

知识表示的五个主要角色

- 一种代理：A KR is a Surrogate。通过对外部事物的内部表示，无需实践即可进行内部推理。知识表示是客观事物的机器标识，每个客观事物应该有一个统一的机器标识，这就如同我们记住万事万物的名字后，才是万事万物的各种联系。通过这个代理（机器标识），可以将外部事物进行统一的内部表示，实现内部推理。
- 一组本体论约定的集合：A KR is a Set of Ontological Commitments。知识表示是一组本体约定和概念模型，通过对万事万物进行抽象，建立有关客观世界的抽象概念模型（本体）。例如，有的学生是计算机学院的，有的学生是人工智能学院的，有的学生是硕士生，有的学生是直博生，但所有同学都是国科大的学生，知识工程课程是专业核心课。
- 智能推理的组成部分：A KR is a Theory of Intelligent Reasoning。知识表示是推理的基础，只有建立了万事万物的概念模型后，才能开始推理。例如，如果知道知识工程学科分类是人工智能，那么就能推断出这门课的授课老师的研究方向之一是人工智能，选这门课的同学都是对人工智能感兴趣的或研究方向是人工智能相关的。
- 高效计算的媒介：A KR is a Medium for Efficient Computation。知识需要进行有效组织，才能保证推理算法的效率。知识表示方法应易于被机器处理，自然语言文本、图像、音频也可以表示知识内容，但不易于被机器处理，因此自然语言文本、图像、音频等不是最适合机器的知识表示方法。
- 人类表达的媒介：A KR is a Medium of Human Expression。基于通用表示框架，方便人们表达和分享对世界的认知，以及人机交互。知识表示应是人类可理解的，这与AI的可解释性要求有关，这是当前数值化知识表示方法的最大问题。

知识表示方法

产生式规则

- 基本术语：规则
 - 确定性规则：通常表示为 $P \rightarrow Q$ 或 IF P THEN Q，其中P是产生式的前提或条件，Q是一组结论或操作，指出前提P所指示的条件被满足时，应该得出的结论或执行的操作。例如，IF 动物有犬齿 AND 有爪 AND 眼盯前方 THEN 该动物是食肉动物。
 - 不确定性规则：通常表示为 $P \rightarrow Q$ （置信度）或 IF P THEN Q（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，按照置信度的要求模糊匹配，并按特定算法将不确定性传递到结论。
- 产生式系统结构
 - 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。
- 正向推理系统：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。推理过程包括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。示例：动物识别产生式系统，通过一系列已知事实和规则推理出A是猎豹。
- 反向推理系统：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。推理过程包括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。示例：动物识别反向推理系统，通过设定目标推理出A是猎豹。

- 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：表示局限性（不能表示结构性或层次性知识）、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）。
- 语义网络
 - 基本概念：语义网络是一种有向图表示的知识系统，节点代表的是概念，而边则表示这些概念之间的语义关系。语义网络中最基本的语义单元称为语义基元，可以用三元组形式表示：<节点1, 关系, 节点2>。
 - 系统结构：语义网络系统由知识库和推理机两部分构成。知识库使用语义网络表示，推理机是求解问题的程序。问题求解过程包括将求解目标表示为一个语义网络子图，其中有些节点或者边的标识是缺失的，表示求解目标。基于求解问题子图，从知识库中搜索可以匹配的网络片段，从而找出目标信息。这种匹配一般是不完全的，具有不确定性，需要考虑不确定匹配问题。
 - 示例：例如，RDF知识图谱查询示例，展示了如何使用语义网络结构化表示通信终端无法驻留在LTE网络上的问题，并进行推理。
 - 特性
 - 优点：使用直观的图结构来描述知识，表达自然，而且方便于计算机的存储和检索，有较为成熟的应用。
 - 缺点：由于缺少形式化的语义定义，不同的语义网络之间难以互相操作，表示不完善。推理过程中通过临时性（ad-hoc）的过程对数据结构进行操作，从而实现推理，推理过程复杂。

框架

- 基本概念：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；侧面表示属性的不同侧面；值是槽/侧面的取值，可以为原子型，也可以为集合型。
- 框架层次结构：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。框架的层次结构包括子类-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。
- 示例：框架名：<灾难>、<地震>、<汶川地震>。<灾难>框架包括时间、地点、伤亡、损失等槽；<地震>框架继承自<灾难>，增加了震级、震源深度、断层等槽；<汶川地震>实例框架进一步具体化了<地震>框架中的值。
- 推理机制：继承推理：框架表示具有分类、分层的特点，下层框架可以继承上层框架的信息。匹配推理：例如找出“震级在7级以上，死亡人数在10万人以上的地震”的实例。
- 特性
 - 优点：框架理论符合人类认知的思维过程，具有自然性；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；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；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：由于真实世界的多样性和复杂性，许多实际情况与框架原型存在较大的差异，在框架设计中难免引入错误或冲突；不能表示过程性知识；缺乏明确的推理机制。

脚本

- **基本概念**：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。
- **示例**：例如，用脚本表示去餐厅吃饭：进入条件是顾客饿了并且有足够的钱；角色包括顾客、服务员、厨师、老板；道具包括食品、桌子、菜单、账单、钱；场景包括进入、点菜、等待、吃饭、离开；结果是顾客吃了饭，不饿了，顾客花了钱，老板赚了钱，餐厅食品少了。
- **推理机制**：匹配推理：例如，鉴别特殊事件的发生。“约翰走进餐厅，他被带到餐桌旁，订了一大块牛排之后，他坐在那儿等了很久，后来，他生气地走了。”这个情节中没有出现“付账单”事件，说明是一个非常规的事件。
- **特性**
 - 优点：在非常狭小的领域内，脚本表示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。
 - 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。

一阶谓词逻辑

- **基本概念**：命题(proposition)是具有真假意义的陈述句，是对客观事实的描述。原子命题是结构上不能再分解出其他命题的命题。逻辑联结词(logical connective)包括否定(\neg)、析取(\vee)、合取(\wedge)、蕴含(\rightarrow)、等价(\leftrightarrow)，连接词优先级依次为 \neg ， \wedge ， \vee ， \rightarrow ， \leftrightarrow 。
- **表示法**：原子命题和复合命题。复合命题是将多个原子命题用逻辑连接词组合而成的命题，描述的是原子命题之间的逻辑关联。例如，如果这款手机配置好且价格低廉，那么它的销量就好($P \wedge Q \rightarrow R$)。
- **推理演算**：等值演算考察逻辑关系符(\leftrightarrow)，即两个命题是否等价，包括等值定理、等值公式、析取范式和合取范式。推理演算考察逻辑关系符(\rightarrow)，即两个命题是否具有蕴含关系，包括基本推理公式(三段论及其证明方法)、基于规则的推理演算、归结推理法。
- **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)、管理困难(缺乏知识的组织原则，知识库管理困难)、推理效率低(一阶逻辑表示不能用多项式时间判断任何逻辑公式的真伪)。

描述逻辑

- **基本概念**：描述逻辑通过概念类别来描述物理世界，又具有形式化的模型论语义，因此称为描述逻辑。为了处理一类对象，一阶逻辑需要借助变量、全称量词和谓词；而描述逻辑舍弃了变量，直接处理概念，避免了一些复杂度较高的操作。虽然描述逻辑的表现力比一阶逻辑要弱，但具备了一阶逻辑不具备的可判定性，推理算法效率更高，是语义网的基础。
- **属性语言AL**：Attribute Language (AL)是最简单的一种描述逻辑，包括概念描述、属性、个体三个基础部分。概念描述(Concept Description)表示一类事物而非个体，类似面向对象编程语言中的“类”。概念描述构造器：交集构造器(\cap)表示两个概念描述的交集，得到一个新概念；并集构造器(\cup)表示两个概念描述的并集；否定构造器(\neg)表示一个概念的否定。属性/角色(Role)作用于概念，以取值来限定概念范围，属性必须要搭配量词(\forall 或 \exists)使用。例如，“所有作者都是中国人的书”记作：书籍 $\cap \forall$ 作者.中国人；“作者不可考的书”记作：书籍 $\cap \forall$ 作者. \perp ；“至少有一个作者的

书”记为：书籍 \sqcap 作者.T。个体（individual）是一个概念的实例，例如“小说(边城)”、“国有银行(农行)”、“智能手机(华为手机)”。

- **知识库**：描述逻辑的知识库包括术语部分（TBox）和断言部分（ABox）。TBox主要包含概念和公理：概念的定义（definition）使用 $=$ 符号定义一个新的概念，例如中国小说 \equiv 小说 \sqcap 中国文学；一般包含公理（general inclusion axiom）用包含于 \sqsubseteq 符号表示两个概念的子集关系，例如中国小说 \sqsubseteq 中国文学。ABox用于描述个体的知识，例如小说（边城），作者（边城，沈从文）；银行（农行），成立时间（农行，1951年）；手机（华为手机），制造于（华为手机，中国）。
- **推理**：TBox相关推理包括判断一个概念是否是另一个概念的子类，检验一个新加的概念C与TBox是否“一致”还是“矛盾”，例如IsA（柴犬，猫）与犬和猫不相交的定义矛盾；检验两个概念是否等价；检验两个概念是否毫不相关（disjoint，即交集为空）。ABox相关推理包括判断某个个体是否属于某个概念，检验ABox是否与TBox相一致，例如中国小说（边城）与美国小说（边城）矛盾。常用推理算法包括Tableau。
- **特性**
 - 优点：形式化语义、推理效率高
 - 缺点：表现力弱于一阶逻辑，但具备可判定性

知识表示方法的衡量

衡量知识表示方法的好坏可以从以下几个维度进行考量：

表达能力

- **强表达能力**：能够准确、全面地描述各种类型的知识，包括静态知识和动态知识。
- **精细度**：能够细致地描述实体、关系和属性，满足复杂知识表示的需求。

推理和计算

- **正确性**：能够保证推理过程的正确性，确保推理结果符合预期。
- **效率**：推理算法需要具有高效性，能够在合理的时间内处理大规模知识库的推理任务。

使用者

- **可读性**：表示方法应具备良好的可读性，使人类用户能够理解和维护知识库。
- **模块性**：表示方法应具有良好的模块性，便于知识库的扩展和管理。

语义网和知识图谱

语义网：从万维网开始

- **Tim Berners-Lee 创造了万维网（World Wide Web）**
 - **时间**：1989-1990年
 - **技术**：
 - 统一资源定位符（URL）：用于标识和定位互联网上的资源。
 - 超文本标记语言（HTML）：用于创建和结构化网页内容。
 - 超文本传输协议（HTTP）：用于在万维网中传输超文本信息。
- **特色**：
 - **超链接**：文本(网页)之间通过超链接进行关联，使用户可以在网上冲浪。
 - **信息检索**：触发了Google等Web搜索引擎的发展。
 - **影响**：为互联网用户提供了一种全新的信息浏览和检索方式。

Web 1.0 → Web 2.0 → Web 3.0

- **Web 1.0:** 服务商产生内容，用户被动接受
- Web 2.0: 用户在服务商平台生成和贡献内容
 - **典型应用:** 维基百科 (Wikipedia)、社区问答 (百度知道) 等。
- Web 3.0: 语义网，赋予Web信息明确的含义，使计算机和人更好地协同工作
 - 特征：
 - 语义标记：网页内容不仅面向人类用户，也能被机器理解。
 - 数据重用：支持数据的共享和复用，促进跨平台的应用和信息整合。

语义网的基本概念和体系结构

- **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：
 - 事物拥有唯一的URI
 - 事物之间链接显式存在，并具有不同语义类型
- 体系结构：
 - **第一层:** Unicode和URI，提供统一编码和资源标识。
 - **第二层:** 语法层，使用XML等格式表示数据内容和结构。
 - **第三层:** 数据交换层，通过RDF描述网络资源。
 - **第四层:** 分类层，使用RDFS扩展RDF，提供类、实例、子类等关系描述。
 - **第五层:** 本体和规则层，使用OWL和RIF定义复杂的语义联系和规则。
 - **第六层:** 统一逻辑层，支持推理任务和验证推理结果。
 - **第七层:** 证明层，建立数据真实性和用户授权的框架。
 - **第八层:** 信任层，提供信任机制，确保身份、内容治理和隐私问题。

核心技术

- **XML:** 用于描述数据的结构和语义，提供可扩展的标记语言。
- **RDF:** 资源描述框架，利用URI标识事物，通过三元组描述资源的属性和关系。
- **RDFS:** RDF Schema，扩展RDF，提供类和属性的层次结构和关系描述。
- **OWL:** Web本体语言，支持复杂语义表达和推理，包含三个子语言 (OWLLite, OWL DL, OWL Full)。
- **SPARQL:** 查询语言，用于对RDF数据进行查询和操作。

知识图谱

- **定义:** 知识图谱是以图的形式表示知识，包含实体、概念及其关系，支持语义查询和推理。
- **起源:** 概念最早由Google提出，用于增强搜索引擎的语义理解能力。
- 特征：
 - **实体:** 知识图谱中的节点，代表具体的对象（如人、地点、事件）。
 - **关系:** 连接实体的边，表示实体之间的关联和语义联系。
 - **属性:** 实体的特征和属性值。

- **应用**：广泛应用于搜索引擎、智能推荐、问答系统、数据整合和语义分析等领域。

主要知识图谱系统

- **Google知识图谱**：通过整合结构化数据和网络信息，提升搜索引擎的语义理解能力。
- **Freebase**：开放的结构化知识库，后来被Google收购并整合到Google知识图谱中。
- **DBpedia**：从Wikipedia抽取结构化信息，支持SPARQL查询和数据访问。
- **YAGO**：结合Wikipedia和WordNet，创建高质量的知识库，包含丰富的实体和关系信息。
- **BabelNet**：结合Wikipedia和WordNet，支持多语言知识表示。
- **Wikidata**：协作编辑的开放知识库，提供结构化数据，支持跨领域和跨语言的知识共享。

知识图谱的构建与应用

- **构建方法**：通过自动抽取、人工编辑和数据融合等方法构建，确保数据的完整性和一致性。
- **应用**：用于增强信息检索和问答系统的语义理解能力，提供更精准和相关的搜索结果。

代表性的符号化知识工程

WordNet

- **起源**：1985年，由普林斯顿大学创建的英文词汇语义知识库。
- **理论基础**：德国语言学家特雷尔的现代语义学理论——语义场理论。词汇在语义上相互联系，构成一个完整的词汇系统。
- **结构**：
 - **同义词集 (Synset)**：将词汇聚类成同义词集，每个同义词集表示一个基本的词汇语义概念。
 - **语义关系**：包括同义关系、反义关系、上下位关系、整体关系、部分关系等。
- **规模**：WordNet包含146,350个单词和111,223个同义词集。
- **应用**：广泛用于自然语言处理、信息检索、机器翻译等领域。

FrameNet

- **起源**：1997年，由加州大学伯克利分校创建的词汇语义知识库。
- **理论基础**：美国语言学家Fillmore的框架语义学理论。认为人类通过已有的知识框架理解词汇和概念。
- **结构**：
 - **语义框架**：表示词汇蕴含的概念和相关的知识预设。
 - **框架元素**：描述框架内的角色，如“烹饪”框架包括厨师、食物、容器和加热设备等。
 - **词语单元**：表示触发框架的词，如烘焙、油炸、蒸煮等。
- **规模**：包含1000多个框架、10000多个词法单元和150000多个标注例句。
- **应用**：用于语义分析、机器翻译和文本理解等领域。

HowNet

- **起源**：由中国计算机学会语言与知识计算专业委员会创建。
- **理论基础**：利用义原，用于描述概念的最小语义单位，描述概念及其关系。
- **结构**：

- **概念**：用义原描述词汇的语义，一个词可以表达多个概念。
- **义原关系**：包括上下位关系、同义关系、反义关系、整体与部分关系等。
- **应用**：用于机器翻译、自然语言处理和知识表示。

Cyc

- **起源**：1984年，由Douglas Lenat创建。
- **目标**：通过编码大量常识建立最大的常识知识库，支持复杂推理任务。
- **结构**：
 - **CycL语言**：基于谓词逻辑，语法类似Lisp。
 - **术语 (Terms)**：定义实体、概念和关系。
 - **断言 (Assertions)**：建立术语之间的关系，包括事实和推理规则。
- **应用**：用于知识推理、专家系统和智能代理等领域。

ConceptNet

- **起源**：由MIT Media Lab的Open Mind Common Sense项目发展而来。
- **目标**：构建一个开放的、多语言的常识知识图谱。
- **结构**：
 - **节点**：表示词语或短语，消除词语歧义。
 - **关系**：定义词语之间的语义关系，包括对称关系和非对称关系。
- **规模**：包含800万个节点和2100万个关系描述。
- **应用**：用于自然语言理解、语义分析和智能助手。

DBpedia

- **起源**：2007年，由社区创建，通过从Wikipedia抽取结构化信息构建。
- **结构**：
 - **抽取方法**：基于属性映射的Infobox抽取、原始Infobox抽取、特征抽取和统计抽取。
 - **Ontology**：社区构建的本体，包含280个类别。
- **应用**：用于数据集成、信息检索和语义搜索。

YAGO

- **起源**：由德国马克斯·普朗克研究所创建。
- **目标**：结合Wikipedia和WordNet，创建高质量的知识库。
- **结构**：
 - **RDFS和OWL描述**：构建具有逻辑定义的知识系统。
 - **类别关联**：将Wikipedia中的类别与WordNet中的类别关联。
- **应用**：用于语义搜索、知识管理和智能系统。

Freebase

- **起源**：由Metaweb公司创建，2010年被Google收购。
- **目标**：基于众包方法构建完全结构化的知识资源。
- **结构**：
 - **知识库**：包含4726万个实体和19亿个实体关系三元组。
 - **与Google知识图谱整合**：成为Google知识图谱的一部分。
- **应用**：用于搜索引擎、数据集成和知识管理。

KnowItAll

- **起源**：由华盛顿大学图灵中心创建。
- **目标**：通过开放信息抽取，从非结构化文本中提取结构化信息。
- **代表系统**：TextRunner和ReVerb。
- **功能**：从文本中识别谓词抽取二元关系，对冗余信息进行评估。
- **应用**：用于信息抽取、知识发现和数据挖掘。

NELL

- **起源**：卡内基梅隆大学的Read the Web项目。
- **目标**：永不停止的语言学习系统（Never-Ending Language Learning）。
- **结构**：
 - **阅读任务**：从Web文本中获取三元组知识。
 - **学习任务**：提升机器学习算法性能。
- **应用**：用于知识抽取、信息集成和自然语言处理。

CogNet

- **起源**：2021年发布的大规模多元知识图谱项目。
- **目标**：以事件为核心，整合语言知识、世界知识和常识知识。
- **结构**：
 - **数据规模**：包含2000万+事件实例和2.5亿+三元组。
 - **语义框架**：包括通用语义框架和细粒度语义框架。
- **应用**：用于知识计算、语义分析和智能应用。

总结

- 符号化知识工程的重要性
 - **知识表示**：提供了一种形式化的方法来表示和组织知识。
 - **推理能力**：支持复杂的推理任务，实现知识的自动化处理。
 - **数据集成**：促进跨领域、跨平台的数据整合和知识共享。
- 未来展望：
 - **技术发展**：随着人工智能和大数据技术的发展，符号化知识工程将在更多领域发挥重要作用。
 - **应用前景**：在智能搜索、语义分析、知识管理和决策支持等方面，符号化知识工程将提供更高效、更智能的解决方案。

小结

.