

常识:知识获取的瓶颈*)

Common Sense: The Bottleneck of Knowledge Acquisition

危 辉 黄旗明 潘云鹤

(浙江大学计算机科学系/人工智能所 杭州 310027)

Abstract Knowledge acquisition is the most important, but also most difficult process in the course of building an intelligent system. And the acquisition of common sense is the neck of knowledge acquisition. Common sense plays a very primary role in the reasoning of human being. At the same time, common sense had generally been paid less attention to and then been ignored, consequently not been formalized into the knowledge base, because its range is various and its form is too abstract. In this paper, how common sense problem comes from, why it is difficult and how to deal with it were discussed.

Keywords Common sense, Knowledge acquisition, Artificial Intelligence

一、常识问题的由来

智能系统之所以能展现智能行为,是因为它形式化包含了某些人的知识,且这些知识是系统行为的核心。智能系统的知识是由知识工程师和领域专家进行交流提取的,这里所指的领域专家也包括由领域专家所撰写的专业文献。这个看似简单的过程其实蕴涵着专家系统构造中的一个瓶颈问题:知识获取,而知识获取的难点就是所谓常识知识的获取。为了说明这个问题的重要性,我们来看几个例子。

例 1 设计一个智能系统实现以下工程设计目标:某公司要求其设计人员想出一个开核桃的自动机械的设计方案。甲的方案以人用螺丝刀撬开核桃为原型;乙的方案以人用榔头砸开核桃为原型;丙的方案以人用钳子夹开核桃为原型;丁的方案以用超高速水流切割核桃为原型,等等。当这些方案提交给方案评审专家审查后,专家提出:“为什么所有这些方案都是从外面用力来分开核桃?”这个激发新设想的疑问一经抛出,许多诸如低温冷却冻裂核桃壳、在真空环境中让核桃自行裂开等具有创新性的方案被

设计出来。

现在我们来看一下让一个专家系统来代替那个评审专家所需具备的知识。要让机器也归纳出前述激发新设计思想的疑问是不容易的:它必须知道¹螺丝刀、榔头、钳子、超高速水流是什么,以及撬、砸、夹、切割是什么意思,即必须具有关于工具和工具使用的知识,其中之一是要知道这些动作的施力方向。现在问题就出来了,除了超高速水流之外,其它的工具和它们的使用知识都是常识²,这些可以说是人人皆知的事情。在方案评审专家系统的知识获取过程中,它们难道不存在被知识工程师和真正的评审专家忽视的可能吗?而且今天是一个开核桃的方案,若明天要评审一个自动削苹果的方案,后天再评审一个自动擦拭窗玻璃的方案呢?系统是不是还得补充关于核桃与苹果的区别、各种刀具、窗玻璃和高层建筑的知识呢?这些评审都是自控和机械领域专家擅长的任务,他们会给知识工程师讲许多关于反馈、传感器、机械设计等方面的知识;与此同时,他们是否会把关于水果的知识、建筑的知识、各类日常的工具等等当作专业知识写进专家系统的知识库呢,

*)本文得到浙江大学与国家计委重大项目“面向市场和创新的产开发技术”的资助。危辉 在站博士后,研究方向:人工智能、认知科学。黄旗明 在站博士后,研究方向:智能 CAD。潘云鹤 工程院院士,研究方向:智能 CAD、人工智能。
¹这种“知道”眼下还只能是某种形式化的存储。“知道我知道”和“知道我不知道”本身就是一个极复杂的自认知问题。²常识,一般的普通的知识(对于专门的知识而言),《新华词典》,商务书馆,1981, P93。

还是把它们当作太普通的常识?而且按这样的逻辑演绎下去,需要表示的常识是不是就没完没了,不可预测了,除非严格限定了这个评审专家系统的使用范围。

例2 积木块世界的搬运问题是具有广泛代表性的通过中间-结果分析求解的问题,在对积木块搬运问题进行形式化描述时遇到三个典型的问题^[2]:限定(qualification)问题:机械手在抓取物体时必须没有抓住别的物体,且不能超重,一个积木块被搬运时必须是最上面的一个,等等;分歧(ramification)问题:没有被机械手搬运的其它积木块保持原状态不变;框架问题:对整个积木块世界的状态要有一个整体的描述。

现在我们来查看其中的常识问题:机械手能搬运物体的条件决不止以上列出的这几项,比如积木块的尺寸不能超大等等。计算机计算的是一个形式化的模型,可以说是虚拟的,一切状态都是计算出来的,如果常识越多,意味着计算量越大。上述后两个问题所反映的正是想通过加入一些“元规则”来减少所需明呈的规则数。

例3 在计算语言学^[3]中常识问题非常明显。“我们吃食堂”和“苹果吃我们”都同样是符合语法且不符合语义的句子,但前者符合语用,而后者不符合语用,因此前者是正确的,后者是错误的。这种语言使用上的习惯都可以看作是常识,新的使用习惯也不断出现,所以数量难以预测,对这样的常识进行表达是困难的。

例4 用于机器人的人工视觉系统是一个图像理解的智能系统,如果它所处理的情景是非针对性的,那么就涉及许多关于环境、照明和光线的知识。在对点、线条和色块进行组织时需要的知识量会随着情景的多样性而变化,其中有很多是关于遮挡、物体结构等方面的常识。

二、造成常识问题的原因

直接了当地说,知识获取的困难来源于知识工程师与领域专家间交流的困难,造成对知识的发现不够充分,只重视明显的专业知识,而忽视了基础知识,这些基础知识中就包括了大量的常识。就人处理某些领域问题而言,这些常识所起的作用可能是不被注意的³,然而“谁都知道的事情,计算机不一定知道”,计算机能做的只有严格的演绎,中间缺一步也

不行。

常识问题可能是不可避免的,那么在什么情况下更容易出现常识问题呢?当专家系统所面临的领域问题非常复杂,所需知识的范围不好预测时,常识问题就会出现。上述例1中所述的问题就是作者在设计一个面向创新的产品设计系统时碰到的。

常识知识的表达也是困难的。在例1中,要得到“都是从外向内用力”的归纳结论,其实是借助了对三种工具使用方法的形象思维,如果脱离这种形象的表示方法就必须对三种工具的使用过程作一个步进式(step by step)的时序/空序描述,然后寻找或抽象出它们相同的模式。要抽象出我们希望的、仅是施力模式上的结论,当然就对描述序列提出了很高的要求,另外,常识知识表达上的困难从各种对信息不完全推理的研究^[4]上还可可见一斑。

再则重要的一点是常识所具有的表现形式是不确定的,也可以说基础知识具有很多不同的变种,人学习到的知识形式和他们再次使用知识的形式可以存在很大的区别,或针对不同的领域,孩子在认知发展过程中,学习到一种极基础、却极重要的认知能力——守恒性(Conservation):爸爸从摇篮的这边走到摇篮的那边,他还是爸爸;一杯牛奶倒进另一个形状不同的杯子里,还是一样多;一个球在地板上滚动,球和地板都各自没有变化,所以球和地板是不同的两样东西,这个世界是由许多不同的东西组成的;……等等。孩子从倒牛奶的例子中学到的守恒性,下次他/她就有可能在分爆米花时用到。毫无疑问,守恒性也是一种常识,然而它的表现形式实在是太丰富了!在专家系统中要表达守恒性可能用一条“元规则”就可以了,然而它是连谓词变元都适用的,所以这条规则所涉及的形式逻辑就至少是二阶的了,一阶逻辑尚且仅具有半可判定的性质,那么二阶逻辑岂不是面临更大的计算上的困难了吗^[5]?

三、对策

当我们遇到常识问题时,最常用的办法是“贴补丁”:补充缺乏的知识。然而这不是解决问题的根本出路,没能充分获取相关知识的可能性总是存在的,也许常识问题的出现就象 SAT 问题的计算复杂度一样,是由问题本身的性质决定的,不是通过技术手段可以绕开的。

在这种情况下,一个对策就是对智能系统的功

³ 常识所起的作用仅仅是被不注意而已,并不表示它们不起作用!基础知识正是因为基础,象树根一样,才显得重要。

能界定要有一个明晰的认识:人做人该做的,机器做机器能做的。如果知识工程师尚不能确保所采集的知识对某一智能行为是充分的;且它们的算法是可行的,那么这一智能行为就不能交给计算机去做,例如,对于象“创新”这样通常有赖于灵感爆发的智能,我们自己尚无法勾勒出其发生的来龙去脉,让计算机去做不就勉为其难了吗?对于这样的问题,计算机采用“随机组合枚举”算法也许比“创新”算法来得现实。换句话说,缩小和明确智能系统的应用领域范围,并将其放在辅助而不是替代人类专家的位置上是目前可行的办法,当然,即使这样我们面对的仍然可能是一个巨量的知识库,这个知识库的组织问题也是非常重要的。

从长远来看,对人工智能的基础学科——认知科学的研究才是问题的出路。例如,从发展心理学^[6]来看,人的认知发展过程是连续的,任何一个智能行为的发生都不是孤立的,不是可分离的,不能割断发展的前后联系性,就某一单个智能行为就事论事地进行模拟是很片面的,在人工智能问题上也应该有历史观。通过对关于 Nature(自然)和 Nurture(训练)问题的讨论,我们可以获得许多关于智能系统的模型构造设计和学习样本设计的极有价值的启

发^[7-8],所以应该给智能系统设计学习子系统,用一种发展的、进化的、循序渐进的观点来实现系统功能的进步^[4]。

参考文献

- 1 姚天顺,等. 自然语言理解. 北京:清华大学、广西科学技术出版社,1995
- 2 Witold L. Non-Monotonic Reasoning. Ellis Horwood Limited, 1990, Introduction
- 3 Ethrington D W. Reasoning with incomplete information, Pitman London, 1988, Introduction
- 4 危辉. 智能脑机制的认知结构核心之元态——表象式语义网及其构造.[北京航空航天大学研究生院博士学位论文], 1998
- 5 王元元. 计算机科学中的逻辑学. 北京:科学出版社, 1989
- 6 Collins W A, et al. Developmental Psychology. New York: Macmillan Publishing Company, 1991
- 7 危辉,何新贵. 基于颞下皮层结构的视知觉不变性模型. 中国智能自动化学会 98 年会, 1997
- 8 危辉,何新贵. 视皮层超柱结构的自组织网络模型. 第五届全国人工智能联合会议, 1998

(上接第 12 页)

环用模拟退火算法初步估计系统的性能,外层循环在 RTL 级进一步修正此值。

参考文献

- 1 Wolf W H. Hardware-Software Co-Design of Embedded Systems. Proc. of IEEE, 1994, 82(7): 967~989
- 2 McFarland M C, et al. The High-Level Synthesis of Digital Systems. Proc. of IEEE, 1990, 78(2): 301~318
- 3 Buchenrieder K, Veith C. CODES: A Practical Concurrent Design Environment. In: Proc. of the Intl. Workshop on Hardware-Software Co-Design. October 1992
- 4 Chou P H, et al. The CHINOOK Hardware-Software Co-Synthesis System. In: Proc. Of ISSS. Sept 1995
- 5 Gupta R K, Micheli G D. Hardware-Software Cosynthesis for Digital Systems. IEEE Design and Test of Computers, 1993, 10(3): 29~41
- 6 Ernst R, et al. Hardware-Software Co-Synthesis for Microcontrollers. IEEE Design and Test of Computers, 1993, 10(4): 64~75
- 7 Kalavade A, Lee E A. A Hardware-Software Codesign Methodology for DSP Application. IEEE Design and Test of Computers, 1993, 10(3): 16~27
- 8 Kumar S, et al. ADEPT: An Unified System Level Modeling Design Environment. In: Proc. of the First Annual RASSP Conference. 114~123
- 9 Murata T. Petri Nets: Properties, Analysis, and Applications. Proc. of IEEE, 1989, 77(4): 541~550
- 10 郭晓东,陈定君,余克清,刘积仁. 嵌入式系统设计的模型及方法学. 计算机科学, 1998, 25(5)