Knowledge Graph: Techniques

from world wide web to knowledge graph

Peng Wang

School of Computer Science and Engineering

online open course: https://github.com/npubird/KnowledgeGraphCourse

Unwatch 22 | Unstar 420 | Fork 150

A systematic course about knowledge graph for graduate students, interested researchers and engineers.

东南大学《知识图谱》研究生课程
时间：2019年春季（2月下旬~5月中旬）
每周五下午2:00~4:30
地点：东南大学九龙湖校区, 纪忠楼Y205
答疑/讨论/建议：请致信 pwang AT seu.edu.cn

课程内容

第1讲 知识图谱概论 （2019-3-1,2019-3-8）

1.1 知识图谱起源和发展
1.2 知识图谱 VS 深度学习
1.3 知识图谱 VS 关系数据库 VS 传统专家库
1.4 知识图谱本质和核心价值
1.5 知识图谱技术体系
1.6 典型知识图谱
1.7 知识图谱应用场景
课件下载:partA partB partC

第2讲 知识表示 (2019-3-15)

2.1 知识表示概念
2.2 知识表示方法

今日头条 首页 / 教育 / 正文

24
转发
微博
Qzone
微信

东南大学2019年最新研究生精品课程《知识图谱》资源分享

原创 深度学习与NLP 2019-09-15 09:50:50

Outline

- 1. Knowledge Representation and Modeling
- 2. Knowledge Extracting
- 3. Knowledge Fusion
- 4. Knowledge Storage

What is knowledge representation?

- **Knowledge representation and reasoning is the field of artificial intelligence (AI) dedicated to representing information about the world in a form that a computer system can utilize to solve complex tasks such as diagnosing a medical condition or having a dialog in a natural language. Knowledge representation incorporates findings from psychology about how humans solve problems and represent knowledge in order to design formalisms that will make complex systems easier to design and build. Knowledge representation and reasoning also incorporates findings from logic to automate various kinds of reasoning, such as the application of rules or the relations of sets and subsets**

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

knowledge representation

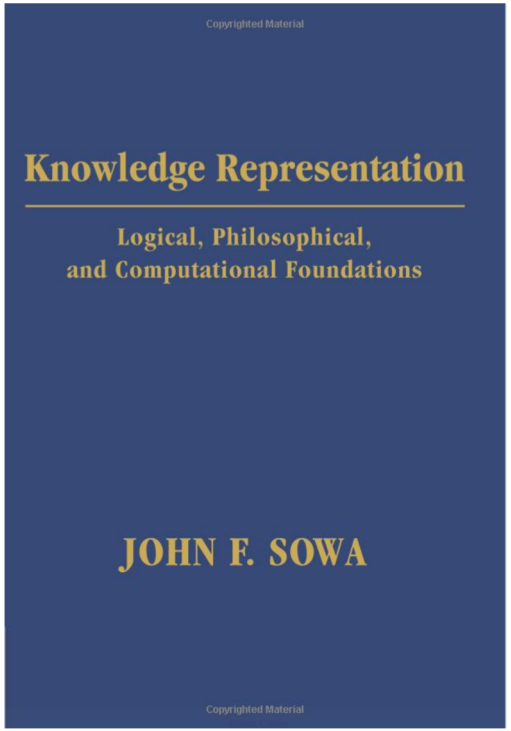
找到约 5,230,000 条结果 （用时**0.11**秒）

Knowledge representation: logical, philosophical, and computational foundations

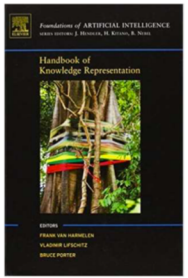
JF Sowa - 1999 - citeulike.org

Sowa integrates logic, philosophy, linguistics, and computer science into this study of knowledge and its various models and implementations. His definitive new book shows how techniques of artificial intelligence, database design, and object-oriented programming help …

☆ 99 被引用次数：3736 相关文章 所有 28 个版本 »



Roadmap of knowledge representation



Brooks *Subsumption*

Hayes&McCarthy *Frame Problem*

ConceptNet

Minsky, Filmore *Frames*

Quillian *Semantic Networks*

McCarthy *Formalizing Commonsense*

Description Logic

Bobrow *STUDENT*

Simon&Newell *General Problem Solver*

Lenant *Cyc*

Winograd *SHRDLU*

2000 1990 1980 1970 1960 1950 1940

SHRUTI

Series of Neural-Symbolic Models

Rumelhart et al *BackPropagation*

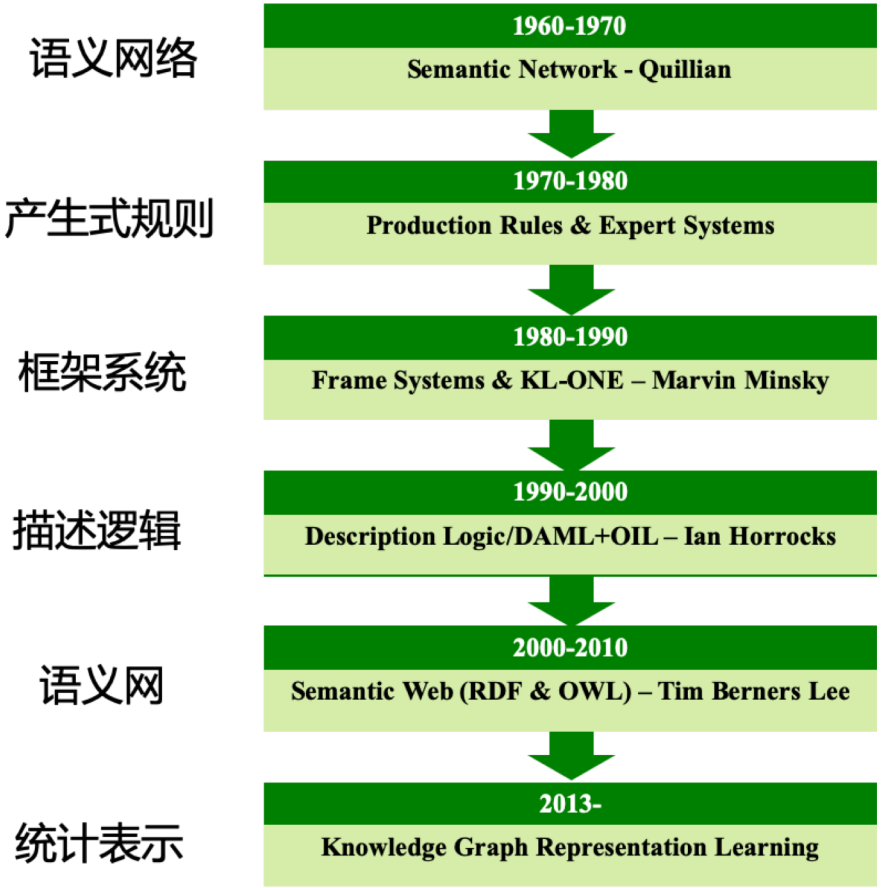
Systematicity Debate

Minsky &Pappert *"Perceptrons"*

McCulloch &Pitts *Artificial Neurons*

SLIDE COURTESY OF DANIEL KHASHABI

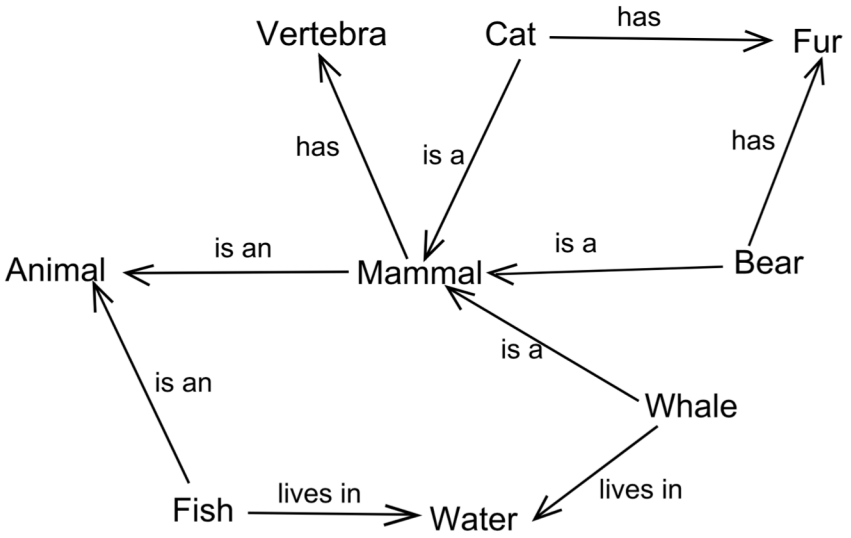
Roadmap of knowledge representation



Semantic network

```
(defun *database* ()
'((canary   (is-a bird)
            (color yellow)
            (size small))
  (penguin  (is-a bird)
            (movement swim))
  (bird     (is-a vertebrate)
            (has-part wings)
            (reproduction egg-laying))))
```

example: semantic network in LISP



example: semantic network in graph

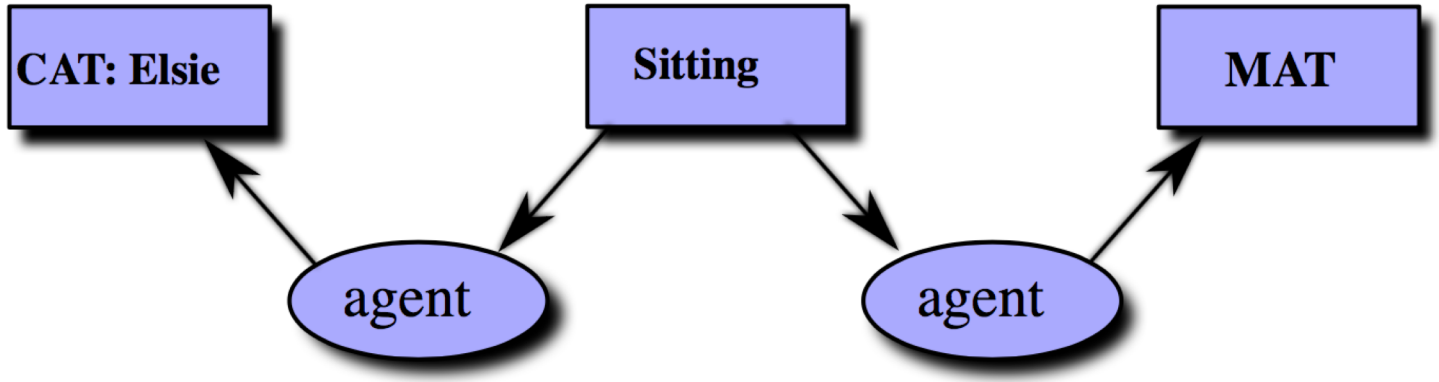
Frame Systems

Slot	Value	Type
MONKEY	_	(This Frame)
ISA	Primate	(parent frame)
SEX	OneOf(Male,Female)	(procedural attachment)
AGE	an integer	(procedural attachment - sets constraint)
HABITAT	Default = Jungle	(default)
FAVORITE_FOOD	Default = Bananas	(default)
CLIMBS	Trees	_
BODY_TYPE	Default = Wiry	(default)
NUM_LEGS	Default = 2	(default)

Marvin Minsky (1974)

Conceptual Graph

(exists ((x Sitting) (y Mat)) (and (Cat Elsie) (agent x Elsie) (location x y)))



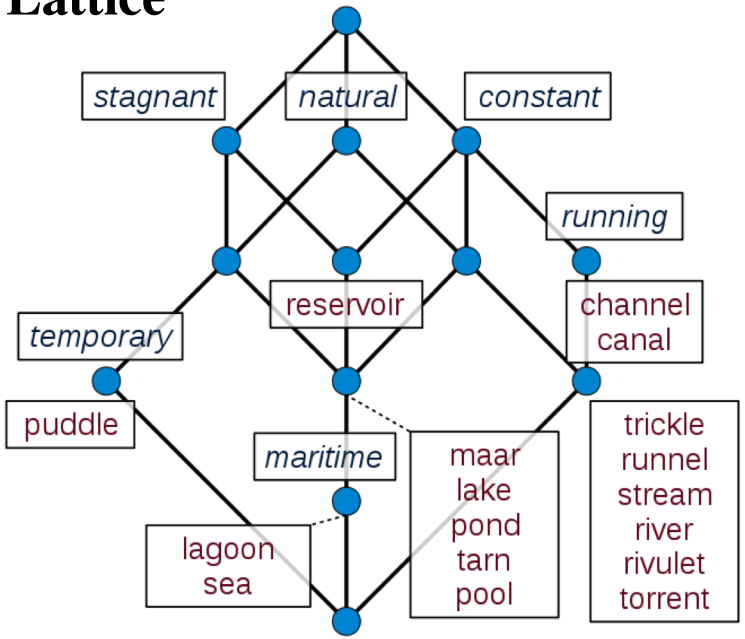
John F. Sowa（1976，1984）

Formal Concept Analysis

Example for a formal context: "bodies of water"

bodies of water	attributes					
objects	*temporary*	*running*	*natural*	*stagnant*	*constant*	*maritime*
canal		X			X	
channel		X			X	
lagoon			X	X	X	X
lake			X	X	X	
maar			X	X	X	
puddle	X		X	X		
pond			X	X	X	
pool			X	X	X	
reservoir				X	X	
river		X	X		X	
rivulet		X	X		X	
runnel		X	X		X	
sea			X	X	X	X
stream		X	X		X	
tarn			X	X	X	
torrent		X	X		X	
trickle		X	X		X	

Lattice



Rudolf Wille (1982)

Description Logic

DLs are more expressive than propositional logic but less expressive than first-order logic.

Decidable!!

定义 3.1：ALC 语法.令 N_C 和 N_R 是可数的不相交的原子概念集和原子关系集。ALC 的概念描述(简称概念)递归定义如下：

1)任意原子概念 $A \in N_C$ 是 ALC 概念；

2)令 C 和 D 是 ALC 概念，R 是 ALC 的原子关系，即 $R \in N_R$，则表达式 $\neg C$ (补)、$C \sqcup D$ (并)、$C \sqcap D$ (交)、$\exists R.C$ (存在约束)和 $\forall R.C$ (全称约束)是 ALC 概念。

Description Logic

定义 3.2：ALC 语义. ALC 解释是一个二元对 $I=(\Delta^I,\cdot^I)$，其中 Δ^I 是代表论域的非空集合，$\cdot^I$ 是解释函数，它将所有 $A\in N_C$ 映射为 Δ^I 的子集，所有 $R\in N_R$ 映射为 $\Delta^I\times\Delta^I$ 的子集，分别称为原子概念 A 和原子关系 R 的解释。对于定义 3.1 中由构造子构成的概念，解释函数 $\cdot^I$ 定义它们的解释为：

$(\neg C)^I$	=	Δ^I / C^I
$(C\sqcap D)^I$	=	$C^I\bigcap D^I$
$(C\sqcup D)^I$	=	$C^I\bigcup D^I$
$(\forall R.C)^I$	=	$\{a\in\Delta^I \mid \forall b((a,b)\in R^I\rightarrow b\in C^I)\}$
$(\exists R.C)^I$	=	$\{a\in\Delta^I \mid \exists b,(a,b)\in R^I\wedge b\in C^I\}$

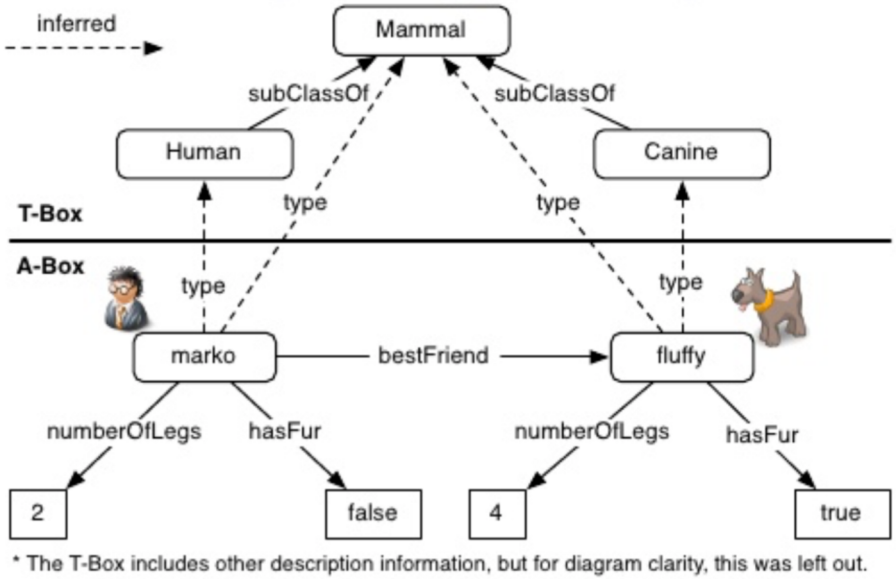
Description Logic

TBox [edit]

- $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I} \models \mathcal{T}$ if and only if $\mathcal{I} \models \Phi$ for every $\Phi \in \mathcal{T}$

ABox [edit]

- $\mathcal{I} \models a : C$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \models (a, b) : R$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
- $\mathcal{I} \models \mathcal{A}$ if and only if $\mathcal{I} \models \phi$ for every $\phi \in \mathcal{A}$



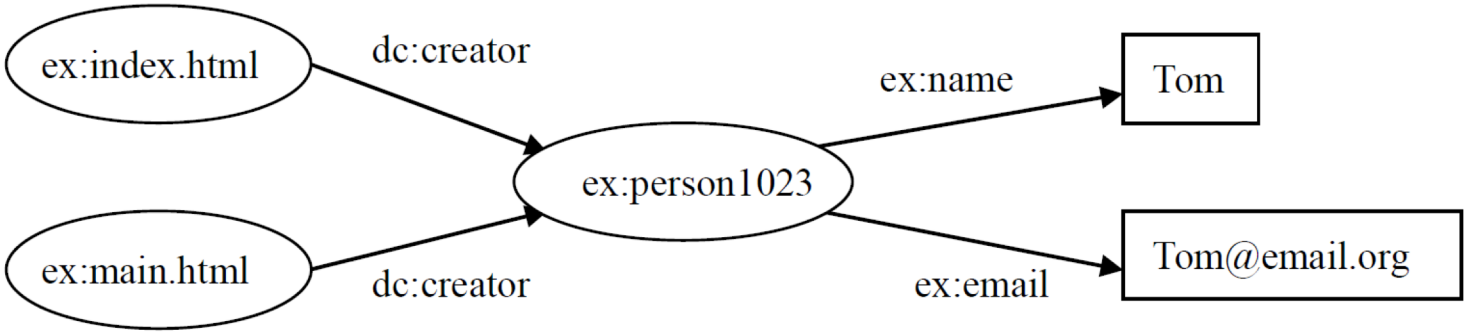
* The T-Box includes other description information, but for diagram clarity, this was left out.

Resource Description Framework

- **RDF model**

  (Resource)：URI标识的所有事物
  (Literal)：字符串或数据类型的值
  (Property)：描述资源特征、属性、或关系
  (Statement)：一个资源加上属性及属性值

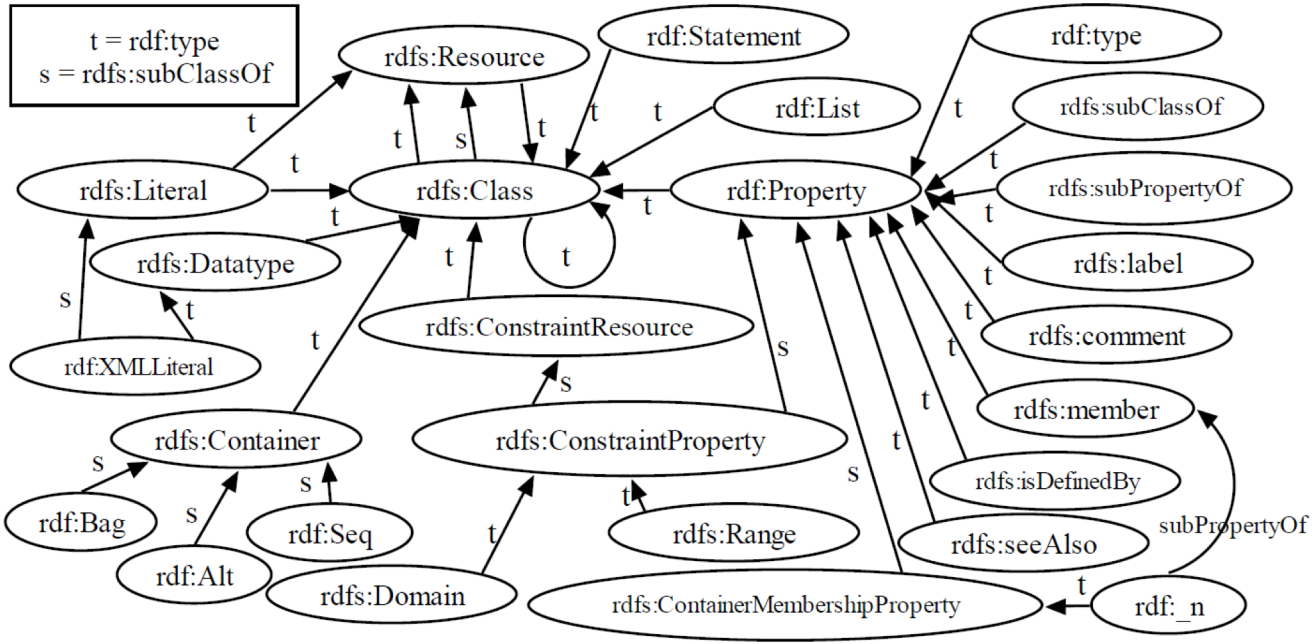
RDF statements: triples



Resource Description Framework

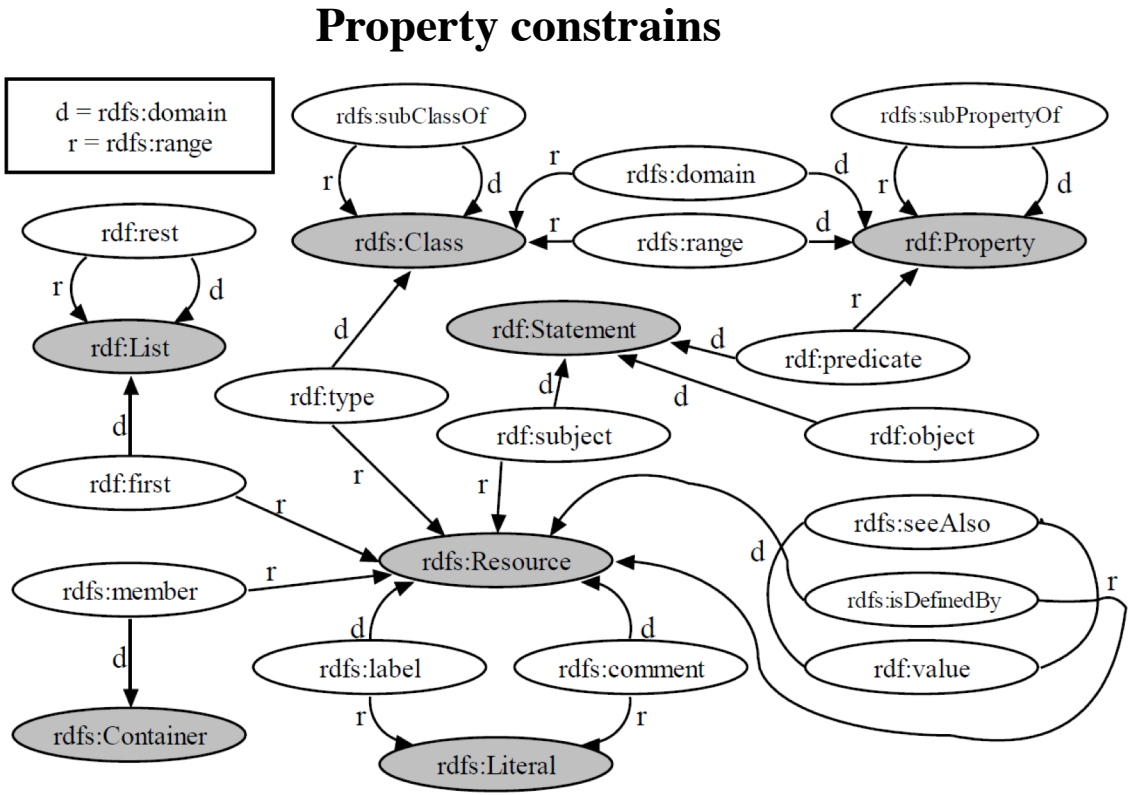
- **RDF schema**

Class and Property



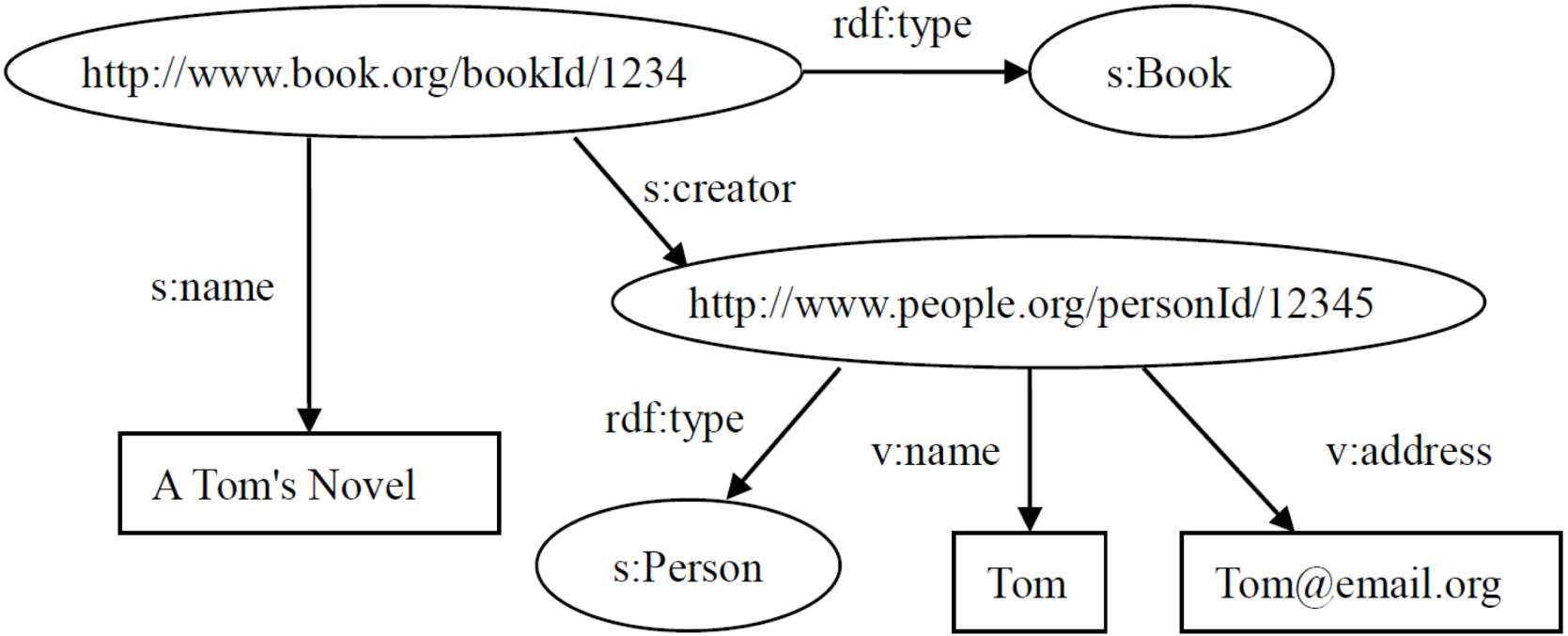
Resource Description Framework

- **RDF schema**



Resource Description Framework

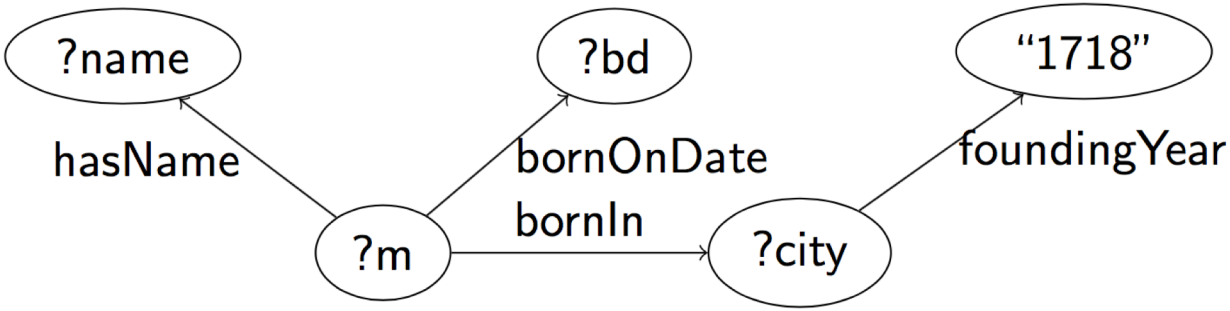
A RDF example:



SPARQL

```
SELECT ?name
WHERE {
   ?m <bornIn> ?city. ?m <hasName> ?name.
   ?m<bornOnDate> ?bd. ?city <foundingYear> ''1718''.
   FILTER(regex(str(?bd),''1976''))
}
```

FILTER(*regex*(*str*(?bd),"1976"))



JSON-LD ()

JOSN

```
{
"name": "Manu Sporny",
"homepage": "http://manu.sporny.org/",
"image": "http://manu.sporny.org/images/manu.png"
 }
```

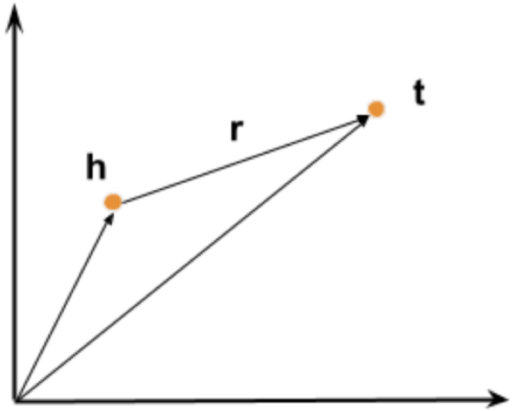
JOSN-LD

```
{
"http://schema.org/name": "Manu Sporny",
"http://schema.org/url": { "@id":"http://manu.sporny.org/" }, "http://schema.org/image":
 { "@id":"http://manu.sporny.org/images/manu.png" }
}
```

KG Embedding (Representation Learning)

- 将三元组<h,r,t>看成h通过r翻译到t的过程

Beijing-China = Pairs-France = Caption-of

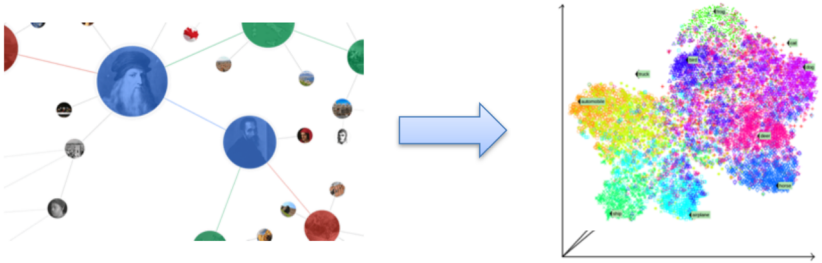


- 优化目标
  - 势能函数

$$f(h,r,t) = ||h + r - t||_2$$

  - 目标函数

$$min \sum_{(h,r,t)\in\Delta} \sum_{(h',r',t')\in\Delta'} [\gamma + f(h,r,t) - f(h',r',t')]_+$$

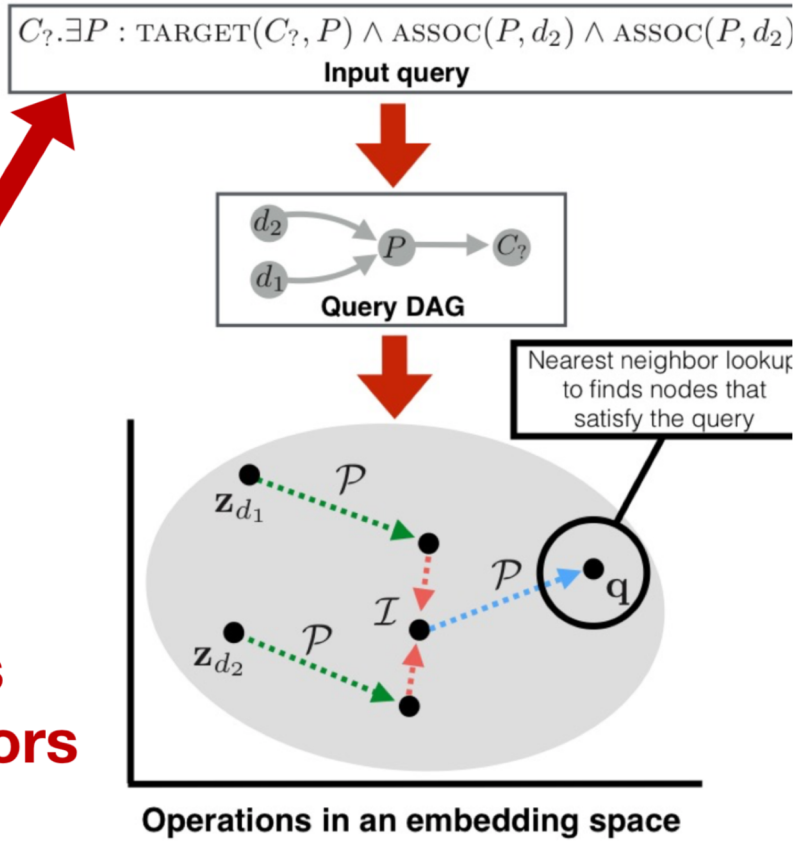


KG Embedding (Representation Learning)

Goal: Answer logical queries

E.g.: "**Predict drugs C** likely *target* proteins X *associated* with diseases d_1 and d_2"

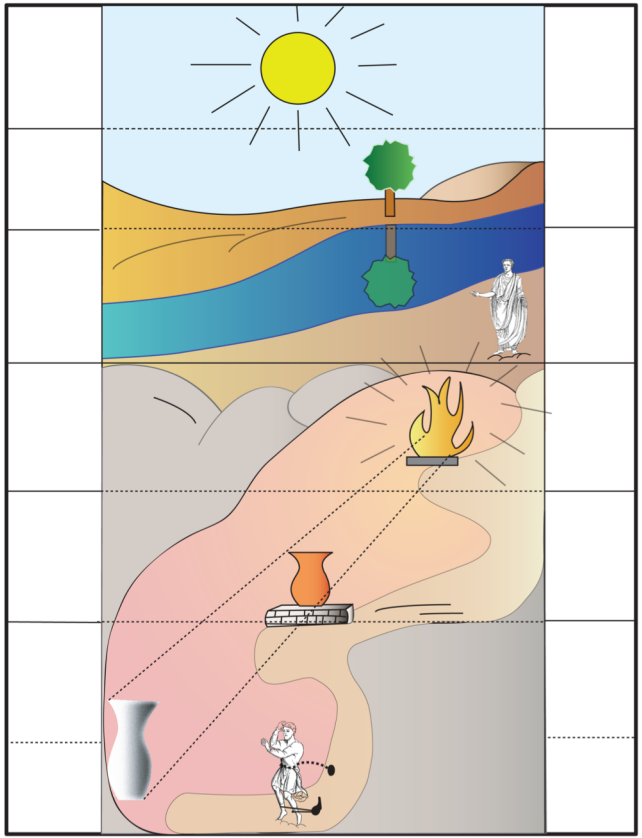
Idea: Logical operators become spatial operators



Embedding Logical Queries on Knowledge Graphs. NeurIPS, 2018.

Ontology

Plato's Cave



本体意味着存在一个外在的完美的知识体系，不依赖人类的认识而存在。人类探索知识的过程就是不断从现实世界的现象中，摸索、推测这个完美知识体系的过程。人类的认识注定只能不断接近，却永远达不到这个完美的知识体系。

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

Ontology

In ***computer science and information science***,

an ontology is ***a formal representation of knowledge*** as ***a set of concepts within a domain***, and the ***relationships between those concepts.***

It is used to ***reason about the entities*** within that domain, and may be used to ***describe the domain***.

In theory, ***an ontology is a "formal, explicit specification of a shared conceptualisation".***

——T. R. Gruber(1993)

Link: http://en.wikipedia.org/wiki/Ontology_(information_science)

Ontology

Ontologies are the structural frameworks for organizing information and are used in ***artificial intelligence***, the ***Semantic Web***, ***systems engineering***, ***software engineering***, ***biomedical informatics***, ***library science***, ***enterprise bookmarking***, and ***information architecture as a form of knowledge representation*** about the world or some part of it.

Link: http://en.wikipedia.org/wiki/Ontology_(information_science)

Ontology

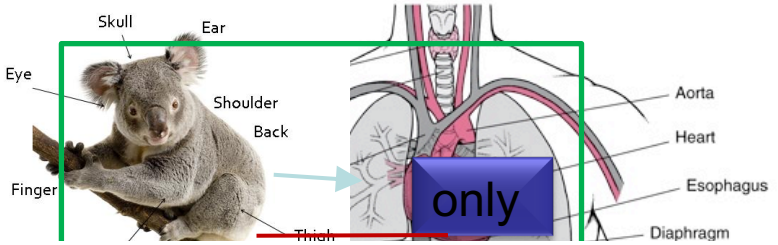
A model of (some aspect of) the world

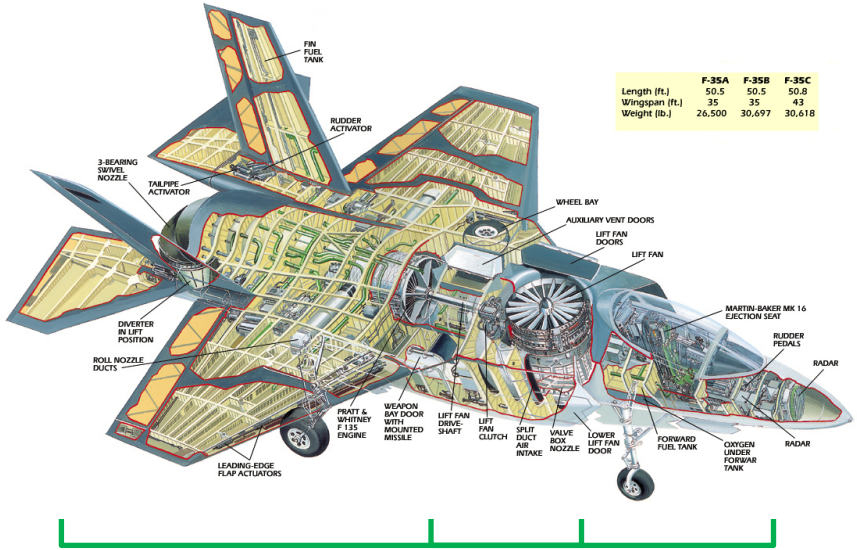
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Aerospace
  - Koala
- Introduces vocabulary
  Specifies meaning (semantics)

$$Koala \sqsubseteq \forall eat.(\exists partof.Eucalypt)$$

  - Koala eat only some part of Eucalypt
  - Eucalypt is Plant

$$Eucalypt \sqsubseteq Plant$$





Ontology Engineering

Ontology engineering in computer science, information science and systems engineering is a field which studies the methods and methodologies for building ontologies: formal representations of a set of concepts within a domain and the relationships between those concepts.

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

知识图谱中需要一个本体来形式化描述和界定它所描述的知识和事实的范围。

本体工程是用工程化规范保证本体质量的方法学。

Ontology Engineering

[PDF] Ontology development 101: A guide to creating your first ontology

NF Noy, DL McGuinness - 2001 - corais.org

In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have …

☆ 99 被引用次数：5513 相关文章 所有 78 个版本 »

Ontology Development 101: A Guide to Creating Your First Ontology

Natalya F. Noy and Deborah L. McGuinness
Stanford University, Stanford, CA, 94305
noy@smi.stanford.edu and dlm@ksl.stanford.edu

https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Ontology Engineering

- **Step1：Determine the domain and scope of the ontology**
  确定本体的领域和范围

这是本体开发的第一步。我们必须首先明确一些基本的问题，诸如:该本体针对什么领 域?用途是什么?要描述什么信息?回答哪一类的问题?谁将使用和维护这个本体?等等。这些问题通常可以借助能力咨询的方法来获得[Gruninger1995]。需要注意的是，随着开发的 进行，这些问题和它的回答可能会发生变化，这时需要考虑什么时候再回到第一步进行迭代开发。

Ontology Engineering

- **Step2：Consider reusing existing ontologies**
  考虑重用现有本体

通常，收集和待开发本体相关的其它本体是有价值的。一方面如果可以精炼、扩充、或修改现有的本体，那么则可以避免很多不必要的开发工作。另一方面即使现有的本体无法满 足当前的应用要求，通常也会从其中得到一些启发和帮助。目前网络上已有一些本体库，从中可以获得很多现有的本体。

Ontology Engineering

- **Step3：Enumerate important terms in the ontology**
  列出本体中的重要术语

将所关心的术语列举出来是非常有用的。这些术语大致表明建模过程所感兴趣的事物、物所具有的属性和它们间的关系等。这些重要术语能保证最终创建的本体不会偏离所感兴 趣的领域。

Ontology Engineering

- **Step4：Define the classes and the class hierarchy**
  定义类和类的继承
  - 确保类的继承正确
  - 分析继承结构中的兄弟类
  - 引入新类的时机
  - 新类或属性值的取舍
  - 实例或类的取舍
  - 范围限制
  - 不相交的子类

类的继承结构的定义可以采用自顶向下的方法，即从最大的概念开始，然后通过添加子 类细化这些概念;也可以采用自底向上的方法，即由最底层、最细的类定义开始，然后找到 它们的父类;当然，也可以采用这两种方法的综合进行定义。

Ontology Engineering

- **Step5：Define the properties of classes—slots**
  定义属性和关系
- **逆属性/关系**
- **缺省属性值**

仅有类不能对很多问题给出回答，通常一旦定义了类，还需要定义概念和概念间的内部联系。这里所指的联系可分为两种:一种是概念自身的属性，称为“内在属性”，如概念“Wine”的味道这种属性可用术语“Flavor”表示，这一类属性通常连接一个概念和一个值，本体语言OWL中，这种属性被表示为DatatypeProperty。内在属性具有通用性，也就是说该类对应的所有实例都具有这种属性，并且这种属性通常能向下传递，即如果一个类 具有一个内在属性，那么它的所有子类都继承了这种属性。这样也就要求在属性建模的过程中，一个属性应该为拥有该属性的最大类所拥有。另一类属性称为“外在属性”，也有的文 献直接称之为关系，通常用于连接概念间的实例，如概念“Worker”的一个外在属性“Workfor”连接了概念“Company”，表明对于一对分别来自这两个概念的实例来说，可能会存在 “Workfor”这个关系。“外在属性”在随后的 OWL 语言中表示为 ObjectProperty。

Ontology Engineering

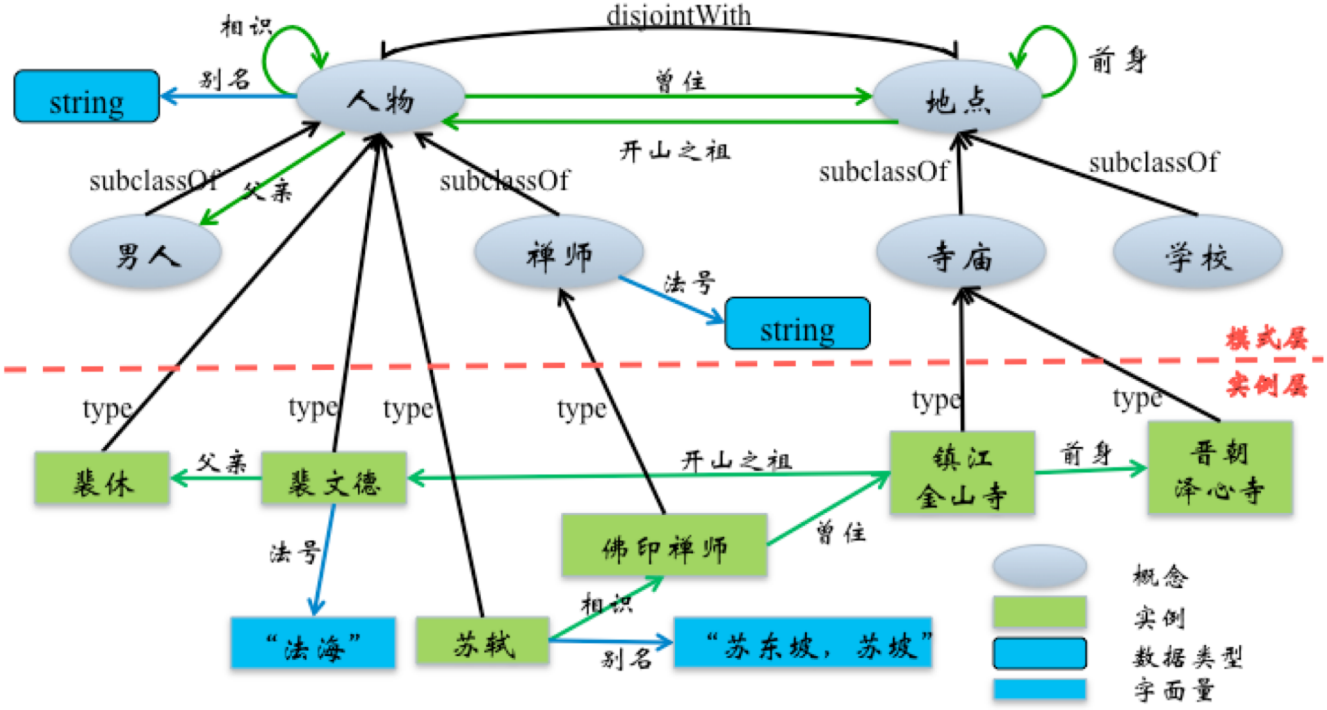
- **Step6：Define the facets of the slots**
  定义属性的限制

在这一步中，需要进一步定义属性的一些限制，包括属性的基数、属性值的类型，以及属性的定义域和值域。

- **Step7： Create instances**
  创建实例

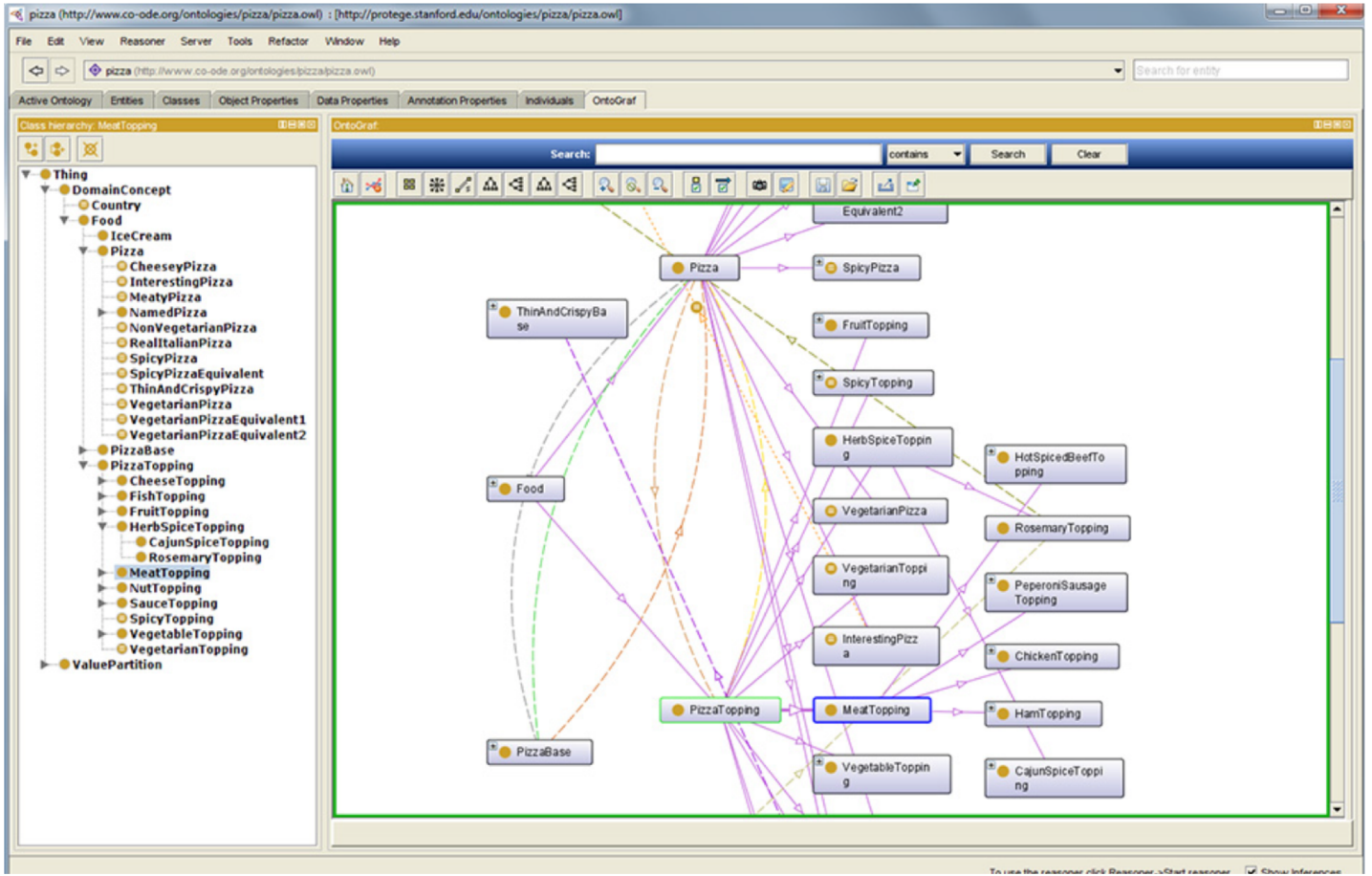
最后，还需要为类创建实例。这需要确定与个体最接近的类，然后添加个体进去作为该 类的一个实例，同时要为实例的属性赋值。

Ontology Building with Protégé

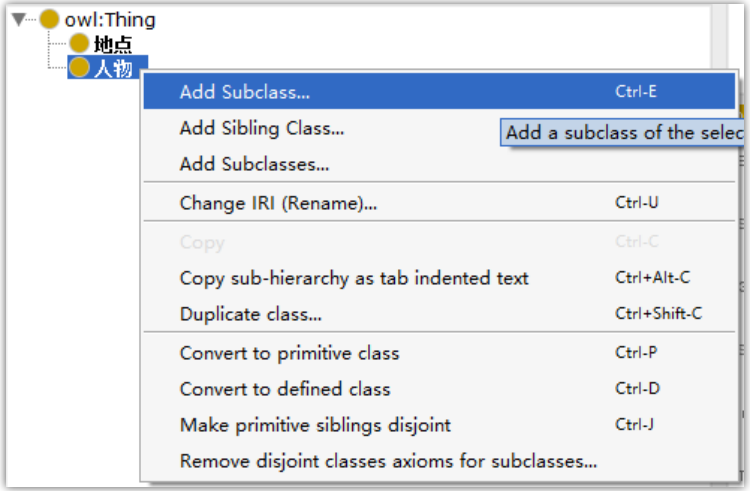


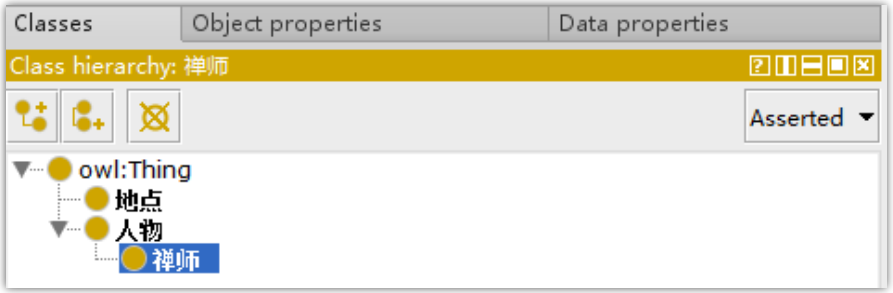
—本体引用自王昊奋博士

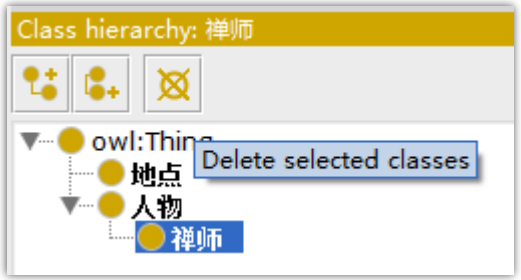
Ontology Building with Protégé



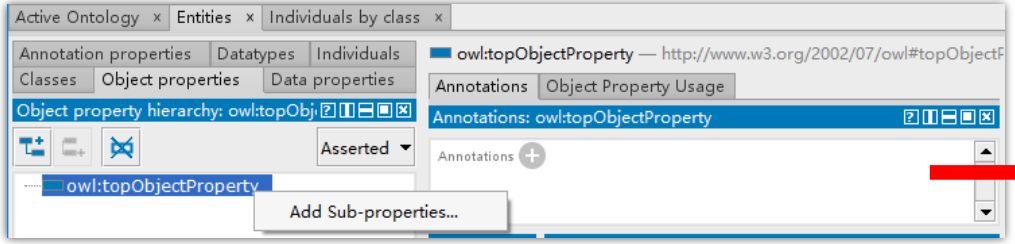
Ontology Building with Protégé

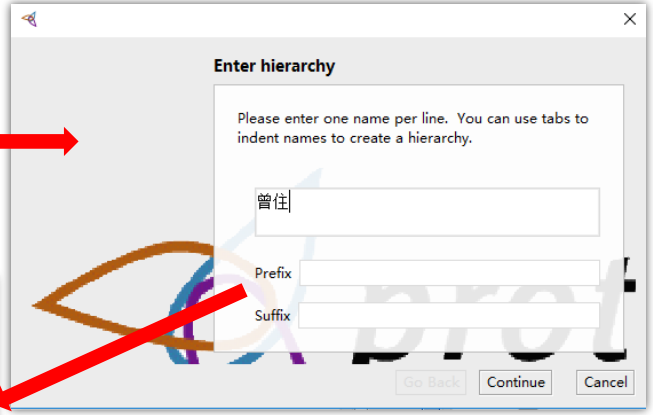


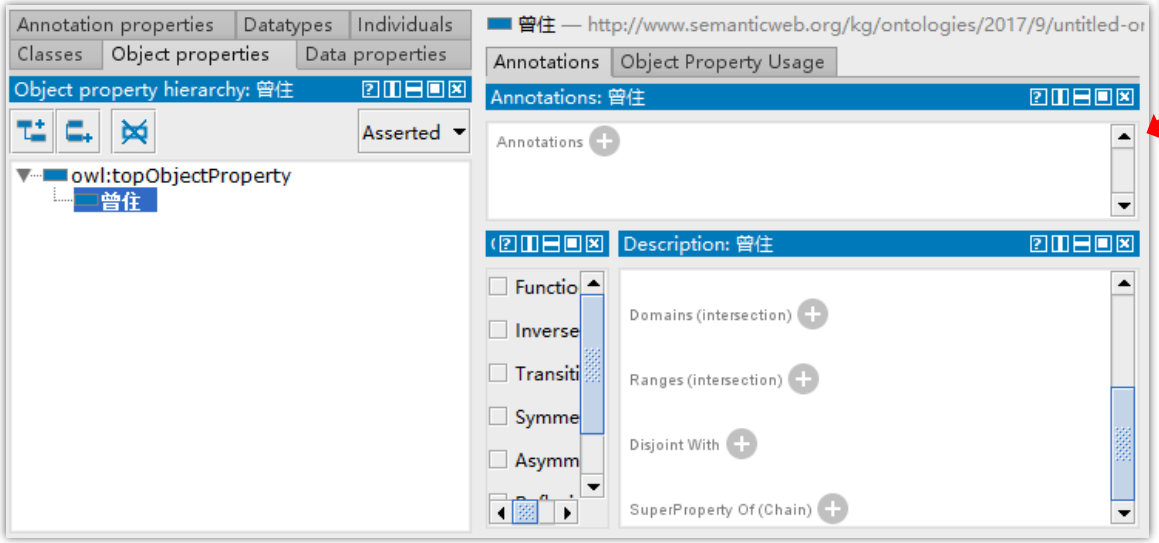




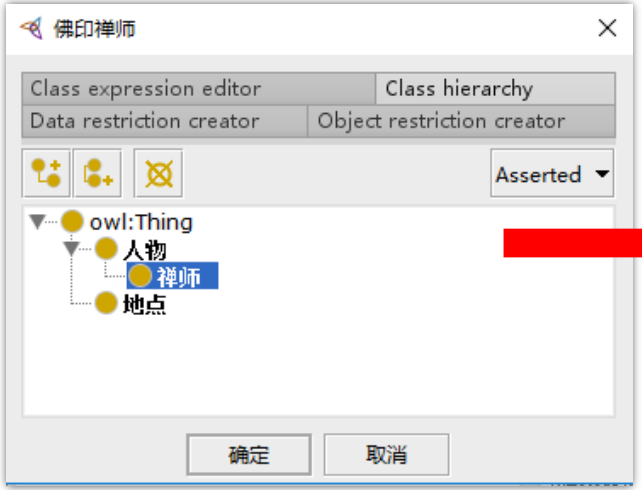
Ontology Building with Protégé

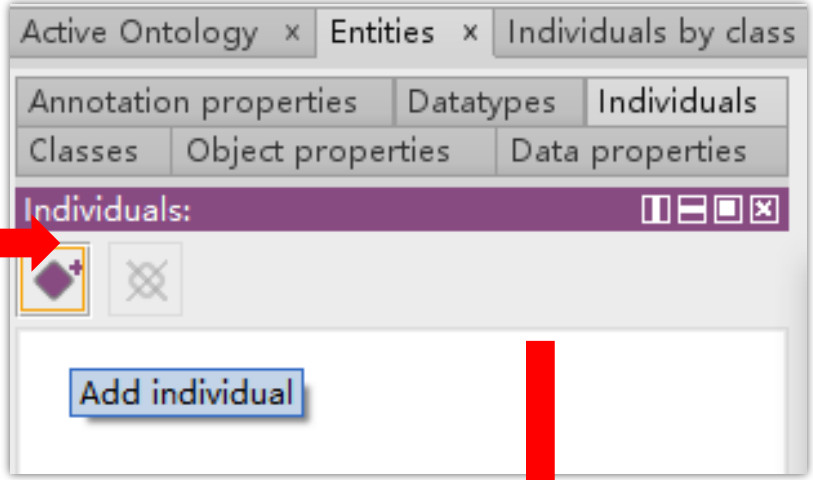


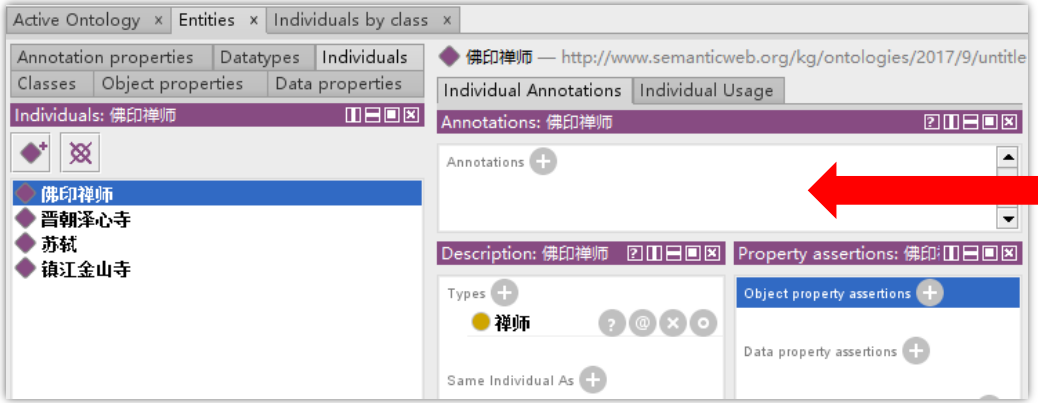


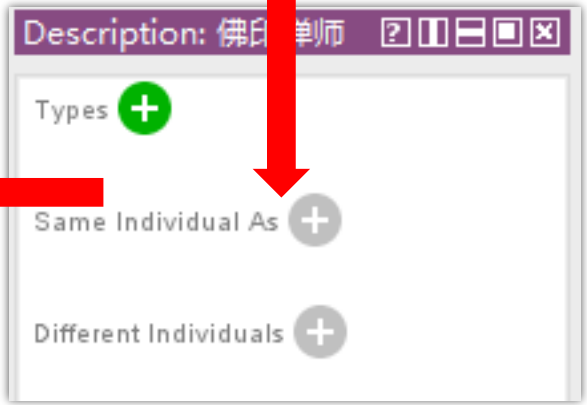


Ontology Building with Protégé









Outline

- 1. Knowledge Representation and Modeling
- 2. Knowledge Extracting
- 3. Knowledge Fusion
- 4. Knowledge Storage

From IE to KE

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents.

—wikipedia

Knowledge extraction (KE) is the creation of knowledge from structured (relational databases, XML) and unstructured (text, documents, images) sources. The resulting knowledge needs to be in a machine-readable and machine-interpretable format and must represent knowledge in a manner that facilitates inferencing.

—wikipedia

John was born in Liverpool, to Julia and Alfred Lennon.

Text

NLP

Lennon..
John Lennon...

the Pool

Mrs. Lennon..
.. his mother ..

his father
he
Alfred

Person Location Person Person

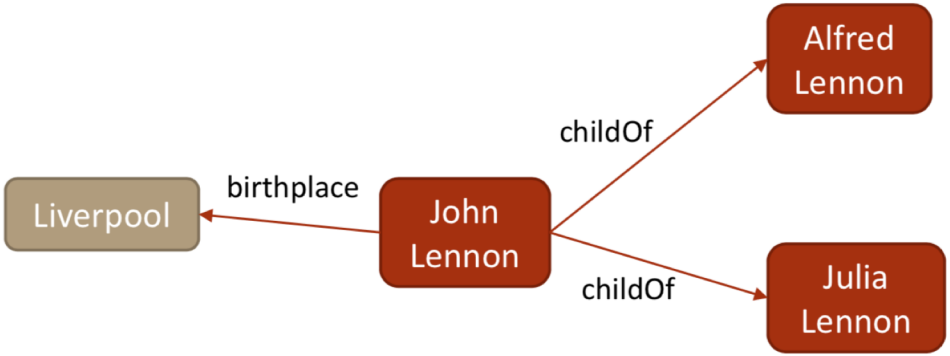
John was born in Liverpool, to Julia and Alfred Lennon.

NNP VBD VBD IN NNP TO NNP CC NNP NNP

Annotated text

Information Extraction

Extraction graph



KE challenge: ambiguous

Beetles， beetles， Beatles

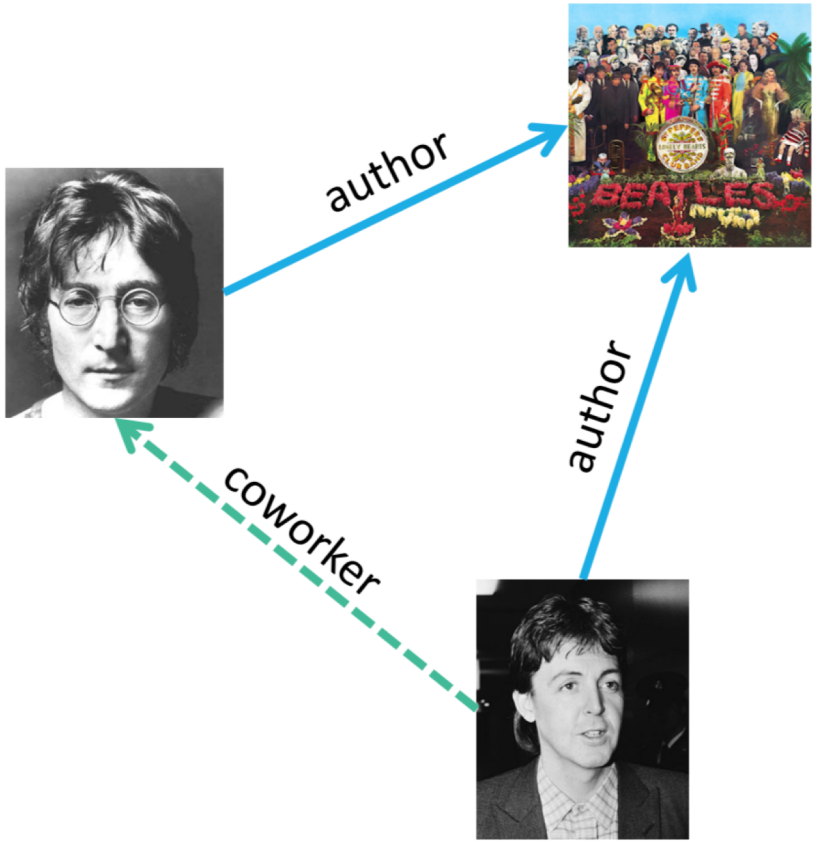
citizenOf, livedIn, bornIn

KE challenge: incomplete

missing relation

missing label

missing entity



KE challenge: inconsistent

Cynthia Lennon, Yoko Ono

exclusive labels (alive, dead)

domain-range constraints

spouse

spouse

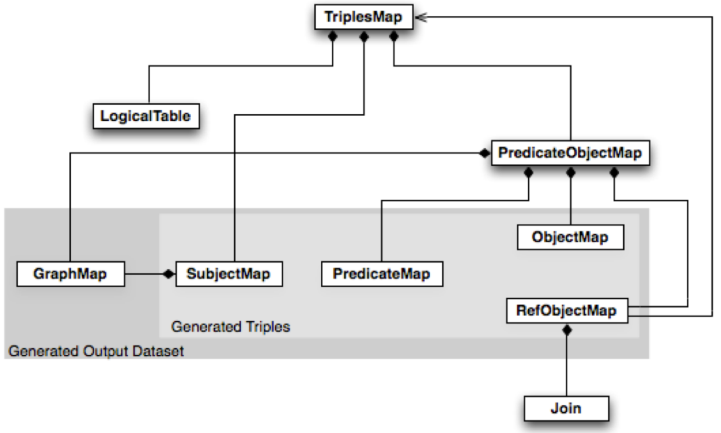
KE from relational database

- **Principle**
  - **Table—Class**
  - **Column—Property**
  - **Row—Resource/Instance**
  - **Cell—Property Value**
  - **Foreign Key—Reference**

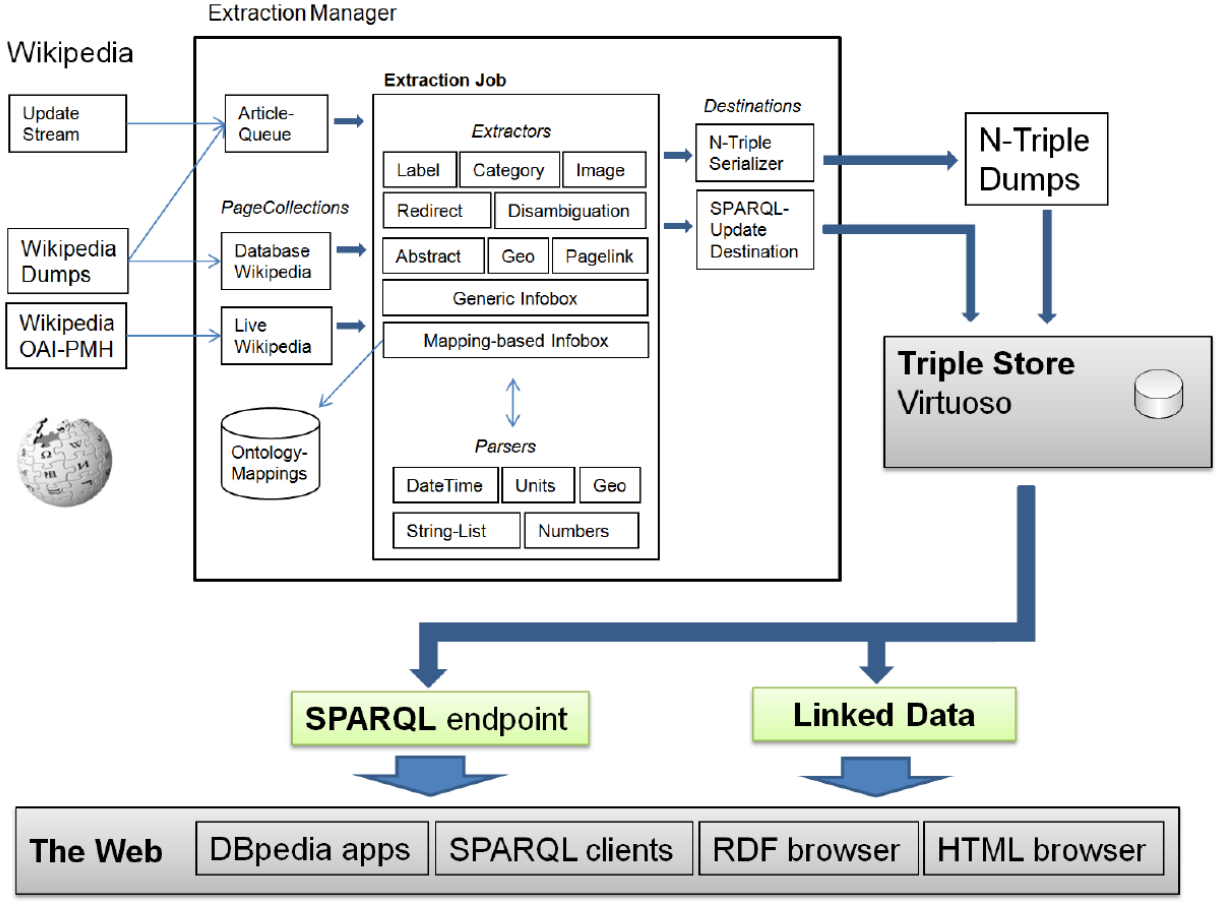
transform a relational database to a knowledge graph based on rules

KE from relational database

- Standard
  - Direct Mapping
  - R2RML
- Tools
  - D2R, Virtuoso, Orcle SW, Morph, …

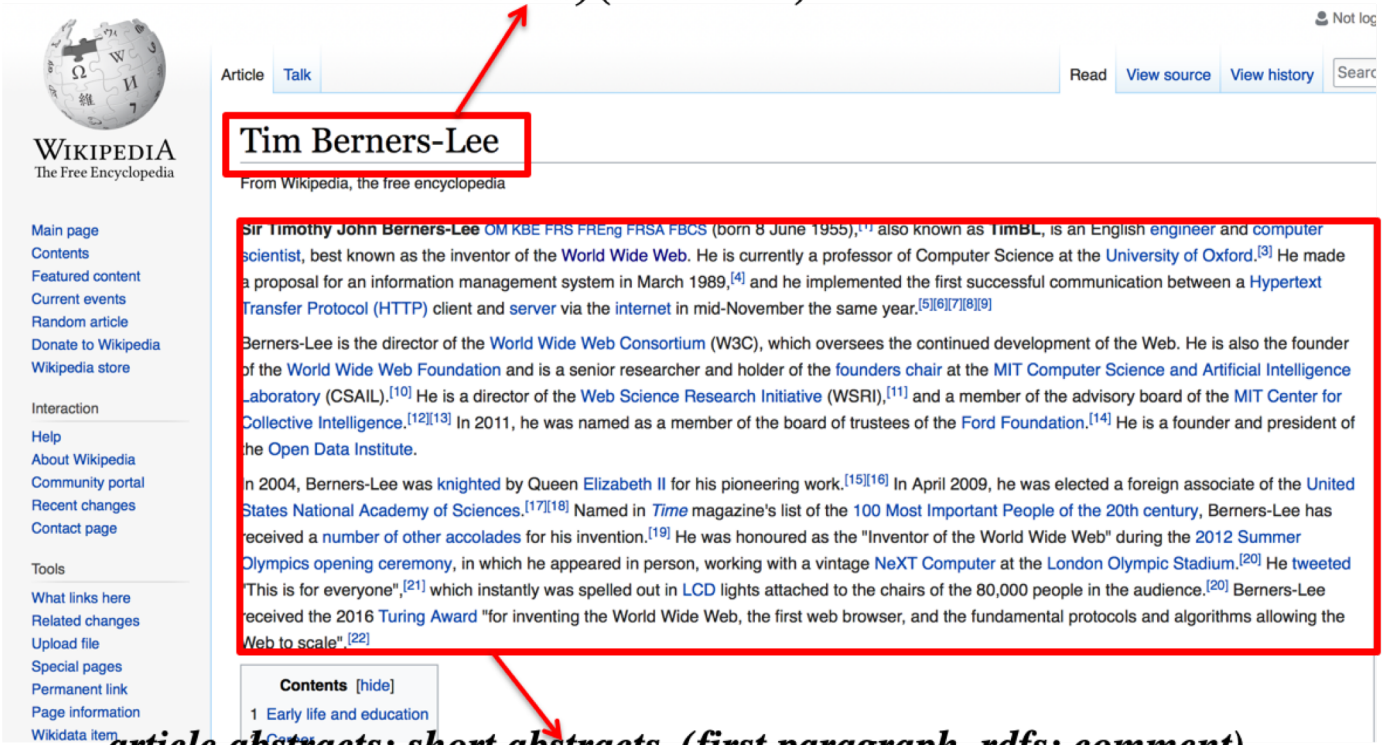


KE from encyclopedic knowledge



KE from encyclopedic knowledge

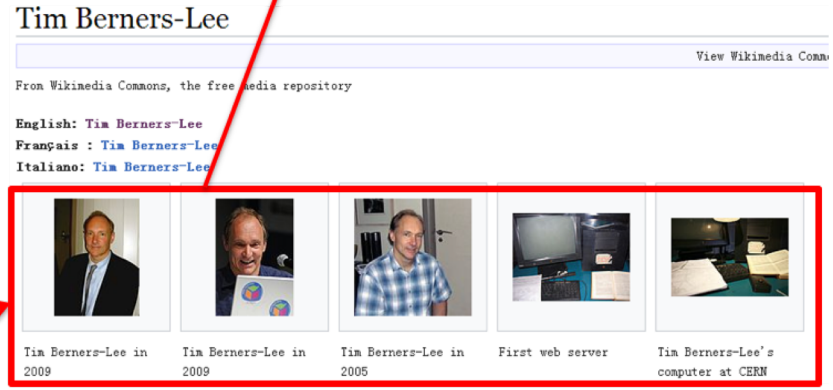
article titles=labels, **(rdfs: label)**



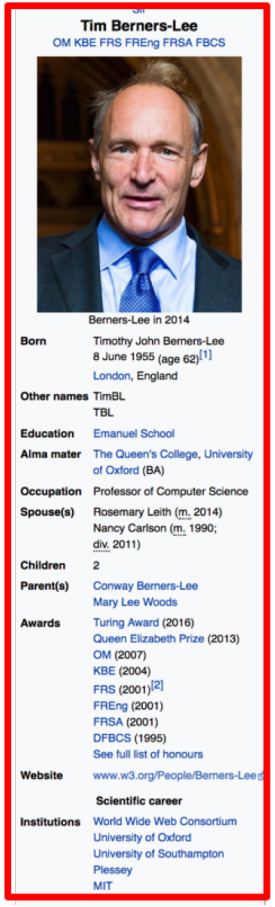
article abstracts: short abstracts (first paragraph, rdfs: comment)
long abstracts (dbpedia: abstract)

KE from encyclopedic knowledge

Wikipedia Commons Images=images,
(foaf: depiction)



KE from encyclopedic knowledge



***infobox template names=instance types*, (rdf: type)**

```
{{pp-move-indef}}
{{pp-semi-vandalism|small=yes}}
{Infobox person
honorific_suffix = {{postnominals|country=GBR|OM|KBE|FRS|FREng|FRSA|FBCS}}
image = Sir Tim Berners-Lee (cropped).jpg
image_size = 220px
caption = Berners-Lee in 2014
alt = blond man in his fifties wearing a blue suit, light blue shirt, and blue
birth_name = Timothy John Berners-Lee
birth_date = {{birth date and age|1955|6|8|df=y}}<ref name="whoswho"/>
birth_place = [[London]], England, UK
education = [[Emanuel School]]
alma_mater = [[The Queen's College, Oxford]] (BA)
awards = {{Plainlist|
[[Turing Award]] (2016)
[[Queen Elizabeth Prize]] (2013)
[[Member of the Order of Merit|OM]] (2007)
[[Knight Commander of the Order of the British Empire|KBE]] (2004)
[[Fellow of the Royal Society|FRS]] (2001)<ref name=frs/>
[[Fellow of the Royal Academy of Engineering|FREng]] (2001)
[[Fellow of the Royal Society of Arts|FRSA]] (2001)
[[Distinguished Fellow of the British Computer Society|DFBCS]] (1995)
[[Awards and honours presented to Tim Berners-Lee|See full list of honours]]}}
spouse = {{Plainlist|
```

***infobox properties=instance properties*, (dbpedia:property/[propertyName])**

KE from encyclopedic knowledge

page links=internal links, (dbpedia: wikilink)

Tim Berners-Lee

From Wikipedia, the free encyclopedia

Sir Timothy John Berners-Lee OM KBE FRS FREng FRSA FBCS (born 8 June 1955),[1] also known as **TimBL**, is an English engineer and computer scientist, best known as the inventor of the World Wide Web. He is currently a professor of Computer Science at the University of Oxford.[3] He made a proposal for an information management system in March 1989,[4] and he implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the internet in mid-November the same year.[5][6][7][8][9]

Berners-Lee is the director of the World Wide Web Consortium (W3C), which oversees the continued development of the Web. He is also the founder of the World Wide Web Foundation and is a senior researcher and holder of the founders chair at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL).[10] He is a director of the Web Science Research Initiative (WSRI),[11] and a member of the advisory board of the MIT Center for Collective Intelligence.[12][13] In 2011, he was named as a member of the board of trustees of the Ford Foundation.[14] He is a founder and president of the Open Data Institute.

Categories: 1955 births | Living people | People from Barnes, London | People educated at Emanuel School | Alumni of The Queen's College, Oxford | Academics of the University of Southampton | Fellows of Christ Church, Oxford | Members of the Department of Computer Science, University of Oxford | People associated with CERN | English computer scientists | English expatriates in the United States | English inventors | English Unitarians | Fellows of the American Academy of Arts and Sciences | Fellows of the British Computer Society | Fellows of The Queen's College, Oxford | Fellows of the Royal Academy of Engineering | Fellows of the Royal Society | Hypertext Transfer Protocol | Internet pioneers | Knights Commander of the Order of the British Empire | MacArthur Fellows | Massachusetts Institute of Technology staff | Members of the Order of Merit | Members of the United States National Academy of Engineering | Turing Award laureates | Members of the United States National Academy of Sciences | Royal Medal winners | Semantic Web people | UNESCO Niels Bohr Medal recipients | Unitarian Universalists | Former Anglicans | Webby Award winners | World Wide Web Consortium | Freemen of the City of London

article categories=topical concepts, (http://purl.org/dc/terms/subject)

KE from encyclopedic knowledge

KE from encyclopedic knowledge

WIKIPEDIA
The Free Encyclopedia

Article Talk Read Edit Vie

Einstein (disambiguation)

From Wikipedia, the free encyclopedia

Albert Einstein (1879–1955) was a German-born theoretical physicist.

Einstein may also refer to:

- Einstein (surname), a surname (including a list of people with the name)

Science and technology [edit]

- 2001 Einstein, a main belt asteroid
- Einstein (crater), a large lunar crater along the Moon's west limb
- Einstein Observatory, the first fully imaging X-ray telescope put into orbit
- Einstein (US-CERT program), an intrusion detection program used by the United States Department of
- Einstein Telescope, a gravitational wave detector under design
- Einstein Tower, an astrophysical observatory in Potsdam, Germany
- Einstein (unit), a physical unit defined as one mole of photons
- Tatung Einstein, a personal computer produced by Taiwanese corporation Tatung

Fictional characters [edit]

- Einstein (Farscape), a character in the science fiction television series *Farscape*
- Einstein (dog), canine pet of Doc Brown in the *Back to the Future* movie trilogy
- Albert Einstein, a character in the television series *Alien Nation*
- Einstein, a Great Dane in the 1988 film *Oliver & Company*
- Nina Einstein, a character from the anime series *Code Geass*
- Agent Einstein, a character in the tenth season of the television series *The X-Files*

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export

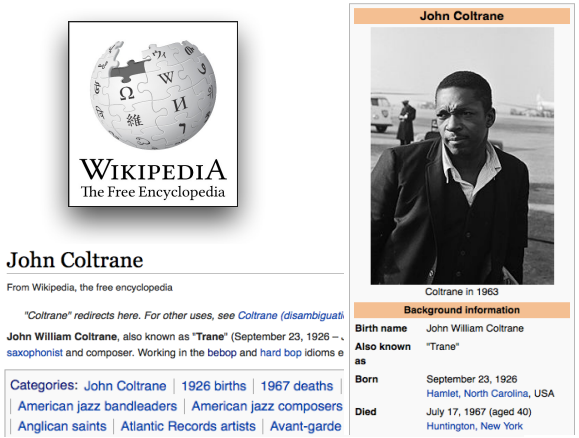
Create a book

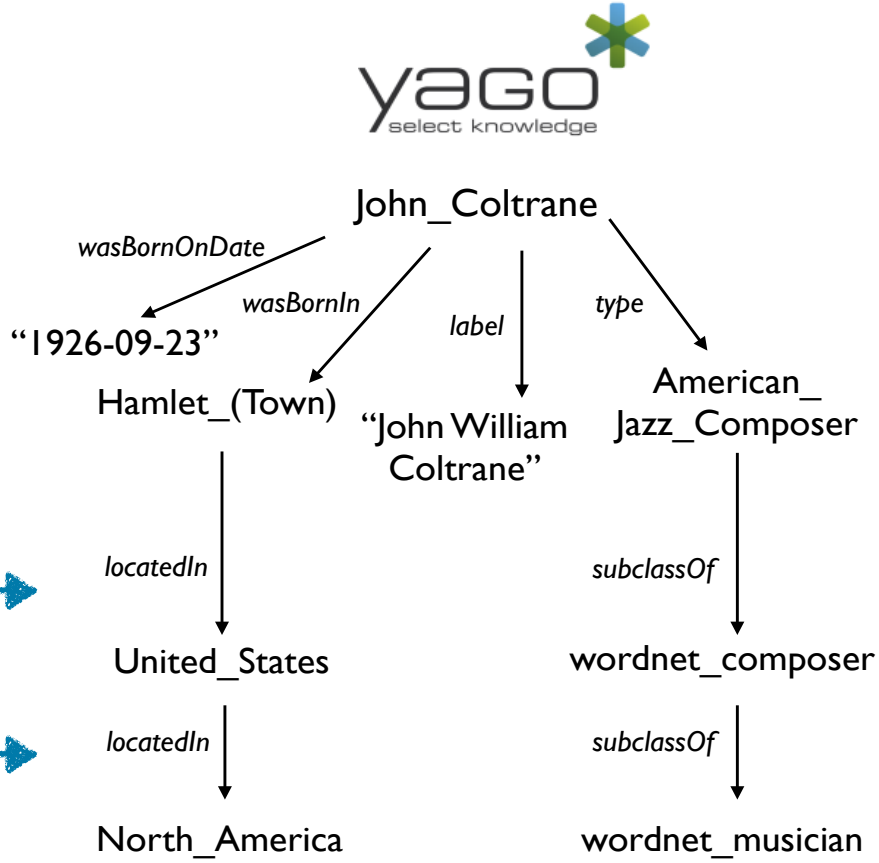
***homonymies=ambiguous instances*, (dbpedia: disambiguates)**

KE from encyclopedic knowledge

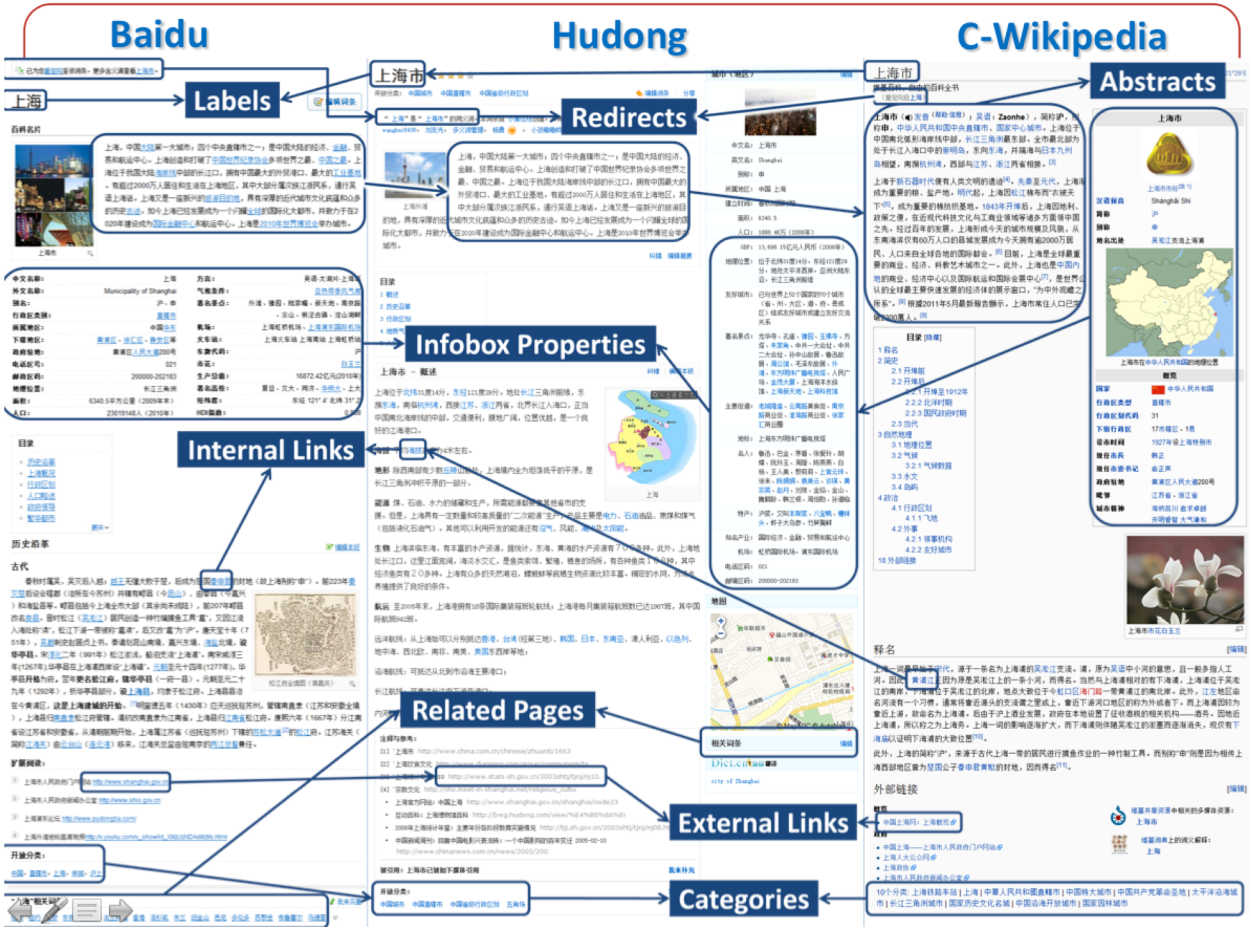
synonyms=redirects, (owl:sameAs)

YAGO


WIKIPEDIA
The Free Encyclopedia
John Coltrane
From Wikipedia, the free encyclopedia
"Coltrane" redirects here. For other uses, see Coltrane (disambiguati
John William Coltrane, also known as "Trane" (September 23, 1926 –
saxophonist and composer. Working in the bebop and hard bop idioms e
Categories: John Coltrane | 1926 births | 1967 deaths |
American jazz bandleaders | American jazz composers
Anglican saints | Atlantic Records artists | Avant-garde
John Coltrane
Coltrane in 1963
Background information
Birth name John William Coltrane
Also known as "Trane"
Born September 23, 1926
Hamlet, North Carolina, USA
Died July 17, 1967 (aged 40)
Huntington, New York


yago
select knowledge
John_Coltrane
wasBornOnDate
wasBornIn
label
type
"1926-09-23"
Hamlet_(Town)
"John William Coltrane"
American_Jazz_Composer
GeoNames
locatedIn
subclassOf
United_States
wordnet_composer
WordNet
A lexical database for English
locatedIn
subclassOf
North_America
wordnet_musician

ZHISHI.ME


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

KE from unstructural data

- **Problems**
  - ✓ **Named entity recognition**
  - ✓ **Relation extraction**
  - ✓ **Event extraction**

Name Entity Recognition

- Person(人名)
- Organization(组织/机构)
- Place(地理位置)
- Time(时间/日期)
- ……

Time
时间

Person
人名

北京时间3月23日0时50分许，美国总统特朗普在白宫正式签署对华贸易备忘录。特朗普当场宣布，将有可能对600亿美元中国出口商品征收关税。

Place
地理位置

Name Entity Recognition

- **Machine Learning Models**
  - **Hidden Markov Model, HMM（隐马尔科夫模型）**
  - **Maximum Entropy Markov Model, MEMM （最大熵马尔科夫模型）**
  - **Conditional Random Fields, CRF（条件随机场）**
  - **Support Vector Machine, SVM（支持向量机）**

Name Entity Recognition

- **Labels**

Token	IO	BIO	BIOES	BMEWO
特	I-PER	B-PER	B-PER	B-PER
朗	I-PER	I-PER	I-PER	M-PER
普	I-PER	I-PER	E-PER	E-PER
在	O	O	O	O
白	I-LOC	B-LOC	B-LOC	B-LOC
宫	I-LOC	I-LOC	E-LOC	E-LOC
签	O	O	O	O
署	O	O	O	O

Name Entity Recognition

- **Labels**

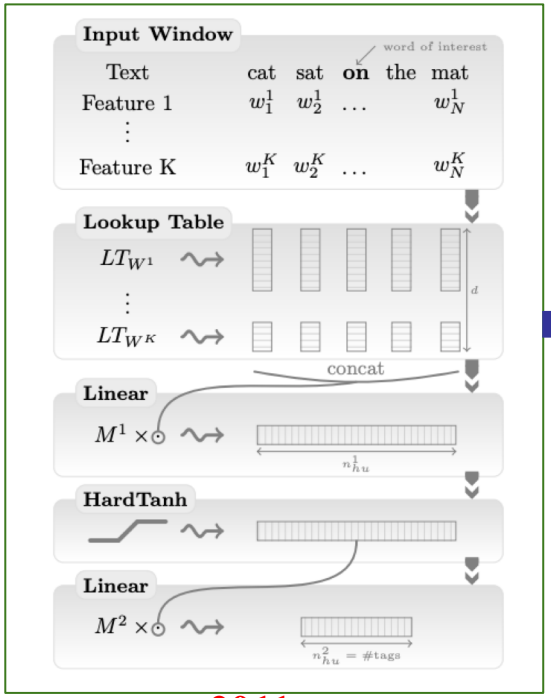
PER a ORG s LOC d DATE f

郑帅帅 ×，男。因涉嫌犯运输毒品罪，于 2016年6月6日 × 被 昆明市公安局官渡分局 × 刑事拘留，2016年7月11日 × 被执行逮捕。现羁押于 昆明市官渡区看守所 ×。辩护人 胡纯蛟 ×、姚芳 ×，云南颐高律师事务所律师。 昆明市官渡区人民检察院 × 以官检公一科刑诉1038号起诉书指控被告人 郑帅帅 × 犯运输毒品罪，于 2016年10月21日 × 向本院提起公诉,并以官检公一科量建820号量刑建议书提出"建议判处被告人 郑帅帅 × 十三年以上十五年以下有期徒刑，并处罚金"。

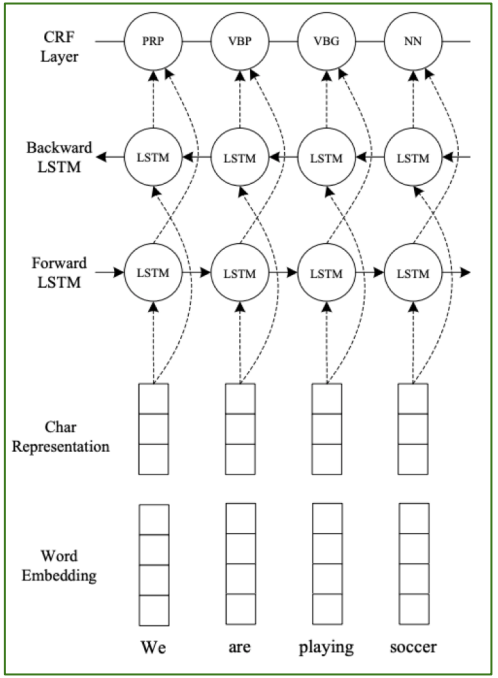
PER a ORG s LOC d DATE f

本院受理后，适用普通程序依法组成合议庭，于 2016年12月22日 ×、2017年1月11日 × 在本院法庭公开开庭审理了本案。 昆明市官渡区人民检察院 × 指派检察员 袁松 ×、代理检察员 詹晶 ×、书记员 何秋虹 × 出席法庭支持公诉，被告人 郑帅帅 × 及其辩护人 胡纯蛟 × 到庭参加诉讼，现已审理终结。经审理查明：2016年6月6日11时许 ×，被告人 郑帅帅 × 以体内藏匿的方式携带毒品可疑物欲乘飞机前往 郑州 ×，后在 云南省景洪市西双版纳国际机场 × 安检时，被民警查获。从其体内排出毒品可疑物净重142克。经鉴定，排出的毒品可疑物检出海洛因成分。

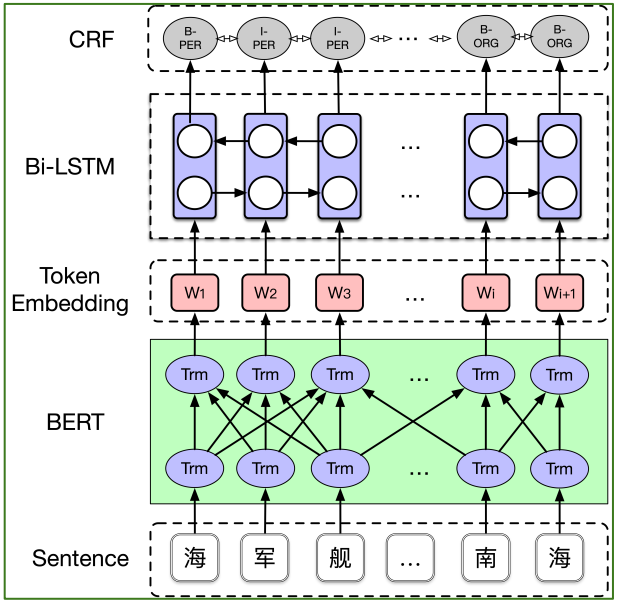
Name Entity Recognition



2011
NN/CNN + CRF
89%

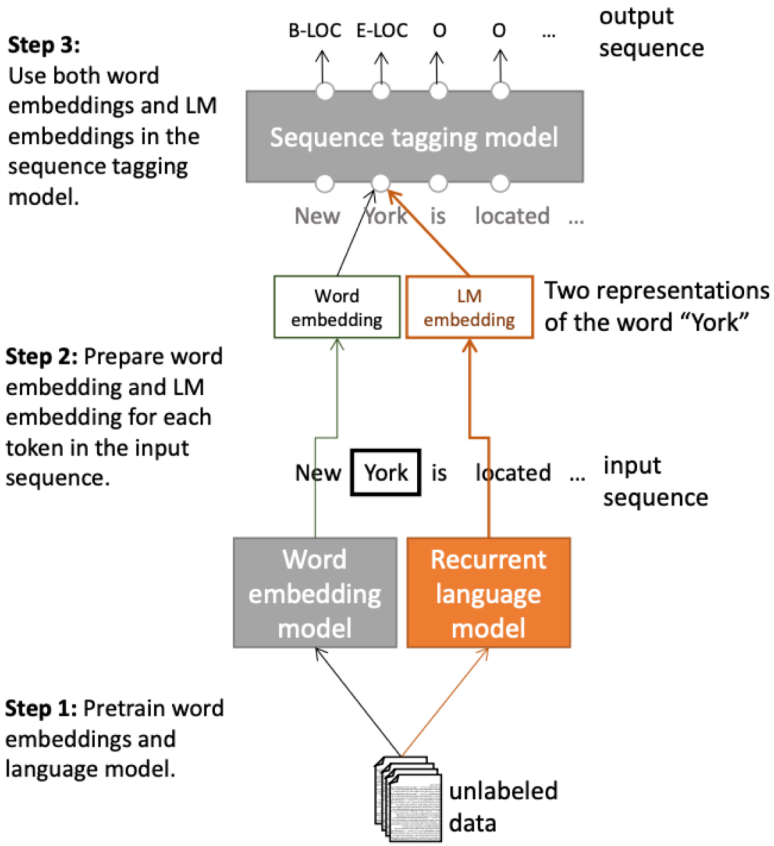


2016
BiLSTM+CRF
91%



2018
BERT+BiLSTM-+CRF
92.8%

Name Entity Recognition

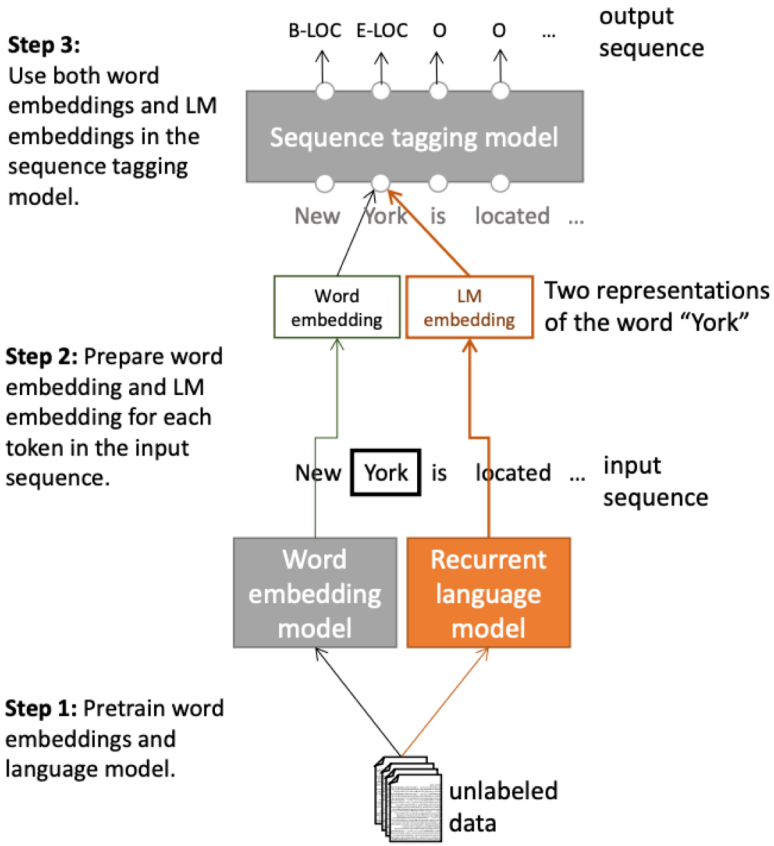


Language Model Augmented Sequence Taggers(TagLM)

a. 使用海量无标注语料训练Bi-LSTM
b. 获取LM embedding和Word embedding
c. 将词的向量和语言模型向量混合输入到序列标注模型中进行预测

Peters et al.[2017]

Name Entity Recognition

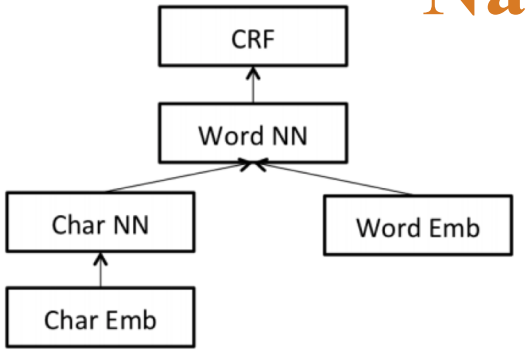


Language Model Augmented Sequence Taggers(TagLM)

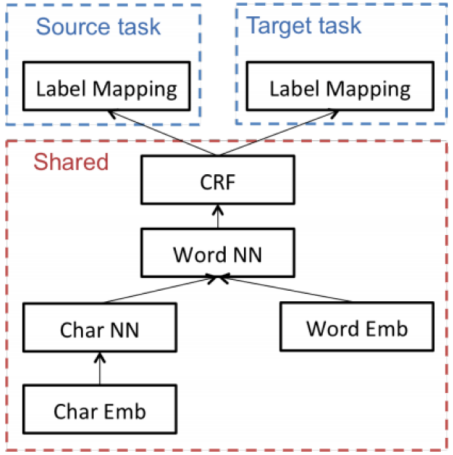
a. 使用海量无标注语料训练Bi-LSTM
b. 获取LM embedding和Word embedding
c. 将词的向量和语言模型向量混合输入到序列标注模型中进行预测

Peters et al.[2017]

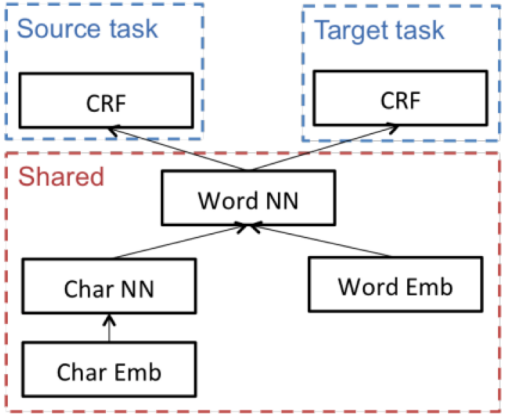
Name Entity Recognition



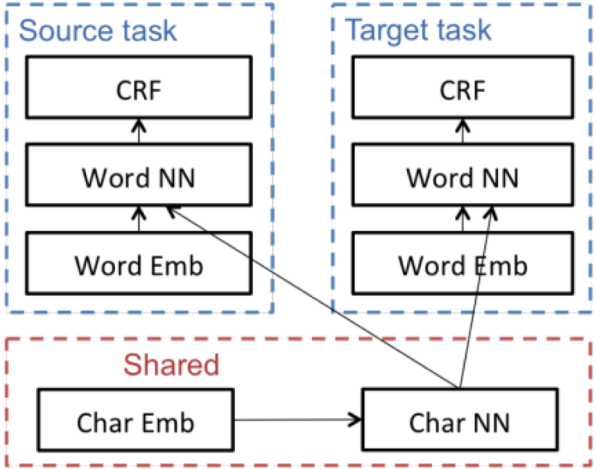
(a) Base model: both of Char NN and Word NN can be implemented as CNNs or RNNs.



(b) Transfer model T-A: used for cross-domain transfer where label mapping is possible.



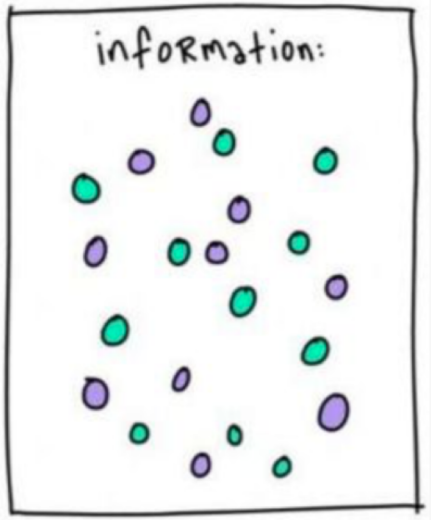
(c) Transfer model T-B: used for cross-domain transfer with disparate label sets, and cross-application transfer.



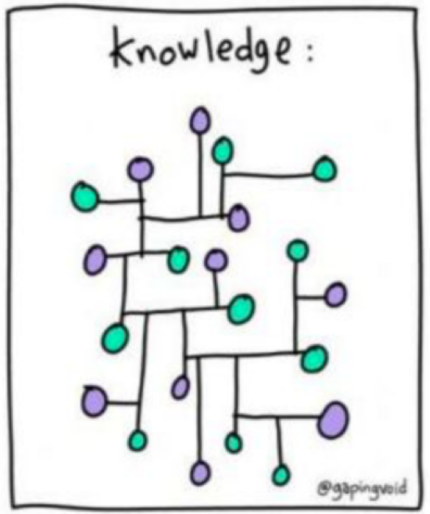
(d) Transfer model T-C: used for cross-lingual transfer.

Yang et al.[2017]

Relation Extraction

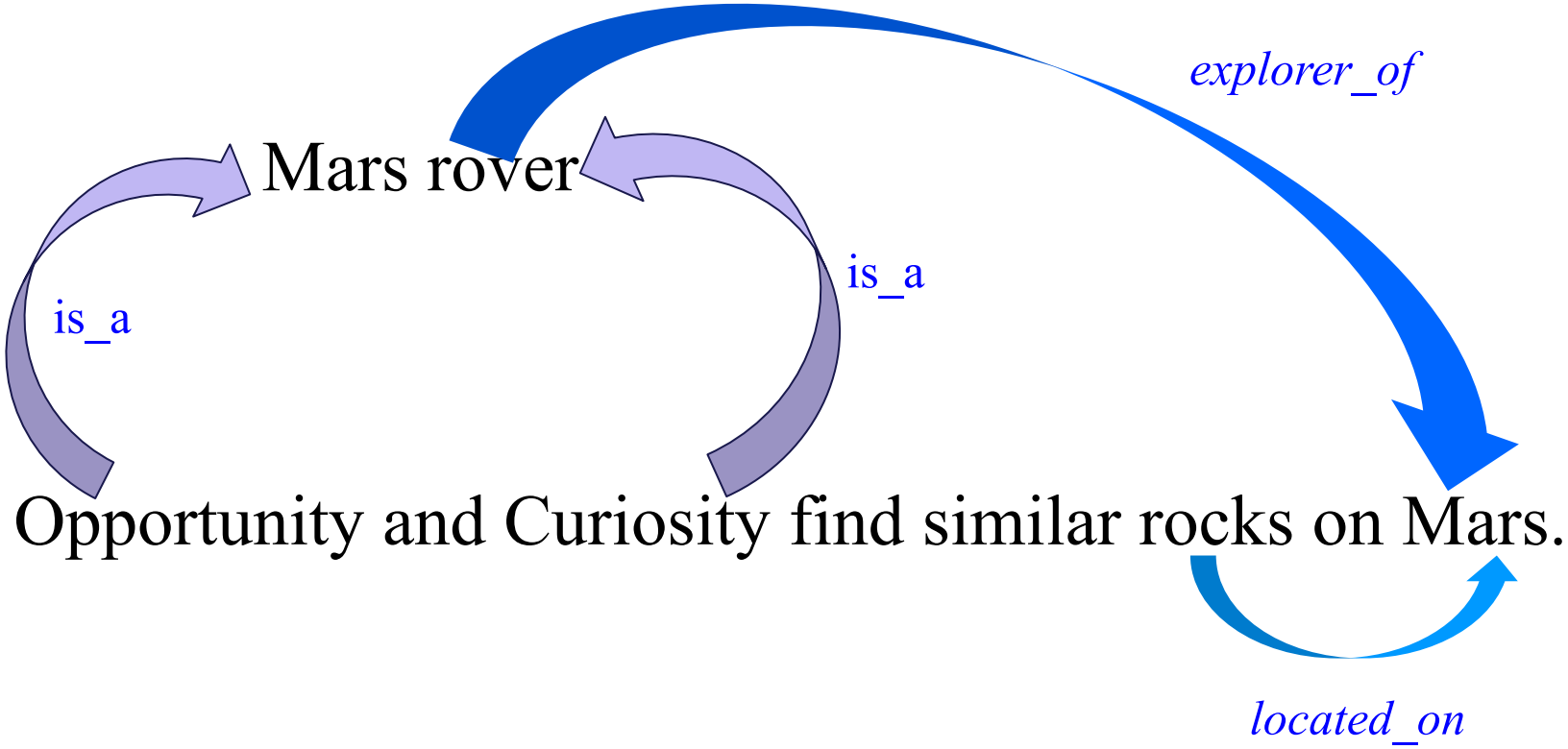


Entity

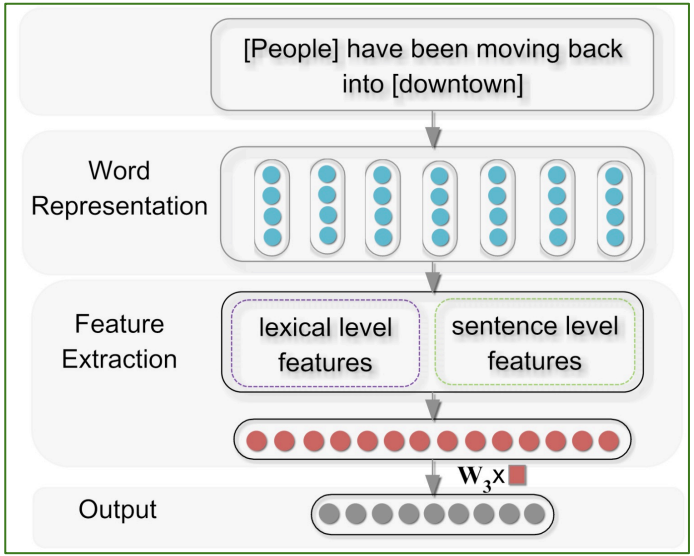


Entity + Relation

Relation Extraction



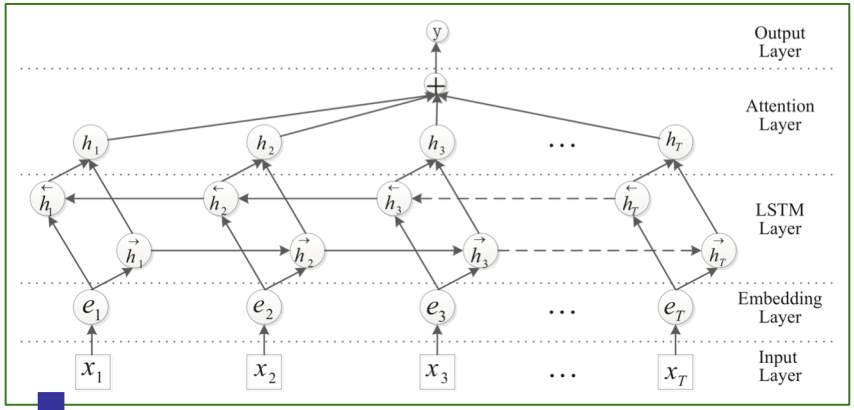
Relation Extraction



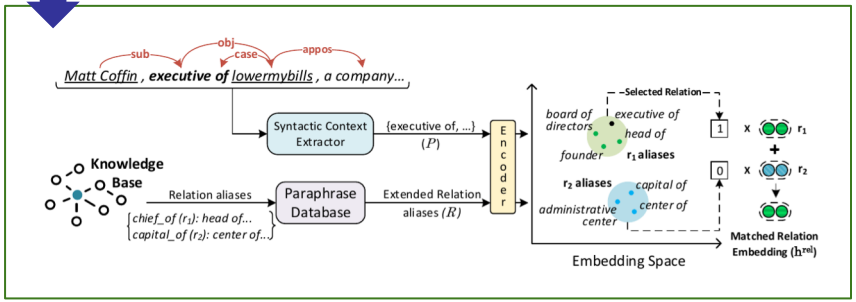
2012-2014

POS+CNN +RNN

82%



2016 Attention+BiLSTM 84%



Relation Extraction

semi-supervised methods

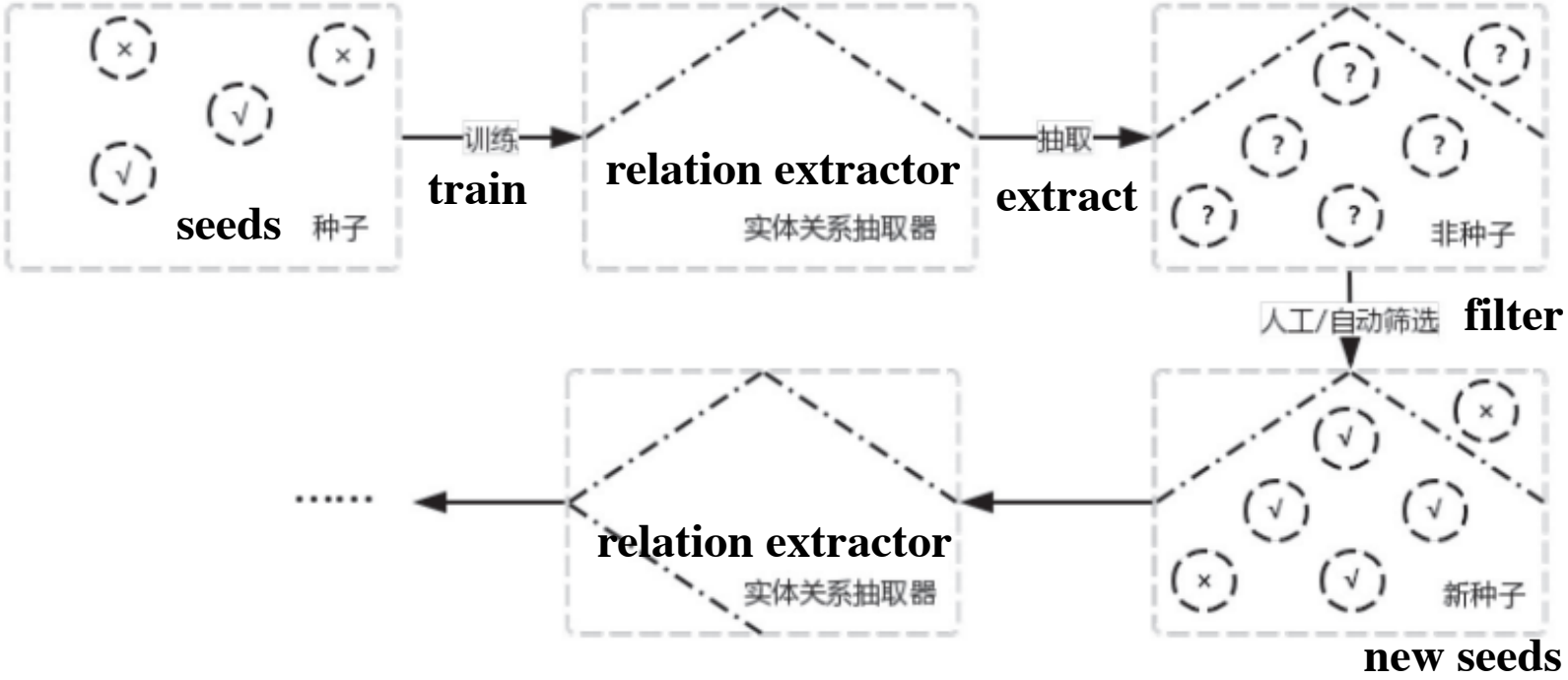


图 2　半监督学习训练过程

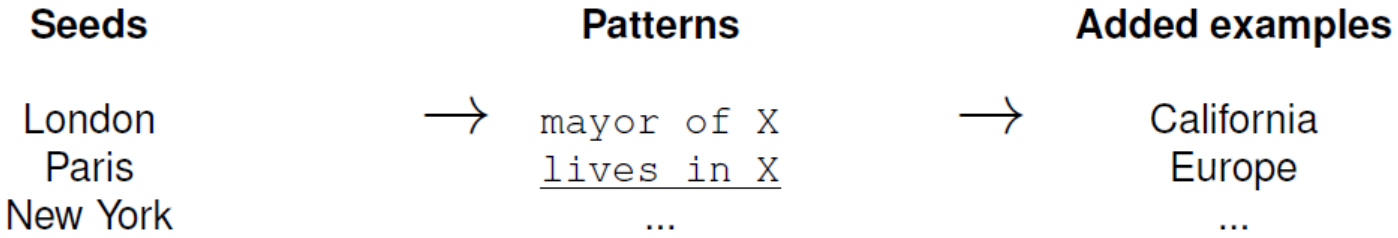
Relation Extraction

weak supervised methods: bootstrapping

SEMANTIC RELATION	SURFACE PATTERNS	$Hits_{sur\_for\ patterns}$
narrow similarity (2)*	X ie Y Xed ie Yed	210,480
broad similarity (2)*	Xed and Yed to X and Y	154,518,326
strength (8)	X even Y Xed even Yed X and even Y Xed and even Yed Y or at least X Yed or at least Xed not only Xed but Yed not just Xed but Yed	1,016,005
enablement (4)	Xed * by Ying the Xed * by Ying or to X * by Ying the to X * by Ying or	2,348,392
antonymy (7)	either X or Y either Xs or Ys either Xed or Yed either Xing or Ying whether to X or Y Xed * but Yed to X * but Y	18,040,916
happens-before (12)	to X and then Y to X * and then Y Xed and then Yed Xed * and then Yed to X and later Y Xed and later Yed to X and subsequently Y Xed and subsequently Yed to X and eventually Y Xed and eventually Yed	8,288,871

Relation Extraction

semantic drifting problem



Relation Extraction

distant supervision

Feature type	Left window	NE1	Middle	NE2	Right window
Lexical	[]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[]
Lexical	[Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[,]
Lexical	[#PAD#, Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[, Missouri]
Syntactic	[]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Edwin Hubble $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Astronomer $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{lex-mod}$,]
Syntactic	[Edwin Hubble $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{lex-mod}$,]
Syntactic	[Astronomer $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{lex-mod}$,]
Syntactic	[]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{inside}$ Missouri]
Syntactic	[Edwin Hubble $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{inside}$ Missouri]
Syntactic	[Astronomer $\downarrow_{lex-mod}$]	PER	[$\uparrow_s$ was $\downarrow_{pred}$ born $\downarrow_{mod}$ in $\downarrow_{pcomp-n}$]	LOC	[$\downarrow_{inside}$ Missouri]

Table 3: Features for 'Astronomer Edwin Hubble was born in Marshfield, Missouri'.

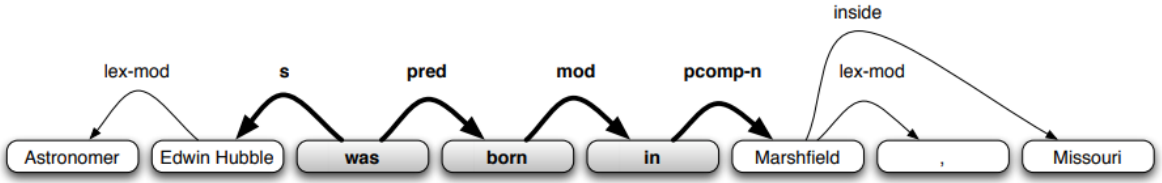


Figure 1: Dependency parse with dependency path from 'Edwin Hubble' to 'Marshfield' highlighted in boldface.

Distant supervision for relation extraction without labeled data [Mintz et al, 2009]

Relation Extraction

unsupervised method

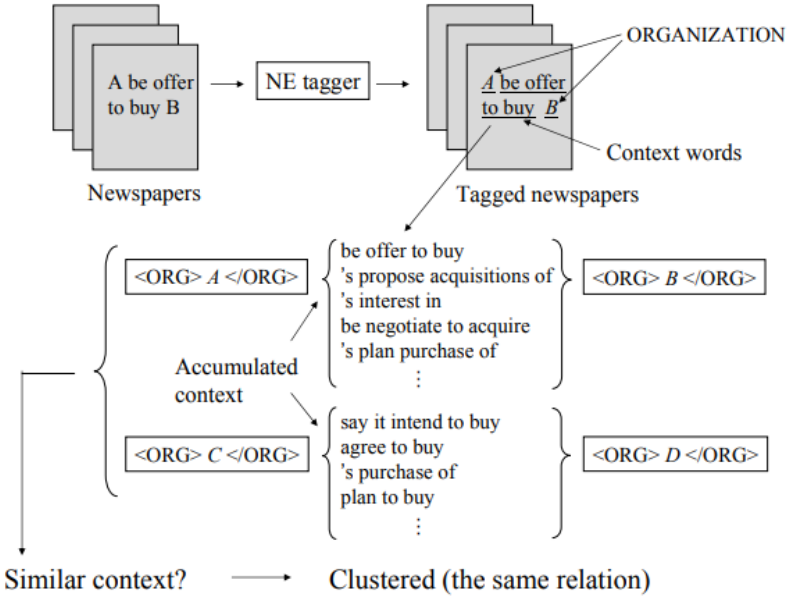


Figure 1: Overview of our basic idea

Our basic idea is as follows:

1. tagging named entities in text corpora
2. getting co-occurrence pairs of named entities and their context
3. measuring context similarities among pairs of named entities
4. making clusters of pairs of named entities
5. labeling each cluster of pairs of named entities

Discovering Relations Among Named Entities from Large Corpora [Hasegawa, 2004]

Relation Extraction

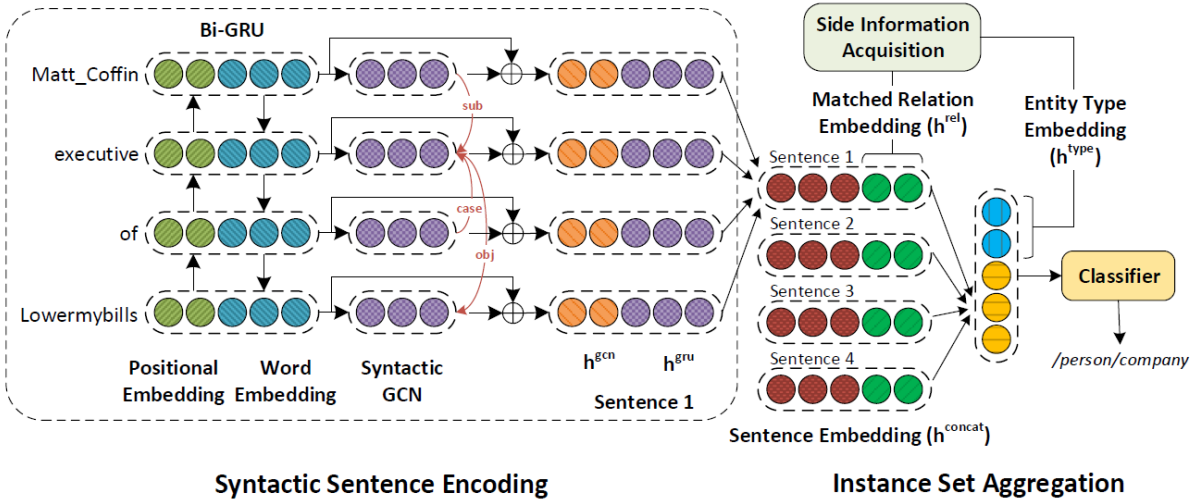
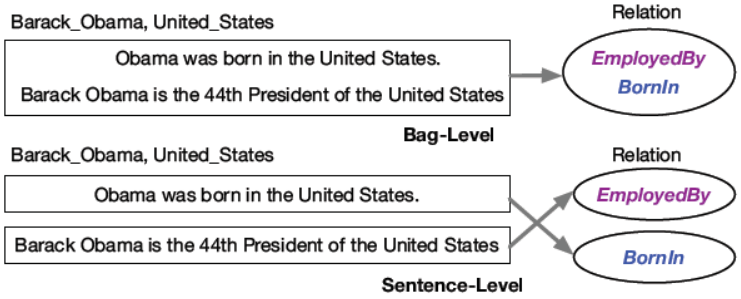
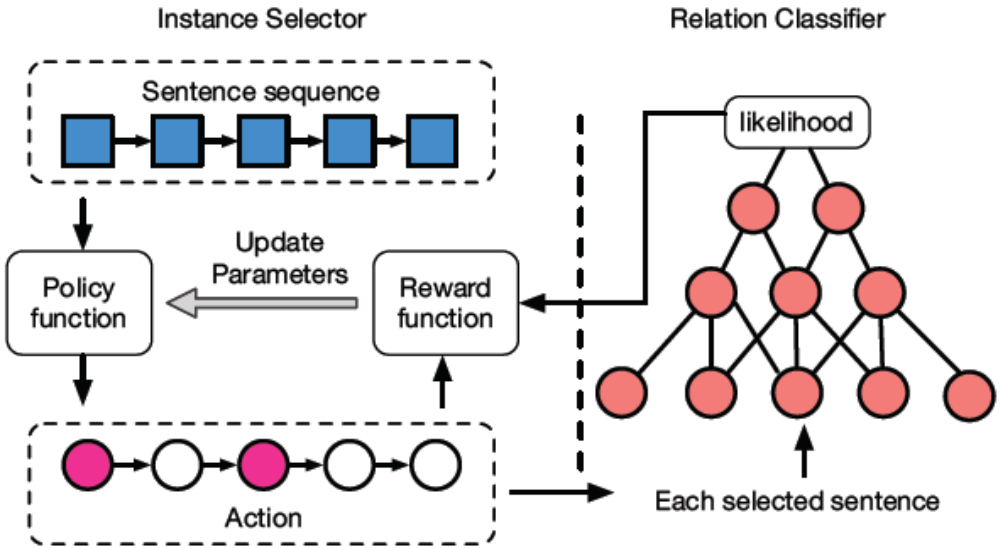


Figure 1: Overview of RESIDE. RESIDE first encodes each sentence in the bag by concatenating embeddings (denoted by $\oplus$) from Bi-GRU and Syntactic GCN for each token, followed by word attention. Then, sentence embedding is concatenated with relation alias information, which comes from the Side Information Acquisition Section (Figure 2), before computing attention over sentences. Finally, bag representation with entity type information is fed to a softmax classifier. Please see Section 5 for more details.

RESIDE: Improving Distantly-Supervised Neural Relation Extraction using Side Information [Vashishth, 2018, EMNLP]

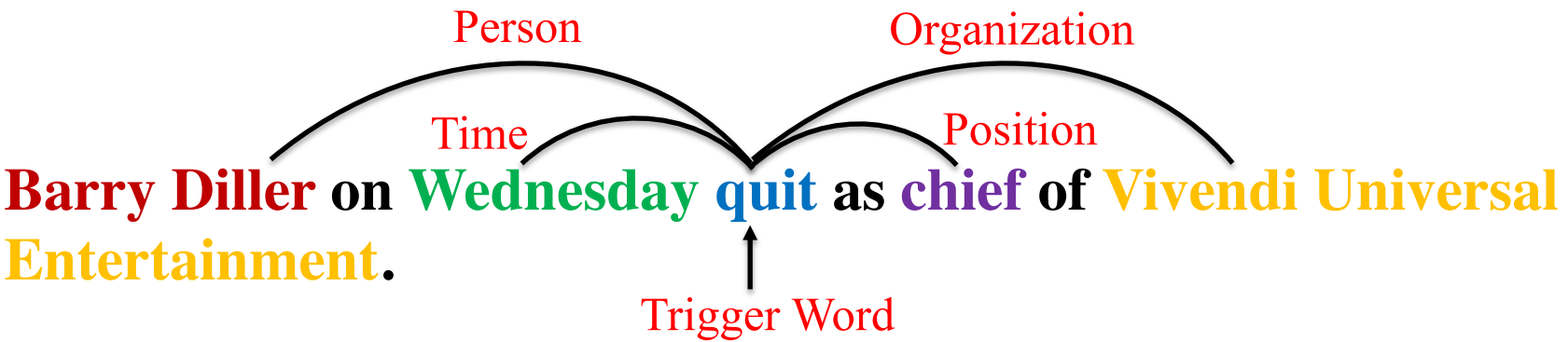
Relation Extraction





Reinforcement Learning for Relation Classification from Noisy Data [Feng, 2018, AAAI]

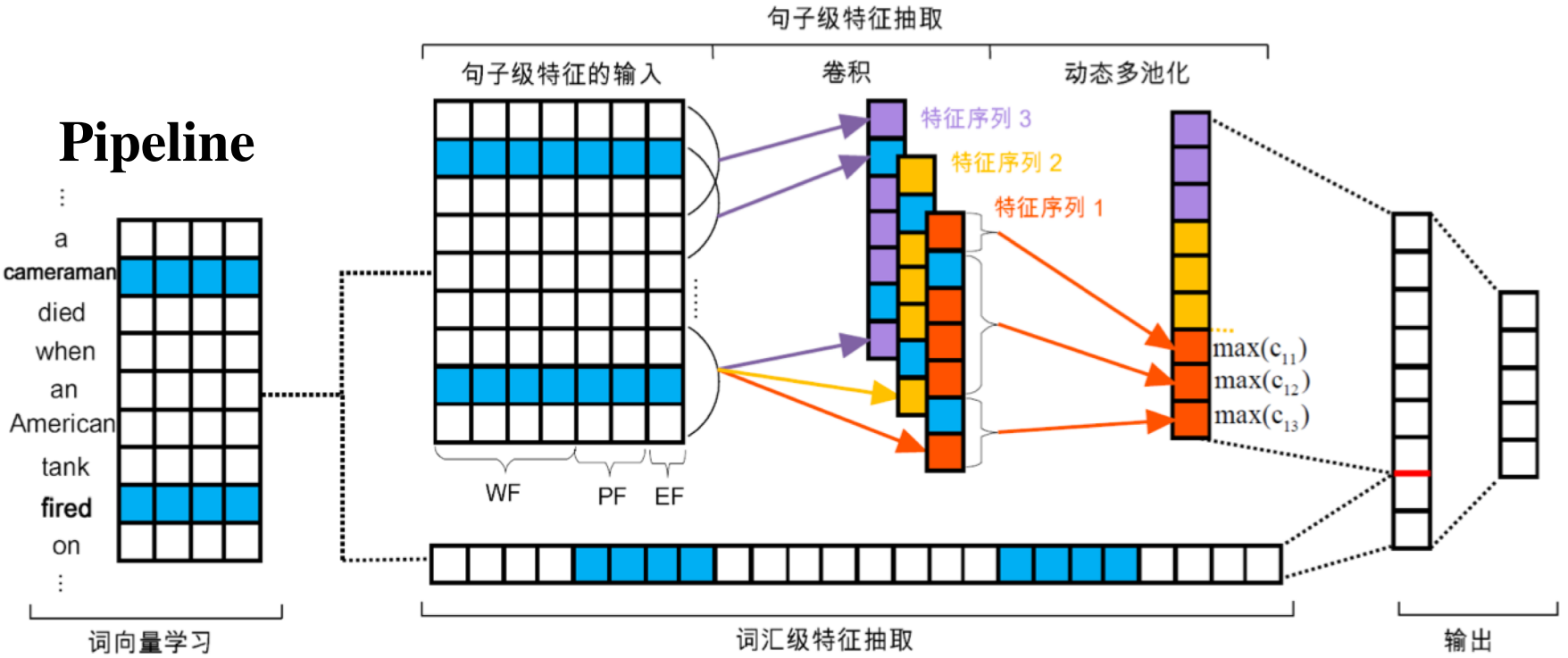
Event Extraction



Trigger	quite("Personnel/End-Position" event)	
Argument	**Role=Person**	Barry Diller
	Role=Organization	Vivendi Universal Entertainment
	Role=Position	Chief
	Role=Time-within	Wednesday(2003-03-04)

Event Extraction

Dynamic Multi-Pooling Convolutional Neural Network(DMCNN)



Chen et al.[2015]

Outline

- 1. Knowledge Representation and Modeling
- 2. Knowledge Extracting
- 3. Knowledge Fusion
- 4. Knowledge Storage

Knowledge Fusion/Integration

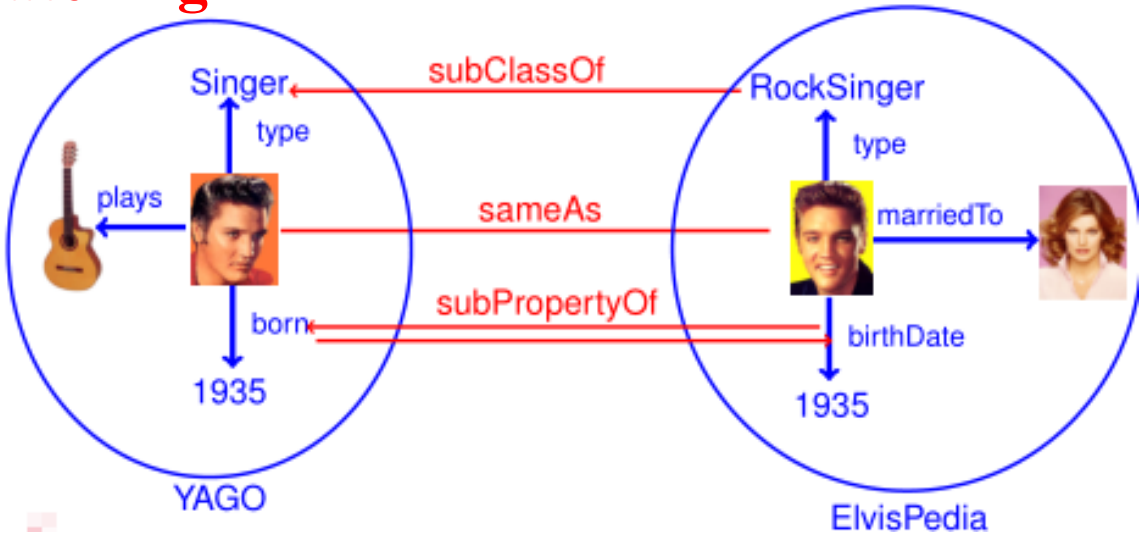
Goal: integrate multiple knowledge graphs

1. ontology matching

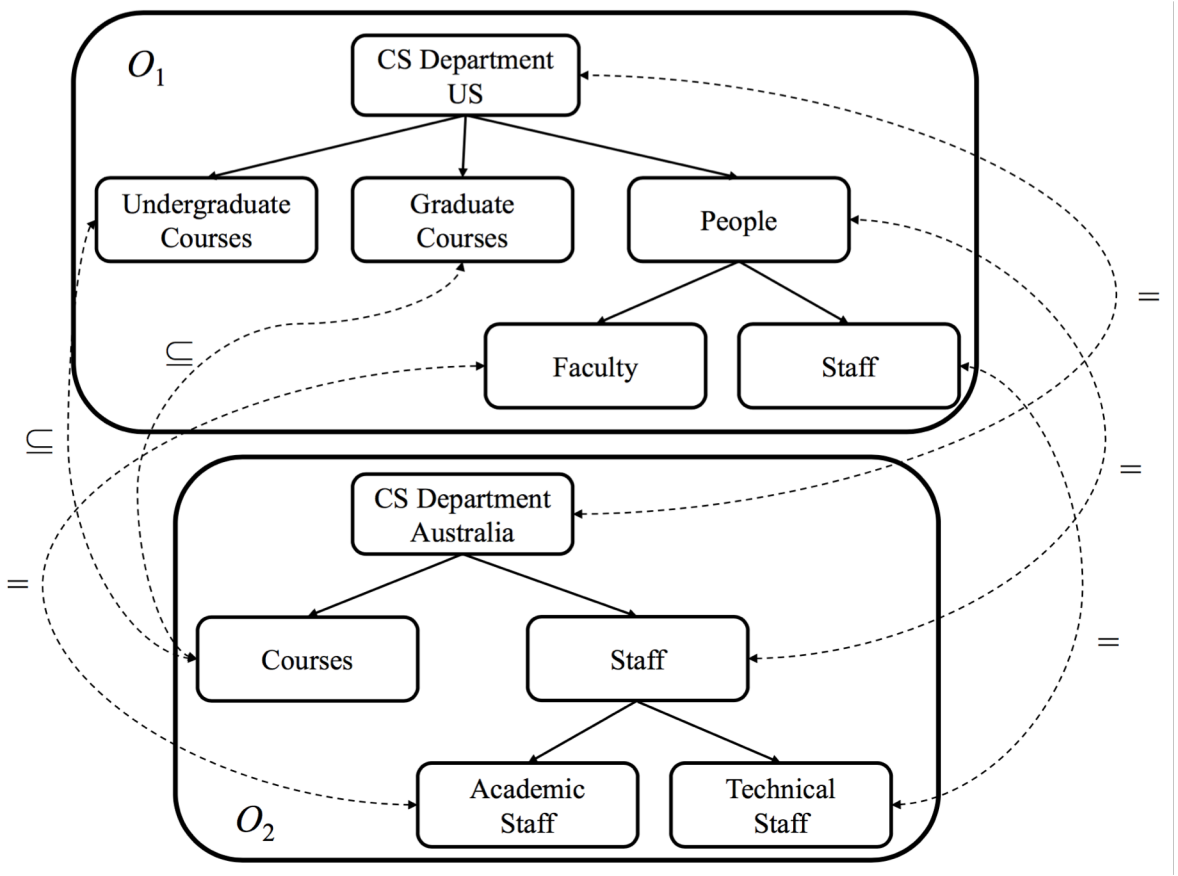
concept-concept matching

property-property matching

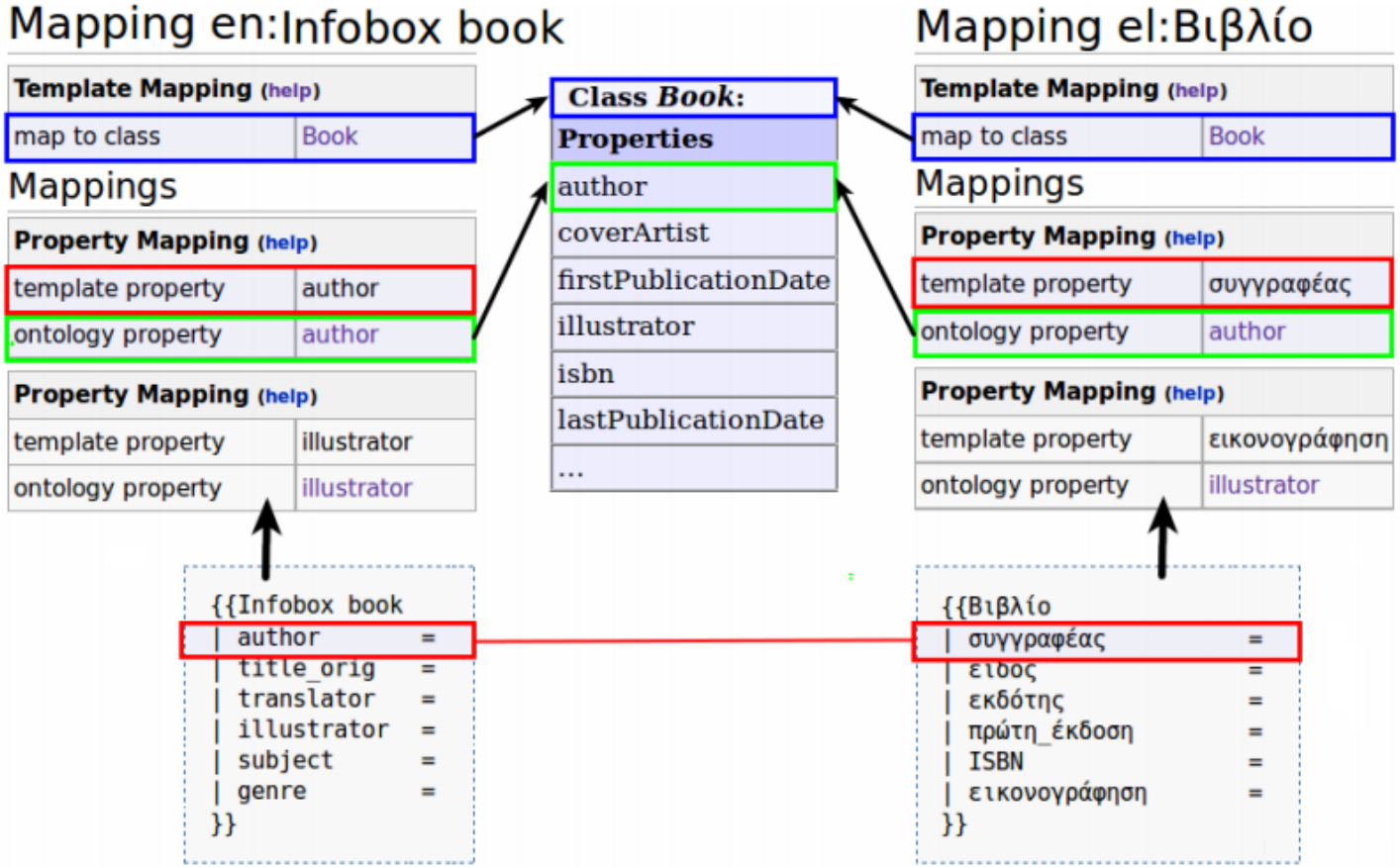
2. entity matching



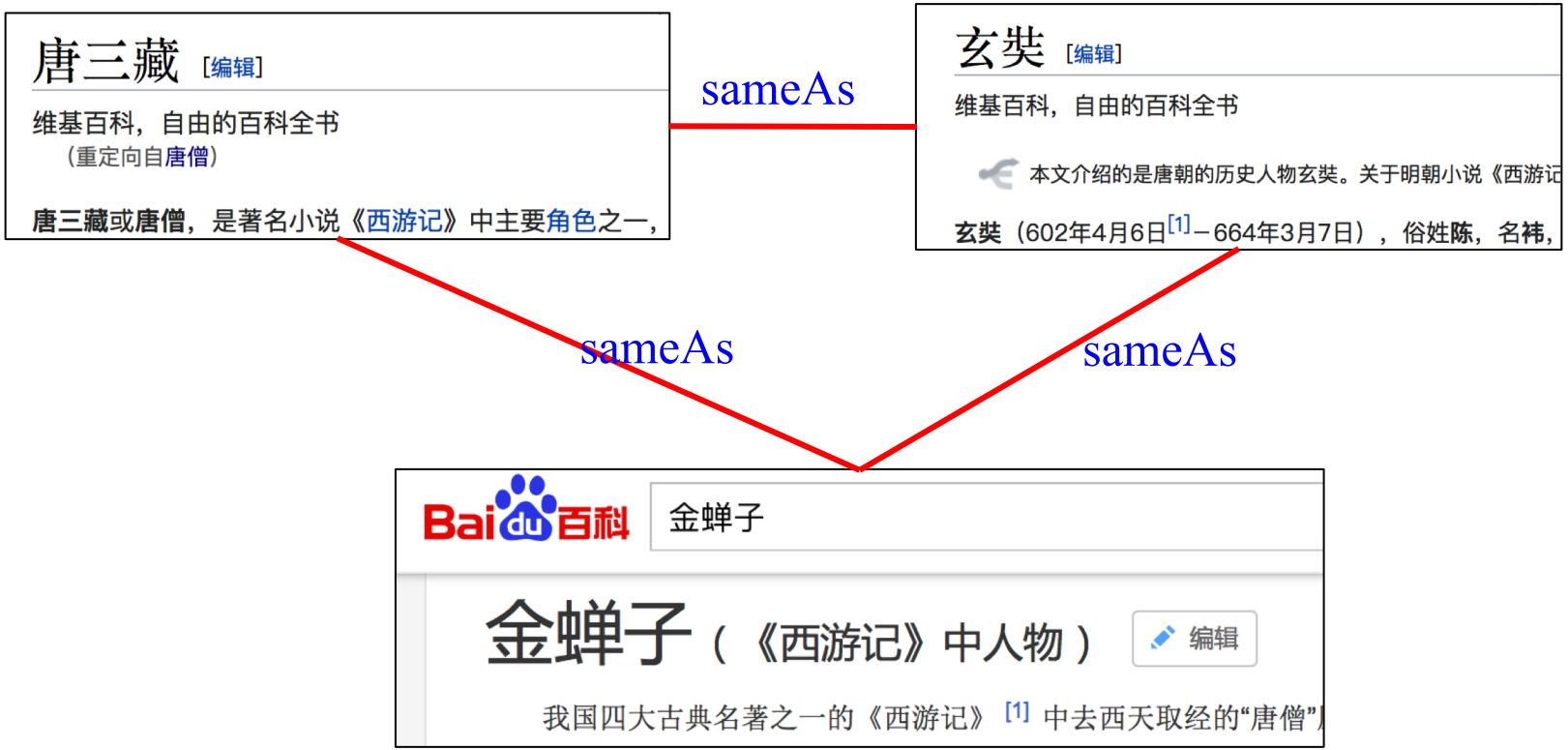
Knowledge Fusion/Integration



Knowledge Fusion/Integration

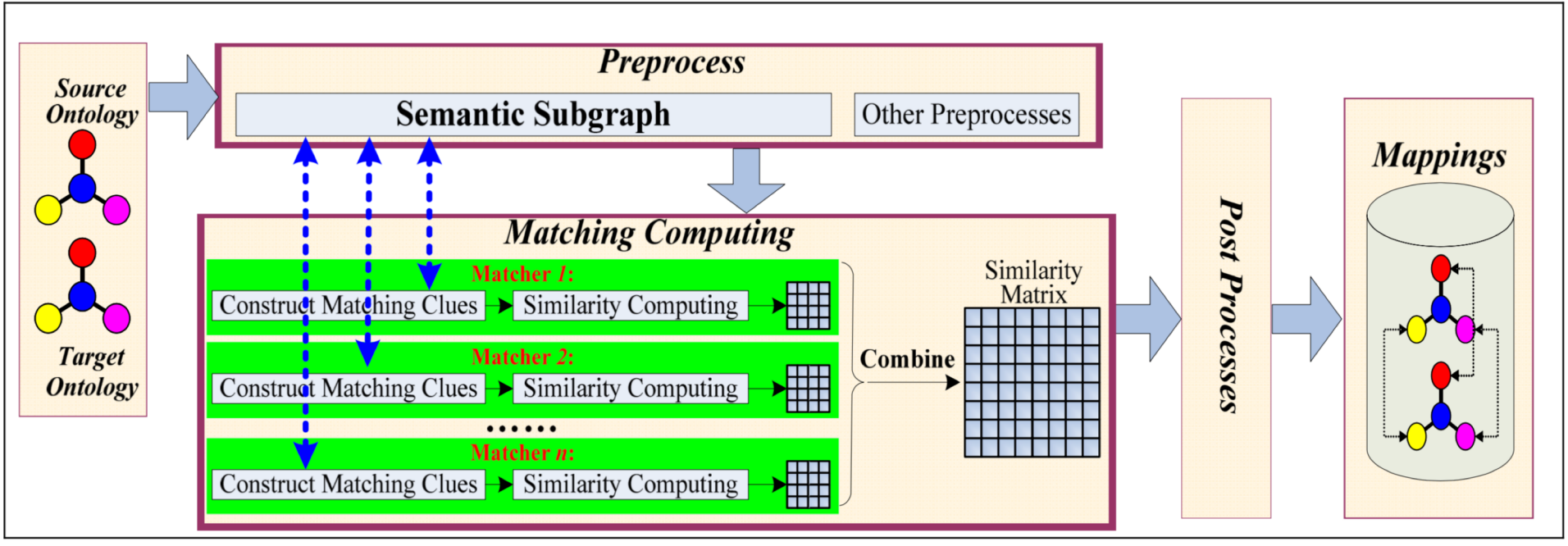


Knowledge Fusion/Integration



Knowledge Fusion/Integration

- **Ontology matching**

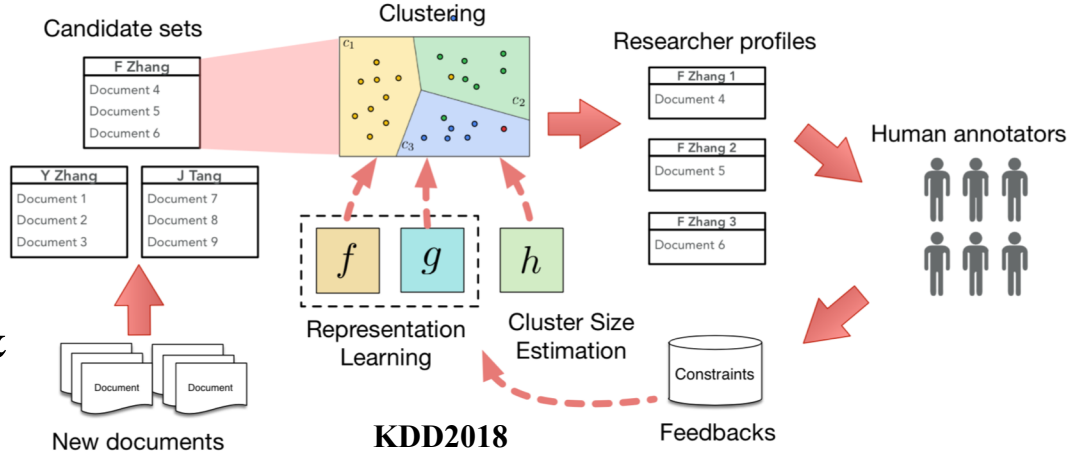


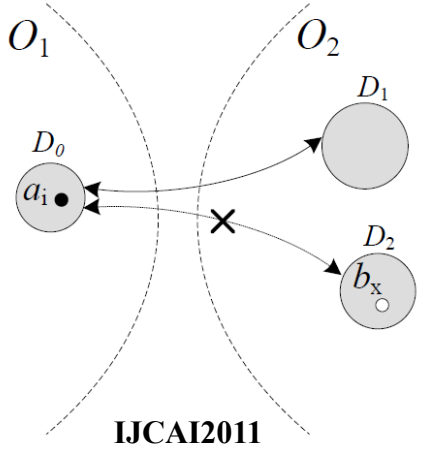
Knowledge Fusion/Integration

- **Large scale instance matching**

分治策略

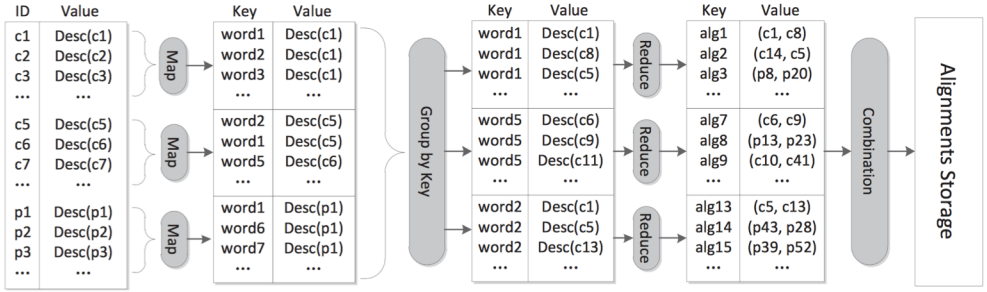
divided & conquer





并行提速

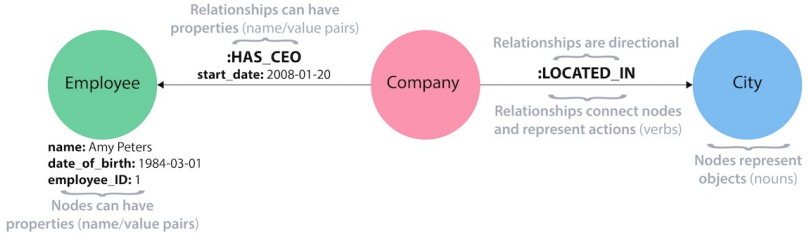
parallel



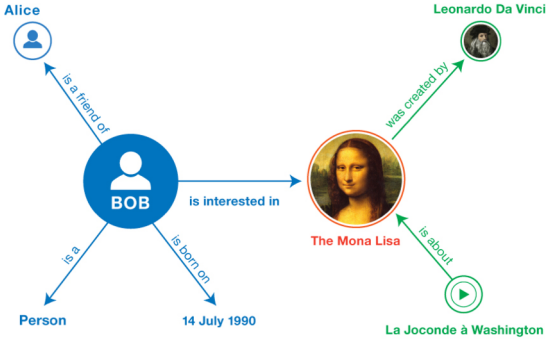
Outline

- 1. Knowledge Representation and Modeling
- 2. Knowledge Extracting
- 3. Knowledge Fusion
- 4. Knowledge Storage

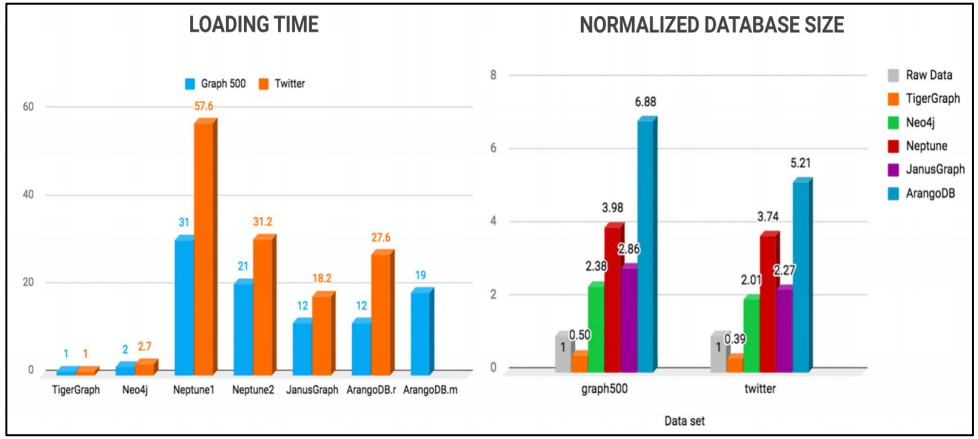
Knowledge Storage



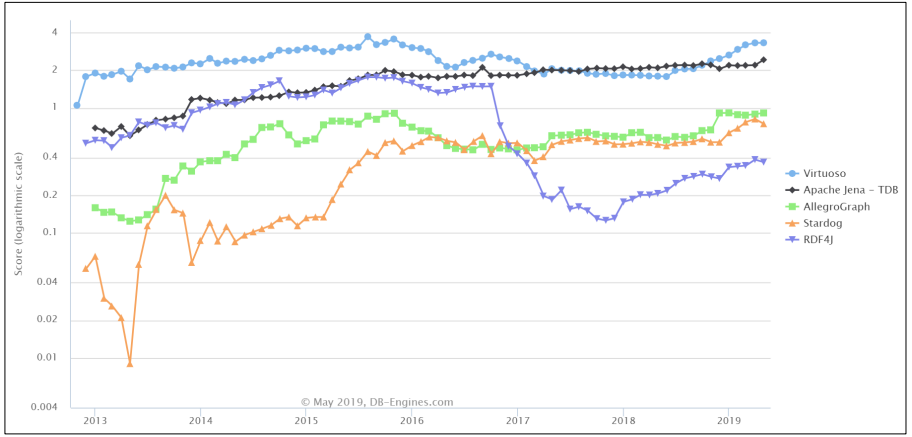
property graph
属性图



RDF graph



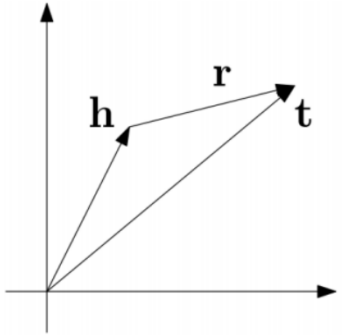
graph database



RDF database

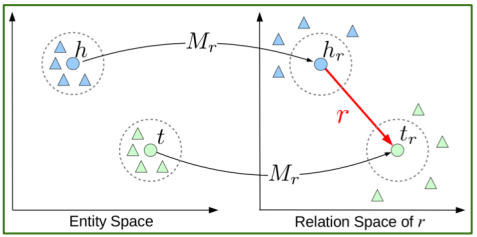
More techniques

Knowledge Graph Representation Learning

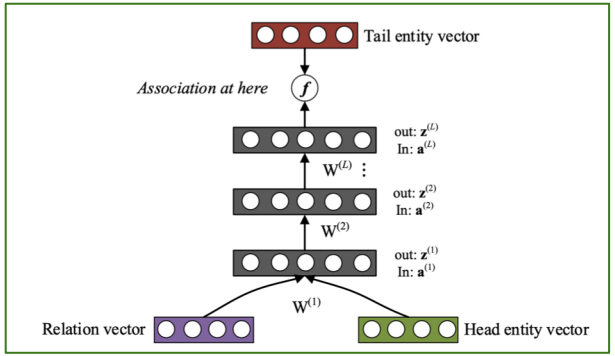


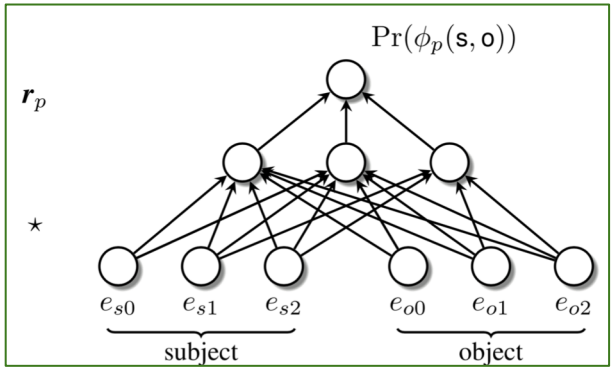
$$f(h,r,t) = \| h + r - t \|_{l1/l2}$$

$$L = \sum_{\xi \in T} \sum_{\xi' \in T'} [\gamma + f(\xi) - f(\xi')]_+$$

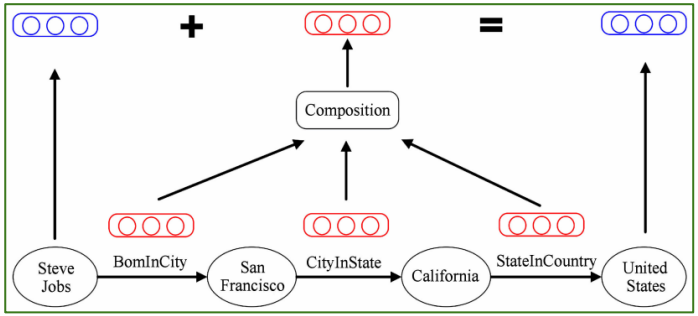


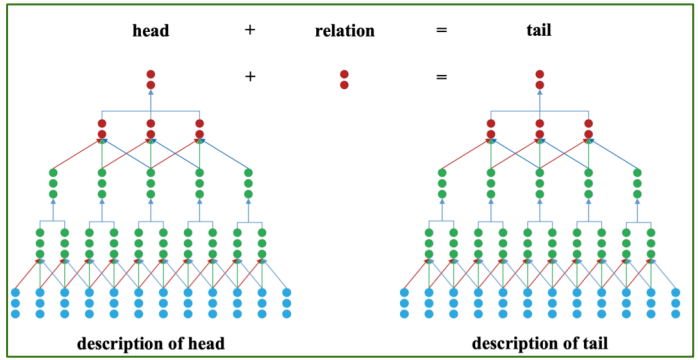
Translation models
TransE TransH TransR





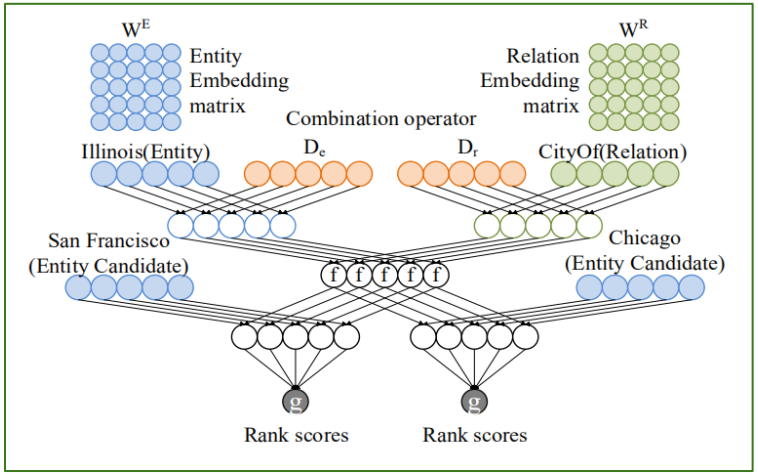
semantic matching models
RESCAL MLP NAM



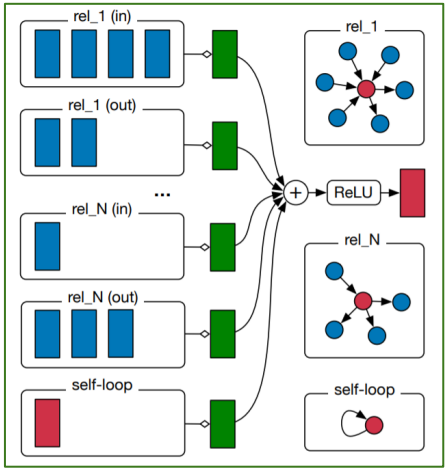


Fusion models
PTransE TKRL DKRL

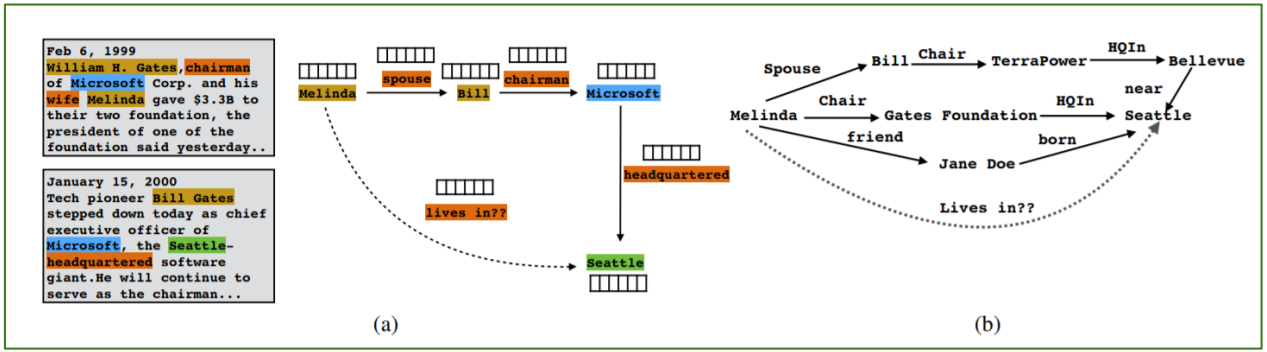
Knowledge Reasoning



NN+semantic-based reasoning



NN+structural-based reasoning



NN+path-based reasoning

Course Project

please take it seriously!

Tasks

- 1.Constructing a knowledge graph for a specific domain
  - Domain: Military or Finance
  - Source data: crawling data from Web (>=2 data sources)
  - Ontology: knowledge representation, concepts, properties, …
  - Knowledge graph:
    - knowledge extraction (entity, relations)
    - knowledge fusion/integration
    - knowledge storage
    - knowledge embedding (optional)
  - Knowledge graph sizes:
    - >100,000 entities
    - >1,000,000 triples

Tasks

- 2. Intelligent applications based on the knowledge graph
  - Semantic search
  - Question answer
  - Visualization (optional)
  - Mining and analyzing knowledge graph (optional)
  - Any other existing applications (use your imagination)

- 3 checkpoints
  - 1: Problem direction, context, outline of algorithm and evaluation
  - 2: Formulation, algorithm, data, preliminary results
  - 3: Additional theory/methods and results, applications

- Final presentation
  - 15 mins presentation + 5 mins questions

- Final report
  - Detailed writeup (latex, <30 pages)

Timeline for the Project

Tasks	Important Date
checkpoint1	October 28
checkpoint2	November 12
checkpoint3	November 26
Final Presentation	December 03
Final Report	December 10

More detail information and requirements (upload in this week)：
https://github.com/npubird/WebScienceCourse


धन्यवाद
Hindi
Hindi
多謝
Traditional Chinese
ขอบคุณ
Thai
Спасибо
Russian
Thank You
English
Gracias
Spanish
شكرا
Arabic
Obrigado
Brazilian Portuguese
Danke
German
Grazie
Italian
多谢
Simplified Chinese
Merci
French
நன்றி
Tamil
Tamil
ありがとうございました
Japanese
감사합니다
Korean