号，则该表中所有行的此列都必须是学生的学号。
  - 属性的值只能是域中的值。例如学生的性别只能在“男”或“女”中取值。
  - 属性必须有唯一的名字，但不同的属性可以出自相同的域。列在表中的顺序可以任意交换。

| 学号 | 姓名 | 性别 | 出生日期 | 入学日期 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 2017-09-01 |
| 1202 | 张红 | 女 | 1995-11-07 | 2017-09-01 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 2017-09-01 |

# 关系数据库基本概念

- 上述二维表格有以下限制：
  - 每一属性必须是基本的、不能再拆分的数据类型，如整型、实型、字符型等。结构或数组不能作为属性的类型。
  - 属性的所有值必须是同类型、同语义的。如果某一列包含学生的学号，则该表中所有行的此列都必须是学生的学号。
  - 属性的值只能是域中的值。例如学生的性别只能在“男”或“女”中取值。
  - 属性必须有唯一的名字，但不同的属性可以出自相同的域。列在表中的顺序可以任意交换。
  - 任意两个元组的值不能完全相同，即不允许有重复的行。行在表中的顺序可以任意交换。

# 关系数据库属性分类

- 候选码：能够唯一标识元组的最小的属性集合。
  - 唯一性：候选码在整个表的范围内必须具有唯一的值，不同元组不能具有相同的候选码值。
  - 最小性：候选码所包括的属性必须都是必不可少的，缺少其中的任何一个都不再具备唯一性。

| 学号 | 姓名 | 性别 | 出生日期 | 身份证号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 310103199509222317 |
| 1202 | 张红 | 女 | 1995-11-07 | 410104199511072437 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 510105199604082223 |

  - 例如，在上表中学生的学号、身份证号分别都可以作为候选码；但是（学号, 身份证号）这一属性组不能作为候选码，因为去掉其中任何一个剩下的属性仍然可以作为候选码，因此不满足最小性。

# 关系数据库属性分类

- 主码：一个数据表可以包含多个候选码，从中任意选取一个作为主码。虽然理论上所有的候选码都可以作为主码，但是实际操作中一般选择单属性组成的候选码作为主码。

| 学号 | 姓名 | 性别 | 出生日期 | 身份证号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 310103199509222317 |
| 1202 | 张红 | 女 | 1995-11-07 | 410104199511072437 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 510105199604082223 |

  - 在该表中有三个候选码：学号、身份证号、姓名（不重名）。
  - 姓名作主码还是有些限制（不能重名）；
  - 身份证号太长，使用不方便；
  - 学号简洁，使用方便，在实际生活中也被用来唯一区分学生，所以选学号作主码最合适。

# 关系数据库属性分类

- 外码：如果数据表中的某个属性或属性组是其它表的候选码，那么称该属性或属性组是当前表的外码。外码能够保证不同数据表之间数据的一致性。

学生表（外表）

| 学号 | 姓名 | 性别 | 出生日期 | 班号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 11 |
| 1202 | 张红 | 女 | 1995-11-07 | 12 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 13 |

班级表（主表）

| 班号 | 班名 | 专业 |
|---|---|---|
| 11 | 数学01 | 应用数学 |
| 12 | 物理01 | 理论物理 |
| 13 | 化学01 | 材料化学 |

  - 班号是班级表的候选码，所以班号是学生表的外码。

- 主属性与非主属性：包含在任何候选码中的属性称为主属性；相反地，不包含在任何候选码中的属性称为非主属性。

# 关系数据库完整性约束

- 关系数据库通过关系完整性约束条件来保证数据的正确性和一致性。所谓关系完整性约束条件主要是指在表和属性（列）上定义的规则，数据库管理系统会依据这些约束条件维护数据完整性。
  - 域完整性规则
  - 实体完整性规则
  - 参照完整性规则

# 关系数据库完整性约束

- 域完整性：在关系数据模型定义时，由用户对属性值的数据类型、长度、单位、精度、格式、值域范围、是否允许为空值等进行限定，规定属性取值必须在其值域中。

| 学号 | 类型定义 | 属性限定 |
|---|---|---|
| 学号 | char(4) | Primary Key |
| 姓名 | nvarchar(20) | Not Null |
| 性别 | nchar(1) | Not Null, 男/女 |
| 出生日期 | datetime | >=1900-01-01 |
| 班号 | char(2) | Not Null |

| 学号 | 姓名 | 性别 | 出生日期 | 班号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 11 |
| 1202 | 张红 | 女 | 1995-11-07 | 12 |
| 1303 | 王鹏 | ~~21~~ | 1996-04-08 | 13 |

# 关系数据库完整性约束

- 实体完整性：元组（记录）在构成主码的属性上不能有空值且主码的值不能相同。实体完整性主要是为了确保主码能唯一标识元组。

| 学号 | 姓名 | 性别 | 出生日期 | 班号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 11 |
| 1202 | 张红 | 女 | 1995-11-07 | 12 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 13 |
| √1102 | 孙丽 | 女 | 1996-01-29 | 11 |
| X1202 | 赵刚 | 男 | 1995-10-20 | 12 |

# 关系数据库完整性约束

- 参照完整性：一个外表的外码取值必须是其主表主码的存在值或空值。

学生表（外表）

| 学号 | 姓名 | 性别 | 出生日期 | 班号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 11 |
| 1202 | 张红 | 女 | 1995-11-07 | 12 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 13 |

班级表（主表）

| 班号 | 班名 | 专业 |
|---|---|---|
| 11 | 数学01 | 应用数学 |
| 12 | 物理01 | 理论物理 |
| 13 | 化学01 | 材料化学 |

  - 学生表中班号的取值必须是空值或在班级表中班号属性中出现过的值。如果学生表的班号字段允许为空，它取空值表示该学生还没有确定班级；否则只能取班级表出现过的班号值。

# 关系数据库操作语言

- 关系数据库的基本操作语言是SQL (Structured Query Language)。
- SQL语言以简洁的语法支持关系数据库的各类操作（插入/修改/删除/查询）。
- SQL语言独立于关系数据库本身，独立于使用的机器、网络和操作系统。

如果选择关系数据库作为知识图谱的存储引擎，那么对知识图谱的所有操作都需要转换为SQL语句才能执行。

# 关系数据库基本操作

- 关系数据库通过SQL语言为用户提一系列的操作接口，其核心功能包括插入、修改、删除、查询四种操作。
  - 插入 (INSERT)：在一个表中插入一条或多条新记录。
  - 修改 (UPDATE)：在一个表中修改满足条件的记录的某些字段的值。
  - 删除 (DELETE)：从一个表中删除一条或多条满足条件的记录。
  - 查询 (SELECT)：从一个或多个表中提取满足条件的数据、生成计算列或汇总数据。

# 常见的关系数据库存储系统

- DB2：DB2是IBM公司于1983年推出的数据库管理系统，能够运行于各种不同的操作系统平台上，如Windows、Linux、VMS、OS/2等。该数据库的特色之处包括支持面向对象的编程、支持多媒体应用、支持分布式数据库的访问等。更多信息请参考DB2官网：https://www.ibm.com/analytics/us/en/technology/db2/
- Oracle：Oracle是甲骨文公司于1979年推出的关系数据库管理系统，是目前最流行的数据库之一。该数据库遵循标准SQL语言，支持多种数据类型。和DB2相同，Oracle同样具有跨平台的特点，可以运行于不同的操作系统之上。更多信息请参考Oracle官网：https://www.oracle.com/database/index.html

# 常见的关系数据库存储系统

- Microsoft SQL Server：Microsoft SQL Server是微软公司推出的应用于Windows操作系统的数据库产品，该数据库实际上是微软基于Sybase公司提供的技术开发而成的，因此完全兼容Sybase数据库。更多信息请参考Microsoft SQL Server官网：https://www.microsoft.com/zh-cn/sql-server/sql-server-2016
- PostgreSQL：PostgreSQL是一个自由软件系统，和上面几个数据库系统相比，该系统具有开源、免费的特点。由于PostgreSQL可以免费地被使用、修改和分发，因此其具有非常完善的功能，不仅支持大部分的SQL标准，还提供外键、复杂查询、触发器等特性。更多信息请参考PostgreSQL官网：https://www.postgresql.org/

# 常见的关系数据库存储系统

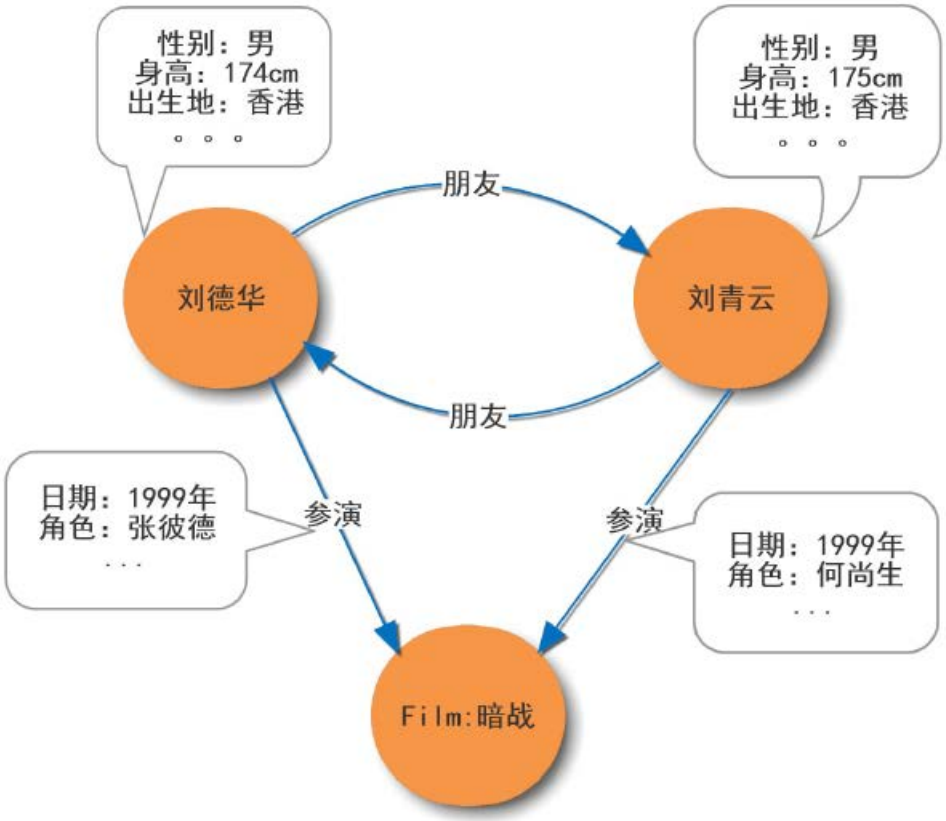
- MySQL：MySQL 也是一个自由软件系统，最早是由瑞典的MySQL AB公司开发的小型关系数据库管理系统，在2008年被Sun公司收购。该系统由于具有体积小、速度快、跨平台、免费开源等特点，是中小型网站最为青睐的数据库。更多信息请参考MySQL官网：https://www.mysql.com/

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 基于图结构的存储模型

- 将实体看做节点，关系看做带有标签的边，那么知识图谱的数据很自然地满足图模型结构。




- **用节点表示实体：**刘德华、刘青云、Film：暗战
- **用边表示实体间的关系：**刘德华和暗战之间的参演关系、刘德华和刘青云之间的朋友关系等
- **节点可以定义属性：**刘德华性别男、身高174cm、出生地香港等
- **边上也可以定义属性：**刘德华参演暗战的时间是1999年，参演角色是张彼得等
- **无向关系需要转化为两条对称的有向关系：**刘德华和刘青云之间互为朋友关系

# 图数据库核心概念

- 图数据库基于有向图，其理论基础是图论。节点、边和属性是图数据库的核心概念。
  - 节点：节点用于表示实体、事件等对象，可以类比于关系数据库中的记录或数据表中的行数据。例如人物、地点、电影等都可以作为图中的节点。
  - 边：边是指图中连接节点的有向线条，用于表示不同节点之间的关系。例如人物节点之间的夫妻关系、同事关系等都可以作为图中的边。
  - 属性：属性用于描述节点或者边的特性。例如人物的姓名、夫妻关系的起止时间等都是属性。

# 常见的图数据库存储系统

- Neo4j：Neo4j是一个开源的图数据库系统，它将结构化的数据存储在图上而不是表中。Neo4j基于Java实现，它是一个具备完全事务特性的高性能数据库，具有成熟数据库的所有特性。Neo4j是一个本地数据库（又称基于文件的数据库），这意味着不需要启动数据库服务器，应用程序不用通过网络访问数据库服务，而是直接在本地对其进行操作，因此访问速度快。因其开源、高性能、轻量级等优势，Neo4j 受到越来越多的关注。更多信息请参考官网：https://neo4j.com/

# 常见的图数据库存储系统

- OrientDB：OrientDB是一个开源的文档-图混合数据库，兼具图数据库对数据强大的表示及组织能力和文档数据库的灵活性及很好的可扩展性。OrientDB具有多种模式可选，包括全模式、无模式和混合模式。全模式要求数据库中的所有类别都必须有严格的模式，所有字段都强制约束；无模式则相反，不需要为类别定义模式，存储的数据记录可以有任意的字段；混合模式则允许为类别定义若干字段，同时支持自定义字段。该数据库同样是本地的，支持许多数据库的高级特性，如事务、快速索引、SQL查询等。更多信息请参考官网：http://orientdb.com/

# 常见的图数据库存储系统

- HyperGraphDB：HyperGraphDB同样是开源的存储系统，并依托于BerkeleyDB数据库。和上述图数据库相比，HyperGraphDB最大的特点在于HyperGraph，即超图。从数学角度讲，有向图的一条边只能指向一个节点，而超图则可以指向多个节点。实际上，HyperGraphDB在超图的基础上更进了一步，不仅允许一条边指向多个节点，还允许其指向其它边。如此以来，HyperGraphDB相较于其它图数据库具有更强大的表示能力。更多信息请参考官网：http://www.hypergraphdb.org/

# 常见的图数据库存储系统

- InfiniteGraph：InfiniteGraph是一个基于Java语言开发的分布式图数据库系统。和MySQL等传统关系数据库类似，InfiniteGraph需要作为服务项目进行安装，应用程序只能通过访问数据库服务对数据库进行操作。Infinite Graph借鉴了面向对象的概念，将图中的每个节点及每条边都看做一个对象。具体地，所有的节点都继承自基类BaseVertex，所有的边则都继承自基类BaseEdge。更多信息请参考官网：http://www.objectivity.com/products/infinitegraph/

# 常见的图数据库存储系统

- InfoGrid：InfoGrid是一个开源的互联网图数据库，提供了很多额外的组件，可以很方便地构建基于图结构的网络应用。InfoGrid实际上是一个基于Java语言的开源项目集，其中InfoGrid图数据库项目是其核心，其它的项目包括InfoGrid存储项目、InfoGrid用户接口项目等。更多信息请参考官网：http://infogrid.org

# 常见的图数据库存储系统

- InfoGrid：InfoGrid是一个开源的互联网图数据库，提供了很多额外的组件，可以很方便地构建基于图结构的网络应用。InfoGrid实际上是一个基于Java语言的开源项目集，其中InfoGrid图数据库项目是其核心，其它的项目包括InfoGrid存储项目、InfoGrid用户接口项目等。更多信息请参考官网：http://infogrid.org

和成熟的关系数据库相比，图数据库的发展较晚，相关的标准及技术尚不完善，在实际使用中可能会遇到一些棘手的问题，因此在选用数据库时除了需要考虑数据库本身的特性、性能等因素以外，还需要考虑数据库的易用性、技术文档的完整性等因素。

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 知识图谱的检索

- 知识图谱的知识实际上是通过数据库系统进行存储的，大部分数据库系统通过形式化的查询语言为用户提供访问数据的接口。
  - 关系数据库：标准查询语言SQL
  - 图数据库：标准查询语言SPARQL
- 由于知识图谱中的数据在逻辑上本身就是一种图结构，因此也可以通过图查询技术来完成特定查询图的查找，其核心问题是判断查询图是否是图数据集的子图，因此也叫子图匹配问题。

《A Performance Comparison of Five Algorithms for Graph Isomorphism》

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# SQL的产生与发展

- SQL（Structured Query Language，结构化查询语言）是介于关系代数和元组演算之间的一种语言，用于管理关系数据库，目前已经成为关系数据库的标准语言。
  - 1972年，IBM公司开始研制实验型关系数据库管理系统SYSTEM R，其配备的查询语言称为SQUARE（Specifying Queries As Relational Expression）语言，语言中使用了较多的数学符号。
  - 1974年，Boyce和Chamberlin把SQUARE修改为SEQUEL (Structured English Query Language) 语言。后来SEQUEL简称为SQL (Structured Query Language)，即结构化查询语言，SQL的发音仍是“sequel”。现在SQL已经成为一个标准。

# SQL的组成

- SQL语言从功能上可以分为四个部分：数据查询、数据操纵、数据定义和数据控制。
  - 数据查询语言，即SQL DQL，用来对已存在的数据库中的数据按照指定的组合、条件表达式或排序进行检索。
  - 数据操纵语言，即SQL DML，用来改变数据库中的数据，分为插入、修改、删除三种操作。
  - 数据定义语言，即SQL DDL，用来创建数据库中的各种对象，包括数据库模式、表、视图、索引、同义词、聚簇等。
  - 数据控制语言，即SQL DCL，用来授予或回收访问数据库的某种特权，控制数据操纵事务的发生时间及效果、对数据库进行监视等。

# SQL的组成

- SQL语言功能极强，但由于设计巧妙，语言十分简洁，完成数据定义、数据操纵、数据控制、数据查询的核心功能只用了9个动词。并且SQL语言语法简单，接近英语口语，因此易学易用。

| 功　能 | 动　词 |
| --- | --- |
| 数据查询 | SELECT |
| 数据操纵 | INSERT, UPDATE, DELETE |
| 数据定义 | CREATE, DROP, ALTER |
| 数据控制 | GRANT, REVOKE |

# 数据查询

- 数据查询是指从数据表中选取满足条件的数据，查询结果存储在一个临时的结果表中。SQL通过SELECT语句完成该功能，其基本语法为：

  SELECT 目标表的列名或列表达式序列
  FROM 基本表名和（或）视图序列
  [ WHERE 行条件表达式 ]
  [ GROUP BY 列名序列 ]
  [ HAVING 组条件表达式 ]
  [ ORDER BY 列名 [ ASC | DESC ], … ]

  - 主语句**SELECT-FROM-WHERE**的含义是：根据WHERE子句的条件表达式，从FROM子句指定的基本表或视图中找出满足条件的元组，再按SELECT子句中的目标列表达式，选出元组中的属性值形成结果表。

# 数据查询

- 完整的**SELECT**语句执行过程如下：
  ① 读取FROM子句中基本表、视图的数据；
  ② 选取满足WHERE子句中给出的条件表达式的元组；
  ③ 按GROUP BY子句中指定列的值分组，同时提取满足HAVING子句中组条件表达式的那些组；
  ④ 按SELECT子句中给出的列名或列表达式求值输出；
  ⑤ ORDER BY子句对输出的目标表进行排序，按附加说明ASC升序排列，或按DESC降序排列。

# 单表查询

- 查询仅涉及一个表，是一种最简单的查询操作：
  - 选择表中的若干列
  - 选择表中的若干元组
  - 对查询结果排序
  - 使用聚合函数
  - 对查询结果分组

# 查询表中的若干列

- 查询指定列：在很多情况下，用户只对表中的一部分属性感兴趣，这时可以通过在SELECT子句的<目标列表达式>中指定要查询的属性列。

例9.1 查询以下学生表中全体学生的学号与姓名。

```
SELECT number, name
FROM Student;
```

| number | name | sex | age | birthplace |
|---|---|---|---|---|
| 20141001 | 朱晓明 | 男 | 26 | 江苏徐州 |
| 20141002 | 王月华 | 女 | 25 | 山西太原 |
| 20141003 | 刘子华 | 男 | 26 | 四川成都 |

Student

| number | name |
|---|---|
| 20141001 | 朱晓明 |
| 20141002 | 王月华 |
| 20141003 | 刘子华 |

# 查询表中的若干列

- 查询全部列：将表中的所有属性列都选出来，可以有两种方法。一种方法就是在SELECT关键字后面列出所有列名；另一种方法是如果列的显示顺序与其在基表中的顺序相同，也可以简单地将<目标列表达式>指定为“*”。

例9.2 查询Student表中全体学生的详细记录。

```
SELECT *
FROM Student;
```

等价于：

```
SELECT number, name, sex, age, birthplace
FROM Student;
```

# 查询表中的若干列

- 查询经过计算的值：SELECT子句的<目标列表达式>不仅可以是表中的属性列，也可以是表达式。

例9.3 查询Student表中全体学生的姓名及其出生年份。

```
SELECT name, 2017-age
FROM Student;
```

| number | name | sex | age | birthplace |
|---|---|---|---|---|
| 20141001 | 朱晓明 | 男 | 26 | 江苏徐州 |
| 20141002 | 王月华 | 女 | 25 | 山西太原 |
| 20141003 | 刘子华 | 男 | 26 | 四川成都 |

Student

| name | |
|---|---|
| 朱晓明 | 1991 |
| 王月华 | 1992 |
| 刘子华 | 1991 |

# 查询表中的若干元组

- 消除取值重复的行：两个本来并不完全相同的元组，投影到指定的某些列上后，可能变成相同的行了，这时可以用DISTINCT消除它们。

例9.4 查询Student表中全体学生的年龄。

| number | name | sex | age | birthplace |
|---|---|---|---|---|
| 20141001 | 朱晓明 | 男 | 26 | 江苏徐州 |
| 20141002 | 王月华 | 女 | 25 | 山西太原 |
| 20141003 | 刘子华 | 男 | 26 | 四川成都 |

```
SELECT age
FROM Student;
```

| age |
|---|
| 26 |
| 25 |
| 26 |

```
SELECT DISTINCT age
FROM Student;
```

| age |
|---|
| 26 |
| 25 |

# 查询表中的若干元组

- 查询满足条件的元组：查询满足指定条件的元组可以通过WHERE子句实现。

| 查询条件 | 谓　词 |
|---|---|
| 比较 | =, >, <, >=, <=, !=, <>, !>, !<, NOT+上述运算符 |
| 确定范围 | BETWEEN AND, NOT BETWEEN AND |
| 确定集合 | IN, NOT IN |
| 字符匹配 | LIKE, NOT LIKE |
| 空值 | IS NULL, IS NOT NULL |
| 逻辑运算 | AND, OR, NOT |

# 查询表中的若干元组

- 查询满足条件的元组：查询满足指定条件的元组可以通过WHERE子句实现。

例9.5 查询Student表中出生于江苏徐州的学生的学号和姓名。

```
SELECT number, name
FROM Student
WHERE birthplace='江苏徐州';
```

| number | name |
|---|---|
| 20141001 | 朱晓明 |

例9.6 查询Student表中年龄大于25岁的学生的学号和出生地。

```
SELECT number, birthplace
FROM Student
WHERE age>25;
```

| number | birthplace |
|---|---|
| 20141001 | 江苏徐州 |
| 20141003 | 四川成都 |

# 对查询结果排序

- 使用ORDER BY子句可以按一个或多个属性列排序。ASC（缺省值）表示按升序排列，DESC表示按降序排列。当排序列含空值时，ASC排序列为空值的元组最后显示，DESC排序列为空值的元组最先显示。

例9.7 查询选修了3号课程的学生的学号及其成绩，查询结果按分数的降序排列。

```
SELECT studentNo, grade
FROM StudentCourse
WHERE courseNo='3'
ORDER BY grade DESC;
```

| studentNo | grade |
|---|---|
| 95010 | |
| 95007 | 92 |
| 95003 | 82 |
| 95010 | 82 |
| 95009 | 75 |
| 95014 | 61 |

# 使用聚合函数

- 聚合函数是涉及整个关系的另一类运算操作。通过聚合函数，可以把某一列中的值形成单个值。
  - 计数：COUNT ( [ DISTINCT | ALL ] * )
    COUNT ( [ DISTINCT | ALL ] <列名> )
  - 计算总和： SUM ( [ DISTINCT | ALL ] <列名> )
  - 计算平均值： AVG ( [ DISTINCT | ALL ] <列名> )
  - 求最大值： MAX ( [ DISTINCT | ALL ] <列名> )
  - 求最小值： MIN ( [ DISTINCT | ALL ] <列名> )

# 使用聚合函数

- 聚合函数是涉及整个关系的另一类运算操作。通过聚合函数，可以把某一列中的值形成单个值。

例9.8 查询选修了课程的学生人数。

```
SELECT COUNT ( DISTINCT studentNo )
FROM StudentCourse;
```

例9.9 计算选修了1号课程的学生的平均成绩。

```
SELECT AVG ( grade )
FROM StudentCourse
WHERE courseNo='1';
```

# 对查询结果分组

- 使用GROUP BY子句分组，细化聚合函数的作用对象：未对查询结果分组，聚合函数将作用于整个查询结果；对查询结果分组后，聚合函数将分别作用于每个组。

例9.10 查询每门课程的选课人数和平均成绩。

```
SELECT courseNo, COUNT (studentNo), AVG (grade)
FROM StudentCourse
GROUP BY courseNo;
```

# 数据操纵

- 数据插入是指向数据表中添加新的数据记录，SQL通过INSERT语句完成该功能。其基本语法为：

  INSERT
  INTO 表名 [(<属性列1> [,<属性列2>, … ])]
  VALUES (<常量1> [,<常量2>, … ])

  - 该语句的功能是将新元组插入指定表中，其中新元组的属性列1的值为常量1，属性列2的值为常量2，依此类推。INTO子句中没有出现的属性列，新元组在这些列上将取空值。如果INTO子句没有指明任何属性列名，则新插入的元组必须在每个属性列上均有值。

# 数据操纵

例9.11 将一个新学生元组 (学号: 20141004; 姓名: 陈东; 性别: 男; 年龄: 24; 出生地: 广东深圳) 插入到Student表中。

```
INSERT
INTO Student
VALUES ('20141004','陈东','男',24,'广东深圳');

INSERT
INTO Student (number, name, sex, age, birthplace)
VALUES ('20141004','陈东','男',24,'广东深圳');
```

| number | name | sex | age | birthplace |
|---|---|---|---|---|
| 20141001 | 朱晓明 | 男 | 26 | 江苏徐州 |
| 20141002 | 王月华 | 女 | 25 | 山西太原 |
| 20141003 | 刘子华 | 男 | 26 | 四川成都 |
| 20141004 | 陈　东 | 男 | 24 | 广东深圳 |

# 数据操纵

- 数据修改是指修改数据表中已有记录某些列的值，SQL通过UPDATE语句完成该功能。其基本语法为：

  UPDATE 表名
  SET <列名>=<表达式> [ , <列名>=<表达式>, … ]
  [ WHERE 行条件表达式 ]

  - 该语句中SET子句给出<表达式>的值用于取代相应的属性列值。如果省略WHERE子句，则表示要修改表中的所有元组。

# 数据操纵

例9.12 将学号为20141003的学生的年龄改为25岁。

```
UPDATE Student
SET age=25
WHERE number='20141003';
```

例9.13 将所有学生的年龄增加1岁。

```
UPDATE Student
SET age=age+1;
```

# 数据操纵

- 数据删除是指从数据表中删除指定的数据记录，SQL通过DELETE语句完成该功能。其基本语法为：

```
DELETE
FROM 表名
[WHERE 行条件表达式]
```

例9.14 删除学号为20141003的学生记录。

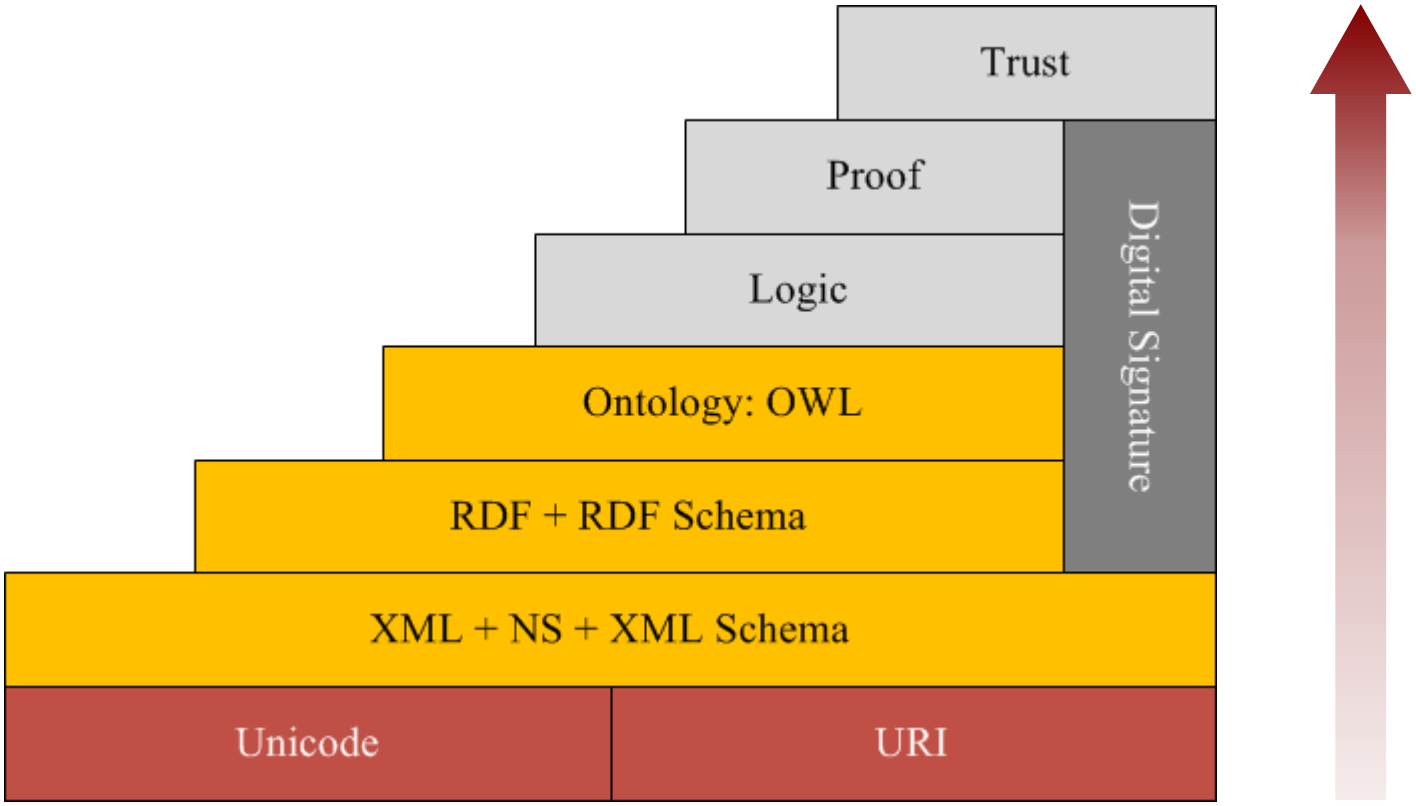
```
DELETE
FROM Student
WHERE number='20141003';
```

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
- 本章小结

# 语义网回顾

- 本质：以Web数据的内容（即语义）为核心，用机器能够理解和处理的方式链接起来的海量分布式数据库。



# RDF回顾

- RDF（Resource Description Framework，资源描述框架）是一种资源描述语言，其核心思想是利用Web标识符（URI）来标识事物，通过指定的属性和相应的值描述资源的性质或资源之间的关系。

```
# Default Graph
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>  ns:director  <http://example/person1> .
```

图1 只包含一个三元组的RDF图示例

# SPARQL语言

- SPARQL是Simple Protocol and RDF Query Language的缩写，是由W3C为RDF数据开发的一种查询语言和数据获取协议，是被图数据库广泛支持的查询语言。和SQL类似，SPARQL也是一种结构化的查询语言，用于对数据的获取与管理。
  - 数据操纵（插入、删除、更新）
  - 数据查询

# 数据操纵

- 数据插入：将新的三元组插入到已有的RDF图中。SPARQL通过INSERT DATA语句完成该功能，其基本语法为：

  INSERT DATA 三元组数据

  - 三元组数据可以是多条三元组，不同的三元组通过“;” 分隔。如果待插入的三元组在RDF图中已经存在，那么系统会忽略该三元组。

例9.15 将以下两条三元组插入到图1所示的RDF图中。

<http://example/movie1> ns:producer <http://example/person2>
<http://example/movie1> ns:writer <http://example/person3>

# 数据操纵

其对应的SPARQL语句为：

```
prefix ns: <http://example.org/ns#>

INSERT DATA
{
<http://example/movie1>  ns:producer  <http://example/person2>;
                         ns:writer  <http://example/person3> .
}
```

执行该语句后，图1中的RDF图变为：

```
# Default Graph
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>  ns:director  <http://example/person1> .
<http://example/movie1>  ns:producer  <http://example/person2>.
<http://example/movie1>  ns:writer  <http://example/person3> .
```

图2 SPARQL插入数据结果示例

# 数据操纵

- 数据删除：从RDF图中删除一些三元组。SPARQL通过DELETE DATA语句完成该功能，其基本语法为：

  DELETE DATA 三元组数据

  - 其中三元组数据可以是多条三元组，不同的三元组通过“;” 分隔。对于给定的每个三元组，如果其在RDF图中，则将其从图中删除，否则忽略该三元组。

例9.16 将以下三元组从图2所示的RDF图中删除。

<http://example/movie1> ns:producer <http://example/person2>

# 数据操纵

其对应的SPARQL语句为：

```
prefix ns: <http://example.org/ns#>

DELETE DATA
{
<http://example/movie1>  ns:producer  <http://example/person2>.
}
```

执行该语句后，图2中的RDF图变为：

```
# Default Graph
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>  ns:director  <http://example/person1> .
<http://example/movie1>  ns:writer  <http://example/person3> .
```

图3 SPARQL删除数据结果示例

# 数据操纵

- 数据更新：更新RDF图中指定三元组的值。和SQL不同，SPARQL没有定义UPDATE操作，也就是说SPARQL语言没有直接更新已有数据的方法。但是，可以通过组合INSERT DATA语句和DELETE DATA语句来实现该功能。
  - DELETE DATA：删除待修改的三元组；
  - INSERT DATA：插入修改后的三元组。

例9.17 将图3中三元组 (<http://example/movie1> ns:director <http://example/person1>) 的客体<http://example/person1>修改为<http://example/person4>。

# 数据操纵

其对应的SPARQL语句为：

```
prefix ns: <http://example.org/ns#>

DELETE DATA
{
<http://example/movie1> ns:director  <http://example/person1>.
};

INSERT DATA
{
<http://example/movie1> ns:director  <http://example/person4>.
}
```

执行该语句后，图3中的RDF图变为：

```
# Default Graph
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>   ns:director  <http://example/person4> .
<http://example/movie1>   ns:writer  <http://example/person3> .
```

图4 SPARQL更新数据结果示例

# 数据查询

- SPARQL提供了丰富的数据查询功能，包括四种形式：
  - SELECT：最为常用的查询语句，其功能和SQL中的SELECT语句类似，从知识图谱中获取满足条件的数据。
  - ASK：用于测试知识图谱中是否存在满足给定条件的数据，如果存在则返回“yes”，否则返回“no”，该查询不会返回具体的匹配数据。
  - DESCRIBE：用于查询和指定资源相关的RDF数据，这些数据形成了对给定资源的详细描述。
  - CONSTRUCT：则根据查询图的结果生成RDF。

# SELECT语句

- SELECT语句的基本语法为：

  SELECT 变量1 变量2 …
  WHERE 图模式
  [ 修饰符 ]

  - SELECT子句中的“变量1 变量2 ...”和SQL中的“列1，列2，...”的含义类似，表示要查询的目标。不同之处在于，SQL处理的数据存储于结构化的二维表中，语句中的“列1，列2，...”和数据表中的列完全对应，因此查询的目标具有明确的语义；而SPARQL处理的数据则具有更加灵活的存储结构，“变量1 变量2 ...”在知识图谱中没有直接的对应。

# SELECT语句

- SELECT语句的基本语法为：

  SELECT 变量1 变量2 …
  WHERE 图模式
  [ 修饰符 ]

  - WHERE子句用于为SELECT子句中的变量提供约束，查询结果必须完全匹配该子句给出的图模式。图模式主要由两类元素组成，一类是三元组，例如“?x a Person”表示变量?x必须是Person的一个实例；另一类是通过FILTER关键字给出的条件限制，这些条件包括数字大小的限制、字符串格式的限制等。
  - 修饰符是可选项，用于对查询结果做一些特殊处理，例如常见的修饰符有用于表示排序的ORDER子句、用于限制结果数量的Limit子句等。

# SELECT语句

例9.18 在图4所示的RDF图中查询资源<http://example/movie1>的导演和编剧。

相应的SPARQL语句为：

```
prefix ns: <http://example.org/ns#>

SELECT ?director ?writer
WHERE
{
<http://example/movie1> ns:director  ?director.
<http://example/movie1> ns:writer  ?writer.
}
```

执行该语句的查询结果为：

| director | writer |
|---|---|
| <http://example/person4> | <http://example/person3> |

# ASK语句

- ASK语句用于测试知识图谱中是否存在满足给定图模式的数据，其基本语法为：

  ASK 图模式

例9.19 在图4所示的RDF图中测试是否存在由<http://example/person4>导演的电影。

相应的SPARQL语句为：

```
prefix ns: <http://example.org/ns#>

ASK
{
?movie ns:director <http://example/person4>.
}
```

该语句返回的结果为“yes”；如果查询是否存在<http://example/person5>导演的电影，查询的结果将会返回“no”。

# DESCRIBE语句

- DESCRIBE语句用于获取和给定资源相关的数据，其基本语法为：

  DESCRIBE 资源或变量
  [WHERE 图模式]

  - DESCRIBE后既可以直接跟确定的资源标识符，也可以跟变量；
  - WHERE子句是可选的，用于限定变量需要满足的图模式。

例9.20 在图4所示的RDF图中获取由<http://example/person4>导演的所有电影的相关信息。

# DESCRIBE语句

其对应的SPARQL语句为:

```
prefix ns: <http://example.org/ns#>

DESCRIBE ?movie
WHERE
{
?movie ns:director <http://example/person4>.
}
```

该语句的执行结果为:

```
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>  ns:director  <http://example/person4> .
<http://example/movie1>  ns:writer  <http://example/person3> .
```

# CONSTRUCT语句

- CONSTRUCT语句用于生成满足图模式的RDF图，其基本语法为：

  CONSTRUCT RDF图模板

  WHERE 图模式

  - RDF图模板确定了生成的RDF图所包含的三元组类型，它由一组三元组构成，每个三元组既可以是包含变量的三元组模板，也可以是不包含变量的事实三元组；图模式则和上述SELECT 等语句中介绍的相同，用于约束语句中的变量。
  - 该语句产生结果的基本流程为：首先执行WHERE子句，从知识图谱中获取所有满足图模式的变量取值；然后针对每一个变量取值，替换RDF图模板中的变量，生成一组三元组。

# CONSTRUCT语句

例9.21 在图4所示的知识图谱上执行如下SPARQL语句：

```
prefix ns: <http://example.org/ns#>

CONSTRUCT
{
?movie a ns:Movie .
?movie ns:director ?director .
}
WHERE
{
?movie ns:writer <http://example/person3> .
?movie ns:director ?director .
}
```

?movie = <http://example/movie1>

?director = <http://example/person4>

# CONSTRUCT语句

该语句的执行结果为:

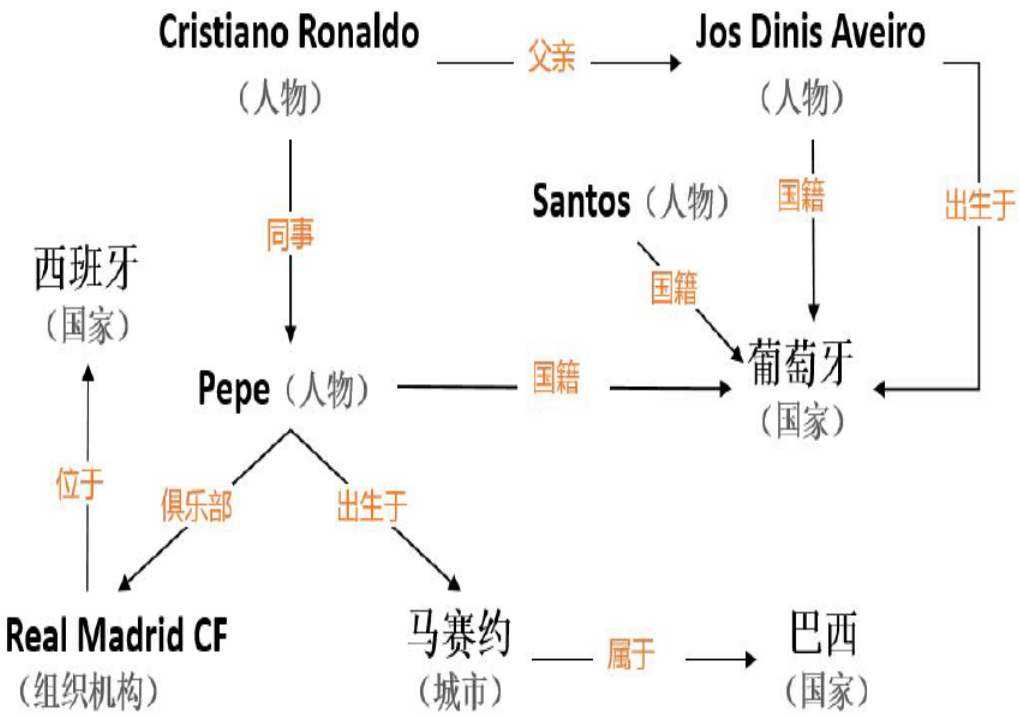
```
@prefix ns: <http://example.org/ns#> .

<http://example/movie1>  a  ns:Movie.
<http://example/movie1>  ns:director  <http://example/person4> .
```

# 提纲

- 概述
- 知识图谱的存储
  - 基于表结构的存储
  - 基于图结构的存储
- 知识图谱的检索
  - 关系数据库查询：SQL语言
  - 图数据库查询：SPARQL语言
  - 图查询技术
- 本章小结

# 知识图谱

- 知识图谱是一种有向图结构，描述了现实世界中存在的实体、事件或者概念以及它们之间的相关关系。




- **实体：**Cristiano Ronaldo, Jos Dinis Aveiro, Real Madrid CF、马赛约等
- **实体类型：**人物、国家、城市、组织机构等
- **属性：**人物有姓名、性别、出生日期、兴趣爱好、职业等属性；国家有国庆日、国家代码、货币、时区、等属性
- **关系：**人物和人物间的同事关系、人物和国家间的国籍关系、城市和国家间的属于关系等

# 知识图谱

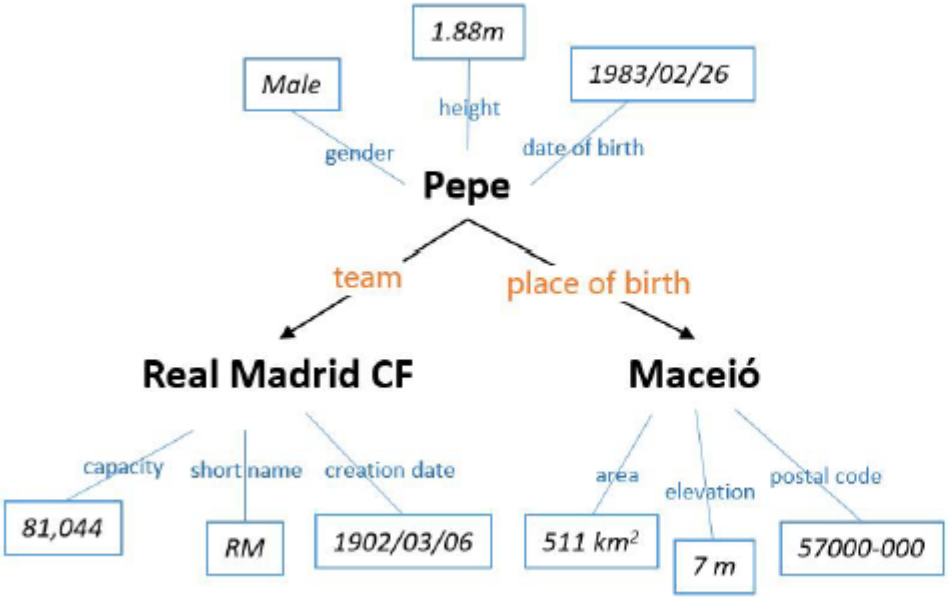
- 知识图谱中的知识是通过RDF的结构进行表示的，其基本构成单元是事实，每个事实被表示为一个形如<subject, predicate, object>的三元组。
- 知识图谱的目标是构建一个能够刻画现实世界的知识库，为自动问答、信息检索等应用提供支撑。因此，对知识的持久化存储并提供对目标知识的高效检索是合格的知识图谱必须具备的基本功能。

# 知识图谱的存储

- 按照存储方式的不同，知识图谱的存储可以分为基于表结构的存储和基于图结构的存储。
  - 基于表结构的存储：利用二维的数据表对知识图谱中的数据进行存储
    - 三元组表、类型表、关系数据库
  - 基于图结构的存储：利用图的方式对知识图谱中的数据进行存储
    - 图数据库

# 三元组表

- 知识图谱中的事实是一个个的三元组，一种最简单直接的存储方式是设计一张三元组表用于存储知识图谱中所有的事实。

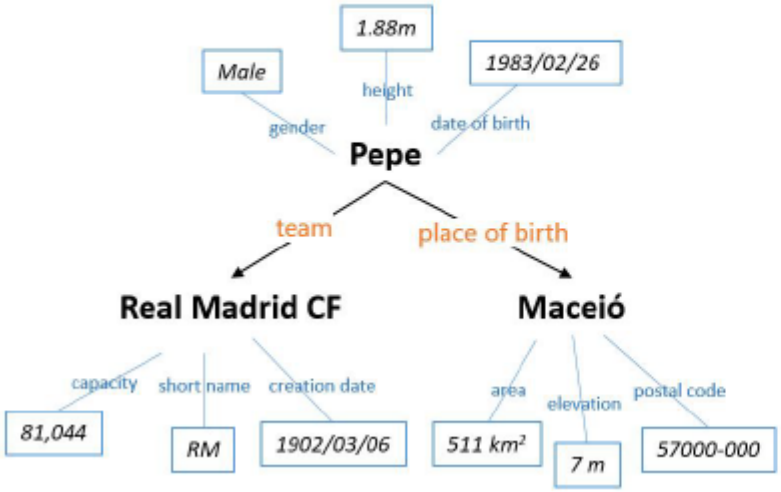



**三元组表**

| S. | P. | O. |
|---|---|---|
| Pepe | gender | male |
| Pepe | height | 1.88m |
| Pepe | date of birth | 1983/02/06 |
| Pepe | team | Ream Madrid CF |
| Pepe | place of birth | Maceió |
| Maceió | area | 511km$^2$ |
| Maceió | elevation | 7m |
| Maceió | postal code | 57000-000 |
| ... | ... | ... |

# 类型表

- 为每种类型构建一张表，同一类型的实例存放在相同的表中。表的每一列表示该类实体的一个属性，每一行存储该类实体的一个实例。




**类型表**

City Table

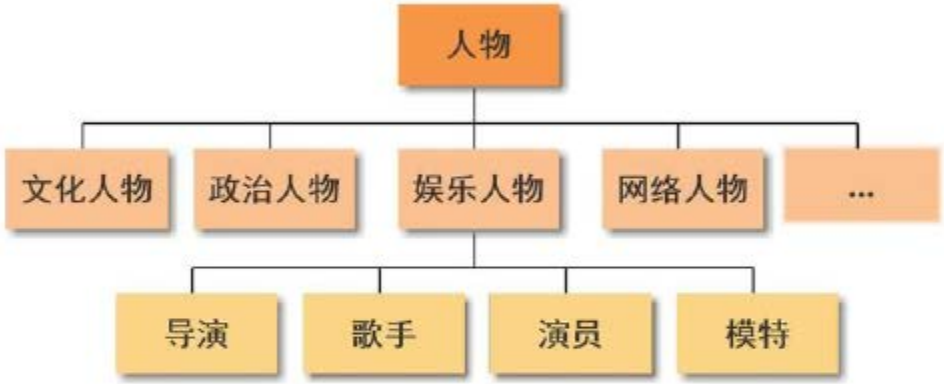
| subject | area | elevation | postal code |
|---|---|---|---|
| Maceió | 511km² | 7m | 57000-000 |

Person Table

| subject | gender | height | date of birth | team | place of birth |
|---|---|---|---|---|---|
| Pepe | male | 1.88m | 1983/02/06 | Ream Madrid CF | Maceió |

# 考虑层级关系的类型表

- 构建数据表时，将知识图谱的类别体系考虑进来。具体来说，每个类型的数据表只记录属于该类型的特有属性，不同类别的公共属性保存在上一级类型对应的数据表中，下级表继承上级表的所有属性。




Person (人物) Table

| subject | 性别 | 出生日期 | 出生地 |
|---|---|---|---|
| 刘德华 | 男 | 1961/09/27 | 香港 |

Entertainer (娱乐人物) Table

| subject | 艺名 | 经纪公司 |
|---|---|---|
| 刘德华 | 刘德华 | 映艺娱乐 |

Actor (演员) Table

| subject |
|---|
| 刘德华 |

Singer (歌手) Table

| subject | 唱片公司 |
|---|---|
| 刘德华 | 东亚唱片 |

# 关系数据库

- 关系数据库以二维表结构对数据进行组织和存储。
  - 属性 (attribute)：表中的每一列称为一个属性（也称字段），用来描述实体集的某个特征。每个属性都有自己的取值范围，称为域。
  - 元组 (tuple)：表中的每一行由一个实体的相关属性取值构成，称为元组（也称记录），它相对完整地描述了一个实体。元组中的一个属性值称为分量。

| 学号 | 姓名 | 性别 | 出生日期 | 身份证号 |
|---|---|---|---|---|
| 1101 | 李强 | 男 | 1995-09-22 | 310103199509222317 |
| 1202 | 张红 | 女 | 1995-11-07 | 410104199511072437 |
| 1303 | 王鹏 | 男 | 1996-04-08 | 510105199604082223 |

# 关系数据库

- 每张数据表可以由任意数量的属性组成，这些属性按照各自的特点可以分为如下几类：
  - 候选码：能够唯一标识元组的最小的属性集合。
  - 主码：一个数据表可以包含多个候选码，从中任意选取一个作为主码。
  - 外码：如果数据表中的某个属性或属性组是其它表的候选码，那么称该属性或属性组是当前表的外码。
  - 主属性与非主属性：包含在任何候选码中的属性称为主属性；相反地，不包含在任何候选码中的属性称为非主属性。

# 关系数据库完整性约束

- 通过完整性约束条件来保证数据的正确性和一致性。
  - 域完整性：在关系数据模型定义时，由用户对属性值的数据类型、长度、单位、精度、格式、值域范围、是否允许为空值等进行限定，规定属性取值必须在其值域中。
  - 实体完整性：元组（记录）在构成主码的属性上不能有空值且主码的值不能相同。实体完整性主要是为了确保主码能唯一标识元组。
  - 参照完整性：一个外表的外码取值必须是其主表主码的存在值或空值。

# 常见的关系数据库存储系统

- DB2 (https://www.ibm.com/analytics/us/en/technology/db2/)
- Oracle (https://www.oracle.com/database/index.html)
- Microsoft SQL Server (https://www.microsoft.com/zh-cn/sql-server/sql-server-2016)
- PostgreSQL (https://www.postgresql.org/)
- MySQL (https://www.mysql.com/)

# 图数据库

- 图数据库基于有向图，其理论基础是图论。节点、边和属性是图数据库的核心概念。
  - 节点：节点用于表示实体、事件等对象，可以类比于关系数据库中的记录或数据表中的行数据。例如人物、地点、电影等都可以作为图中的节点。
  - 边：边是指图中连接节点的有向线条，用于表示不同节点之间的关系。例如人物节点之间的夫妻关系、同事关系等都可以作为图中的边。
  - 属性：属性用于描述节点或者边的特性。例如人物的姓名、夫妻关系的起止时间等都是属性。

# 常见的图数据库存储系统

- Neo4j (https://neo4j.com/)
- OrientDB (http://orientdb.com/)
- HyperGraphDB (http://www.hypergraphdb.org/)
- InfiniteGraph (http://www.objectivity.com/products/infinitegraph/)
- InfoGrid (http://infogrid.org)

# 知识图谱的检索

- 知识图谱的知识实际上是通过数据库系统进行存储的，大部分数据库系统通过形式化的查询语言为用户提供访问数据的接口。
  - 关系数据库：标准查询语言SQL
  - 图数据库：标准查询语言SPARQL
- 由于知识图谱中的数据在逻辑上本身就是一种图结构，因此也可以通过图查询技术来完成特定查询图的查找，其核心问题是判断查询图是否是图数据集的子图，因此也叫子图匹配问题。

# SQL的组成

- SQL语言从功能上可以分为四个部分：数据查询、数据操纵、数据定义和数据控制。

| 功　能 | 动　词 |
| --- | --- |
| 数据查询 | SELECT |
| 数据操纵 | INSERT, UPDATE, DELETE |
| 数据定义 | CREATE, DROP, ALTER |
| 数据控制 | GRANT, REVOKE |

# 数据查询

- 数据查询是指从数据表中选取满足条件的数据，查询结果存储在一个临时的结果表中。SQL通过SELECT语句完成该功能，其基本语法为：

  SELECT 目标表的列名或列表达式序列
  FROM 基本表名和（或）视图序列
  [ WHERE 行条件表达式 ]
  [ GROUP BY 列名序列 ]
  [ HAVING 组条件表达式 ]
  [ ORDER BY 列名 [ ASC | DESC ], … ]

  - 主语句**SELECT-FROM-WHERE**的含义是：根据WHERE子句的条件表达式，从FROM子句指定的基本表或视图中找出满足条件的元组，再按SELECT子句中的目标列表达式，选出元组中的属性值形成结果表。

# 数据操纵

- 数据插入是指向数据表中添加新的数据记录，SQL通过INSERT语句完成该功能。其基本语法为：

  INSERT
  INTO 表名 [(<属性列1> [,<属性列2>, … ])]
  VALUES (<常量1> [,<常量2>, … ])

- 数据修改是指修改数据表中已有记录某些列的值，SQL通过UPDATE语句完成该功能。其基本语法为：

  UPDATE 表名
  SET <列名>=<表达式> [,<列名>=<表达式>, … ]
  [WHERE 行条件表达式]

# 数据操纵

- 数据删除是指从数据表中删除指定的数据记录，SQL通过DELETE语句完成该功能。其基本语法为：

```
DELETE
FROM 表名
[WHERE 行条件表达式]
```

# SPARQL语言

- SPARQL是Simple Protocol and RDF Query Language的缩写，是由W3C为RDF数据开发的一种查询语言和数据获取协议，是被图数据库广泛支持的查询语言。和SQL类似，SPARQL也是一种结构化的查询语言，用于对数据的获取与管理。
  - 数据操纵（插入、删除、更新）
  - 数据查询

# 数据操纵

- 数据插入：将新的三元组插入到已有的RDF图中。SPARQL通过INSERT DATA语句完成该功能。
- 数据删除：从RDF图中删除一些三元组。SPARQL通过DELETE DATA语句完成该功能。
- 数据更新：更新RDF图中指定三元组的值。和SQL不同，SPARQL没有定义UPDATE操作，也就是说SPARQL语言没有直接更新已有数据的方法。但是，可以通过组合INSERT DATA语句和DELETE DATA语句来实现该功能。

# 数据查询

- SPARQL提供了丰富的数据查询功能，包括四种形式：
  - SELECT：最为常用的查询语句，其功能和SQL中的SELECT语句类似，从知识图谱中获取满足条件的数据。
  - ASK：用于测试知识图谱中是否存在满足给定条件的数据，如果存在则返回“yes”，否则返回“no”，该查询不会返回具体的匹配数据。
  - DESCRIBE：用于查询和指定资源相关的RDF数据，这些数据形成了对给定资源的详细描述。
  - CONSTRUCT：则根据查询图的结果生成RDF。

谢谢！

wangquan@iie.ac.cn