

大規模言語モデル(LLM)シリーズ

シリーズリーダー 鵜林 尚靖 (九州大学 教授)

講師陣

中井 悦司 (グーグル・クラウド・ジャパン)
山口 光太 (サイバーエージェント)
坂本 一憲 (東京通信大学 准教授、
WillBooster(株))
吉岡 信和 (早稲田大学/NII)

大規模言語モデル(LLM)シリーズ

シリーズ全体のゴール

LLMのベースとなる生成モデルの仕組み, LLMを組み込んだアプリ開発の手法, LLMのソフトウェア開発への適用, について基礎から応用までのスキルを幅広く習得する

基礎 2科目

生成モデルの基礎

画像や自然言語の生成モデルを理解するための基礎知識を習得

DCGAN, Diffusion モデル, Transformer モデル, マルチモーダル生成モデル等

大規模言語モデルを組み込んだアプリ開発

LLMを組み込んだ新しいタイプのアプリを実装するスキルを習得

APIを介したLLM活用, チャットボットやQ&Aボットのプログラミング

応用 1科目

大規模言語モデルのソフトウェア開発への応用

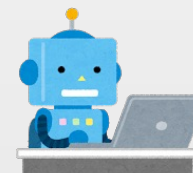
要求抽出, 設計, プログラミング, デバック, テストへ LLMを応用する技術を習得

プロンプトパターンカタログの活用法, Generative Agent のソフトウェア開発への応用

人間と機械の
新たな知的インタラクション



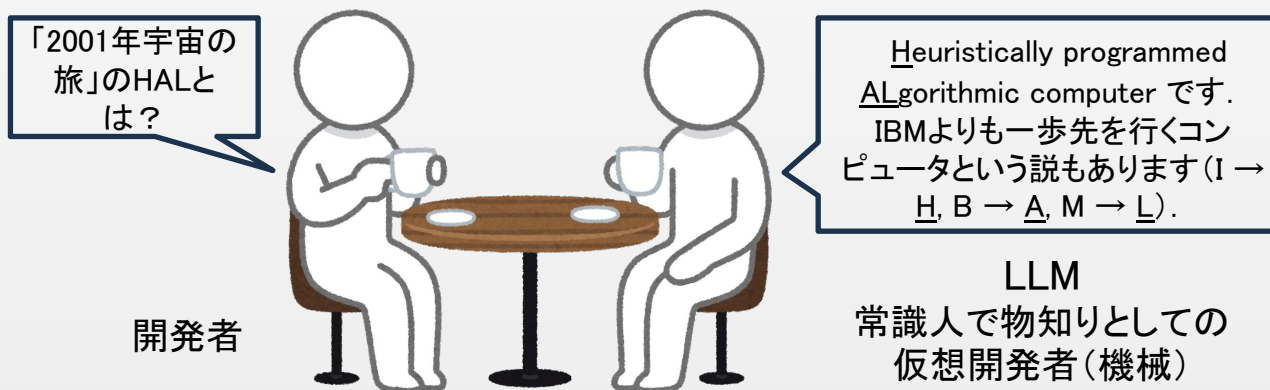
人間
(開発者)



機械
(LLM)

LLM時代のソフトウェア開発とは

LLM = 自然言語コンピューティング
 新たな知性(人間以外の知的存在)との出会い



ソフトウェア開発へのインパクト

- タスク自動化から仮想開発者群による自動化への転換
- 自然言語によるプログラミング(ノーコード)

ポイント

- 常識人としての相談相手(合理的な判断を行う)
- 仮想開発者群による仮想プロジェクトチーム(LLMベースのマルチエージェントシステム)



研究事例： ソフトウェア開発プロジェクトの自動化

METAGPT: META PROGRAMMING FOR MULTI-AGENT COLLABORATIVE FRAMEWORK

Sirui Hong¹ Xiawu Zheng² Jonathan Chen¹ Yuheng Cheng³ Ceyao Zhang³ Zili Wang⁴
 Steven Ka Shing Yau¹ Zijuan LIN² Liyang Zhou⁵
 Chenyu Ran⁶ Lingfeng Xiao⁶ Chenglin Wu^{1*}
¹DeepWisdom ²Xiamen University ³The Chinese University of Hong Kong, Shenzhen
⁴Nanjing University ⁵University of Pennsylvania
⁶University of California, Berkeley

ABSTRACT

Recently, remarkable progress has been made in automated task-solving through the use of multi-agents driven by large language models (LLMs). However, existing works primarily focus on simple tasks, lacking exploration and investigation in complicated tasks mainly due to the hallucination problem. This kind of hallucination gets amplified infinitely as multiple intelligent agents interact with each other, resulting in failures when tackling complicated problems. Therefore, we introduce MetaGPT, an innovative framework that infuses effective human workflows as a meta programming approach into LLM-driven multi-agent collaboration. In particular, MetaGPT first encodes Standardized Operating Procedures (SOPs) into prompts, fostering structured coordination. And then, it further mandates modular outputs, bestowing agents with domain expertise paralleling human professionals to validate outputs and reduce compounded errors. In this way, MetaGPT leverages the assembly line work model to assign diverse roles to various agents, thus establishing a framework that can effectively and cohesively deconstruct complex multi-agent collaborative problems. Our experiments conducted on collaborative software engineering tasks illustrate MetaGPT's capability in producing comprehensive solutions with higher coherence relative to existing conversational and chat-based multi-agent systems. This underscores the potential of incorporating human domain knowledge into multi-agents, thus opening up novel avenues for grappling with intricate real-world challenges. The GitHub repository of this project is made publicly available on: <https://github.com/geekan/metagpt>

1 Introduction

Multi-agent systems that utilize Large Language Models (LLMs) offer remarkable prospects for replicating and enhancing human workflows. However, existing systems tend to oversimplify the complexities inherent to real-world applications, as illustrated in recent studies [10, 11, 12, 13, 14, 15, 16]. These systems primarily struggle to foster effective collaboration through conversational and tool-based interactions, which leads to challenges such as achieving coherent interactions, mitigating unproductive feedback loops, and guiding meaningful collaborative engagements [11, 12, 35, 17, 29]. Multifaceted workflows demand well-structured Standardized Operating Procedures (SOPs) to ensure efficacy. A comprehensive understanding and integration of real-world practices is essential. Addressing these prevalent limitations and integrating these insights can facilitate the emergence of an innovative paradigm in the design and organization of LLM-based multi-agent systems, thereby enhancing their effectiveness and applicability.

Furthermore, through prolonged collaborative practice, humans have developed widely accepted SOPs across many domains [1, 2, 3]. These SOPs play a critical role in supporting task decomposition and efficient coordination. For instance, in software engineering, the waterfall methodology delineates orderly phases of requirements analysis, system design, coding, testing, and deliverables. This consensus workflow enables effective collaboration among multitudes of engineers [1, 2]. Moreover, human roles possess specialized expertise tailored to their assigned responsibilities:

*Chenglin Wu (E-mail: alexanderwu@fuzhi.ai) is the corresponding author with DeepWisdom, 361000, China

エージェント
(アーキテクト)

エージェント
(エンジニア)

エージェント
(プロダクト
マネージャ)

エージェント
(プロジェクト
マネージャ)

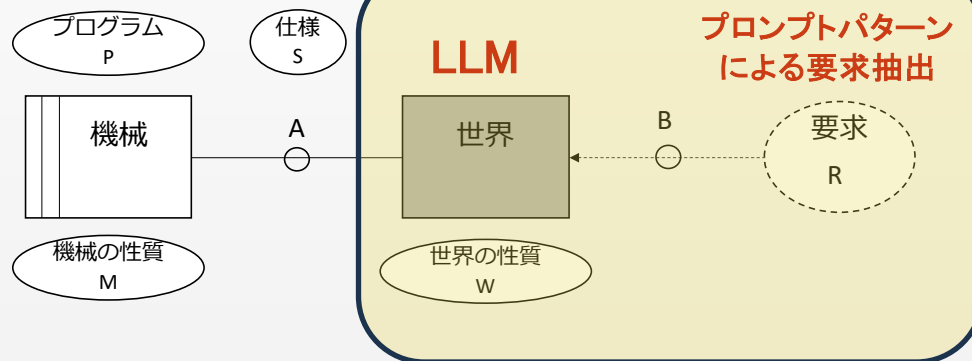
エージェント
(QAエンジニア)

LLMベースのマルチエージェントシステムによる自動化

プロンプトパターンカタログ

ペルソナ・プロンプトパターン

Contextual Statements
Act as persona X
Provide outputs that persona X would create



ペルソナの例(人間)

From now on, **act as a security reviewer**. Pay close attention to the security details of any code that we look at. **Provide outputs that a security reviewer would regard the code.**

ペルソナの例(人間以外)

You are going to **pretend to be a Linux terminal** for a computer that has been compromised by an attacker. When I type in a command, you are going to **output the corresponding text** that the Linux terminal would produce.

[1] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, Douglas C. Schmidt, A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT, arXiv:2302.11382, 2023

科目履修の流れ

まずは, LLMで何ができるかプログラミングを通じて学ぼう!

大規模言語モデルを組み込んだアプリ開発 (坂本)



次に, LLMのソフトウェア開発への適用について議論しよう!

大規模言語モデルのソフトウェア開発への応用 (鵜林)



最後に, LLMの原理について理解を深めよう!

生成モデルの基礎 (中井, 山口)

大規模言語モデルを組み込んだアプリ開発

学習のゴール

- 近年、大規模言語モデル(LLM)を活用したソフトウェアの普及が急速に進んでいます。
- LLMはチャットボットや要約などの自然言語処理が中心のソフトウェアで活用できるだけでなく、ユーザに自然言語ベースのユーザインターフェース(UI)を提供したり、LLMに行動の選択肢を選ばせることで、複雑な処理の自動化を実現したりできます。
- 本科目は、**LLMを組み込むことでソフトウェアの価値を向上させたり、これまで存在しなかった新しいソフトウェアを実現したりするスキルを習得**することを目的とします。
- 具体的には、Python言語やTypeScript言語と、LangChainなどの著名なLLM向けのフレームワークを組み合わせ、**LLM搭載アプリケーションのプログラミング演習を実施**します。

講義内容

- 1限目： LLMの基礎
- 2限目： プロンプティング演習
- 3限目： APIを介したLLM活用(基礎編)
- 4限目： LLM APIを活用したプログラミング演習(基礎編)
- 5限目： APIを介したLLM活用(発展編)
- 6限目： LLM APIを活用したプログラミング演習(発展編)
- 7限目： 発表とLLM最新動向の紹介

大規模言語モデルのソフトウェア開発への応用

学習のゴール

- 大規模言語モデル(LLM)の出現は社会の様々な活動に大きな影響を与えています。
- ソフトウェア開発も例外ではありません。LLMにより、プログラムを自動生成したり、バグを自動修正したりすることがある程度可能になっています。プログラムの自動生成、バグの自動生成の研究自体は2010年代におけるソフトウェア工学研究のメインピックの一つで様々な方式がこれまで提案されてきました。しかし、LLMはこれらの方式よりも優れた性能を発揮しており、**現在、伝統的なソフトウェア工学研究は大きな転換期に差し掛かっています。**
- 本科目では、**現在進行形で発展しているLLMのソフトウェア開発への応用について学びます。**本科目が、これからのソフトウェア開発のあるべき姿について考え直す機会になれば幸いです。

講義内容

- 1限目: LLMとソフトウェア開発
- 2限目: LLMの基礎
- 3限目: プロンプトパターンカタログとソフトウェア開発への応用
- 4限目: 演習(自身のソフトウェア開発にLLMがどう活用できそうかについてグループディスカッション)
- 5限目: LLMによるソフトウェア開発環境の拡張
- 6限目: 演習(LLMの出現により従来のソフトウェア工学がどう変化するかについてグループディスカッション)
- 7限目: LLMとソフトウェア工学研究最前線



生成モデルの基礎

学習のゴール

- 実用レベルの生成モデルについて、さまざまな学術論文が出版されていますが、これらの**モデルの仕組みを理解するための基礎知識を学びます**。
- 数学的な説明は最低限に留めて、モデルの主要な機能をシンプルに実装した**サンプル実装のコードを通し、「それぞれのモデルは何をしているのか」「なぜそれが生成モデルとして機能するのか」といった点を理解**します。
- 最後に、実用レベルの生成モデルの紹介と実務利用に向けた取り組みについても紹介します。

講義内容

- 1限目： 前提知識
- 2限目： 画像生成モデルの基礎
- 3限目： 自然言語モデルの基礎
- 4限目： 演習
- 5限目： より高性能な生成モデル
- 6限目： 演習
- 7限目： 生成AIの実務利用

本シリーズのポイント

- ① LLMを活用したソフトウェア開発の最前線が学べる
- ② 演習を通じてLLMのアプリ開発が体験できる
- ③ LLMの基礎理論について学べる

