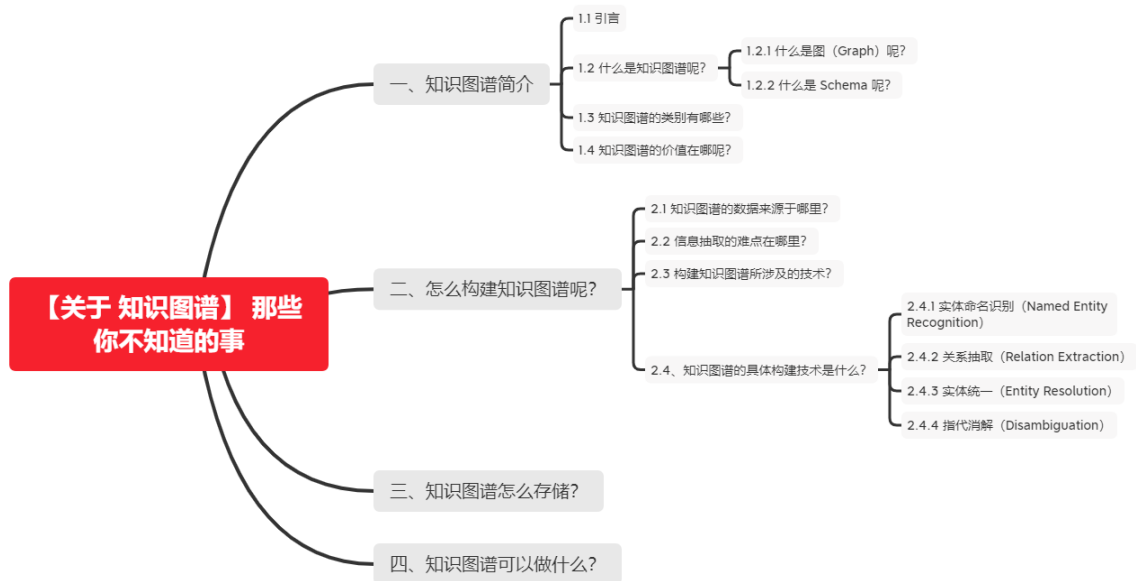


# 【关于 知识图谱】 那些你不知道的事



## 一、知识图谱简介

### 1.1 引言

从一开始的Google搜索，到现在的聊天机器人、大数据风控、证券投资、智能医疗、自适应教育、推荐系统，无一不跟知识图谱相关。它在技术领域的热度也在逐年上升。

早在 2010 年微软就开始构建知识图谱，包括 Satori 和 Probase；2012 年，Google 正式发布了 Google Knowledge Graph，现在规模已超 700 亿。目前微软和 Google 拥有全世界最大的通用知识图谱，Facebook 拥有全世界最大的社交知识图谱，而阿里巴巴和亚马逊则分别构建了商品知识图谱。



图 1 业内布局

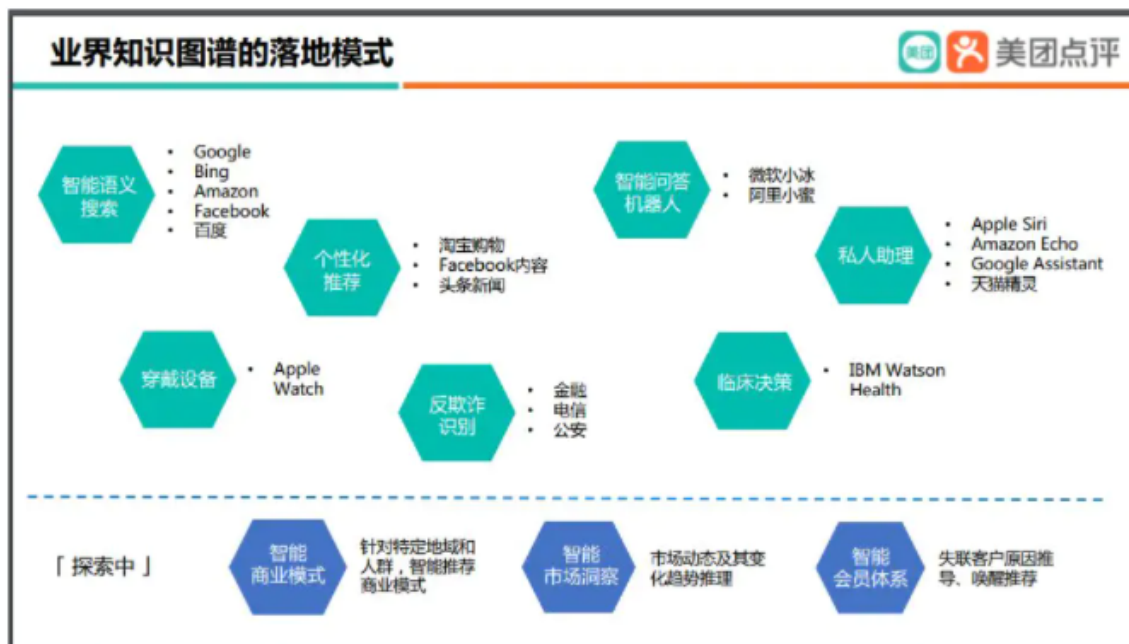


图 2 业内应用

本章以通俗易懂的方式来讲解知识图谱相关的知识、介绍从零开始搭建知识图谱过程当中需要经历的步骤以及每个阶段。本次组队学习还将动手实践一个关于kg在智能问答中的应用。

## 1.2 什么是知识图谱呢？

知识图谱是由 Google 公司在 2012 年提出来的一个新的概念。从学术的角度，我们可以对知识图谱给一个这样的定义：“知识图谱本质上是语义网络（Semantic Network）的知识库”。但这有点抽象，所以换个角度，从实际应用的角度出发其实可以简单地把知识图谱理解成多关系图（Multi-relational Graph）。

### 1.2.1 什么是图（Graph）呢？

图（Graph）是由节点（Vertex）和边（Edge）来构成，多关系图一般包含多种类型的节点和多种类型的边。实体（节点）指的是现实世界中的事物比如人、地名、概念、药物、公司等，关系（边）则用来表达不同实体之间的某种联系，比如人-“居住在”-北京、张三和李四是“朋友”、逻辑回归是深度学习的“先导知识”等等。

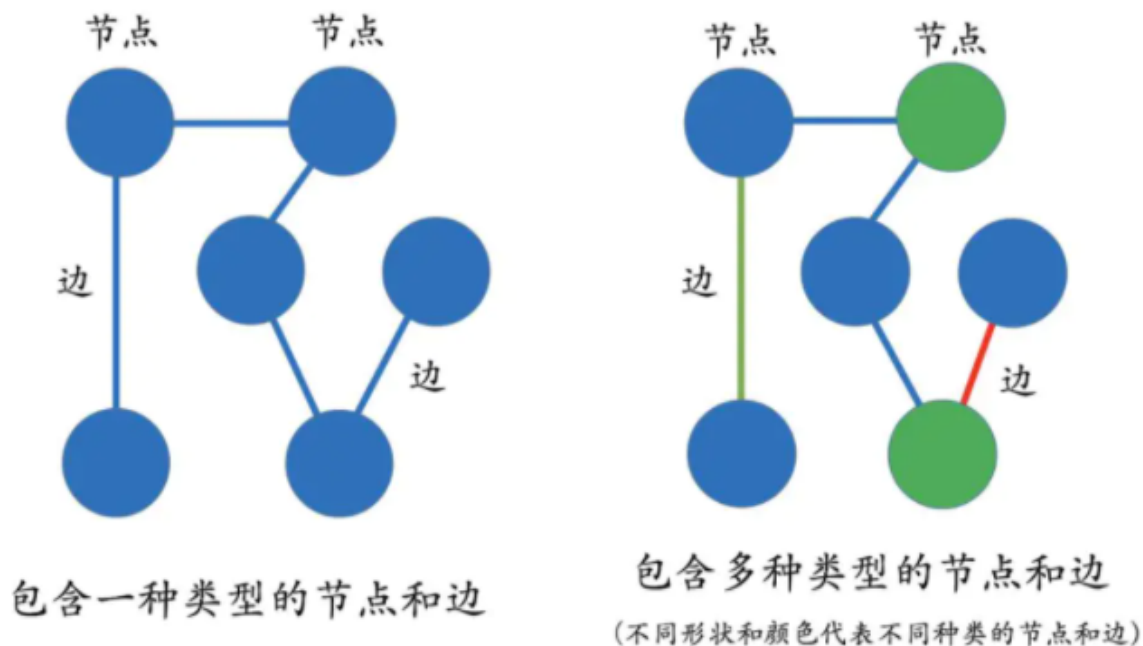


图 3 图 (Graph) 介绍

### 1.2.2 什么是 Schema 呢?

- 知识图谱另外一个很重要的概念是 Schema:
  - 介绍: 限定待加入知识图谱数据的格式; 相当于某个领域内的数据模型, 包含了该领域内有意义的概念类型以及这些类型的属性
  - 作用: 规范结构化数据的表达, 一条数据必须满足Schema预先定义好的实体对象及其类型, 才被允许更新到知识图谱中, **一图胜千言**
    - 图中的DataType限定了知识图谱节点值的类型为文本、日期、数字 (浮点型与整型)
    - 图中的Thing限定了节点的类型及其属性 (即图1-1中的边)
  - 举例说明: 基于下图Schema构建的知识图谱中仅可含作品、地方组织、人物; 其中作品的属性为电影与音乐、地方组织的属性为当地的商业 (eg: 饭店、俱乐部等)、人物的属性为歌手
  - tips: 本次组队学习不涉及schema的构建



图 4 Schema定义

## 1.3 知识图谱的类别有哪些？

先按知识图谱应用的深度主要可以分为两大类：

- 一是通用知识图谱，通俗讲就是大众版，没有特别深的行业知识及专业内容，一般是解决科普类、常识类等问题。
- 二是行业知识图谱，通俗讲就是专业版，根据对某个行业或细分领域的深入研究而定制的版本，主要是解决当前行业或细分领域的专业问题。

## 1.4 知识图谱的价值在哪呢？

从图5中可以看出，知识图谱是人工智能很重要的一个分支，人工智能的目标为了让机器具备像人一样理性思考及做事的能力 ->

在符号主义的引领下，知识工程（核心内容即建设专家系统）取得了突破性的进展 ->

在整个知识工程的分支下，知识表示是一个非常重要的任务 ->

而知识图谱又恰恰是知识表示的重要一环

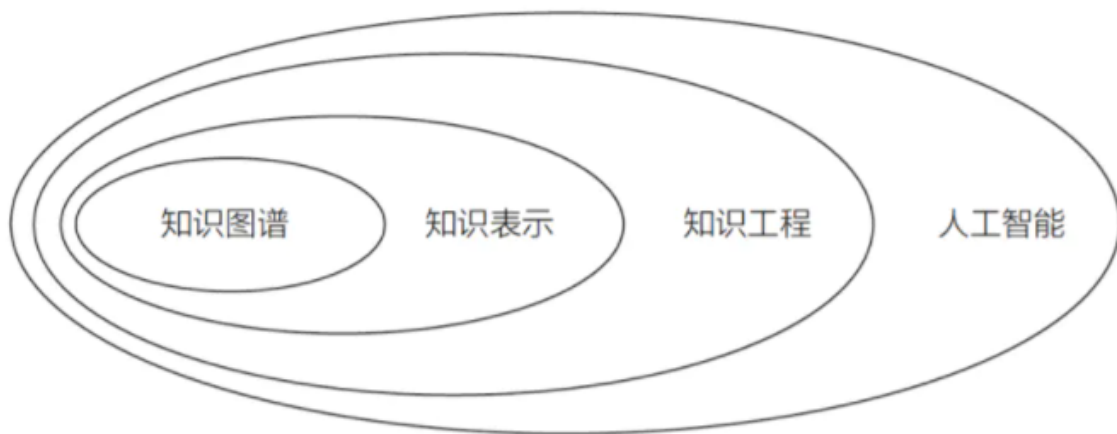


图 5 学科概念

## 二、怎么构建知识图谱呢？

### 2.1 知识图谱的数据来源于哪里？

知识图谱的构建是后续应用的基础，而且构建的前提是需要把数据从不同的数据源中抽取出来。对于垂直领域的知识图谱来说，它们的数据源主要来自两种渠道：

- 第一种：业务本身的数据。这部分数据通常包含在公司内的数据库表并以结构化的方式存储，一般只需要简单预处理即可以作为后续AI系统的输入；
- 第二种：网络上公开、抓取的数据。这些数据通常是以网页的形式存在所以是非结构化的数据，一般需要借助于自然语言处理等技术来提取出结构化信息。

数据库表(结构化数据)



网页 (非结构化数据)

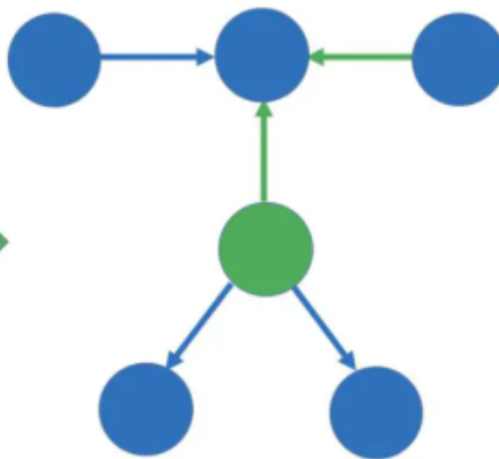


图 6 数据来源

比如在下边的搜索例子里，Bill Gates和Malinda Gate的关系就可以从非结构化数据中提炼出来，比如维基百科等数据源。

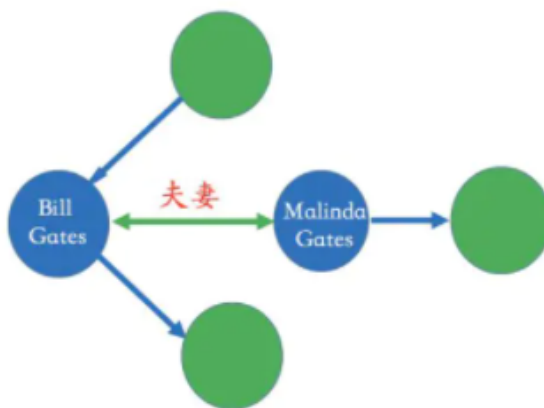


图 7 举例说明

## 2.2 信息抽取的难点在哪里？

信息抽取的难点在于处理非结构化数据。在下面的图中，我们给出了一个实例。左边是一段非结构化的英文文本，右边是从这些文本中抽取出来的实体和关系。

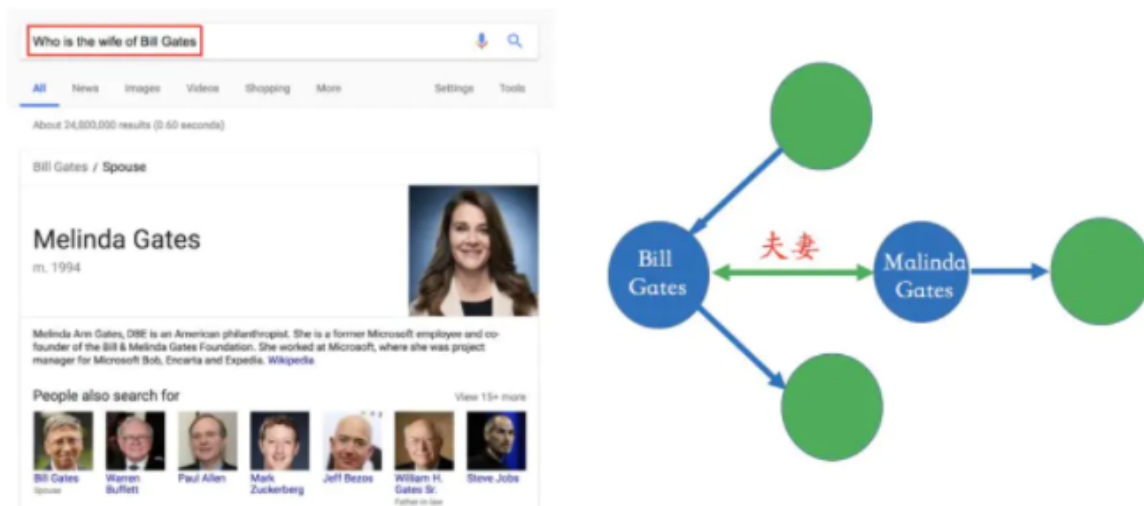


图 8 信息抽取的难点举例

## 2.3 构建知识图谱所涉及的技术？

在构建类似的图谱过程当中，主要涉及以下几个方面的自然语言处理技术：

1. 实体命名识别 (Name Entity Recognition)
2. 关系抽取 (Relation Extraction)
3. 实体统一 (Entity Resolution)
4. 指代消解 (Coreference Resolution)
5. ...

## 2.4、知识图谱的具体构建技术是什么？

下面针对每一项技术解决的问题做简单的描述，至于这些是具体怎么实现的，不在这里一一展开，后续课程和知识图谱第二期的课程将会慢慢展开：

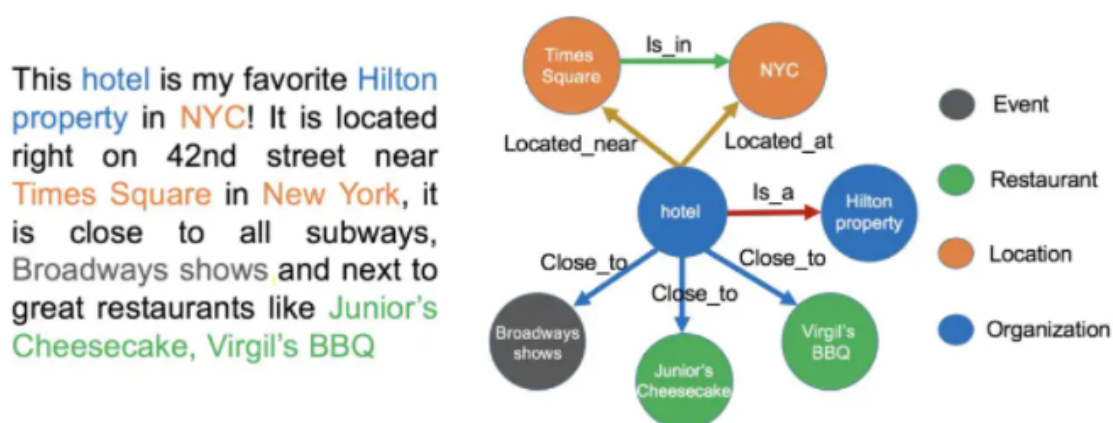


图 9 具体构建技术 示例



### 2.4.1 实体命名识别 (Named Entity Recognition)

- 实体命名识别 (英语: Named Entity Recognition) , 简称NER
  - 目标: 就是从文本里提取出实体并对每个实体做分类/打标签;
  - 举例说明: 比如从上述文本里, 我们可以提取出实体-“NYC”, 并标记实体类型为 “Location”; 我们也可以从中提取出“Virgil's BBQ”, 并标记实体类型为“Restarant”。
  - 这种过程称之为实体命名识别, 这是一项相对比较成熟的技术, 有一些现成的工具可以用来做这件事情。

### 2.4.2 关系抽取 (Relation Extraction)

- 关系抽取 (英语: Relation Extraction) , 简称 RE
  - 介绍: 通过关系抽取技术, 把实体间的关系从文本中提取出来;
  - 举例说明: 比如实体“hotel”和“Hilton property”之间的关系为“in”; “hotel”和“Time Square”的关系为“near”等等。



图 9 NER 和 RE 示例

### 2.4.3 实体统一 (Entity Resolution)

- 实体统一 (英语: Entity Resolution) , 简称 ER
  - 介绍: 对于有些实体写法上不一样, 但其实是指向同一个实体;
  - 举例说明: 比如“NYC”和“New York”表面上是不同的字符串, 但其实指的都是纽约这个城市, 需要合并。
  - 价值: 实体统一不仅可以减少实体的种类, 也可以降低图谱的稀疏性 (Sparsity) ;

### 2.4.4 指代消解 (Disambiguation)

- 指代消解 (英语: Disambiguation)
  - 介绍: 文本中出现的“it”, “he”, “she”这些词到底指向哪个实体, 比如在本文里两个被标记出来的“it”都指向“hotel”这个实体。

This hotel is my favorite Hilton property in NYC! It is located right on 42nd street near Times Square in New York, it is close to all subways, Broadway shows and next to great restaurants like Junior's Cheesecake, Virgil's BBQ

实体统一

This hotel is my favorite Hilton property in NYC! It is located right on 42nd street near Times Square in New York, it is close to all subways, Broadway shows and next to great restaurants like Junior's Cheesecake, Virgil's BBQ

指代消解

图 10 ER 和 Disambiguation 示例

### 三、知识图谱怎么存储？

- 知识图谱主要有两种存储方式：
  - 一种是基于RDF的存储；
  - 另一种是基于图数据库的存储。

它们之间的区别如下图所示。RDF一个重要的设计原则是数据的易发布以及共享，图数据库则把重点放在了高效的图查询和搜索上。其次，RDF以三元组的方式来存储数据而且不包含属性信息，但图数据库一般以属性图为基本的表示形式，所以实体和关系可以包含属性，这就意味着更容易表达现实的业务场景。其中Neo4j系统目前仍是使用率最高的图数据库，它拥有活跃的社区，而且系统本身的查询效率高，但唯一的不足就是不支持准分布式。相反，OrientDB和JanusGraph（原Titan）支持分布式，但这些系统相对较新，社区不如Neo4j活跃，这也就意味着使用过程中不可避免地会遇到一些棘手的问题。如果选择使用RDF的存储系统，Jena或许一个比较不错的选择。

- 存储三元组 (Triple)
- 标准的推理引擎
- W3C标准
- 易于发布数据
- 多数为学术界场景

RDF

- 节点和关系可以带有属性
- 没有标准的推理引擎
- 图的遍历效率高
- 事务管理
- 基本为工业界场景

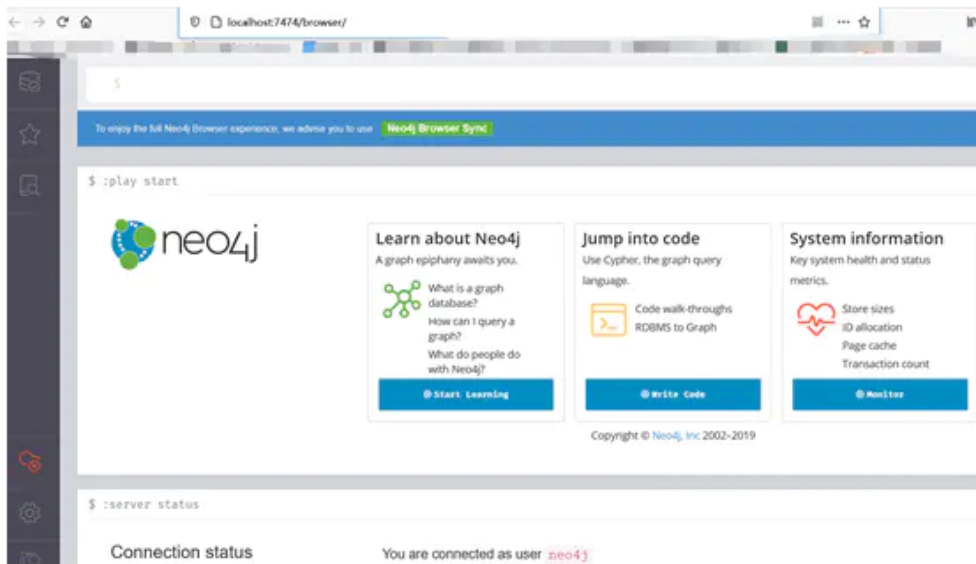
图数据库

图 11 RDF的存储 和 基于图数据库的存储 的区别

### 四、知识图谱可以做什么？

- 通用知识图谱的应用
  - 搜索引擎搜索（百度搜索、搜狗搜索等）
  - 搜索推荐（根据搜索内容进行推荐）
  - 问答





- 行业知识图谱的应用
  - 人脉路径查询。基于两个用户之间的关联实体（比如：所在单位、同事、同学、朋友、家人等）找到两者之间的关联路径。
  - 企业社交图谱查询。基于投资、任职、专利、招投标、涉诉关系以目标企业为核心心向外层层扩散，形成一个网络关系图，直观立体展现企业关联。
  - 辅助信贷审核。基于知识图谱数据的统一查询，全面掌握客户信息；避免由于系统、数据孤立、信息不一致造成信用重复使用、信息不完整等问题。
  - 征信系统。根据用户已有信息（例如：教育信息、身份信息、联系方式、担保或被担保人信息）关联多家平台信用记录。
  - ...

## 参考资料

1. [干货 | 从零到一学习知识图谱的技术与应用](#)
2. [手把手教你快速入门知识图谱 - Neo4j教程](#)
3. [python操作图数据库neo4j的两种方式](#)
4. [Neo4j之导入数据](#)
5. [schema 介绍](#)
6. [知识图谱Schema](#)
7. [美团大脑：知识图谱的建模方法及其应用](#)
8. [肖仰华. 知识图谱：概念与技术. 北京：电子工业出版社, 2020. 2 - 39.](#)
9. [知识图谱的前世今生](#)