

Part I: Lean KG

精益知识图谱方法论

鲍捷

部分基于丁海星，王丛，段清华，郑锦光的工作



摘要

知识图谱是一系列结构化数据的处理方法，它涉及知识的提取、表示、存储、检索等诸多技术。从渊源上讲，它是知识表示与推理、数据库、信息检索、自然语言处理等多种技术发展的融合。但传统的知识处理方法，在实际的工程应用，特别是互联网应用中，面临实施成本高、技术周期长、熟悉该类技术的人才缺乏、基础数据不足等诸多现实制约。实战中的知识图谱，需要充分利用成熟的工业技术，不拘泥于特定的工具和方法，特别是不盲目追求标准化、技术的先进性或者新颖性，以实际的业务出发，循序渐进推进工程的实施。在本教程中，我们首先回顾知识图谱从理念到工程逐步落地的发展历程，梳理各技术与知识图谱应用的关系。然后，我们结合工程上具体的实例，来展示知识图谱的核心理念和技术如何在成本约束下实施，包括结构化数据生成，可维护的知识结构，海量知识的数据库管理，和多层次的语义检索等。

下载: <http://blog.memect.cn/?p=2005>

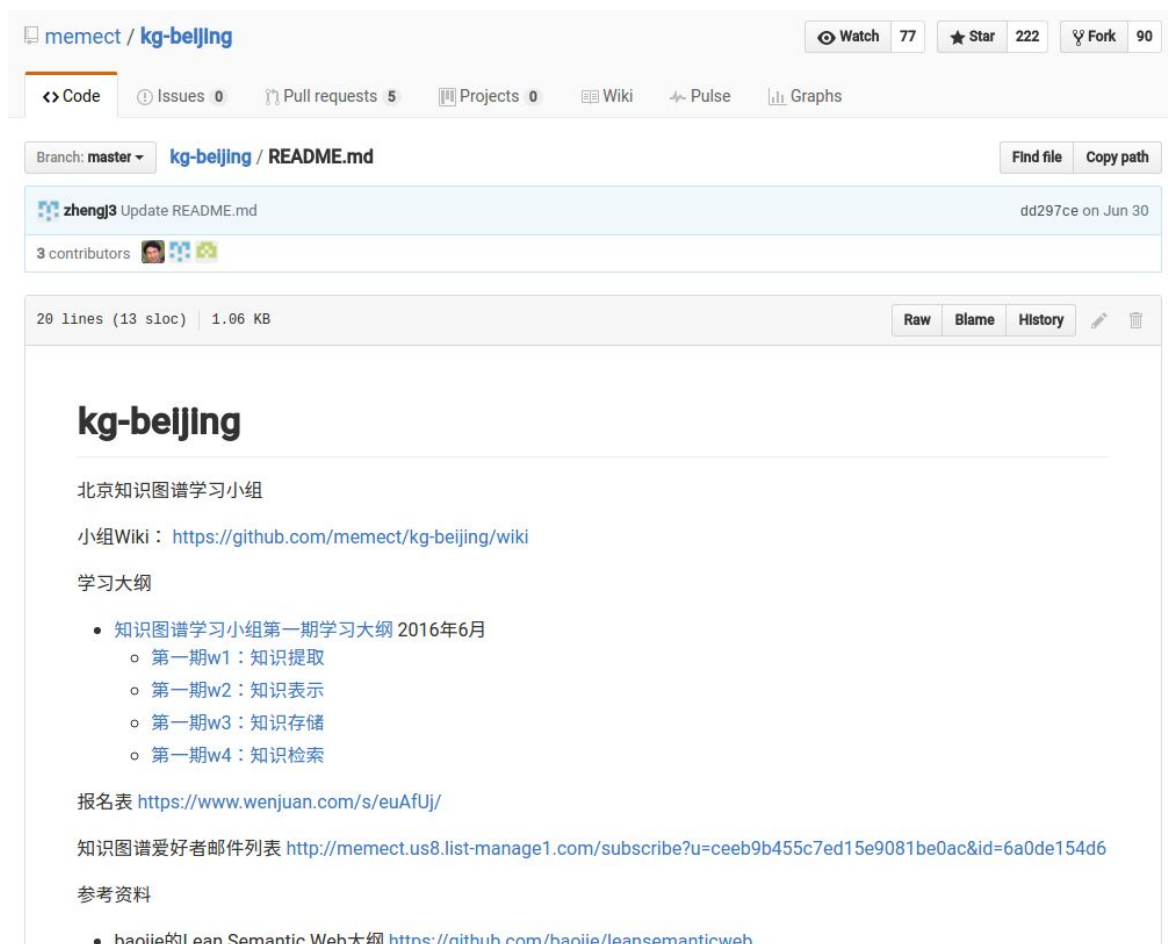


相关项目经历

- 1999 放射科图像元数据, PACS
- 2003 蛋白质本体, 基因本体
- 2004 生物信息学数据语义集成 (INDUS)
- 2005 动物特征本体; 内窥镜本体; 协作本体编辑器COB-Editor
- 2006 模块化本体开发, 分布式推理
- 2007 推理中的隐私保护
- 2008 地图元数据 (RPI Map)
- 2008 W3C OWL本体语言工作组成员
- 2009 语义维基和情报分析, 受控自然语言
- 2009 AIR语言和法规建模
- 2010 XBRL (电子化年报) 语义化
- 2010 会议元数据 (ISWC)
- 2011 语义用户画像 (三星)
- 2012 三星SVoice个人语音助手
- 2013 语义电子邮件
- 2014 Emma可视化书签
- 2015 好东西传送门, 基于知识库的新闻推荐
- 2016 金融知识图谱 @ 文因互联

丰富的失败经验

北京知识图谱学习小组



memect / kg-beijing

Watch 77 Star 222 Fork 90

<> Code Issues 0 Pull requests 5 Projects 0 Wiki Pulse Graphs

Branch: master kg-beijing / README.md Find file Copy path

zhengj3 Update README.md dd297ce on Jun 30

3 contributors

20 lines (13 sloc) 1.06 KB Raw Blame History

kg-beijing

北京知识图谱学习小组

小组Wiki: <https://github.com/memect/kg-beijing/wiki>

学习大纲

- 知识图谱学习小组第一期学习大纲 2016年6月
 - 第一期w1: 知识提取
 - 第一期w2: 知识表示
 - 第一期w3: 知识存储
 - 第一期w4: 知识检索

报名表 <https://www.wenjuan.com/s/euAfUj/>

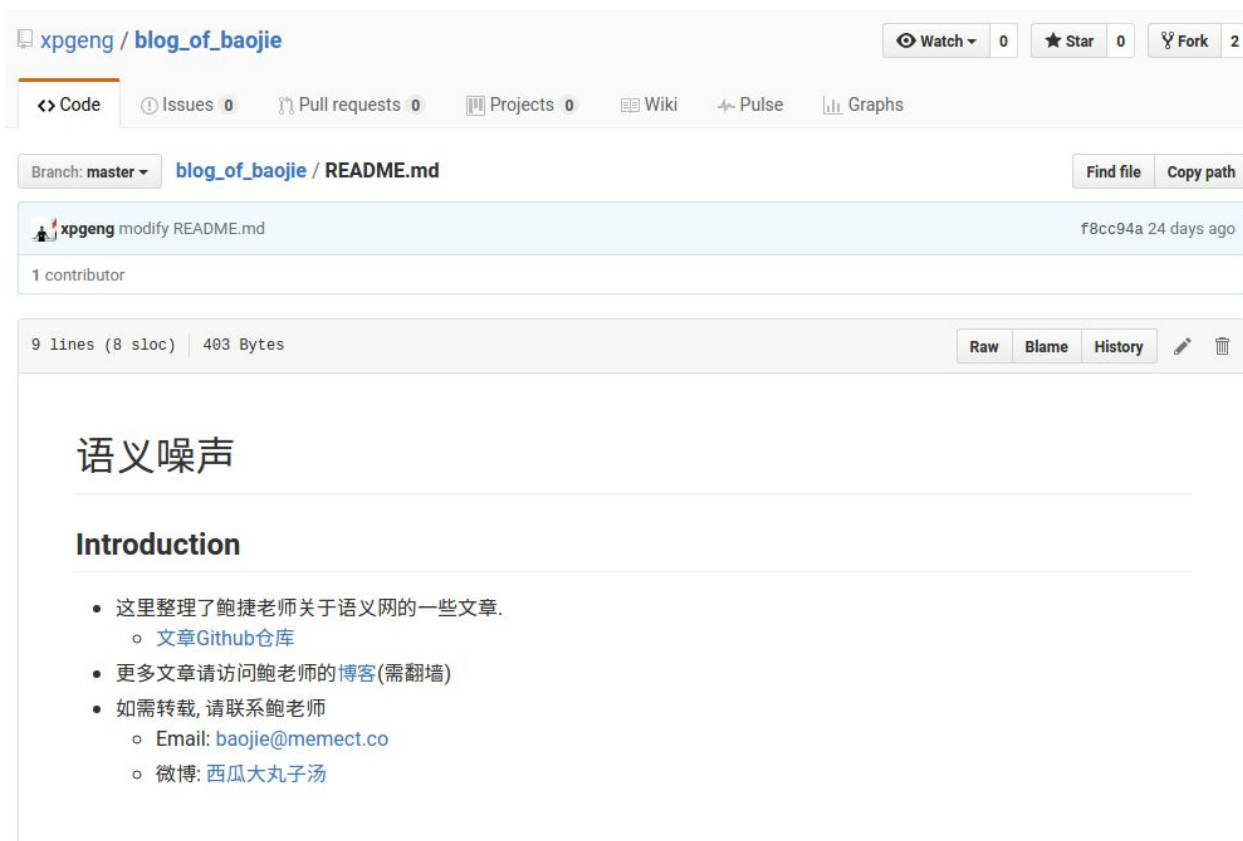
知识图谱爱好者邮件列表 <http://memect.us8.list-manage1.com/subscribe?u=ceeb9b455c7ed15e9081be0ac&id=6a0de154d6>

参考资料

- baojie的Lean Semantic Web大纲 <https://github.com/baojie/leansemanticweb>

<https://github.com/memect/kg-beijing>

语义噪声



https://github.com/xpgeng/blog_of_baojie (耿新鹏整理)

<http://baojie.org/blog>

Lean Semanticweb

The screenshot shows the GitHub repository page for 'baojie / leansemanticweb'. At the top, the repository name is displayed with navigation links for 'Code', 'Issues' (1), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. Below this, the repository description 'Lean Semantic Web tutorials – Edit' is shown. A summary bar indicates 39 commits, 1 branch, 0 releases, and 1 contributor. Action buttons include 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table lists recent updates to 'Bibliography.md', 'README.md', and 'Syllabus.md'. The 'README.md' section is expanded, showing the title 'leansemanticweb' and the subtitle 'Lean Semantic Web tutorials', followed by a bulleted list of links to 'Syllabus' and 'Bibliography'.

baojie / leansemanticweb

Unwatch 23 Star 57 Fork 26

<> Code Issues 1 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Lean Semantic Web tutorials – Edit

39 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

baojie	Update Syllabus.md	Latest commit bdd0f63 on Feb 13, 2014
Bibliography.md	Update Bibliography.md	4 years ago
README.md	add links to Bibliography and Syllabus	4 years ago
Syllabus.md	Update Syllabus.md	3 years ago

README.md

leansemanticweb

Lean Semantic Web tutorials

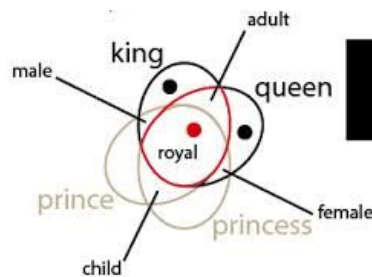
- [Syllabus](#)
- [Bibliography](#)

<https://github.com/baojie/leansemanticweb>

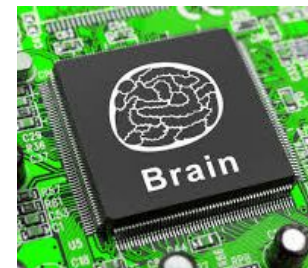
知识图谱



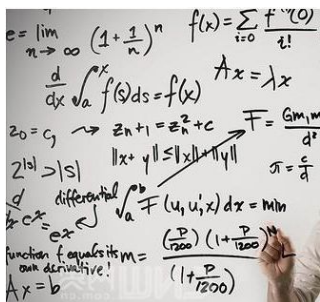
媒体报道的



学术界发表的



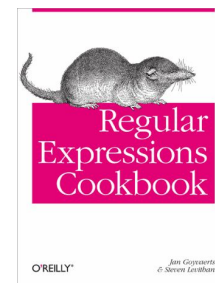
大公司宣传的



别人以为我干的



我以为我干的



我实际干的

知识图谱



知识图谱

+Dave Search Images Maps Play YouTube News Gmail Documents Calendar More

Google

marie curie

Search

8 personal results, 17,300,000 other results (0.32 seconds)

Everything

Images

Maps

Videos

News

Shopping

Books

More

Mountain View, CA

Change location

Any time

Past hour

Past 24 hours

Past 2 days

Past week

Past month

Past year

Custom range...

Marie Curie - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Marie_Curie

Marie Skłodowska-Curie (7 November 1867 – 4 July 1934) was a French-Polish physicist and chemist famous for her pioneering research on radioactivity.

Marie Curie Cancer Care - Pierre Curie - Irène Joliot-Curie - Radioactive decay

Marie Curie - Biography - Nobelprize.org

www.nobelprize.org/nobel_prizes/physics/_/marie-curie-bio.html

Short profile from the foundation that awards the Nobel Prize.

Marie Curie - Biography - Nobelprize.org

www.nobelprize.org/nobel_prizes/chemistry/_/marie-curie-bio.html

Marie Curie, née Maria Skłodowska, was born in Warsaw on November 7, 1867, the daughter of a secondary-school teacher. She received a general education ...

Images for marie curie - Report images



Marie Curie and The Science of Radioactivity

www.aip.org/history/curie/

The life of Marie Curie, from the AIP Center for History of Physics. Text by Naomi Pasachoff and many illustrations describe Curie's contributions to the science of ...

Marie Curie



es.wikipedia.org

Marie Skłodowska-Curie was a French-Polish physicist and chemist famous for her pioneering research on radioactivity. She was the first person honored with two Nobel Prizes—in physics and chemistry. Wikipedia

Born: November 7, 1867, Warsaw

Died: July 4, 1934, Sancellemoz

Spouse: Pierre Curie (m. 1895–1906)

Children: Irène Joliot-Curie, Ève Curie

Discovered: Radium, Polonium

Education: École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris, University of Paris

People also search for



Albert Einstein



Pierre Curie



Ernest Rutherford



Louis Pasteur



John Dalton

Report a problem

面向人的知识图谱



欧阳询

为人民服务

为人民服务

- 数据在哪里？谁提供数据？
- 元数据会一并提供吗？
- 用户会不会提供元数据？他们会因此得到什么好处？他们的心理是什么？到什么程度他们会烦？
- 产生数据和元数据的代价是多大？用户需要付出多大的努力？
- 用户需要多长时间才会从提供元数据得到好处？
- 总用户里，有多少会主动提供元数据？在这些用户里，又有多少会提供高质量的元数据？
- 数据的依赖性如何？是不是要等很多用户都来提供数据了，系统才能有用？单个用户的数据有多少用？
- 需要多少数据/元数据系统才会有一点用？明显有用？要几天，几个月，还是几个世纪才能达到？
- 什么样的元数据是对用户/客户有价值的？如果没有，有多大可能用工具获得？
- 对现有系统，需要多大的改动？现有的工作模式是不是会被改变？这些人有动机做这种改变吗？会不会有抵触？怎么解决？
- 在这个应用的全生命周期，哪些人会参与？有多少人需要重新培训？培训的代价是多少？

<http://baojie.org/blog/2011/12/20/semantic-web-for-human/>

知识图谱技术的组成与源流



知识提取
(自然语言处理, 机器学习)



知识存储
(数据库)

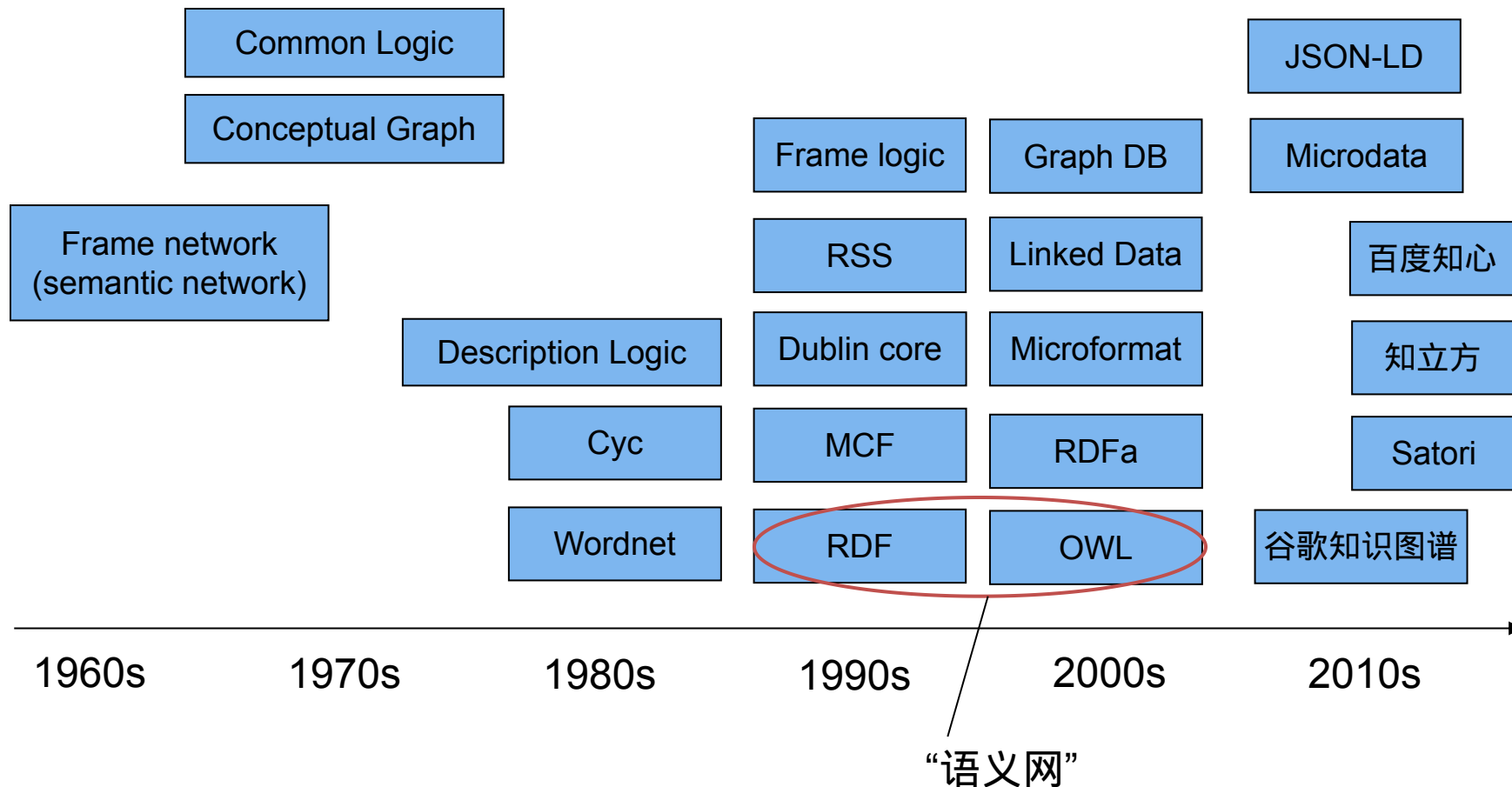


知识表现
(逻辑, 语义网)

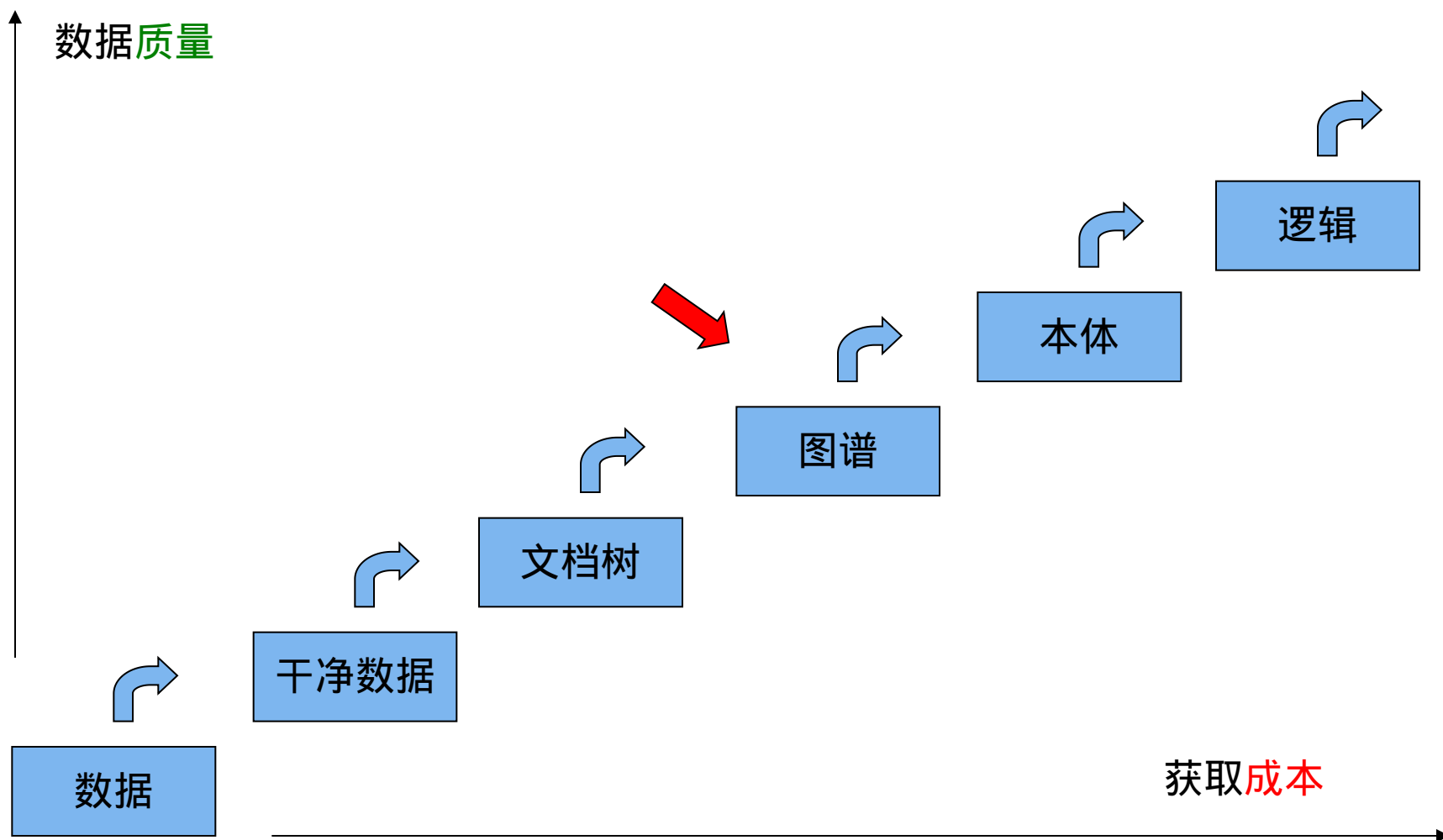


知识检索
(信息检索, 人机交互)

知识图谱的前身



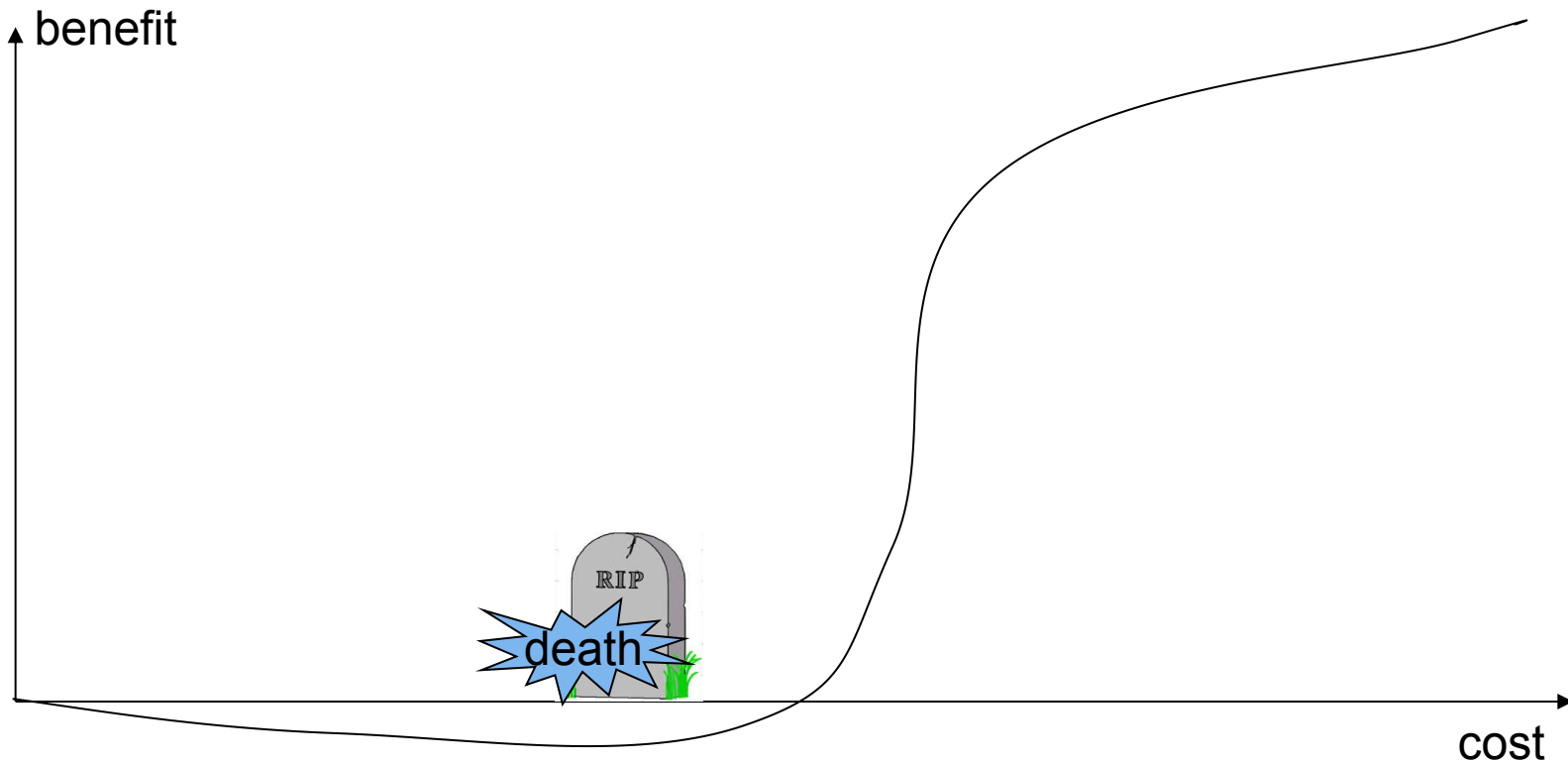
知识图谱



阵亡名单



“重”的知识项目



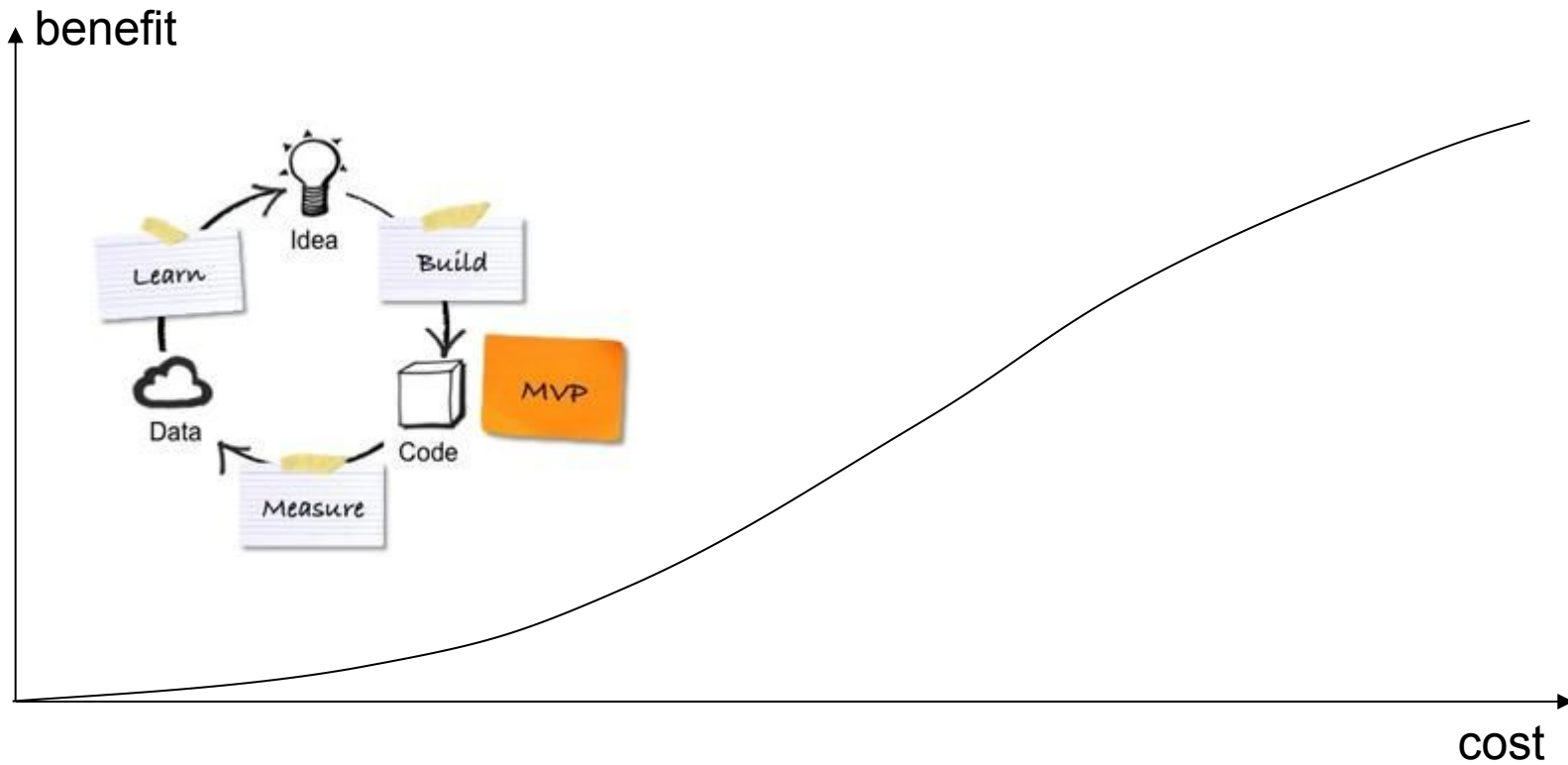
Twine

The screenshot shows the Twine website interface. At the top, there's a navigation bar with links like 'My Interests', 'My Profile', 'Explore', and 'Inbox'. A search bar is present with the text 'Find Everything by name or interest'. The main content area is titled 'Web 3.0 - Semantic Web' with a blue arrow pointing to it from the word 'Subject' in red. Below the title, there's a 3D Rubik's cube image and a description: 'This twine is about the emerging Semantic Web -- and Web 3.0, the next-evolution of the Web.' It also shows '3870 Items | 22545 Members' and a list of tags including 'Web', 'World Wide Web', 'internet', 'dataweb', 'Web 3.0', 'Semantic Web', 'semantics', 'Web 3G', 'linked data', 'hyperdata', 'GGG', 'Giant Global Graph', 'world wide database'. A red arrow points to the 'Items' tab, which is labeled 'Items' in red. The 'Recent Activity' section lists several items, including 'The Sweet Compendium of Ontology Building Tools » AI3:::Adaptive Information' and 'Case Study: Use of Semantic Web Technologies on the BBC Web Sites'. On the right side, there are buttons for 'Invite People', 'Leave This Twine', 'Started Oct. 4, 2007', 'Rules of this twine', and 'Post Items by Email'. A 'Recommended Twines' section shows links to 'Linked Data', 'Semantic Web', and 'Apps :: On Semantic Web & Related ...'. At the bottom right, there's a 'People (16)' section listing 'Alex Iskold'.

Twine的尸检报告

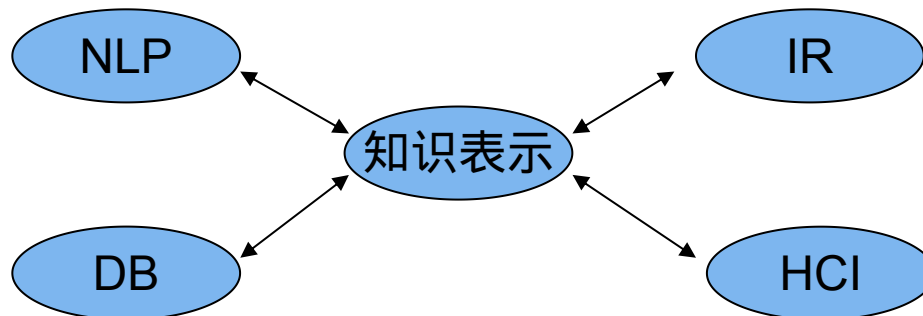
- 一次只做一件事，革新的步子不要太大。
 - 开发分布式语义数据库，Web级别的语义搜索，对于一个40个人的小公司，战线太长了。
- 不要坚持和W3C标准兼容。
- 图数据库是未来，但不要指望一步到位，要演化。
- 产品功能要集中，不要挑战用户稀缺的注意力和理解力。
- 产品和市场要循序渐进，周期要切短。

“轻”的知识项目

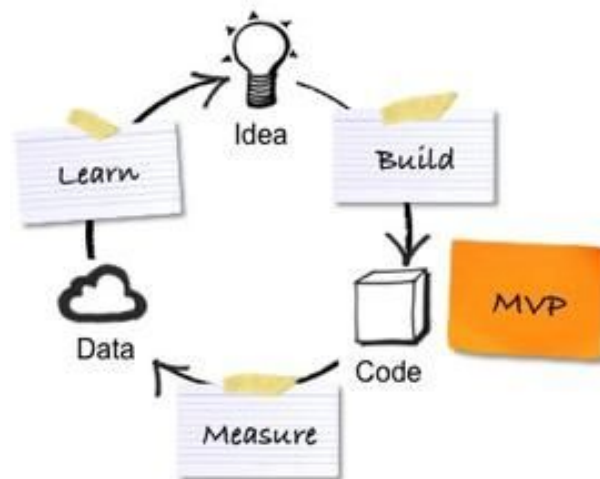


降低成本

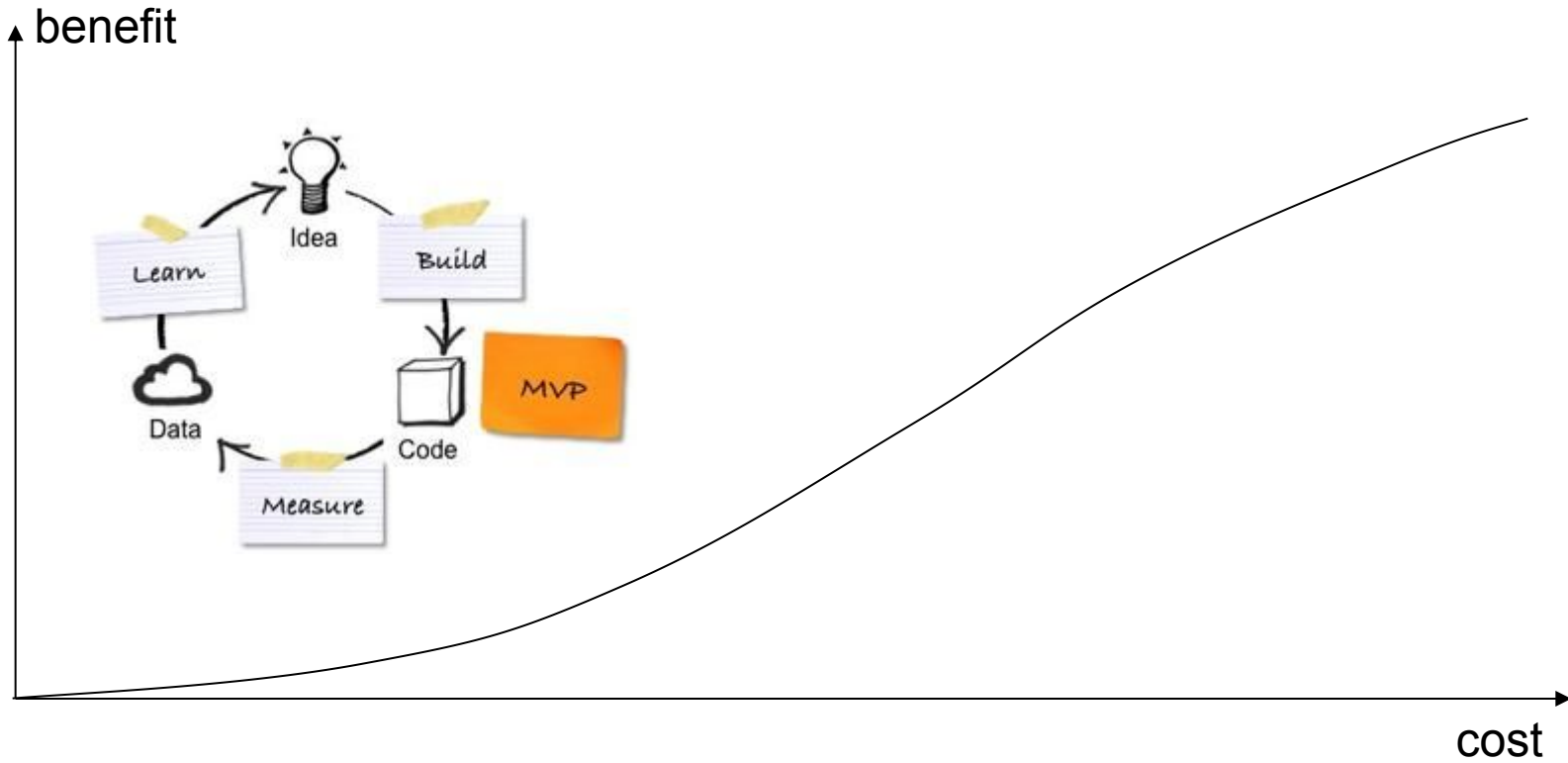
- 依托成熟技术



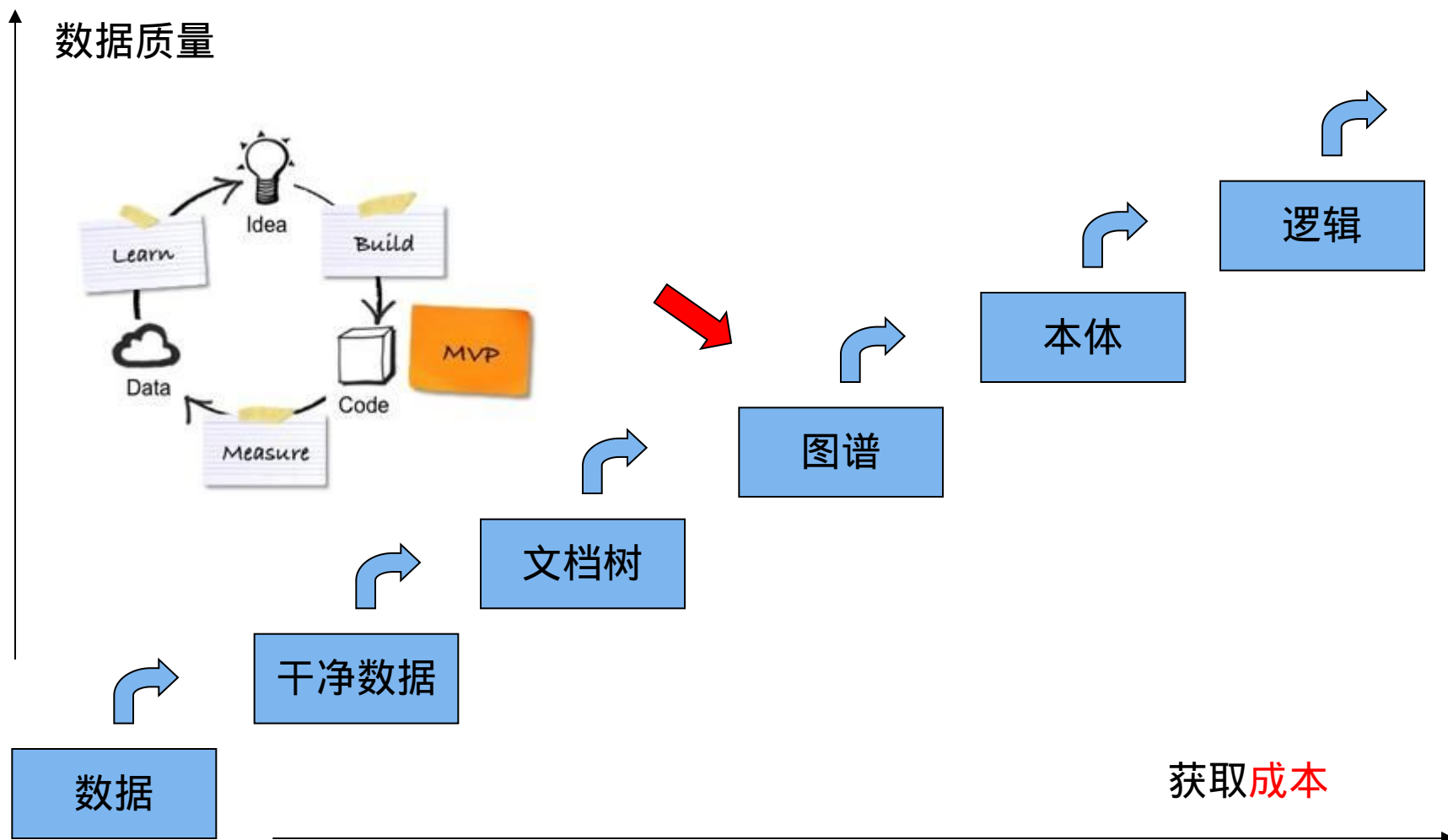
- 迭代



迭代！



迭代！



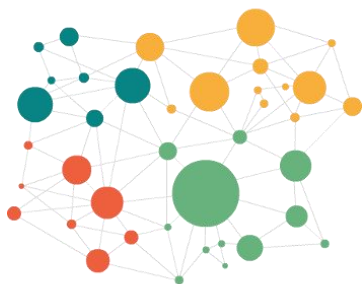
（回顾）



知识提取
（自然语言处理，机器学习）



知识存储
（数据库）



知识表现
（逻辑，语义网）



知识检索
（信息检索，人机交互）

知识提取

知识提取：为人民服务



欧阳询

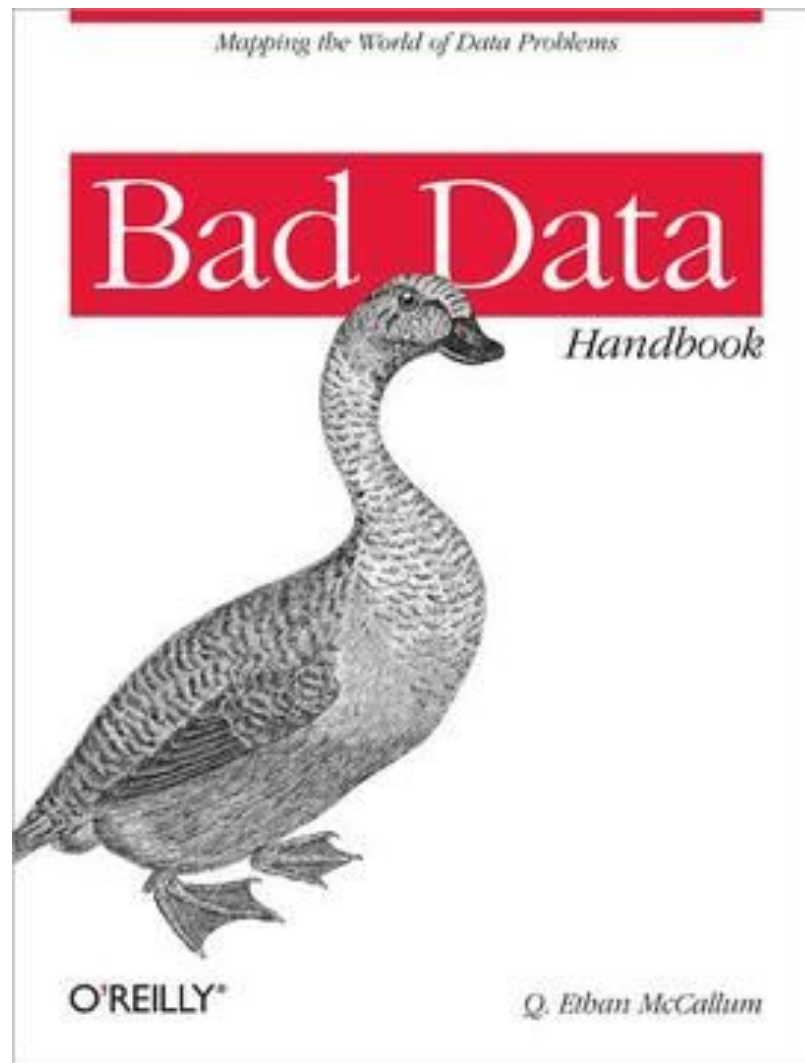
知识提取

- 知识提取是要解决结构化数据生成的问题。
- 但是广义上讲，知识提取是数据质量提升中的一环，各种提升数据质量的方法，都可以视为某种知识提取。
- 学术上一般是用自然语言处理的方法，但在实践中通常是利用规则。

知识提取的多种方法

- 知识提取的成本
- 数据清洗
- Regex, regex, regex ! (正则)
- 结构提取
- 浅度解析Shallow parsing
- 实体提取和链接
- 关系抽取

数据清洗=80%的工作



数据清洗：为人民服务

Table of Contents	Product Details	About the Author	Colophon
Time Series Data			
Conclusion			
Chapter 3 : Data Intended for Human Consumption, Not Machine Consumption			
The Data			
The Problem: Data Formatted for Human Consumption			
The Solution: Writing Code			
Postscript			
Other Formats			
Summary			
Chapter 4 : Bad Data Lurking in Plain Text			
Which Plain Text Encoding?			
Guessing Text Encoding			

正则表达式

- 正则语法的反人类（关键字缩写，没有缩进、空格）容易让我们把注意力集中到语法本身上。
- 正则是一门语言，我们不仅要学习它的语法、词汇，更需要用它写出了漂亮的文章。
- 在自然语言处理中，正则只是不太友好的强大工具，重要的是对自然语言的思考和理解。

正则自我修养

- 自然语言理解能力：句法，变形，同义词，复杂性理解和处理能力
- 工具：编辑器，快捷键，正则可视化
- 代码库：统计排序，相似度比较，常用词典，常用词表，句法分析工具，符号级数据清理，编码处理工具，聚类算法

正则误区

- 不要想利用已有的词典解决问题。
- 不要想建立自己用的通用词表。
- 现有NLP工具要在适当场合，恰当地使用。
比如：词性，实体识别，句法分析
- One more thing ... 规则离不开统计，机器学习是正则表达式的基友。

文章结构是最好的语义

- Structure extraction
 - 语文训练是人们最熟悉的写作方式，也是最低成本的数据结构化方式
 - 文章结构本身有巨大的价值。例如：基于股转书的自动报告生成。
 - 我们基于Wikipedia制作了“WikiMeme”数据库
- 其他技术
 - Shallow parsing
 - Entity extraction, entity linking
 - Relation extraction

知识表现

知识表现：为人民服务



欧阳询

逻辑、本体、推理以及实战工具与方法

王丛

文因互联 - Memect

2016 年 6 月 17 日



https://github.com/memect/kg-beijing/blob/master/2016-06-19_KR%E8%AE%B2%E4%B9%89.pdf

知识表现

- 知识表示（Knowledge Representation, KR, 也译为知识表现）是如何将结构化数据组织，以便于机器处理和人的理解的方法。
- 从结构推导出新的结构，这就是推理。
- 经典的教科书中的KR，主要关注的是如何方便机器处理。但是在现实的工程中，如何方便人的理解也是极为关键的。在工程实践中，人才是知识不能被处理好、不能快速交换、不能规模化的核心。

数据优先，逻辑靠边

- 知识表现的瓶颈不在于机器处理能力的不足，而在于人的认知能力的不足。
- 因此，我们在学习知识表现方法的时候，要始终牢记知识的可读性、可维护性要远远比它的表达力、计算速度重要。
- 知识是为人阅读而设计的，只是偶尔被机器执行。

从数据表现到知识表现

- 什么是数据和结构？
- 非结构化数据的价值
- 结构化数据的价值
- 数据建模的代价
- 可读知识的艺术
- 命名资源
- 定位资源
- 关联资源
- 语法: XML, JSON, YAML, RDF, OWL, Python etc.

知识表现与知识交换

- Portablility
- Protocol Buffers and Thrift
- Email and MIME
- JSON-RPC
- XMPP and Google Wave Protocol
- REST API design
- Some most important data APIs

知识表现与知识推理

- Cost of reasoner and index (IR, DB, KB)
- Just-in-time knowledge
- Practical rule modeling
- Inference as graph operations
- Inference using databases
- Inference using functional programming

“大”知识

- Measuring semantics in data
- Small is beautiful in knowledge
- Small knowledge: in-memory graph models
- Big knowledge: knowledge bases on clusters
- Datasets: Freebase, DBPedia, LOGD, Factual etc.

知识表现的数据结构

- 不是所有的数据结构我们会用知识工程的方法来处理
 - 集合、哈希表、队列、堆栈、链表等等，丢给软件工程
 - 二维表，丢给数据工程
- 传统上，我们把那些能够导出新的关系的关系（比如“爸爸的爸爸是爷爷”，这里面“爸爸”和“爷爷”都是关系）看成知识。
- 知识表现的数据结构，一般来说是那些“复杂”的结构，最常见的就是图（graph）和树（tree）。

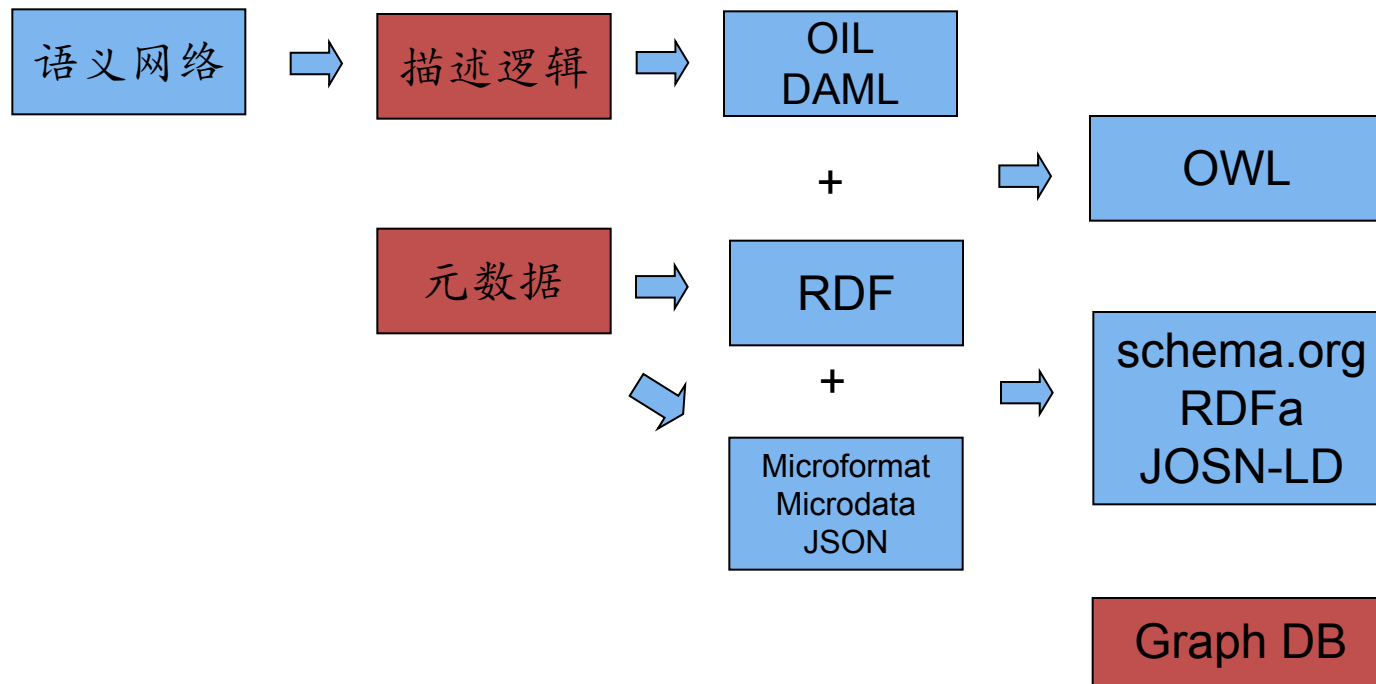
知识表现为图

- 知识表现的图，是“有类型的边”（typed edge）
 - 分析方法和一般的图论和社交关系图谱中分析的无类型的边很不相同。
- 语义网试图给链接加上类型说明，
 - 如某人主页声明工作于某公司，这就有“工作”关系。
 - 这里，类型就是边的“元数据”（metadata）
- 在应用中还需要给边和节点，添加更多的元数据，这就形成了有类型的边构成的图谱结构，上面每个边和每个节点都拥有元数据。

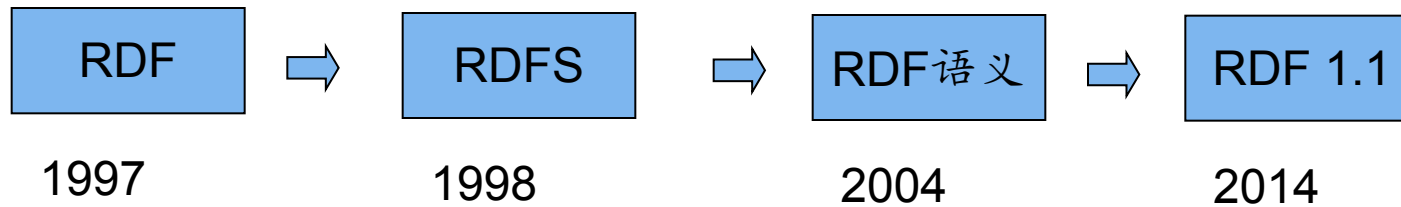
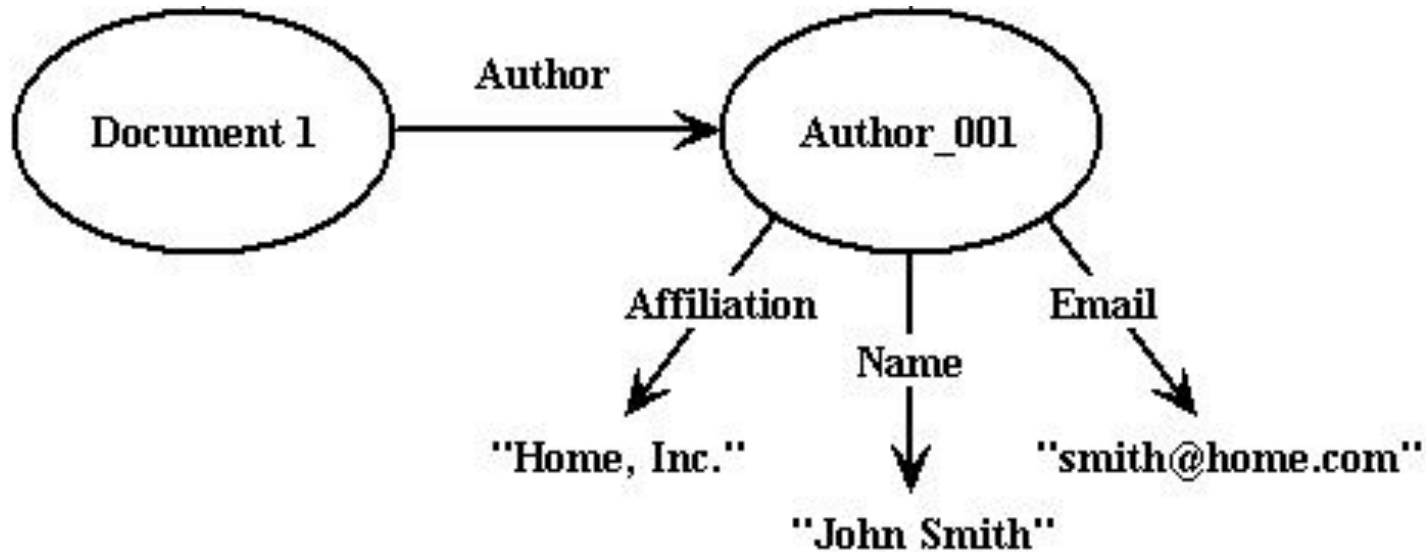
图的两个流派

- 一个是RDF图，一个是属性图（Property Graph）。
- RDF图是W3C的官方标准，得到了政府资金和一些大公司的大力支持，但是最终市场表现平平。
 - 科学顶层设计出来的
- 属性图是草根自发的，最终得到了市场的认可，现在主要是中小企业、创业公司在用。
 - 工程实战中总结出来的
- 好的架构一般是总结出来的，不是凭空设计出来的。

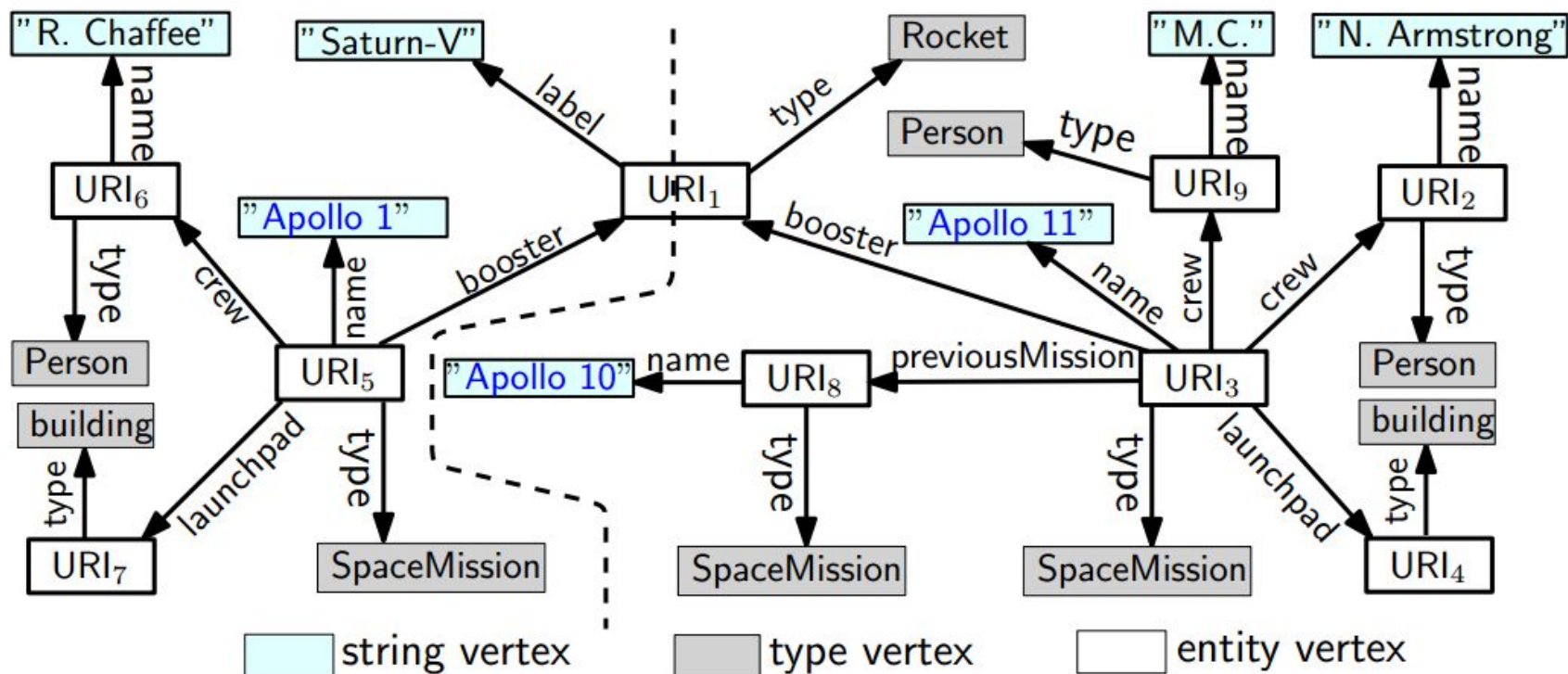
知识图谱KR的演化（简化）



RDF



数据变成三元组



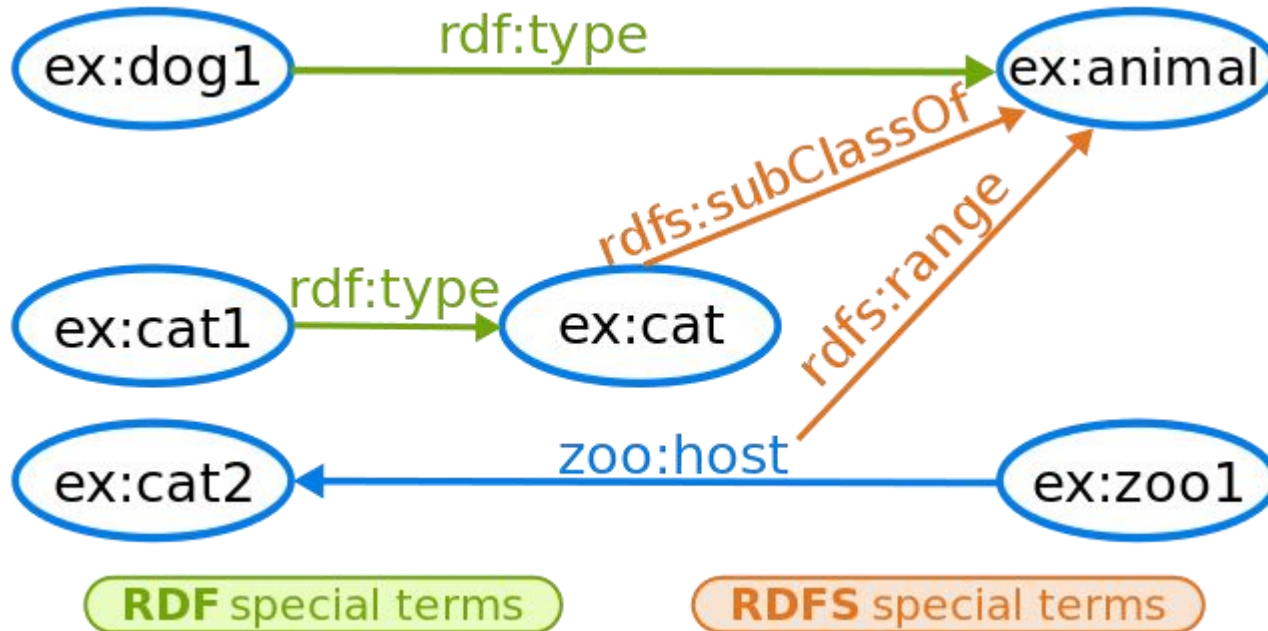
RDF

- RDF的基本单元是三元组（triple）。
- 每个三元组是（主语 谓语 宾语）这样的元组 tuple。主谓宾的取值称为“资源”（Resource，也就是RDF里的R）。
- 资源可以是一个网址（URI），一个字符串或数字（严格来讲都是带类型的字符串，称为 literal），或者一个“空节点”（blank node）。
- 有两种特殊类型的资源。rdfs:Class代表类。rdf:Property代表二元关系。有一种特殊的关系叫 rdf:type，声明一个资源属于某一个类。
- 主谓宾有一些限制，即其“schema”

RDF

- 一个三元组就是一个关系。在RDF里我们可以声明一些规则，从一些关系推导出另一些关系。这些规则我们称为“schema”，所以有了RDFS（RDF Schema）。
- 这些规则用一些词汇（可以类比编程语言里的保留字，不过RDF里任何词汇都可以被重定义和扩展）表示，如subClassOf subPropertyOf domain range。
- RDF里的推理规则有十几条，其中最常用的大概就是父类子类关系（subClassOf）。

RDFS



图片来自 https://en.wikipedia.org/wiki/RDF_Schema

RDF只提供了概念和关系的基本描述能力，并没有推理的能力。
RDFS则提供了简单的推理“schema”

RDF语义

http://www.slideshare.net/baojie_iowa/rdf-semantics

RDF Semantics

by Patrick Hayes
W3C Recommendation

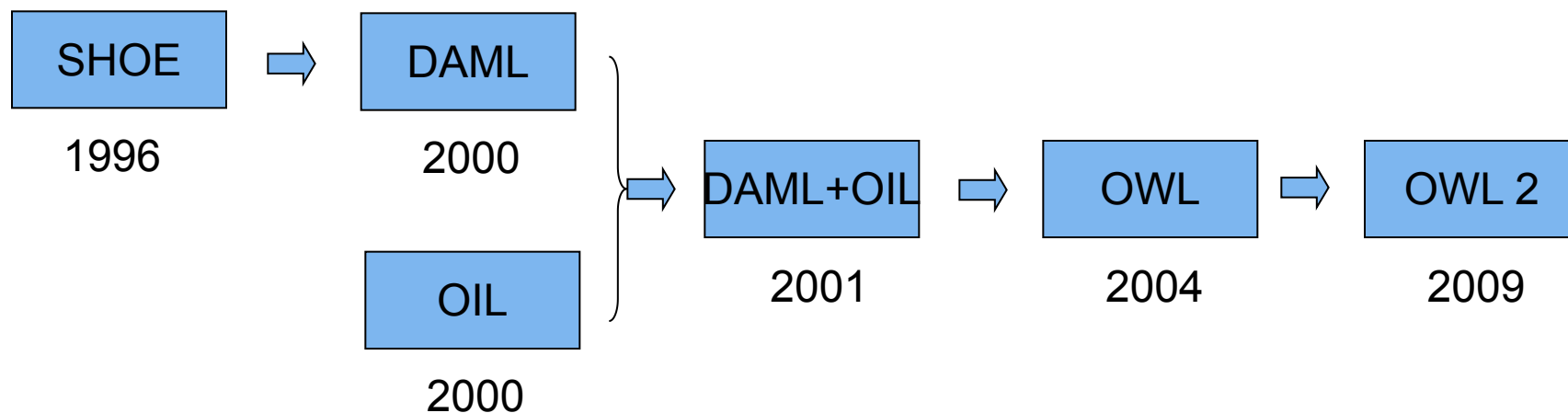
<http://www.w3.org/TR/rdf-mt/>

Presented by Jie Bao
RPI
Sept 4, 2008

Part 1 of RDF/OWL Semantics Tutorial
[http://tw.rpi.edu/wiki/index.php/RDF and OWL Semantics](http://tw.rpi.edu/wiki/index.php/RDF_and_OWL_Semantics)

RDF引入（一个高阶的可能会产生悖论的）语义也许是一个错误！

OWL



- OWL提供了一组适合Web传播的描述逻辑的语法。
- 其开放世界语义当初被认为适合刻画Web的开放性

OWL的语义

http://www.slideshare.net/baojie_iowa/2008-0906-owl-full-semantics

OWL Full Semantics

-- RDF-Compatible Model-Theoretic Semantics

by Peter F. Patel-Schneider, Patrick Hayes and Ian Horrocks
W3C Recommendation, 2004

<http://www.w3.org/TR/owl-semantics/rdfs.html>

Presented by Jie Bao

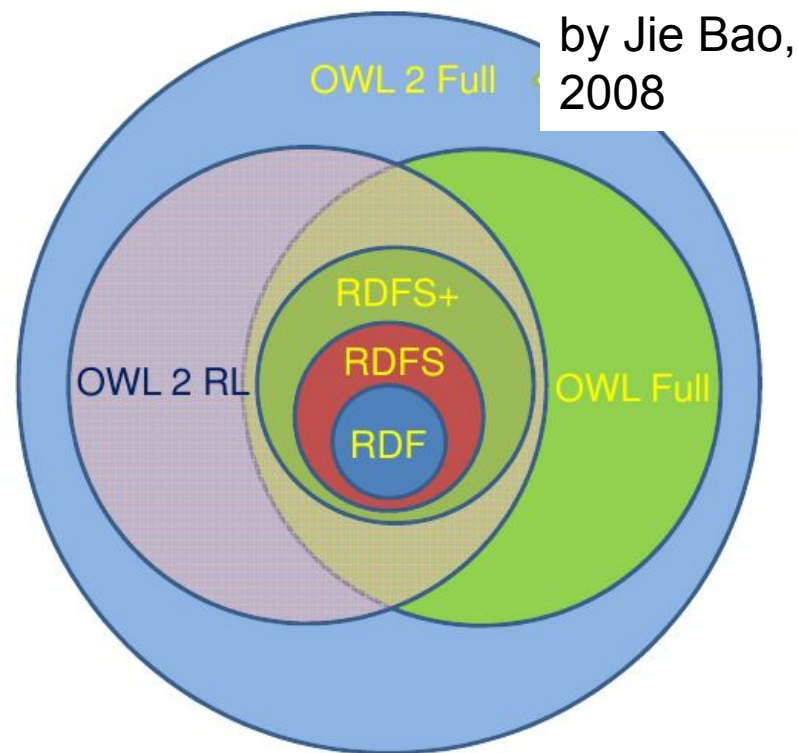
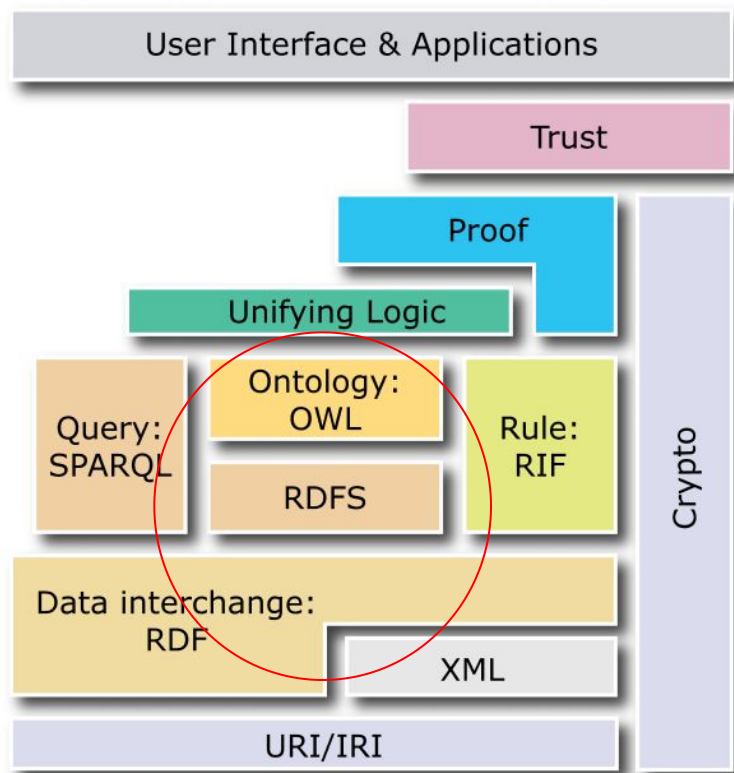
RPI

Sept 11, 2008

Part 2 of RDF/OWL Semantics Tutorial

[http://tw.rpi.edu/wiki/index.php/RDF and OWL Semantics](http://tw.rpi.edu/wiki/index.php/RDF_and_OWL_Semantics)

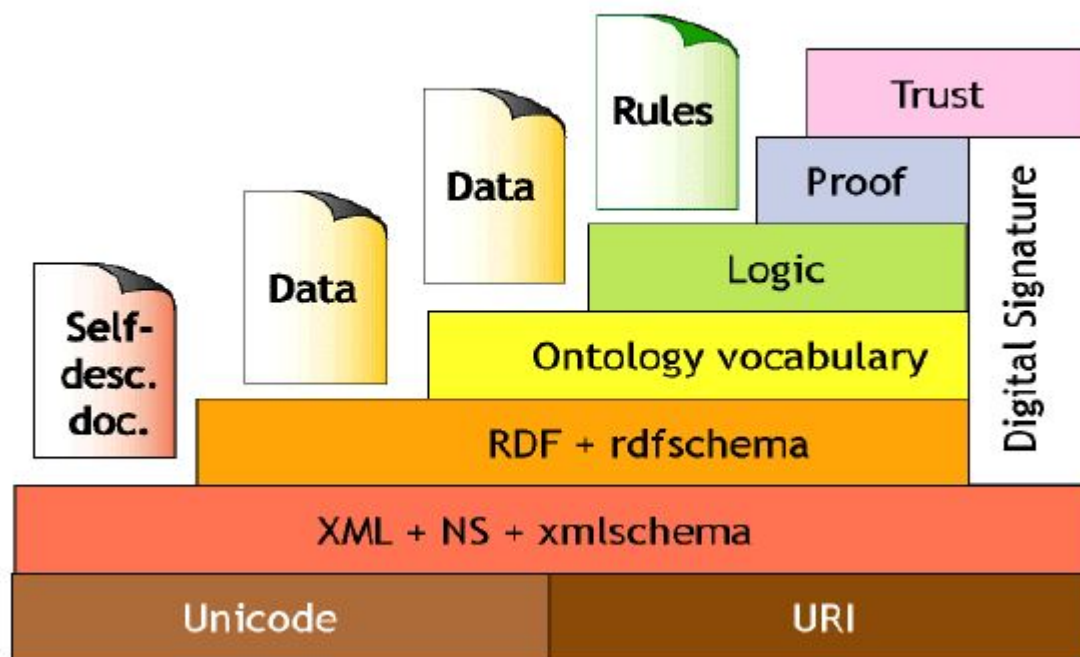
RDF和OWL的关系



RDF和OWL的进阶介绍

- RDF和OWL语义
 - <http://blog.memect.cn/?p=871> 我的两个ppt, 讲解了RDF和OWL的模型论语义
- RDF和OWL的语法和基本使用, 可以看官方的文档
 - RDF 1.1 Primer <https://www.w3.org/TR/rdf11-primer/>
 - OWL 2 Primer <http://www.w3.org/TR/owl2-primer/>
- 语义网速查表
<http://ebiquity.umbc.edu/resource/html/id/94/> (丁力写的)
- OWL语法速查表 <https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/> (我写的)

语义网层次蛋糕



历史版本的回顾 <http://baojie.org/blog/2011/04/25/layer-cake-sw-2/>

语义网层次蛋糕...3D

The Semantic Web Technology Stack
(not a piece of cake...)

Most apps use only a subset of the stack

Querying allows fine-grained data access

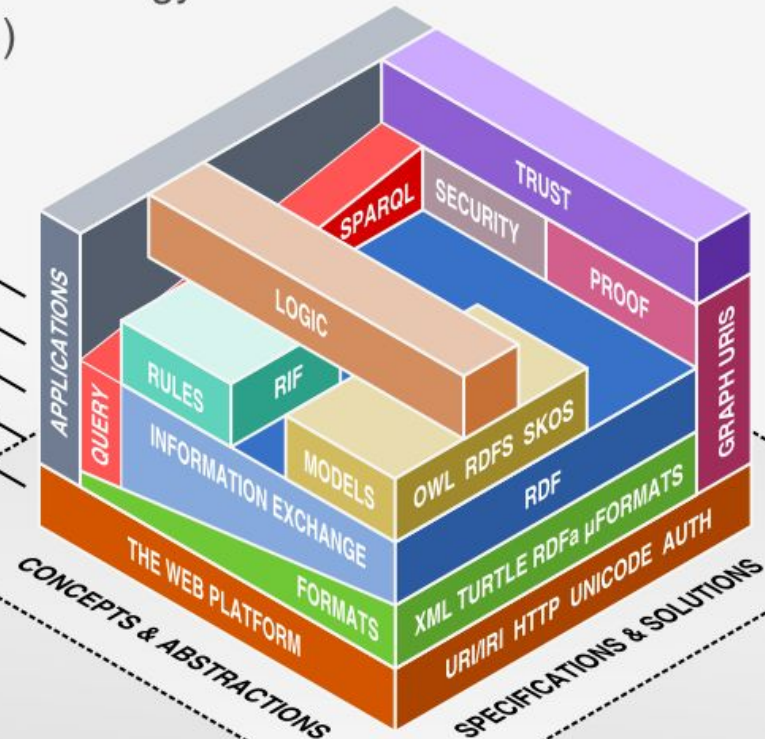
Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small
selection of technologies

LINKED DATA



RDF的Python工具

- Python里的rdflib包可以很方便处理RDF。推荐按rdflib的文档过一遍例子，加深对RDF的理解 <http://rdflib.readthedocs.io/en/stable/>
- 这个2011年的综述，提到了各种RDF相关的Python包：RdfLib RdfAlchemy Fuxi ORDF Django-RDF Djubby Redland SuRF PySparql Sparta Oort Virtuoso pySesame pynappl HTTP4Store py4s
 - Survey of Pythonic tools for RDF and Linked Data programming <http://www.michelepasin.org/blog/2011/02/24/survey-of-pythonic-tools-for-rdf-and-linked-data-programming/>
- 本文没包括的还有：rdfQuery PySWIP pyDatalog PyLog FLiP seth sparrow pymantic pyRDFa djubby pySPARQL。

JSON-LD

- JSON-LD是RDF的JSON语法，其中LD代表Linked Data。
- 它要解决的是RDF没有好的Web兼容语法问题。
 - 经典的RDF语法是XML的，不仅罗嗦和丑陋，也集成了XML“重”的一些特征，适合“企业级”应用。
- JSON-LD就是想提供一个和互联网事实标准更兼容的、“轻”的语法。

JSON-LD

JSON ✓	Copy JSON	YAML ✓
<pre>1 { 2 "@context": { 3 "@vocab": "http://schema.org/", 4 "first_name": "givenName", 5 "last_name": "familyName", 6 "alias": "alternateName", 7 "email": "email" 8 }, 9 "first_name": "Benjamin", 10 "last_name": "Young", 11 "alias": "BigBlueHat", 12 "email": "byoung@bigbluehat.com" 13 }</pre>		<pre>1 --- 2 "@context": 3 "@vocab": http://schema.org/ 4 first_name: givenName 5 last_name: familyName 6 alias: alternateName 7 email: email 8 first_name: Benjamin 9 last_name: Young 10 alias: BigBlueHat 11 email: byoung@bigbluehat.com 12 </pre>

JSON-LD基本思想

- 尽可能用对人友好的字符串来写作，而不是象在传统RDF里用难以理解的URI。为了解释字符串，就引入了@context，把字符串定义在一定的上下文下——这些上下文本身一般是URI。这是比XML domain更友好的设计，增强了可读性。
- 引入模块化组织，加强可读性和可维护性。传统的RDF的组织粒度太低，在三元组层面。JSON-LD把同一个主语的三元组组织在一个{}块下，方便写作和理解。块的主语可以用@id 属性声明。更高层面上它还提供了@graph声明，你可以把它理解成一个子模块，模块里的内容可以共享一些元数据（比如上下文和注释）。

对人友好

- JSON-LD体现了两个对人友好的特性：可读性和模块化。
- 第一代的Web知识语言如RDF和OWL，可以类比为知识的“汇编语言”，对机器很友好，对人不友好。
- Turtle和JSON-LD这类第二代语言，开始“高级化”，注意了方便人来写作和阅读，注意了引入适应人的认知需要的模块。

知识图谱的高级语言

- 我对Web上知识表示的展望
- 两个大胆假设
 - RDF是一门低级语言，只适合机器使用——如同机器语言或者汇编语言
 - 语义网需要一门高级语言，面向工程师（人），用来做大规模知识库的写作、重用

RDF的问题

- 用URL来寻址并不错。但是把精确寻址的任务交给人类，要求人类来设计URL，就如同在C编程中要求人类对每个变量赋予内存地址。RDF是一个“平坦”(flat)的语言，缺少内部的组织单元。
- RDF的语法，即使是Turtle，也没有可读性，理解和重用起来非常困难。
- RDF缺少“宏”或者构造高层次组织的能力。其实SPARQL弥补了一点，就是graph pattern；一些语言如SPIN，把graph pattern作为可重用的单元，甚至可以生成新的数据。如果把这个能力作为RDF原生的能力就好了。

高级语言要有什么特征

- 支持多粒度的知识/数据组织和重用
- 用字符串而不是URL来寻址。不追求addressing uniqueness, 而是probable and eventual addressing uniqueness
- 支持知识的分布式传输（按一定粒度）
- 使用目前主流程程序员熟悉的语法形式。
- 尽可能少重新发明轮子
- 支持结构化和非结构化数据的混合表达（RDF有Literal，不过那个太局限了）
- 这个语言的文档不要提什么“语义”（有几个程序员关心SQL的语义？），不要规定什么schema
- 把推理转化为图的操作或者编程语言内置的运算。在这之外的推理都先不考虑。
- 从一开始就设计成在cluster上能运行的语言
- 用程序员看的懂的语言和例子写文档。

知识存储

知识存储：为人民服务



欧阳询

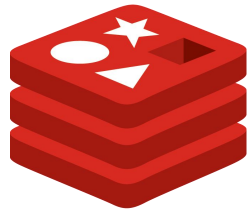
知识存储

- 当我们经过知识提取得到了结构化的数据，并选择了适当的知识表现语法后，下一步就是如何持久化存储这些数据
- 知识存储解决如何管理大量的结构化数据。
- 我们不讨论高大上的T级别、P级别的“企业级”知识存储。
 - 大多数情况下，我们只需要几G、几十G、几百G规模的数据，甚至更少，纯内存处理都可以。不需要去想“大数据”问题。

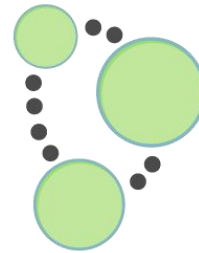
不拘一格

- 现代的关系数据库可能可以解决大多数需要知识图谱的场合。某些特殊场合，我们需要图数据库。
- 在一些场合，我们甚至不需要数据库做这些事情。如果你的数据只需要按“key”来查找而不是按值查找，也不需要“join”，那用文件系统就可以做数据后端。
- 稍稍复杂的处理是把key放在redis这样的键值数据库里来管

不拘一格



redis



Neo4j



PostgreSQL



Stardog



elasticsearch.

三种选择

- 带JSON扩展的关系数据库（首选）
- 图数据库（ TinkerPop stack, Neo4j, OrientDB ）
- RDF数据库
- 其他选择
 - Document database: MongoDB, Elastic Search etc
 - Graph database: Graph batch processing: Pregel, Hama, GraphChi etc

理解成本！

知识在数据库中的链接

- 知识图谱之所以是图谱，是因为一些节点会被另一些节点“引用”。各种数据库有不同的引用方式，会有不同的成本和收益。
- 三种方式字符串、外键、URI
 - 用“key”做数据查找，就是按“字符串”来引用数据。有一些用redis实现的图数据库，其实就是这种方法。字符串可读性很好，实现成本最低，但可能会出现歧义的情况。如果能把歧义控制在可接受的范围内，字符串是很不错的选择。
 - 在关系数据库里我们可以用“外键”来关键数据。其实这也是一种很灵活的方式，可以在一个context下面解释一个“key”，很容易扩展。
 - URI是语义网的思路，在整个web的context下来解释一个字符串，也就是URI网络资源定位。好处是数据互联性好，坏处是可读性不好，扩展性不好，和现有技术栈差别太大，最终实施成本很高。
- 如果我们有很好的“entity resolver”（实体解析器），那字符串就是很好的工程选择。没有的时候，关系数据库的外键也是不错的

PostgreSQL

- 关系数据库 + JSON 是最好的（小规模）知识图谱存储选择。可用工具多、稳定性好、速度快、可join、容易演化。
- 优先推荐使用PostgreSQL 9.3以后版本，直接支持JSON
<https://www.postgresql.org/docs/9.4/static/datatype-json.html>
- 用Psycopg包操作PostgreSQL
<http://initd.org/psycopg/docs/>

PostgreSQL

PostgreSQL 与 JSON 实战

段清华

文因互联 - Memect

2016 年 6 月 26 日



<https://github.com/memect/kg-beijing/tree/master/KG-PostgreSQL/KG-PostgreSQL>

图数据库



- 两种图建模：RDF图和属性图（Property Graph）有融合的趋势
- 推理可以看成图上的遍历和构造
- 属性图标准Tinkerpop已经获得很大的市场成功

Neo4j

- Neo4j 是目前用的最多的图数据库
 - <https://neo4j.com/>
 - Python接口 <https://neo4j.com/developer/python/>
- 两个查询语言
 - Cypher <https://neo4j.com/developer/cypher-query-language/> Neo4j自己的查询语言，更符合图的特点
 - Gremlin
<http://gremlindocs.spinalmallette.documentup.com/> 图数据库的“标准”查询语言

OrientDB

- 我个人最喜欢的是OrientDB，我认为完美达到我需要的知识图谱数据库的基本功能要求 1) 用类SQL查询语法，降低学习成本 2) 直接读写JSON，方便和Web API导入导出 3) 支持图的遍历和gremlin查询 4) 支持blueprints标准 5) 部署简单 6) 还在积极维护
 - <http://orientdb.com/orientdb/>
 - pyorient 二进制Python接口 <https://github.com/mogui/pyorient>
 - compass HTTP Python接口 <https://github.com/emehrkay/Compass> 项目已经停止维护了，不过还能用
- OrientDB的查询语言就是SQL，加一些图的扩展，非常容易入手。
- 但是OrientDB性能调优需要比较多的经验，由于用的人比neo4j少，出现问题不太容易解决。

Stardog

- 如果需要查询RDF，或者某些SPARQL比SQL更好使的场合，试试Stardog
 - <http://stardog.com/>
- 一个不错的SPARQL教程
 - <http://www.cambridgesemantics.com/semantic-university/sparql-by-example>
- Stardog也支持图数据库查询语言Gremlin
- 不推荐用Virtuoso，Sesame，Jena这些。都是上一代的老产品，复杂不好用。

知识检索

知识检索：为人民服务



欧阳询

知识检索人机交换

（详细检索技术请见林德康老师部分）

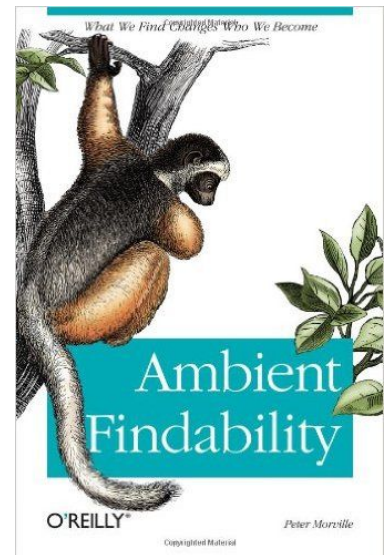
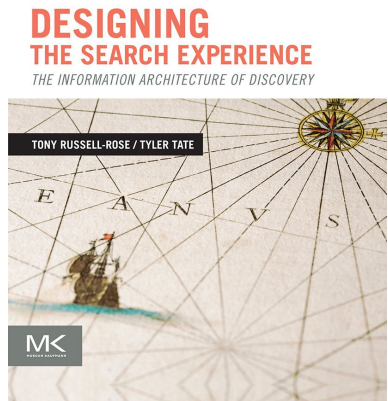
元数据如何促进人机交换？

User Interaction

- It's about **people**, not machine
- Guided data exploration and discovery:
why semantics is part of the solution
- Query formulating
- Faceted Browser
- Mobile search, Voice interface and
personal assistants

Search and Findability

- Database vs search engine
- Cost of inverted index
- Extend inverted index to model semantic relations
- Understanding user queries: from keywords to sentences
- Faceted search: Elastic Search and Solr
- Graph search



Visualization

- Cognitive background
- Exhibit and others
- D3 (and other JavaScript lib)
- NetworkX (and other Python lib)

知识检索：数据人机流动

- 动态的数据建模，不需要什么schema或者shema可以**快速演进**（这是和数据库和XML比的好处）
- 让数据**流动**起来，而不是封固在一个个的网站或者应用里。
- 支持更多的**任务自动化**。（怎么做到？一些数据可机器处理，一些推理，一些查询）

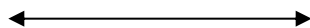
流动性

快！

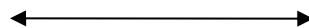


天下武功，无坚不破，唯快不破

数据摩擦力



人-机界面
的摩擦



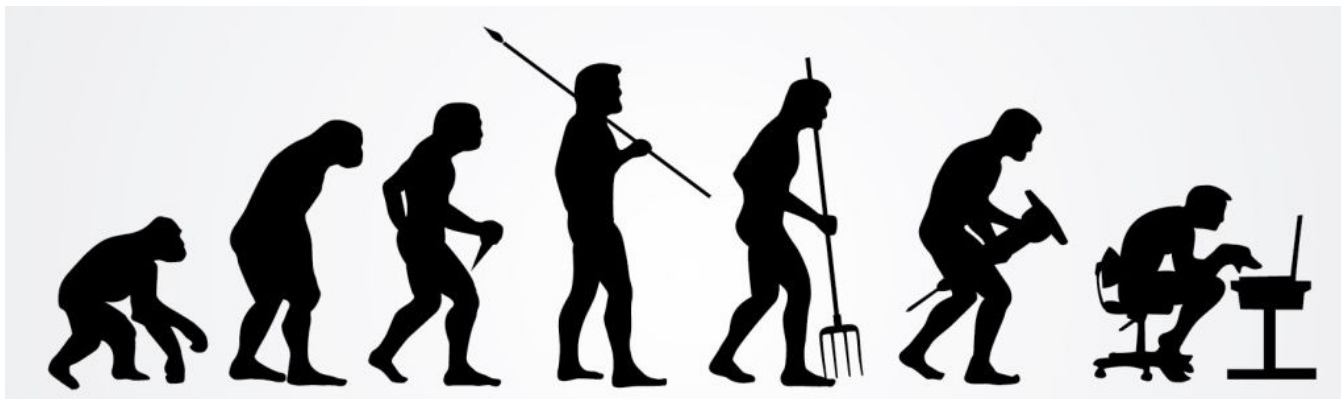
机器-机器界面
的摩擦



人-人界面
的摩擦
(今天不讨论)



人-机界面的数据摩擦



人：

- 5万年未大变的硬件
- 通讯速度慢 几个比特/秒
- 懒
- 依赖直觉

电子表格：减少人机数据摩擦

VisiCalc



C11 (L) TOTAL C1

25

	A	B	C	D
1	ITEM	NO.	UNIT	COST
2	---	---	---	---
3	MUCK RAKE	43	12.95	556.85
4	BUZZ CUT	15	6.75	101.25
5	TOE TONER	250	49.95	12487.50
6	EYE SNUFF	2	4.95	9.90
7				
8			SUBTOTAL	13155.50
9		9.75%	TAX	1282.66
10				
11			TOTAL	14438.16

数据库关注的是机器效率问题，不是人和机器一起工作的效率问题。
电子表格解决的是人机系统的效率问题。

机器-机器界面的摩擦



手段

- 减少数据流动摩擦，加快数据流动速度的手段，也就是着眼减少人机界面的摩擦和机器与机器的摩擦这两个方面。
- 这两个方面都是依靠为网页添加可查询的注释。

例子：寄包裹



怎么样找到
小黄人？



例子：寄包裹

一个不太好的URI

套男女情侣
红色:170/

单位名称 如果俄将变成回忆

广西壮族自治区 南宁市 横县 横州镇海棠桥菜市场近公厕那边两摊猪肉中间摊卖青菜的一个女人

180 3

邮政编码

联系电话(非常重要)

邮政编码

当日达 ☐ 次日达 ☐ 隔日达 ☐ 经济快件 ☐

保价 是 ☐ 否 ☐ 保价金额 万 仟 佰 元(小写)¥

重量(kg)

长 X 宽 X 高 cm

商品 ☐

签收短信通知 ☐ 资费 ¥ 派送 ☐ 资费 ¥5元/票

支付方式 现金 ☐ 协议结算 ☐

收件人签名:

代收人姓名(签字):

证件类型: 证件号: 年 月 日 时

5185 1211

横县

©广西日报 weibo.com/gxrb2013

URI的两个作用



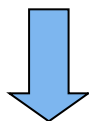
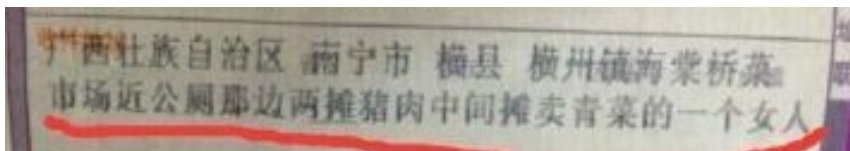
寻址



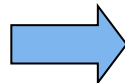
命名

机器可处理的自描述的URI

machine-processable, self-describing dereferenceable URI



<http://j.map.baidu.com/6bbv7>



降低机器-机器界面的摩擦

元数据减少人机界面摩擦



作业：手绘二维码及求你的心理阴影面积

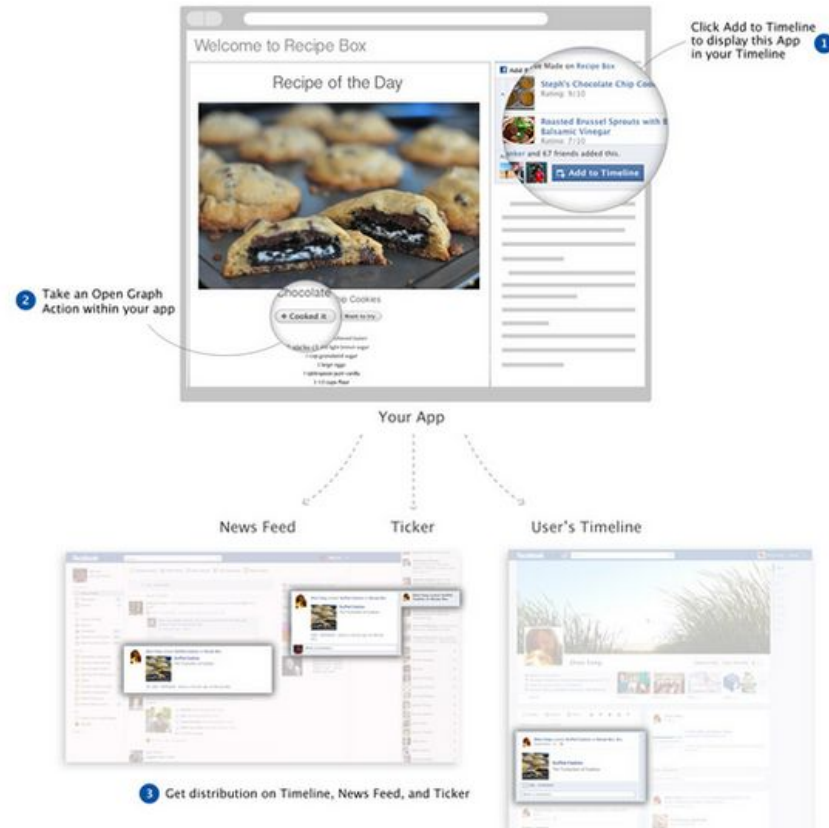
元数据减少人机界面摩擦

探索

易见

理解

易见： Twitter Card, Facebook OpenGraph



探索：Faceted Browser

The screenshot displays the LinkedIn search results page for the keyword "linkedin". The interface is designed as a faceted browser, allowing users to refine their search results using various filters. A red box highlights the "Refine By" sidebar on the left, which contains the following sections:

- Location**: Includes checkboxes for "All Locations", "United States (20762)", "San Francisco Bay Area (2444)", "India (2303)", "United Kingdom (2101)", and "Greater New York City Area (2079)".
- Relationship**: Includes checkboxes for "All LinkedIn Members", "1st Connections (192)", "2nd Connections (2706)", "Group Members (860)", and "3rd + Everyone Else (30390)".
- Industry**: Includes checkboxes for "All Industries", "Information Technology and Services (4219)", "Marketing and Advertising (2709)", "Staffing and Recruiting (2301)", "Computer Software (1947)", and "Internet (1782)".
- Current Company**: A section for filtering by current employer.
- Past Company**: A section for filtering by previous employers.
- School**: A section for filtering by educational institutions.
- Groups**: A section for filtering by LinkedIn groups.
- Profile Language**: A section for filtering by the language of the profile.

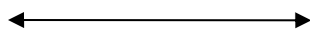
The main content area displays a list of search results, sorted by "Relationship" and viewed in "Basic" format. The results show 33,778 matches for the search term "linkedin". The first few results are:

- Reid Hoffman**: Partner at Greylock, San Francisco Bay Area | Internet. In Common: 120 shared connections, 2 shared groups.
- David Hahn**: Director of Product Management, LinkedIn, San Francisco Bay Area | Internet. In Common: 83 shared connections.
- Candy Chastain Mielke**: Consultant at LinkedIn, Colorado Springs, Colorado Area | Internet. In Common: 128 shared connections, 2 shared groups.
- Wade Burgess**: Business Leader, Greater Omaha Area | Internet. In Common: 44 shared connections, 2 shared groups.
- Alex Vauthey**: Director of Engineering, Monetization at LinkedIn, San Francisco Bay Area | Information Technology and Services. In Common: 103 shared connections.
- Ruslan Belkin**: Sr. Director of Engineering at LinkedIn, San Francisco Bay Area | Computer Software. In Common: 95 shared connections.
- Elizabeth Reaves**: Product Manager at LinkedIn, San Francisco Bay Area | Internet. In Common: 92 shared connections, 4 shared groups.
- Riccardo Ferretti**: Principal Engineer at LinkedIn, San Francisco Bay Area | Computer Software. In Common: 121 shared connections.
- Shirley Xu**: Data Scientist at LinkedIn, San Francisco Bay Area | Internet. In Common: 108 shared connections.

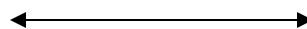
The right sidebar contains "Search tools" (Advanced search, Saved Searches), "Ads by LinkedIn Members" (S&OP Summit), and "Channel Not Performing?" (On demand solutions for channel performance optimization).

[illegible]

以人为核心



人-机界面
的摩擦



机器-机器界面
的摩擦



semantic
WEB



SEMANTIC
web

为人民服务



欧阳询



个人微信 baojie_memect

文因互联期待新的加盟者

- 数据经理
- 运维工程师
- 机器学习/NLP工程师
- 语义网工程师
- 搜索工程师

联系邮箱 contact@memect.co