

大規模言語モデル(LLM)シリーズ

シリーズリーダー 鶴林 尚靖 (早稲田大学)

講師陣

中井 悦司	(Google Cloud Japan 合同会社)
山口 光太	(株式会社サイバーエージェント)
坂本 一憲	(東京通信大学, WillBooster株式会社)
吉岡 信和	(QAML株式会社, 早稲田大学)
溝渕 裕司	(富士通株式会社)
近藤 将成	(九州大学)

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)

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, テキストデータの分析, 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, について幅広く習得する

基礎・応用系 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

言語モデルの基礎と応用

言語モデルの基礎的概念と活用法を習得

LLMの基礎概念、特性、活用、応用等

開発・実践系 2科目

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等

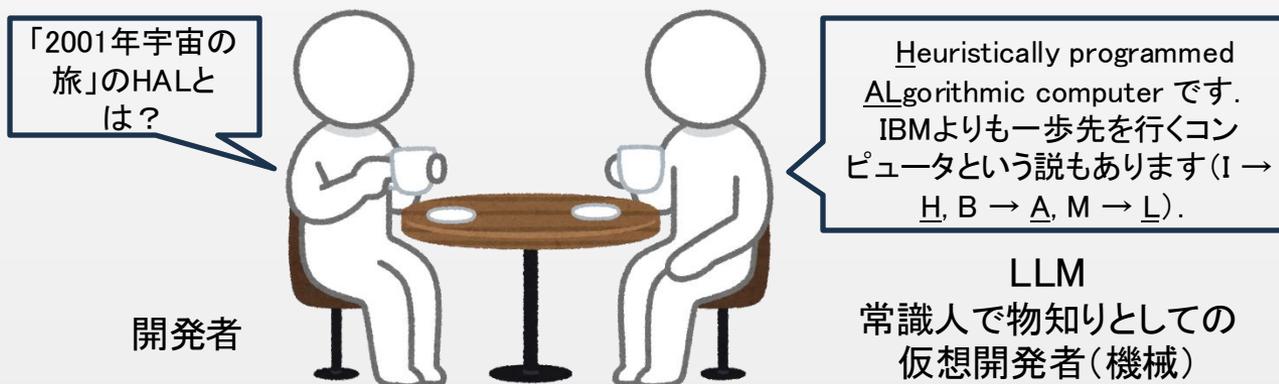
大規模言語モデルのソフトウェア開発への応用

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, LLMエージェントのソフトウェア開発への応用等

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 Xiwu Zheng² Jonathan Chen¹ Yuheng Cheng³ Ceyao Zhang³ Zili Wang¹
Steven Ka Shing Yau¹ Zijuan LIN² Liyang Zhou¹
Chenyu Ran¹ Lingfeng Xiao¹ Chenglin Wu^{1*}
¹DeepWisdom ²Xuaren University ³The Chinese University of Hong Kong, Shenzhen
¹Nanjing University ¹University 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/geekant/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 [1, 2, 3, 4, 5, 6]. 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 [7, 8, 9, 10, 11]. 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 [12, 13]. 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 [14, 15]. 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

arXiv:2308.00352v2 [cs.AI] 2 Aug 2023

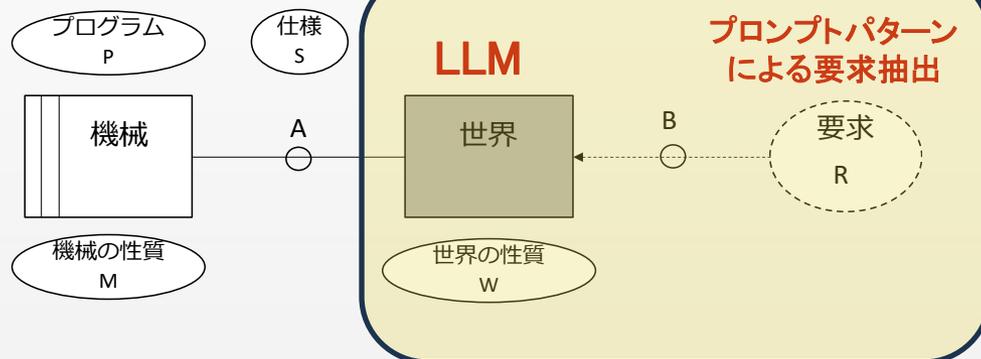
プロンプトパターンカタログ

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, 吉岡)



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 (鷓林, 近藤)



最後に、LLMの原理とテキストデータの活用・解釈について理解を深めよう！

生成モデルの基礎 (中井, 山口)

言語モデルの基礎と応用 (溝渕)

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

- 第1回: 生成AIおよび大規模言語モデル(LLM)に関する基礎
- 第2回: チャットボットの開発演習
- 第3回: LangChain Expression Language
- 第4回: LLM搭載アプリのAI倫理・セキュリティに関する要求とその担保
- 第5回: Retrieval-Augmented Generation (RAG)
- 第6回: Chainlitと自律型AI
- 第7回: グループ演習を通したLLMを組み込んだアプリの検討(1)
- 第8回: グループ演習を通したLLMを組み込んだアプリの検討(2)



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

- 第1回：LLMとソフトウェア開発(概要紹介)
- 第2回：グループ討議
- 第3回：プロンプトパターンとソフトウェア開発への応用
- 第4回：グループ討議
- 第5回：LLMとソフトウェア工学研究の最前線(LLMエージェントによる自動化など)
- 第6回：グループ討議
- 第7回：LLMによるこれからのソフトウェア開発環境(ソフトウェア開発実践演習での取り組み例の紹介)

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

- 第1回：前提知識(Kerasによる画像分類モデルの実装)
- 第2回：画像生成モデルの基礎(オートエンコーダーと潜在空間)
- 第3回：自然言語モデルの基礎(LSTMによるテキスト分類・テキスト生成)
- 第4回：演習
- 第5回：より高性能な生成モデル(Transformerモデル, Diffusionモデル, など)
- 第6回：演習
- 第7回：生成AIの実務利用例

言語モデルの基礎と応用

学習のゴール

- 初期の言語モデルから近年の大規模言語モデルにおける基礎的内容から応用まで俯瞰し、テキストデータの生成AIにおける実践的な活用・解釈を目指す。
- 本科目の受講生は、以下を習得済みであることが望めます。
 - (1) データサイエンスシリーズのテキストデータの基礎と応用
 - (2) 簡単な統計解析
 - (3) Pythonの基本的な操作

講義内容(仮)

- 第1回：言語モデルの基本概念・トレンド
- 第2回：分布意味論(Embedding)、LLMのアーキテクチャ(Transformer, Attention mechanism)
- 第3回：LLMの学習(LLMのPre-training, Fine-tuning, RLHF (Instruction Tuning))
- 第4回：演習(時間調整)
- 第5回：LLMの活用(Prompt Engineering、RAG、ReACT、Agent、データ合成)
- 第6回：LLMの様々な特性(Scaling law, ハルシネーション、説明・解釈可能性など)
- 第7回：LLMの応用(ソフトウェア開発への応用)
- 第8回：演習(時間調整)

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

