

大模型: 基于自然交互的人机协同软件开发与演化工具带来的挑战*



李戈¹, 彭鑫², 王千祥³, 谢涛¹, 金芝¹, 王戟⁴, 马晓星⁵, 李宣东⁵

¹(北京大学 计算机学院, 北京 100871)

²(复旦大学 计算机科学技术学院, 上海 200438)

³(华为技术有限公司, 北京 100095)

⁴(国防科技大学 计算机学院, 湖南 长沙 410073)

⁵(南京大学 计算机科学与技术系, 江苏 南京 210023)

通信作者: 李宣东, E-mail: lxid@nju.edu.cn

摘要: 以自然语言生成为核心的大模型技术正在人工智能领域掀起热潮, 并持续向更多的领域穿透其影响力. 以 ChatGPT 为代表的自然语言生成大模型 (以下简称大模型), 已经在软件工程的多项活动中展示出其自然交互方式给人提供一定程度帮助的能力和潜力, 正在发展成为一种基于自然交互的人机协同软件开发与演化工具. 从人机协同软件开发与演化的视角, 大模型作为一种软件工具呈现出了两大特征: 其一是基于自然语言的人机交互, 在相当大程度上拓展了人机协同的工作空间、提高了人机协同的效率和灵活性; 其二是基于已积累的软件开发和演化知识, 针对给定软件开发和演化任务的预测性内容生成, 可以对软件开发和演化工作提供一定程度的支持和帮助. 然而, 由于大模型本质是基于概率与统计原理和训练数据所形成的数学模型, 具有不可解释性和内生不确定性, 其生成的是缺失可信性判断的预测性内容, 而人在软件开发与演化中所需要完成的是具有可信保障的决策性任务, 所以大模型作为一种软件工具, 在人机协同的软件开发和演化工作环境中给人提供帮助的同时, 也带来了诸多的挑战. 围绕如何构造对软件开发与演化更有帮助的代码大模型、如何引导大模型生成对软件开发与演化更有帮助的预测性内容、如何基于大模型生成的预测性内容开发与演化高质量的软件系统等大模型带来的挑战进行分析和阐述.

关键词: 软件开发与演化; 大语言模型; 人机协同

中图法分类号: TP311

中文引用格式: 李戈, 彭鑫, 王千祥, 谢涛, 金芝, 王戟, 马晓星, 李宣东. 大模型: 基于自然交互的人机协同软件开发与演化工具带来的挑战. 软件学报, 2023, 34(10): 4601–4606. <http://www.jos.org.cn/1000-9825/7008.htm>

英文引用格式: Li G, Peng X, Wang QX, Xie T, Jin Z, Wang J, Ma XX, Li XD. Challenges from LLMs as a Natural Language Based Human-machine Collaborative Tool for Software Development and Evolution. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4601–4606 (in Chinese). <http://www.jos.org.cn/1000-9825/7008.htm>

Challenges from LLMs as a Natural Language Based Human-machine Collaborative Tool for Software Development and Evolution

LI Ge¹, PENG Xin², WANG Qian-Xiang³, XIE Tao¹, JIN Zhi¹, WANG Ji⁴, MA Xiao-Xing⁵, LI Xuan-Dong⁵

¹(School of Computer Science, Peking University, Beijing 100871, China)

²(School of Computer Science, Fudan University, Shanghai 200438, China)

³(Huawei Technologies Co. Ltd., Beijing 100095, China)

⁴(College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)

⁵(Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

* 收稿时间: 2023-08-08; 采用时间: 2023-08-10; jos 在线出版时间: 2023-08-28

CNKI 网络首发时间: 2023-08-30

Abstract: The generative pretrained transformer-based large language models (LLMs) are setting off a wave in the field of artificial intelligence and continue to penetrate their influence into more fields. The LLMs such as ChatGPT have demonstrated their ability and potential to provide people with a certain degree of assistance through natural language-based interaction in many software engineering tasks, and they are developing into a natural language-based human-machine collaborative tool for software development and evolution. From the perspective of human-machine collaborative software development and evolution, the LLMs, as a software tool, present two major features. One is the natural language-based human-machine interaction, which greatly expands the human-machine collaboration workspace and improves the efficiency and flexibility of human-machine collaboration. The second is to generate predictive contents based on accumulated knowledge of software development and evolution, targeting a given software development and evolution task, which can provide a certain degree of support and assistance for the software development and evolution task. However, since LLMs are essentially mathematical models based on probability and statistical principles and training data, with inexplicability and uncertainty, the contents generated by LLMs are predictive and lack the judgments for trustworthiness. As opposed to the tasks that humans need to perform in software development and evolution, which are typically decision-making tasks with trustworthiness guarantees, LLMs, as a software tool, not only provide assistance to people in software development and evolution featuring human-machine collaboration but also bring many challenges. This study analyzes and clarifies the challenges brought by the LLMs, such as how to construct LLMs that are more helpful for software development and evolution, how to guide LLMs to generate predictive contents that are more helpful for software development and evolution, and how to develop and evolve high-quality software systems based on the predictive contents generated by LLMs.

Key words: software development and evolution; large language model (LLM); human-machine collaborative

1 引言

软件是一种人工制品,也是人类制造的一类最复杂制品.软件开发与演化是人类针对所要解决问题而进行的创新性逻辑思维活动,因而软件也是一种逻辑制品,其表达了人脑逻辑思维所形成的问题解决方案.软件的创造过程和制造过程合二为一,每一个软件系统都是独一无二的创造^[1].

人是软件开发与演化的主体,所追求的目标是高效、高质量、低成本地开发和演化软件系统,所要解决的核心问题是如何使得软件系统满足需求、摆脱缺陷.软件需求是人类在认识复杂世界的基础上形成的主观意图,由于软件的创造过程与制造过程相互融合,因而软件系统开发出来之前没有参考样品,只有软件需求;人类对复杂需求的认识是循序渐进的,需要经历反复迭代、不断探求的过程,并且软件需求随着人类认识的不断深入还会不断发生变化,因而人类往往难以一蹴而就、一劳永逸地认识和理解软件需求,经常陷入“按照错误的需求开发出了自认为正确的软件系统”这样的困局.问题的另一方面,由于人在软件开发与演化过程中是会犯错的,因而软件系统存在缺陷无法避免.

软件需求在软件开发与演化之初通常用自然语言表达,软件开发与演化过程直观地说就是把自然语言描述的需求转换为满足需求的程序代码,这个过程的工作不仅相当复杂而且人力成本很高,因而采用软件工具给人以帮助和支持、不断提高软件开发与演化的自动化程度、进而建立人机协同的软件开发与演化工作环境一直是软件工程领域学术界和工业界共同追求的目标.从软件工具以及提高软件开发与演化自动化程度的视角,由于将自然语言描述的需求转换为满足需求的程序代码这样的问题理论上已经归属为不可判定的计算问题^[2,3],因而现有的软件工具很少考虑按照自然语言描述的需求直接生成程序代码.从人机协同软件开发与演化工作环境的视角,现有的软件工具都难以采用自然语言接口与人交互,从而使得人机协同的工作空间以及有效性和灵活性都受到相当的限制.

近年来以自然语言生成为核心的大模型技术取得了重大突破.以 ChatGPT^[4]为代表的自然语言生成大模型,通过类似自然语言的方式同时处理程序语言,不仅突破了人机协同软件开发与演化过程中的自然语言交互障碍,而且可以按照自然语言描述的需求直接生成预测性的程序代码,已经展示出在软件工程的各项活动中通过自然交互方式给人提供一定程度帮助的能力和潜力,正在发展成为一种基于自然交互的人机协同软件开发与演化工具.

2 大模型: 基于自然交互的人机协同软件开发与演化工具

以 ChatGPT 为代表的大模型(以下简称大模型)通常是基于 Transformer^[5]架构的生成式预训练模型.这种预

训练模型接收文本序列作为输入,产生关于下一个词的概率分布作为输出.这种简单的自回归和自监督的学习方式可以高效地利用互联网上和各种图书资料中积累的海量文本内容进行训练,从而使得大模型在汇聚丰富人类社会知识的基础上形成了强大的自然语言生成能力,可以针对自然语言提问生成预测性内容回答.在此基础上,大模型进一步通过指令学习(instruction learning)的方式进行微调(fine tuning),即通过明确的指令(如进行文本分类)引导模型给出正确的回答.指令学习可以使得大模型在经过多种任务的微调后对齐到人类的交互方式,从而能够更好地理解和处理各种任务.为了使得生成的内容更符合人类价值观,大模型还利用来自人类反馈的强化学习(reinforcement learning from human feedback)来提高针对生成内容的翔实性、公平性、安全性等方面的辨识能力^[6].此外,在使用大模型过程中还可以通过提示工程(prompt engineering)进一步激发模型的能力,即通过恰当的提问、科学的提示和引导使得大模型生成的内容更符合人类的预期^[7].特定方式的提示还可以激发大模型的情景学习(in-context learning)能力(即根据样例现学现用、举一反三)和思维组织(chain/tree of thought)能力(即将复杂问题分解为多个中间步骤并逐一回答).

通过类似自然语言的处理方式,大模型也采用了大量的程序语言相关语料进行训练,因而也具备了生成软件开发与演化相关的预测性内容的能力.从软件开发与演化的视角看,大模型可以通过自然语言与软件开发与演化人员进行交互,基于已积累的软件开发与演化知识,针对给定的软件开发与演化任务进行预测性内容的生成.已有使用案例显示大模型所生成的预测性内容涉及软件需求、设计、编码、测试、调试及维护等各个阶段,这些预测性内容对软件开发与演化人员完成相应任务可以提供不同程度的帮助,从而充分说明大模型正在发展成为一种基于自然交互的人机协同软件开发与演化工具.

3 大模型带来的挑战

大模型作为一种人机协同的软件开发与演化工具呈现出了两大特征:其一是基于自然语言的人机交互,在相当程度上拓展了人机协同的工作空间、提高了人机协同的效率和灵活性;其二是基于已积累的软件开发与演化知识、针对给定软件开发与演化任务的预测性内容生成,可以对软件开发与演化提供一定程度的支持和帮助.从人机协同软件开发与演化的视角,一方面大模型突破了人机自然交互的障碍、通过生成软件开发与演化任务相关的预测性内容给人提供支持和帮助;另一方面,由于大模型本质是基于概率与统计原理和训练数据所形成的数学模型,具有不可解释性和内生不确定性,其生成的预测性内容缺失可信性判断,因而也给人机协同的软件开发与演化工作环境带来了如下挑战.

3.1 如何构造对软件开发与演化更有帮助的代码大模型

由于采用了大量程序语言相关语料并按照类似自然语言的处理方式进行训练,大模型具备了生成软件开发与演化相关的预测性内容的能力.考虑到程序语言相对自然语言而言具有特殊性,人们对大模型作为软件工具对软件开发与演化提供的支持和帮助还有更大、更好的期盼,大模型支持软件开发与演化的能力还有进一步提升的空间.如何在自然语言生成大模型的基础上充分利用程序语言的特点构造对软件开发与演化更有帮助的代码大模型,自然成为伴随大模型作为软件工具所带来的基础性挑战.

首先,程序语言相较于自然语言具有更强的结构性,程序代码可以被映射为抽象或具体语法树,具备明确的结构化特征;与自然语言相比,程序语言可以利用语法产生式进行严格定义,程序代码对处理逻辑的描述更加规范且更加受到语法的限制,这种限制使得程序代码的表达方式更加明确和准确,从而在构造代码大模型过程中应该充分考虑和利用并进行相应处理.其次,相比于自然语言,程序代码具有更加明显的环境相关性:一方面,程序代码中各个组成元素的语义往往高度依赖于它们所处的上下文环境,脱离了具体上下文环境相应的语义可能会被丢失,从而给代码大模型的训练和学习带来困难,需要进行针对性的处理;另一方面,程序代码往往是在特定的软件框架环境中编写的,其语义通常依附于既有的软件架构,在构造代码大模型时应该考虑特定软件框架环境对程序代码语义产生的影响,以促使大模型能够准确地学习到程序代码中所包含的处理逻辑和语义.

3.2 如何引导大模型生成对软件开发与演化更有帮助的预测性内容

大模型是采用互联网上和各种图书资料中的海量文本内容进行训练所形成的基于概率与统计原理的数学模

型,在汇聚和蕴含丰富人类社会知识的基础上形成了强大的自然语言生成能力.然而,这种能力需要通过有效的引导才能得到充分的发挥.大模型根据自然语言提问生成预测性内容作为回答,不同的提问将产生不同的回答,提问的内容、视角和顺序等都将影响大模型最终生成的预测性内容.大模型能力的边界尚未得到足够的认知,经验型方法影响着大模型的生成内容和质量.因此,从大模型使用的角度,就需要研究如何通过提示工程(prompt engineering)恰当地提问、科学的提示和引导使得大模型生成的内容更符合人的预期;从人机协同软件开发与演化的视角,就需要研究如何通过基于自然语言人机交互方式不断引导大模型生成对给定软件开发与演化任务更有帮助的预测性内容,进而通过人机协同的方式更加高效地完成软件开发与演化任务.

在软件开发与演化过程中,大模型可以作为软件工具通过自然交互方式生成预测性内容给软件开发与演化人员提供支持和帮助,从而形成人机协同的软件开发与演化工作环境.一方面,软件开发与演化人员对给定任务及其相关领域具有深入的认识与理解并且具备丰富的软件开发与演化经验,将十分有助于引导大模型充分发挥能力,从而能够在人机协同的软件开发与演化过程中驾驭大模型这样的工具以提高工作效率.另一方面,由于人机交互提问的内容、视角、顺序等都将影响大模型生成的内容,因而需要探索如何正确有效地引导大模型,进而在此基础上建立基于大模型的人机协同软件开发与演化方法.

3.3 如何基于大模型生成的预测性内容开发与演化高质量的软件系统

人类在现实世界中所承担完成的任务可分为预测性任务和决策性任务两大类:预测性任务通常不需要承担风险责任、没有可信保障需求;决策性任务通常需要承担相应的风险责任、有相应的可信保障需求.人是软件开发与演化的主体,在软件开发与演化中所完成的是决策性任务,需要对任务完成正确与否做出可信性判断并承担相应的风险责任,以此构成一定的可信保障.从软件开发与演化的视角,大模型尽管采用了大量人类已有的软件开发与演化活动中所积累的程序语言相关语料进行训练,但其本质是基于概率与统计原理和训练数据所形成的数学模型,具有不可解释性和内生不确定性,大模型所生成的是缺失可信性判断的预测性内容,因而只具备辅助和支持软件开发与演化人员工作的作用,不能简单直接地作为软件开发与演化中具有相应可信保障的决策性任务结果加以使用.大模型并没有改变人在软件开发与演化中的主导地位,大模型所生成的预测性内容的内涵和语义是什么、是否满足给定相关软件开发与演化任务的需求、是否存在缺陷、是整体可用还是部分可用、是否需要修改等,都需要软件开发与演化人员加以分析、理解、辨别、判断、选择、修改和确认.只有正确地分析、理解、辨别、判断、选择、修改和确认大模型所生成的预测性内容,才能在此基础上完成有相应可信保障的决策性任务、进而开发与演化高质量的软件系统,这对软件开发和演化人员来说是一项新的重要挑战.

具体从人机协同编程的视角来说,尽管大模型可以按照给定自然语言描述的需求直接生成代码,但是大模型所生成的代码不仅是预测性的、而且缺失可信性判断,这样的代码可能不一定满足给定自然语言描述的需求、可能近似满足给定自然语言描述的需求、可能部分满足给定自然语言描述的需求、也可能不满足给定自然语言描述的需求,因而需要编程人员加以深入分析、理解、修改和确认后才能加以使用、进而在此基础上完成编程这项需要可信保障的决策性任务.从人工编程的视角,一般来说程序员所要完成的任务包括编写代码和文档以及针对所写代码的各项可信保障任务(静态分析、动态测试、形式验证等).然而,程序员编写代码的过程是把人脑逻辑思维所形成的问题解决方案用程序语言表达出来,这个过程将程序设计、分析和理解融为一体,因而可以支撑程序员对所写代码正确与否做出可信性判断并承担相应的风险责任、进而与完成其他代码可信保障任务一起构成相应的可信保障.更加准确地说,人工编程时程序员所完成的任务包括编写代码与文档、分析与理解代码、以及其他相关的可信保障任务(静态分析、动态测试、形式验证等).相比于人工编程,尽管对大模型生成的预测性代码可以通过静态分析、动态测试和形式化验证等途径进行可信保障,但是大模型生成代码的过程缺乏可信保障(一方面由于大模型的不可解释性和内生不确定性,另一方面由于缺失了程序员写代码过程所包含的深度程序分析与理解环节).因此,在人机协同编程过程中,如果缺失了人的深入分析与理解环节,大模型生成的预测性代码的可信度是难以被接受的、所造成的后果难以预料.因此,在基于大模型的人机协同编程过程中,程序员所要完成的任务应该包括引导大模型生成预测性代码、分析与理解代码、修改和确认代码、以及其他相关的可信保障活动

(静态分析、动态测试、形式验证等),其中程序员分析与理解大模型生成的预测性代码成为工作重点和难点.由于对程序员而言,通常分析和理解代码的难度要高于编写代码的难度,所以将面临的难题和挑战是:程序员对大模型生成的预测性代码的理解程度如何达到如同自己写代码一样的理解程度?更进一步,如何判断程序员对大模型生成的预测性代码的理解程度达到如同他自己写代码一样的理解程度,也将成为亟待解决的难题和挑战.综上所述,编程人员只有具备了分析和理解代码的能力并且进一步加强对大模型生成的预测性代码的分析和理解,才能在人机协同编程过程中充分发挥大模型的能力,从而提高编程效率并且保障代码质量.

4 结 语

软件开发与演化是人类针对所要解决问题进行的创新性逻辑思维活动.大模型在汇聚丰富人类社会知识的基础上形成了强大的自然语言生成能力,并通过利用大量人类已有软件开发与演化活动中所积累的程序语言相关材料进行训练,因而能够通过自然交互方式、针对给定软件开发与演化任务生成相关的预测性内容,给人主导的软件开发与演化工作提供一定程度的支持和帮助,从而突破了人机协同软件开发与演化过程中的自然交互障碍、开启了基于自然语言交互的人机协同软件开发与演化时代.然而,以大模型作为主要软件工具的人机协同软件开发与演化工作环境仍然面临挑战,一方面的挑战相关于“机(工具)”,在自然语言生成大模型的基础上亟待构造能够充分利用程序语言特性、对软件开发与演化更有帮助的代码大模型;另一方面的挑战相关于“人”,大模型并没有改变人在软件开发与演化中的主导地位,软件开发与演化人员不仅要能够有效引导大模型生成对软件开发与演化更有帮助的预测性内容,而且要能够正确分析、理解和确认大模型生成的预测性内容并在此基础上完成决策性任务、进而开发和演化高质量的软件系统.

致谢 本文起源于2023年4月6-8日由中国计算机学会软件工程专业委员会和华为公司共同组织召开的“未来软件工具”研讨会,感谢与会讨论的软件工程专委会和华为公司的老师和同事们.

References:

- [1] Ma XX, Liu XZ, Xie B, Yu P, Zhang T, Bu L, Li XD. Software development methods: Review and outlook. *Ruan Jian Xue Bao/Journal of Software*, 2019, 30(1): 3–21 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5650.htm> [doi: 10.13328/j.cnki.jos.005650]
- [2] Zhang XX. *Encyclopedia of Computer Science and Technology*. 3rd ed., Beijing: Tsinghua University Press, 2018 (in Chinese).
- [3] Wood D. *Theory of Computation*. New York: Harper & Row, 1987.
- [4] Wolfram S. What Is ChatGPT Doing...and Why Does It Work? 2023. <https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work>
- [5] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: *Proc. of the 31st Int'l Conf. on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 6000–6010.
- [6] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R. Training language models to follow instructions with human feedback. arXiv:2203.02155, 2022.
- [7] Liu PF, Yuan WZ, Fu JL, Jiang ZB, Hayashi H, Neubig G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv:2107.13586, 2021.

附中文参考文献:

- [1] 马晓星, 刘讚哲, 谢冰, 余萍, 张天, 卜磊, 李宣东. 软件开发方法发展回顾与展望. *软件学报*, 2019, 30(1): 3–21. <http://www.jos.org.cn/1000-9825/5650.htm> [doi: 10.13328/j.cnki.jos.005650]
- [2] 张效祥. *计算机科学技术百科全书*. 第3版, 北京: 清华大学出版社, 2018.



李戈(1977—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为程序生成, 程序理解, 深度学习.



金芝(1962—), 女, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为需求工程, 知识工程, 基于知识的软件工程.



彭鑫(1979—), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为软件智能化开发与运维, 人机物融合泛在计算, 智能网联汽车.



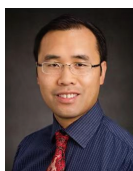
王戟(1969—), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为可信智能与新兴软件技术和系统.



王千祥(1970—), 男, 博士, 高级技术专家, CCF 杰出会员, 主要研究领域为软件工程.



马晓星(1975—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为软件工程, 智能软件系统, 软件方法学.



谢涛(1975—), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为软件工程, 系统软件, 软件安全, 可信人工智能.



李宣东(1963—), 男, 博士, 教授, CCF 会士, 主要研究领域为软件工程, 系统软件, 可信软件, 形式化方法.