从可解释性悖论到计算性诗学：人工智能文本生成的解释策略

朱恬骅

摘　　要　对人工智能“可解释性”的要求表现为一个悖论，即在技术原理上有良好定义的人工智能模型在造成的技术事实上是欠解释的。可解释性人工智能求助于技术物的设计存在根本的困难，显示出围绕“解释”裁量权的冲突。为承认使用者的解释权并使之能形成解释，有必要建构一种解释策略。“计算性诗学”基于“计算的诗学”对“诗学”概念的延展，并对此加以施行性改造。它立足使用者直接的技术经验，以计算过程而非表征性的编码格式为中心，面向计算中临时生成的解释对象，以功能表现为根据、实验性操作为方法，为人工智能模型提供情境中的解释。

关 键 词　人工智能；可解释性；诗学；解释策略；施行性

伴随人工智能技术的日渐成熟，特别是以语言大模型(large language model)文本生成为代表的具有较强实用性成果的出现，人工智能的可解释性(explainability)问题受到了广泛关注，并成为人工智能治理中的一项关键诉求。近年来，我国先后提出《新一代人工智能伦理规范》《全球人工智能治理倡议》等倡议与要求，强调了包括可解释性在内的多项规范性要求。2016 年，欧盟在《普通数据保护条例》中确立了“获得解释的权力”，要求信息服务方就“决策所涉及逻辑”提供“有意义的信息”。围绕人工智能的可解释性“如何”达成、“为何”达成，技术和产业界就基本概念、评价指标、实现方式等展开了持续而广泛的研究，但始终难以达成共识①。

对人工智能的可解释性要求，实际上蕴含了一组悖论，即人工智能系统在技术原理上具有完备定义(突出体现在人工智能模型各构件的数学定义上)，但在技术事实中即与人的互动中呈现出“欠解释”(under-explained)的状况。对这一组悖论的分析引出的可解释性问题涉及一组事关“解释”的权力冲突，而人工智能的使用者居于不利地位。丹尼斯·特能(Dennis Tenen)在对计算机系统中文本的表征和显示中，提出扩展“诗学”的概念而形成“计算的诗学”，对于解释文本生成的人工智能系统有启发意义②。不过，在人工智能的文本生成中，优先需要得到解释的并非文本的编码表征，而是其动态的计算过程。因此，解释策略需要从以表征为中心转向以施行(performance)为中心，重新

作者简介：朱恬骅，男，博士，上海社会科学院文学研究所助理研究员(上海，200235)。

基金项目：国家社科基金项目“计算机艺术历史生成问题的人类学美学研究”(项目编号：21CA169)。

①Zachary C. Lipton, “The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery,” *ACM Queue*, vol. 16, no. 3(2018), pp. 31-57.

②Dennis Tenen, *Plain Text: The Poetics of Computation*, Stanford: Stanford University Press, 2017.

定位技术物在世界中的位置和意义。本文提出构建“计算性诗学”,并从中明确计算性诗学这一解释策略的基本对象和方法,以求为语言模型以及一般意义上的人工智能系统提供基于使用者经验的解释策略。

一、可解释性悖论:解释裁量权的冲突

社会对人工智能“可解释性”的要求突出了当前人工智能中“解释”的缺乏,而从历史来看,这种缺乏与“解释”本身的复杂性相关联。在人工智能发展的不同阶段,专家系统、机器学习、推荐系统以及神经—符号学习和推理方法技术路径的差异,使所能提供的“解释”存在根本差别:可能是基于因果关系,也可能是基于社会关系;可能是通过知识图谱等符号方式给出的语义性的说明,也可能是针对人工智能决策过程给出的相关性表述。这些技术路径的差异凸显了导致“解释”缺乏的现实因素。但与此同时,我们又清楚地知道,人工智能的模型和算法在数学上有着完备定义,否则它们根本无法通过固定的计算方式达成设定目标。在后一种情况下,人们大可以认为人工智能系统不仅是“可解释”的,而且在根本上不应存在任何未经解释的部分,不然它就会无法实现。

这样,在数学的“解释”和社会对人工智能系统提出的“可解释性”要求之间,就呈现出一种悖论:虽然人工智能模型的每一个组成部分都有详尽规定,但它仍然是欠解释的。关于“可解释性人工智能”的研究不能接受这一悖论式的状况,而将“可解释性”的概念进行分层处理。一项针对“可解释性人工智能”的回顾性研究指出,“可解释性”概念可区分为“可模拟的、可分解的、算法的”三个层面:可模拟的,即人工智能系统可以被看作一个整体,人们可以将其直接视为心智的模型;可分解的,即可被分解为可模拟的部分,是处于中间位置的一种可解释性,其程度根据模型大小或所需计算长度来衡量;算法的,即在数学上是可理解的,也是人工智能系统中普遍存在的一种可解释性形式,它被认为是最弱的可解释性①。

可模拟、可分解、算法对应了人工智能系统的整体、部件和实现三个层次,所讨论的也是对应层次上技术对象的属性,而这一分层结构中解释性的高下根据与使用者的远近亲疏来确定。从它所提供的等级次序来看,算法的解释之所以是一种弱解释,原因在于它试图用一种使用者并不一定理解的带有技术原理的语言,来描述一个本身被使用者要求提供“解释”的事物。以生成式预训练变换器(GPT)类语言模型为例:从算法所对应的具体实现方式来看,也包括在一定程度上就人工智能的部件而言,语言模型是“可解释”的,因为它们由数学上良好定义的“变换器”(Transformer)构成。训练语言模型的算法经过数学上的论证和实践的检验,能够有效调节各参数的数值,在可接受的运算时间和存储空间下,得到有用的数值解。构成语言模型的各个部件,以及部件内部“人工神经网络”的层级,也拥有严格的结构配置、激活函数定义,并用合适的目标函数衡量模型生成出的文本与人类撰写的自然语言文本之间的相似程度。

然而,提出“可解释性”这一要求,恰恰证明了如下的状况:对于技术部件的知识不足以提供预测性的规律,以了解人工智能将以怎样的行为响应某个具体的输入情况。因此,即便能够穷尽这些部件和实现层面的关系,人们仍然无法从中获知其与系统整体功能表现(performance),也就是其施行性(performative)特征之间的关联——譬如,在 GPT 结构中的任何一个部分都找不到应用自然语言进行推理的部分,但是生成出仿佛进行了推理一般的文本却是可以切实观察到的功能表现。它允许了进行文本生成的可能性,却无法回溯性地解释具体生成出的文本。

---

①Gesina Schwalbe and Bettina Finzel, “A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts,” *Data Mining and Knowledge Discovery*, vol. 38 (2023), pp. 3043-3101.

对此,设计者们声称,语言模型的“推理能力”是“涌现”出来的,因为它包含大量参数,又在包含大量逻辑推理过程的文本基础上进行了“训练”。然而,如果我们将“推理”和“训练”这些充满拟人意味的用词,还原为文本的计算性生成和模型参数按规则的调整,那么“涌现”实则只是重复了上述有待解释的技术事实,即模型表现出设计时未有意设置的功能,从而也就无从在技术部件中寻找相应的技术原理加以解释。分层的“可解释性”概念只是将悖论推延到“可模拟”的层面,实际上并不能为面对系统整体功能表现的使用者提供任何具有相关性的解释。横亘在部件与整体、原理与事实之间的鸿沟仍然存在,因而从中产生的“解释”,终究无法推论出系统整体施行的技术事实。面对技术原理同技术事实之间的分裂,“可解释人工智能”企图通过改良技术对象的设计来加以弥合,但实际上只能制造出新的分裂,并因此显现为缘木求鱼。

这就让“可解释人工智能”设计者声称的“以使用者为中心”显得不那么可靠。就这一口号的字面意义来看,一方面,它强调的是“可解释性”与人工智能的具体应用场景相关联。此时,使用者期望得到的“解释”不仅不是单一形态的(甚至难以列出完备的清单和分类谱系),而总是临时的、突发的,难以被归类和预设。但在另一方面,技术上追求“可解释性”的前提,是使用者对“解释”的关切最终都可以在设计中得到满足。提出并实践这种视角的设计者,实际上是面向一类想象的(对设计者而言)理想的使用者进行“解释”,后者寻求“解释”的范围不超出设计者在人工智能系统面世并投入运作前,所能设想的范围。

这就显现出一种权力上的不对等:什么构成一种“解释”,这一点为何必须由设计者来决定?在理想状况下,使用者可以提出自己的“解释”,以同设计者提供的“解释”相争辩、调和。然而,发现人工智能“欠解释”性质的正是使用者。他们无法满足于技术原理层面的“解释”,又缺乏从自身对人工智能系统功能表现的整体认识出发,得出有效“解释”的策略。于是,使用者除了否定设计者给出的“解释”的有效性之外,似乎难以提出建设性意见。甚至,为开发者认可、为开发商推广的隐喻,在人们描述人工智能系统时占据了主导地位,深刻影响了使用者对自身与人工智能系统交互过程的表述,也就难以从中得出自己的洞见。这突显出使用者被剥夺权力的状况——那些不了解技术细节、无法参与人工智能系统设计的人们,自始至终都被排除在有关人工智能系统的“解释”活动之外。

二、“诗学”的延展:从话语到技术对象

“可解释性”的悖论性局面,突显了使用者和设计者之间的不对等。丹尼斯·特能在其《纯文本:计算的诗学》(*Plain Text*: *The Poetics of Computation*)一书中,同样关注到了这一类现象。他发现,大部分使用者并不清楚数字化的文本在计算机系统内部的流转,这超出了他们认知的习惯。人们更倾向于看到模仿书籍排版、显示出的文本内容(特能称之为“表面内容”),几乎下意识地“翻页”“跳转”,而没有认识到使阅读和操作成为可能的图形界面、文本内容之下,支撑这一系统运转的各个部件,与阅读纸质书籍进行翻页动作时面对的物质基础已大相径庭。他指出,对于技术上发生过程的无知、对物质性差异的熟视无睹会造成新的区隔,产生新的“选择性文盲”,特别是“当我们把事物误认为有生命的行为者时,我们进行批判性分析或集体行动的能力也就削弱了”①。

正如特能所承认但未作具体引用的,他有关批判性能力丧失的观点来自弗卢塞尔(Vilém Flusser)。在《书写还有未来吗》中,弗卢塞尔就已经提出了这样的忧虑:“我们所担心的,是阅读的衰落,也就是批判性解译的衰落。我们担心,在未来,所有的消息,尤其是感知和体验的模型,都会被

①Dennis Tenen, *Plain Text*: *The Poetics of Computation*, Stanford: Stanford University Press, 2017, p. 11.

不加批判地接受,担心信息革命会把人变成不加批判地排列组合各种消息的接收者。”①和弗卢塞尔一样,特能反对“用批判性理解换取面对技术时的舒适”②, 并指出只有走出这一令人舒适的范围,拉开与日常习惯的距离,才能重新感知并认识到先前熟视无睹的物质环境。

什克洛夫斯基(Viktor Shklovsky)的“陌生化”理论由此进入了特能的视野。从基本立场上看,“陌生化”概括了重建批判性理解所需的疏离,也就是和习惯成自然的“自动化”相对抗。什克洛夫斯基将后者界定为“动作一旦成为习惯,就会自动完成”,指出这种下意识的“自动化”省略了对事物的感受,将它们化约为某一特征,吞没了人、事、物,而“生活就是这样化为乌有”③。人们下意识地与软件交互,在电子设备上完成诸如“翻页”这样的操作而不反思其背后实际的技术运作,正是“自动化”的体现。对技术内容的习焉不察僵化了人的经验,而陌生化则是重新激活这些经验的方式。

从具体方法来看,“陌生化”作为一种艺术手法,旨在创造一种有别于日常语言的艺术语言,把熟悉的事物“当作第一次看见的事物来描写,描写一件事则好像它是第一次发生”④。在数字化文本的境况中,这首先意味着对一些习以为常的行为加以剖析。例如,不用“翻页”一笔带过,而是注意到实际发生的是指尖在触摸屏的滑动,这个滑动的动作造成了物理量的变化而得到测量,触发了程序的中断,改变了屏幕上的显示……在不同层面和不同部件所发生的事情,比通常所说的“翻页”要丰富得多。特能将中断“隐喻的无摩擦推进”列为自己的第一项任务⑤,以展示表面上的模拟(如模仿书籍的外观)如何隐匿了实际发生的数据交换和计算性过程。

在隐喻消解的基础上,还要反思这些修辞手法得以流行的原因。特能从对话语方式的考察,转入了对技术来源的思想史考察。特能追溯了图灵机、维特根斯坦假想的“阅读机器”,以及图灵在提出“图灵测试”时设想的莎士比亚诗歌鉴赏的对话,认为计算机技术的思想根源与文学息息相关。在他看来,对计算机器、智能机器的设想,都涉及“表达机器状态的记号系统”“承载记号的存储媒介”和“将记号转换为机器状态的机制”⑥,文字、纸张和阅读理解则是三者在纸质媒介上文学活动中的对应物。从这种历史的亲缘性中,特能认为可以将诗学延展到技术对象上,因为后者同样是思想的体现,也需要借助巴赫金(M. M. Bakhtin)所说的“从事物到思想的回归”来加以解释⑦。

话语和技术对象这两个主要方面,构成了特能所提出的“计算的诗学”(poetics of computation)。它以揭示“使表面内容具有意义的平台和基础设施”为使命⑧,强调数字环境中的文本阅读不应只关乎文本本身,因为“控制机制不再能与所传达的消息完全分离”,“内容与媒介相互交织”⑨,计算机技术的介入不仅改变了媒介,而且让程序介入了我们的理解过程,影响了我们所能读到的文本内容及其呈现方式。“计算的诗学”提供的是一种微观的分析策略,将电子形态的文本分解为基本的要素。为此,他细致描绘了可见的文本是如何一步步进入不可见的技术对象、刻写到存储芯片内电

①Vilém Flusser, *Die Schrift*: *Hat Schreiben Zukunft*?, Göttingen: European Photography, 2002, p. 76.

②Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, p. 8.

③[苏]维·什克洛夫斯基:《散文理论》,百花洲文艺出版社 1994 年版,第 9—10 页。

④[苏]维·什克洛夫斯基:《散文理论》,百花洲文艺出版社 1994 年版,第 11 页。

⑤Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, p. 26.

⑥Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, pp. 80-81.

⑦Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, p. 64.

⑧Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, p. 6. 特能实际上没有区分“计算的诗学”和“计算性诗学”(computational poetics)。本文用其著作副标题中的“计算的诗学”代指他的解释策略,而将“计算性诗学”保留为下文所提出的解释策略。

⑨Dennis Tenen, *Plain text*: *The poetics of computation*, Stanford: Stanford University Press, 2017, p. 89.

子在栅极中的状态,又如何从中重新变得可见、出现在屏幕上的完整过程,并以形式主义诗学为参照,予以观念层面的阐发。这与文化分析等宏观层面的考察方式互为补充,恢复了文本在计算机中的形式性和物质性,为理解基本的计算机文本处理技术提供了一种有效策略。

三、重新定位"解释":对"计算的诗学"的施行性改造

在诸多技术对象中,特能尤其关注文本在计算机中的"存储和编码"①。"格式",作为所有可见与不可见内容存储和编码、解码和显现的依据,操控了文本的形式,而且表达了特定的权力。特能举了两个例子:"无边距地格式化文本也意味着拒绝边注。而一种阻止再媒介化的文本格式,则是拒绝共有文化的形成。"②按照特能的表述,这些解释都指向"嵌入的表征"(embedded representation),只是按照它们所处的媒介语境所作的分析有所不同③。

语言模型当然也是一种文本处理技术,但是"计算的诗学"中至关重要的"表征"——文本的编码和格式,却不再具有核心地位,甚至几乎处于缺席状态。在语言模型中,文本首先被词元化(tokenize)。这里的"词元"仅是根据字符串的形式,并不必然按照自然语言的语言学形式(词组、词或词素)进行~~的~~分割。其后,词元进入"向量嵌入"(embedding)阶段,成为等待拼合或运算的一个向量。这种运算过程彻底无效化了语言学的分析,因为它不是人类使用语言和思考意义单元的方式。这些向量不表征任何一种既定意义上的语言"符号",而只有在参与运算时发挥相对性作用。例如,对不同词元的嵌入向量之间进行点乘运算,可以得出两个词元的"相似度"。此时这种"相似度"不由实际的、词典式的语义决定并得到表征,而仅是对它们在语料中共现频率等统计参数的近似,并因此间接反映了这一文本片段的用法。

经过词元化并转换为嵌入向量,参与模型内运算的,不再是原本意义上的"编码"。它们不具有编码所要求的可分性等特征,不同的语言模型,甚至是同一语言模型的不同参数、不同版本、不同运行状态,都可能对同样的输入给出不同的中间"向量嵌入"结果。对于向量嵌入的含义只能从概率分布的角度加以推断。譬如,一个词元的嵌入向量或许表示了以它为中心的上下文的概率分布,而连同其位置信息在内,语言模型对于"文本"的操作是通过操纵这些对"上下文"的概率性擘画而间接地完成的。这些概率分布或许过于庞杂,以至于难以系统地加以列举,而随着所考察的技术事实继续向技术原理抽象,所谓的上下文概率同样显现为某种修辞。实际存在的是矩阵的运算、向量的归一化(normalize)等运算,其结果只是在数学形式上满足对"概率"的要求。而作为这一系列运算的反向过程,人工智能的文本生成是对"概率化"了的上下文的"采样",依照最大似然性的原则选取词元,并按顺序展开:将选取出的词元添加到输入的末尾并重复计算的过程。因此对语言模型而言,不再有"文本"和"编码",只有人为认定的概率分布之间的相互关系,这种认定本身就是一种观念的投射,是一种阐释。

"计算的诗学"在语言模型中失去解释的对象,实际上再次呈现了我们在"涌现"等情形中观察到的那种状况:向着技术原理还原、回溯,与语言模型中需要"解释"的现象本身南辕北辙。但是,这并不意味着"计算的诗学"完全失去意义。在延展"诗学"的过程中,"计算的诗学"有意将什克洛夫斯基所讨论的诗歌艺术、巴赫金所谈论的"审美创造"向技术和技术发明迁移,也就是让诗学在论题上从"(诗歌)艺术作品"延展到"人工制品",在经验类型上从"审美经验"转换到"技术经验"。其潜在的逻辑是,人们可以像对待艺术作品一样对待技术对象等人造物。而正如在艺术中的"陌生化"

①Dennis Tenen, *Plain Text: The Poetics of Computation*, Stanford: Stanford University Press, 2017, p. 52.
②Dennis Tenen, *Plain Text: The Poetics of Computation*, Stanford: Stanford University Press, 2017, p. 192.
③Dennis Tenen, *Plain Text: The Poetics of Computation*, Stanford: Stanford University Press, 2017, p. 39.

促进了人的感知,在技术中的“陌生化”也提供了深入感知技术的方式。

而就在为特能所引用的有关意识层面“自动化”的段落下方,什克洛夫斯基对“艺术作品”提出了一句格言式的论断:“艺术是对事物的制作进行体验的一种方式,而已制成之物在艺术之中并不重要。”①在“计算的诗学”所完成的延展基础上,这个命题可以相应改写为:“技术是对事物的制作加以认知的一种方式,而已制成之物在技术之中并不重要。”当然,正如原句并不意味着艺术作品是没有意义的,而是本身就以“陌生化”的方式提请注意“事物的制作”(根据语境,实际是指阅读过程中心智的创造性活动)②;在技术中,需要解释的也不是人们在运用技术时创造、使用的拟人形象或其他喻体,而是在与技术对象交互的经验中之所以形成这些形象、采用这些隐喻的原因。这包含两方面的内涵。一方面,它要求解释应面向技术经验而非技术原理。技术经验可以提供稳定的锚点,而不引入新的、内涵上不易把握的技术事项。相比于借形式主义诗学阐述技术原理,从技术经验出发可以更直接、更具体地呈现待“解释”的疑问——不如说,正是和技术对象打交道的直接经验要求了“解释”。而且,技术经验原则上是人人都能拥有的,这也符合我们寻求一种无需技术知识储备的解释策略初衷。

另一方面,侧重技术对象交互的过程,同时也就包含了对技术对象运行过程的动态考察。这也和“计算的诗学”中强调静态的程序代码、编码格式等大相径庭。在面对巨量参数的语言模型时,我们无法再追踪相对固定的编码、格式,巨细靡遗地把握数据在不同技术部件中的处理方式,而是不得不面对语言模型的“概率化”计算过程,面对某个具体的人工智能系统、某个语言模型的某个版本、某个状态。

这两方面的内涵印证了科学哲学出现的“施行性转向”③。按照安德鲁·皮克林(Andrew Pickering)的阐述,现代科学习惯以表征为媒介,以非对称的方式与处于“被动、静态”地位的世界关联。它蕴含了一种注重“认识世界”的生存方式,但施行性转向实际是原则的彻底转换:它召唤一种“落实在行动中的本体论”(ontology in action),承认世界的不可知性(unknowability),并试图在这一世界中通过与周遭事物互动而寻求主体自身的生存。不仅如此,“现代科学”本身也是这种“行动”的产物,只是其在叙述自身时倾向于掩盖这一点。皮克林建议,“我们应该把科学(自然包括技术)视为一种与物质力量较量的持续与扩展。更进一步,我们应该视各种仪器与设备为科学家如何与物质力量进行较量的核心”④。

运行语言模型的计算机,正是这种技术人员与“物质力量”较量的核心装置,同时也是使用者所面对的某个特定版本和状态下人工智能模型的物质性体现。它不断地响应人的操作,无论是设计者的改造还是使用者的输入,并随之给出新的输出结果。对于这一不断变化的状态,我们无法像把握“技术原理”那样抵达“普遍性”,也就是一些设计者梦寐以求的普遍有效的“解释”,但始终可以基于经验得出一种在某一过程中有效的情境性解释,即便它们在另外的视角看来并无充足依据。在谈论数字格式存储的文本时,特能也已经无意识地触及了这一点:“大多数当代文本仍然从键盘传递到屏幕,……屏幕随后模拟了已灭绝的阐释可供性(interpretative affordance)。”⑤从技术原理角度来看,文本在计算机系统内的流转可以清楚地解释,因而特能称屏幕所能提供的只是一种“模拟”。但是屏幕之所以能够“模拟”,实质在于人只能通过屏幕观察并从中形成技术经验,而它也确

①[苏]维·什克洛夫斯基:《散文理论》,百花洲文艺出版社 1994 年版,第 10 页。

②Alexandra Berlina, *Viktor Shklovsky: a reader*, New York: Bloomsbury Academic, 2017, p. 57.

③[美]安德鲁·皮克林:《实践的冲撞:时间、力量与科学》,南京大学出版社 2004 年版,第 7 页。

④[美]安德鲁·皮克林:《实践的冲撞:时间、力量与科学》,南京大学出版社 2004 年版,第 7 页。

⑤Dennis Tenen, *Plain Text: The Poetics of Computation*, Stanford: Stanford University Press, 2017, p. 198.

实显现了各种技术部件的施行性运作——计算。

接受施行性转向,就是将解释的对象从静态的格式等要素转向计算过程的外在显现,将解释的依据从原理转移到经验,并在必要时抛弃无法追踪的表征。从物质的角度来看,语言模型等人工智能系统在世界中消耗着能源,其生成的文本以可感知的形式呈现给人,“始终不停地处在行事之中(doing things)”①。对它作出解释在根本上就是对它所行之事的解释,是对计算过程和结果,而非计算方法进行“解释”。这意味着,在执行计算之前不存在“解释”的对象,也就遑论“解释”的范围和“解释”本身。这就明确了计算性诗学的解释对象。至于人工智能系统产生了“有意义”的结果,也不是原理上的预先注定,而是人在情境中对于计算结果作出的反应使然。

从而,“计算的诗学”得以改造为真正意义上的“计算性(computational)诗学”。后者继承了前者对“诗学”范畴的延展,即提供有关技术对象的解释策略。它继续了“计算的诗学”在人工智能系统中所未尽的“中断隐喻”这一任务,消解盛行的拟人修辞,关注计算过程本身的意义维度。但不同于“计算的诗学”,“计算性诗学”从同样的诗学文献中得到启示,以实际技术事实在计算过程中的产生为前提,关注的是人面对人工智能系统运转的经验过程,而不是它们“背后”的原理和设计;它强调了直接的技术经验的重要性,而不再围绕预先认定的技术要素展开。

四、对黑箱的实验性操作:使用者的解释之道

从实践中取得效果的“可解释性人工智能”技术来看,它们没有发展对“技术原理”的解释,而是借助模型的运作生成新的有待解释的技术对象。如在一个名为“Ecco”的项目中,研究人员提供了可视化方式,以追踪代词所指及其在语言模型内部结构中的信息传递。他们中断了通常语言模型的计算工作,并使这一计算过程可见,重新使“代词”等通常的语言学范畴获得解释上的效力。(如图 1)

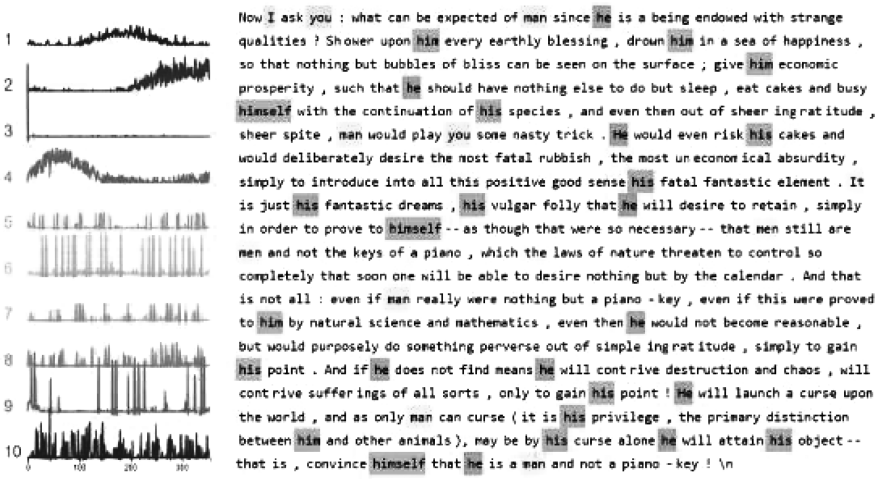

图 1　Ecco 项目运行界面截图

资料来源:J. Alammar, “Ecco: An Open Source Library for the Explainability of Transformer Language Models,” *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2021, pp. 249-257.

对于“Ecco”这样的项目而言,全部的解释都在机器与机器、机器与人的互动中展开,通过可视

①[美]安德鲁·皮克林:《实践的冲撞:时间、力量与科学》,南京大学出版社 2004 年版,第 6 页。

化的界面而呈现,最终指向某一个模型实例。因而“解释”看起来是为了提供“认识”,但当这种“认识”面向具体的计算过程,也就不可避免地改变了原本语言模型的计算——它改变了程序的代码,增加了中间值的输出,编制相应的可视化界面,从而使某一特定的欠解释的语言模型,在继续生产文本层面的技术事实的同时,也制造出新的数据、新的示意图等各种“副产品”来。正是这些出现在“屏幕”上的“副产品”成为使用者解释的对象:曲线图、单词高亮条等具体化了数值计算文本生成中的数值,成为“代词”及指代关系的载体,排除了抽象的、作为“规则”或“设定”的技术原理,也同时中断了文本生成作为“书写”或“创作”的拟人修辞。

不过,这似乎重新提出了使用者与设计者的知识差异:不具备技术知识的使用者何以获得修改人工智能模型的方式?对此的解答有两个方面。从一个方面来说,使用者可以将他人设计的计算过程按照一定方式嵌入其中,只需关注二者如何对接,无需对各自的内部结构及原理有所把握。这种“调用”所需的知识前提远少于自行开发技术组件。

而从另一个方面来说,即便人们缺乏这类直接介入计算过程的手段,模型仍然可以运转并不断生产出新的技术事实,使用者可以通过构造特殊的输入以不断观察模型的反应,并与正常使用的情况相比较、对照,从效果上替代直接的技术改动。例如,向语言模型给出一些精心设计甚至违背常规的提示词,也就是所谓的“对抗性样本”(adversarial samples)。这样做的目的在于干扰语言模型的“正常”运作,观察其输出结果同通常情况之间的差异。尤其是对于开源的语言模型,可以通过自行部署模型实例,使此类对抗性输入对人工智能系统的干扰限制在可控范围内,而不影响他人的使用,规避了法律上的风险。如此,使用者能够以自己的亲身经验认识到,人工智能的实际运作同设计者意图之间的落差,从而探查出应用范围和方式的边界。

上述两个方面实际上都包含同样的立场,也就是将人工智能视为“黑箱”并实验性地操作它。不同于流行的观点,此处的“黑箱”并不强调“不公开、不透明、不可知”等性质,而是回归英国控制论学家罗斯·艾什比(W. R. Ashby)在提出这一术语时所采取的立场:将某物视为一个可以操作的整体,从“做”而非所“是”的角度,有目的地考察技术物。在他假想的密封箱情形中,“上面有些输入接头,可以随意通上多少电压,电击或任何别的干扰;此外有些输出接头,(电机师)可以借此作他所能作的观察”①。通过改变输入并观察输出,最终能够了解密封箱的运作模式,建立输出与输入之间的关联。他笔下电机师面对“黑箱”采取操作以寻找合适的“接头”及其关系与规律,代表了人们从已知求取未知、调用现成之物而制作尚未存在之物的一般方式:不断尝试并根据输出的情况加以调整,直到在某个特定的条件和情境中让它运作起来。

流行话语中的“黑箱”是寂静主义的,凸显出人们的无知,并因此认定人们的无能为力。但原初意义上的“黑箱”则截然相反,它恰恰要求了积极的行动,采取创造性的行动以补偿对内部构造的无知,从而指导下一步对这一“黑箱”的利用方式。我们也可以借“黑箱”蕴含的视觉隐喻理解这一点:“黑箱”并不“透明”,是说它不能成为眼睛“认识”的对象,阻断了对“普遍知识”的要求,却使它重新成为手所使用的对象和工具。② 可以说,“黑箱”在认识论上的负面形象,服务于其在行动上的积极作用。“黑箱化”使人们可以专注于用自己的方式,扰动语言模型等人工智能系统的施行,对其行为表现加以描述,进而找到使事情得以继续的理由和方向。正如艾什比提请注意的,“所有的实物实际上都是黑箱,并且我们从小到老一辈子都在跟黑箱打交道”③,它将日常生活中不自觉的态度转变

①[英]W. R. 艾什比:《控制论导论》,科学出版社 1965 年版,第 86 页。

②苗思萌:《“透明性”与“黑箱化”:新媒体“界面”的技术现象学》,载高建平主编:《外国美学》第 40 辑,江苏凤凰教育出版社 2024 年版,第 66—80 页。

③[英]W. R. 艾什比:《控制论导论》,科学出版社 1965 年版,第 111 页。

为自觉的实验性操作,服务于实践的目的,让人们在面对一个其内部机构“不能完全让我们细察”的机器系统时①,得以继续自己所要完成之事。

作为可操作整体的黑箱本身就是对“制作”的召唤,不仅呼应于“诗学”,而且还将什克洛夫斯基诗学中心智的“创造”落实为切实的技术行动。以“黑箱化”为方法,就是在情境中根据具体的状况生产出解释对象并进行创造性的解释。有时,这种创造性的活动能够超出“黑箱”本身最初被制造出来的目的。至于那些未被充分拆解的事物,它们虽仍嵌套在被拆开的系统之中而保持不变,但随着人们与它不断地展开交互,其内部各个组成部分在技术原理层面上的原有意义也被覆盖了,成为直接技术经验中无关的部分;正如人们可以直接谈论他人的动机,而无需从人作为生物个体、他人大脑作为器官的生理构造和物理性质谈起。这回应了“可解释性”得以提出的一个微观动机:让使用者停止怀疑与自我怀疑,停止对于待解释的语言模型或其他人工智能系统内部“是什么”的不必要追问。只要系统或模块的实际运行表明它的确合乎自己构想的意图,就可以认定它们“做什么”,从而快速定位到对自己有用的“接头”。如果在后续操作中,这种认定能够在人工智能系统的行为表现上得到确认,人们就能够将它视为一种“解释”。

五、结语

计算性诗学接受“施行性转向”而以行动对待人工智能系统的“黑箱”,阻断了表征性知识的普遍性追求。这意味着,算法解释等从技术原理角度作出的“解释”并不只是“弱解释”,而是同人们实际关心的“解释”缺乏必然关联的无关项。相反,情境化的、施行性的知识,对于构建面向人工智能行为表现的“解释”,具有决定性的地位。黑箱化的实验性操作,可以在不追究技术细节的前提下实现对计算过程的介入和更改,从中获得有关人工智能系统运作的情境知识。使用者正可在后一方面大加作为,在“对此行事”和“以此行事”中生成前所未有之物并从中生产出“解释”,从而专注于使用人工智能系统所需完成的事情本身。这种知识地位上的倒转,使权力冲突和不平等获得了消除的可能。“黑箱化”可以有效打破基于技术原理知识所建立的对“解释”的垄断。

这凸显出计算性诗学具有赋(复)权的品格。通过施行性转向,它整体性地考察有文本相关性的计算技术的运作过程,为语言模型等人工智能系统的文本生成提供基于使用者视角的解释;借助“黑箱化”的方法,它可以让使用者不再为自身在技术原理上的无知而困扰。对于可解释性悖论中的权力冲突而言,这也就意味着使用者恢复了在面对设计者提供的“解释”时,加以争辩和协商的权力,让使用者获得言说自身技术经验的合法性。

(责任编校:马延炜)

①[英]W. R. 艾什比:《控制论导论》,科学出版社 1965 年版,第 86 页。