



# 从自然语言需求到程序代码

大语言模型打开的空间和带来的挑战



李宣东 南京大学  
2023年12月6日





# 提 纲



- 软件、软件开发与演化
  - 人类针对所解决问题的创新性思维活动
  - 从自然语言需求到满足需求的程序代码
    - 有可信保障需求的决策性任务
- 大语言模型打开了软件开发与演化的新空间
  - 从自然语言需求到程序代码
    - 预测性代码的生成，打开了解决问题的新空间
- 软件工程专业能力培养面临新的挑战
  - 基本能力：程序、算法、系统、工程
  - 综合能力：解决问题、专业知识学习、**驾驭AI**
- 结语



# 软件是人类制造的最复杂制品



- 软件开发和演化是人类针对所解决问题的**创新性思维活动**
  - 科学家**探索世界**、工程师**改造世界**
  - 计算机科学家、软件工程师**创造世界**
- 软件是人类大脑逻辑思维活动的体现
  - 认识和理解人的思维活动非常困难
    - **认识与理解软件系统非常困难**
  - 管理和规范人的思维活动非常困难
    - 管理和控制软件开发过程相当困难
    - 软件开发进展情况难以衡量，质量难以评价
- 软件的创造过程与制造过程合二为一
  - 软件系统开发出来之前没有参考样品
    - **难以估算成本，投资软件具有相当大的风险和不确定性**



# 人是软件开发与演化的主体



- 软件开发与演化要遵循人的认知规律
  - 人对复杂事物的认识不是一蹴而就的
    - 需要经历循环反复、不断探求的过程
  - 软件开发与演化
    - 把现实世界复杂的问题模型转换为计算模型
    - 目标：高效、高质量、低成本
    - 问题：如何满足需求、摆脱缺陷？
- 人是软件需求产生、认识、确认的主体
  - 需求是人类在认识复杂世界的基础上形成的主观意图，经常不断变化
  - 我们经常陷入“按错误的需求开发出所谓正确的软件”的困局
- 人在软件开发与演化过程中是会犯错的
  - 软件系统缺陷难以避免，软件可信保障成本很高



# 软件的质量观：从正确到可信



- 任何人和机构都不能确保软件质量
  - 软件产品没有“三包”
  - 以正确性为目标（绝对正确，客观认识）
    - 软件系统是封闭、静态环境中的系统
      - 软件系统可以看成是一个数学系统
    - 正确性是基于客观**证明**基础之上的**客观认识**，
      - 软件需求由形式化规约描述，形成人对软件系统的**确认**关系
  - 以可信性为目标（相对可信，主观认识）
    - 软件系统是开放、动态环境下不断演化的系统
      - 软件无法看成是一个数学系统，由于规模和复杂性不断增大，客观的证明（需要投入巨大的成本）是现实无法完成的
    - 可信性是基于客观**证据**（部分客观证明）基础之上的**主观认识**
      - 软件需求由自然语言描述，形成人对软件系统的**信任**关系



# 软件开发与演化任务



- 人类在现实世界中所承担完成的任务
  - 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
    - 需要对软件系统是否满足需求、摆脱缺陷做出可信性判断
  - 承担相应的风险责任，以此构成一定的可信保障



# 软件开发与演化任务



- 从自然语言需求到满足需求的程序代码
  - 规约、设计、编码、测试、调试、.....
    - 工作难度大、人力成本高
  - 涉及两种语言：自然语言，程序语言
    - 语言与计算具有紧密的联系，计算理论问题被转化为语言问题
- 人机协同的软件开发与演化
  - 软件工具给人提供帮助和支持
    - 提高软件开发与演化各个阶段的自动化程度
  - 软件工具很少考虑按照自然语言需求直接生成程序代码
    - 将自然语言描述的需求转换为满足需求的程序代码
      - 不可判定的计算问题
  - 软件工具难以突破自然交互障碍
    - 缺乏自然语言接口与人交互
    - 人机协同的工作空间、有效性和灵活性受到限制



# 算法：软件系统解决问题的基础



- 可判定计算问题与不可判定计算问题
- 算法
  - 求解问题需要遵循的、被清楚指定的简单指令的集合
  - 输入和输出、确定性、有穷性、能行性
- 算法复杂性
  - P类问题，NP类问题
    - 随机算法、近似算法
- 基于逻辑设计的算法
  - 内在可解释、正确性可证明（**输出结果满足需求规约**）



# 神经网络模型：预测性算法



## ■ 基于概率与统计原理和训练数据形成的数学模型

### ■ 输出的是预测性结果

- 预测性判别，预测性内容生成
- 不一定满足需求规约
- 缺失可信性判断

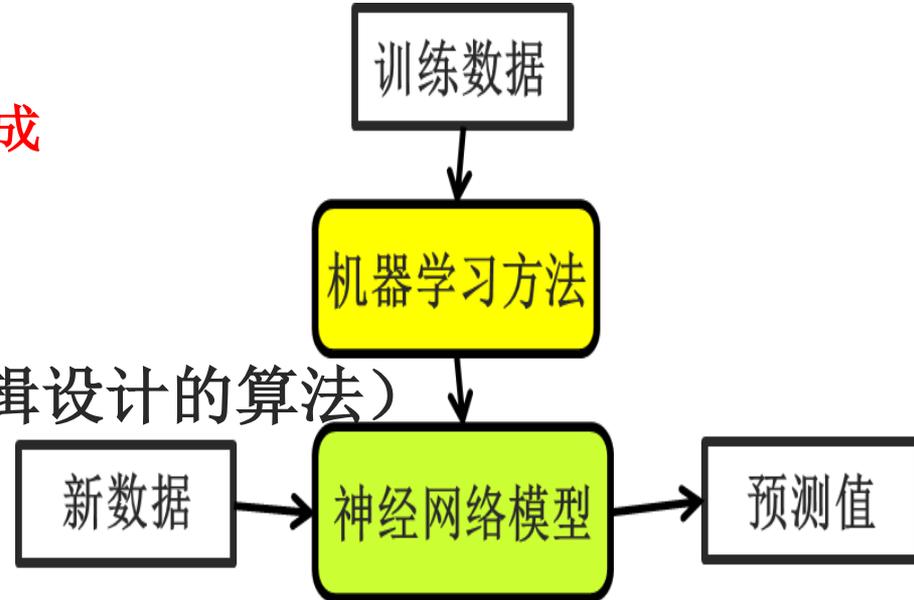
### ○ 基本特征（对比与基于逻辑设计的算法）

- 内在不可解释
- 易受数据扰动

### ■ 输出结果不一定满足需求规约、缺失可信性判断

## ■ 打开了解决问题的新空间，给软件可信保障带来挑战

- 基于给定数据进行预测，基于预测解决实际问题
- 可以用于决策性任务，但直接用于决策存在风险





# 大语言模型



- 以**ChatGPT**为代表的自然语言生成大模型
  - 基于**Transformer**架构的生成式预训练模型
    - 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
    - 简单的自回归和自监督的学习方式
    - 高效地利用互联网上和图书资料中积累的海量文本内容进行训练
  - **在汇聚丰富知识的基础上形成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
  - 指令学习方式进行微调以对齐人类交互方式
  - 来自人类反馈的强化学习提高各方面辨识能力
  - 提示工程使得生成的内容更符合人类的预期
    - 进一步激发出情景学习能力、思维组织能力



# 大模型：基于自然语言交互的软件工具



- 汇聚积累了丰富的软件开发与演化知识
  - 通过类似自然语言的处理方式
  - 采用了大量的程序语言相关语料进行训练
- 基于自然语言交互的人机协同软件开发与演化工具
  - 通过自然语言与软件开发与演化人员进行交互
  - 针对给定的软件开发与演化任务生成预测性内容
  - 所生成的预测性内容涉及软件开发与演化各个阶段
    - 软件需求、设计、编码、测试、调试及维护等
    - 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



# 大模型：基于自然语言交互的软件工具



- 基于自然语言交互的人机协同软件开发与演化工具
  - 基于自然语言的人机交互
    - 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
  - 针对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
  - 生成的预测性内容缺失可信性判断，需要人工分析、理解和确认
    - 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
    - 对预测性内容进行可信性判断需求要成本



# 大语言模型 = 大预言模型



- 大模型生成的是缺失可信性判断的预测性内容
  - 只具备辅助和支持软件开发和演化人员工作的作用
  - 不能直接用作软件开发与演化中决策性任务的结果
    - 预测性内容需要可信性判断后方可使用
- 软件开发和演化人员面临新的重要挑战
  - 如何基于大模型生成的预测性内容开发和演化高质量软件？
    - 基于大模型生成的预测性内容完成软件开发与演化中的决策性任务
      - 分析、理解、修改和确认大模型生成的预测性内容
      - 在此基础上完成有相应可信保障的决策性任务
    - 进而开发和演化高质量的软件系统



# 大模型打开从需求到代码的空间



- 大模型：支持从自然语言需求生成预测性代码
  - 将自然语言描述的需求转换为**满足需求**的程序代码
    - 不可判定的计算问题
  - 大模型所生成的预测性代码**不一定满足需求**
    - 生成的程序代码近似满足需求
    - 生成的程序代码有可能满足需求
    - 生成的程序代码部分满足需求
- 大模型**放弃可信约束条件**生成预测性代码
  - **打开了解决问题的新空间，带来了可信保障难题和挑战**
    - 如何基于大模型的预测性代码得到满足需求的程序代码？



# 从预测性代码到满足需求的程序代码



- 构造能够生成满足需求代码的大模型？
  - 目前看难以改变生成代码的预测性质
- 通过测试等手段对预测性代码做出可信性判断？
  - 大模型生成代码 + 测试 = 人工编写代码？
    - 大模型生成代码的过程是不可理解的、不可信的
    - 人工编写代码的过程是可理解的、可信的
    - 可信度（大模型生成代码 + 测试） < 可信度（人工编写代码 + 测试）
- 基于大模型的人机协同编程
  - 编程人员引导大模型生成预测性代码
  - 编程人员对预测性代码做出可信性判断和决策
    - 分析、理解、修改和确认预测性代码
  - 编程人员完成其它相关的可信保障活动
    - 静态分析、动态测试、形式化验证等



# 基于大模型的人机协同编程



## 人工编程

- 编程人员编写代码和文档
- 编程人员分析与理解代码、修改和确认代码
  - 与编写代码融合一体
  - 支持编程人员对所写代码做出可信性判断并承担相应的风险责任
- 编程完成其它相关的可信保障任务
  - 静态分析、动态测试、形式化验证等

## 基于大模型的人机协同编程

- 编程人员引导大模型生成预测性代码
- 编程人员分析与理解预测性代码、修改和确认代码
  - 若缺失则无法支持对代码做出可信性判断并承担相应的风险责任
- 编程人员完成其它相关的可信保障活动
  - 静态分析、动态测试、形式化验证等



# 基于大模型的人机协同编程



## ■ 问题与挑战

- 大模型生成的预测性代码不可缺失人的分析与理解
  - 相比与人工编程
- 编程人员理解代码的难度通常高于自己写代码的难度
  - 编程人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？
  - 如何判断编程人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？
- 人性所致的编程人员松懈、偷懒和犯错
  - 提交预测性代码应付差事、不认真分析与理解预测性代码



# 基于大模型的人机协同编程



## ■ 解决问题、应对挑战的途径

### ○ 技术途径

- 降低编程人员分析与理解代码的难度和复杂性
  - 研制各类技术与工具辅助编程人员分析与理解代码
- 避免编程人员直接分析与理解代码
  - 引导大模型生成预测性代码
  - 基于预测性代码**自动**逆向生成更高抽象的形式化模型
  - 编程人员修改、确认形式化模型
  - 基于编程人员确认的形式化模型**自动**生成可信程序代码

### ○ 教育途径

- 提升编程人员专业能力



# 大模型如同专业相机



- 大模型是傻瓜相机还是专业相机？
  - 自然语言交互能力使得大模型成为傻瓜相机
    - 人人都能使用，极大地拓展了人机交互的工作空间
  - 预测性内容生成能力使得大模型成为专业相机
    - 生成的预测性内容缺失可信性判断
    - 需要人工分析、理解、修改与确认
- 理解代码需要比写代码更强的专业能力
  - 大模型作为软件工具要求使用者具有更强的专业能力



# 人工编程 & 人机协同编程



- 人工编程
  - 编程人员编写代码的过程可理解、可信
  - 对编程人员的专业能力要求相对低
  - 效率相对低、成本高
- 基于大模型的人机协同编程
  - 大模型生成代码的过程不可理解、不可信
    - 需要程序员分析与理解预测性代码、修改和确认代码
  - 对编程人员的专业能力要求相对高
  - 能够提升效率、降低成本
- 编程人员达不到专业能力要求建议采用人工编程
  - 编程人员需要具备更强的专业能力才能使用大模型工具
  - 对软件工程专业能力培养带来了新的挑战



# 软件工程专业能力培养需求



## ■ 基本能力

### ○ 程序能力

- 程序设计、程序分析、程序理解

### ○ 算法能力

- 算法设计与分析、数理逻辑、计算理论初步
- 现代算法：近似算法、随机算法、数据驱动的算法

### ○ 系统能力

- 有效处理和控制系统复杂性的能力
- 计算机系统、编译系统、操作系统、数据库系统

### ○ 工程能力

- 高效、高质量、低成本地开发和演化大规模复杂软件系统
- 神经-符号融合软件：新的复杂软件形态

- 有机结合基于逻辑设计的符号算法和数据使能的神经网络模型
- 解决开放、动态和不确定场景下的复杂应用问题



# 软件工程专业能力培养需求



- 综合能力（创新能力）
  - 解决问题能力
    - 解决软件工程前沿挑战问题
    - 解决应用领域软件难点问题
  - 专业及专业相关知识的学习能力
    - 不断自我学习新的专业知识及专业相关知识
    - 终身学习是从事软件工程专业职业的必要条件
    - 专业知识的学习能力决定了学生将来职业生涯的生命力
  - 驾驭AI的能力
    - 能用AI，用好AI，用对AI
    - 能够对大模型生成的预测性内容进行可信性判断



# 软件工程专业能力培养



- 软件工程专业能力培养
  - 基本能力：程序能力、算法能力、系统能力、工程能力
  - 综合（创新）能力：解决问题、专业知识学习、**驾驭AI**
- 专业能力培养过程
  - **专业知识传授**
  - **专业知识学习能力培养**
  - **综合（创新）能力培养**
  - **三位一体、融合贯通**
- 在解决问题的过程中培养综合（创新）能力
  - 软件工程前沿挑战问题
  - 应用领域软件难点问题
- **通过全面提升专业能力获得驾驭AI的能力**



# 推动软件发展的三大维度



## ■ 外部环境

- 软件运行环境不断演进
  - 硬件、网络、外设的发展和普及
    - von Neuman计算机、网络计算机、GUC、应用场景计算机
- 信息领域技术浪潮不断推动
  - 云计算、物联网、大数据、人工智能、**大模型**、**X...**
    - **Software for X**、**X for Software**
- 拓展软件需求空间、推动软件方法技术进步

## ■ 内在动力

- 高效、高质量、低成本地开发和演化软件系统
  - 结构化、对象化、构件化、模型驱动、服务化

## ■ 人本属性

- 软件开发和演化要遵循人的认知规律
  - 原型开发、敏捷、**DevOps**
- 有效获取、管理和组织人力资源
  - 开源、生态、供应链、群智开发

人本属性

满足需求  
核心问题  
摆脱缺陷

外部因素

内在动力



# 结 语



- 软件、软件开发与演化
  - 人类针对所解决问题的创新性思维活动
  - 从自然语言需求到满足需求的程序代码
    - 有可信保障需求的决策性任务
- 大语言模型打开了软件开发与演化的新空间
  - 从自然语言需求到程序代码
    - 预测性代码的生成，打开了解决问题的新空间
- 软件工程专业能力培养面临新的挑战
  - 基本能力：程序、算法、系统、工程
  - 综合（创新）能力：解决问题、专业知识学习、**驾驭AI**



问题? ? ?



谢谢!

