

LLM を用いた複数レシピに対する調理計画手法の検討

山口泰弘

クックパッド株式会社

yasuhiro-yamaguchi@cookpad.com

概要

複数の料理を同時かつ効率的に調理するには、利用可能な調理器具の制約と調理・物理現象に対する常識的な理解が不可欠である。自然言語で書かれたレシピから自動的に効率的な調理計画を組み立てるには大規模言語モデル (LLM) がもつ推論能力は非常に有用と考えられる。本研究では LLM を活用して同時調理の工程を計画する手法を検討し、解くべきタスクと課題の考察を行った。

1 はじめに

日常的な料理シーンにおいては複数の料理を同時に調理する場合が多く、複数のレシピから効率的な調理工程を計画するシステムは作業者の支援に有用である。しかし、一般的なレシピは人間が読むことを前提に作成されているため、こうしたシステムの構築にはいくつかの課題が存在する。

例として“ブロッコリーは塩ゆでし、水気を切って1口大に分けておく”という手順について考える。この記述には茹でる工程と切る工程の2つがあり、他のレシピと並行して調理する状況においてはこれらを分割して行うのが効率的な場合が考えられる。また、記載がないものの茹でる工程において通常はコンロを使用するため、作業者の利用する環境や他の工程における使用状況を考慮して計画を立てる必要がある。そして茹でる工程を始める前にはお湯が沸くまでの時間も考慮されるべきである。効率的かつ現実的なプランニングを行うには、こうした非明示的な情報を正しく補う能力が求められる。

計画を立てる段階においては、個々の作業だけでなく片付けにかかる時間や料理が完成するタイミングなどに関する常識的な判断を必要とする。

本研究では複数のレシピを同時かつ効率的に調理する工程の計画を立てることを目指し、大規模言語モデル (LLM) を用いた手法の提案と評価した結果を示した。

2 関連研究

調理工程の計画をタスクスケジューリング問題として定式化する選考事例として Nakabe ら [1] の研究がある。手順に紐づくリソースの情報を元に、複数レシピの並列調理可能にする効率的な計画を決定する。また、Kanazawa ら [2] は調理ロボットへの応用に向けて LLM を活用して自然言語のレシピからロボットの動作を計画する方法を提案している。

最近では LLM を用いて旅行や会議の計画といったプランニングタスクを解くための手法や評価のためのベンチマークがいくつか提案されているが、そうしたタスクは非常に困難なことが知られている。Zheng ら [3] は LLM のための旅行・会議・カレンダーに対するプランニングタスクを提案し、各種モデルや prompting 手法の評価を行った。Xie ら [4] は旅行の計画を立てるベンチマークタスクを提案し、制約について複数の観点で評価した。

3 タスクの定義

本研究では、1 人の作業者がある献立に含まれる複数のレシピを同時に調理するのに適切な作業工程を計画するタスクを考える。レシピは自然言語で書かれた複数の手順の連なりから構成されるものとし、与えられた複数のレシピの手順を組み替えて実現可能かつ効率的な作業工程を提案する。

一般に、多くの作業者にとっては全体の作業時間が短い方がより好ましい計画である。調理工程の途中に発生する焼く・煮るといった待ち時間の間に他の作業を進めることで、時間効率のよい計画を立てられる。一方で、温かい料理を作る場合は完成してから時間が経ち冷めてしまうのは好ましくないため、料理の完成のタイミングにも注意を払う必要がある。調理工程の計画においては時間の効率だけでなくこうした質的な側面も重要である。

より現実的な状況を考慮すると作業者が同時に利用することのできる調理器具の数は限られる場合が

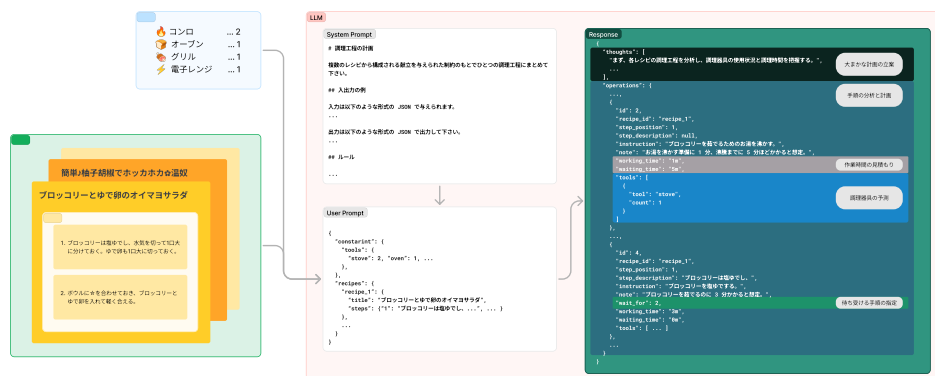


図 1: LLM による調理工程の計画手法

ほとんどである。並列に実行する作業を含む計画を立てる際には、利用可能な調理器具の個数を超えないことが求められる。

調理工程の計画においては、調理や時間・物理現象に対する常識的な理解も非常に重要になる。レシピに書かれた手順を実行するには、使用する調理器具や作業時間・待ち時間といった情報が必要になるが、そうした細かな情報は明示的に記述されない場合も多い。計画を立てる前に、非明示的な情報を適切に推測・補完する必要がある。

上記の要件・制約を踏まえて、本研究では調理工程の計画タスクを **手順の解析** と **工程の計画** の 2 つに分けて考える。

3.1 手順の解析

手順から計画に必要な情報を得るために次のサブタスクを定義する。

手順の分割 通常、レシピは作業者が理解しやすいように適当な粒度の作業に分割された複数の手順の連なりとして記述されるが、その分割が複数のレシピを同時に調理する場合においても適当であるとは限らない。使用する調理器具や待ち時間を含む作業の単位に分割することで、より効率的な調理計画が 1 能になると考えられる。

手順の補完 レシピに書かれた手順の中には、暗黙的に作業時間や待ち時間・道具の使用を含む工程が含まれる場合がある。物理的に可能な計画を立てるためには、非明示的な工程を適切に補完が必要がある。

所要時間の推定 分割された各工程について所要時 1 を予測する。焼く・煮込むといった工程においては待ち時間が発生する場合が多く、その間に別の

工程を進められることを考慮して、ここでは所要時間を作業時間と待ち時間に分けてそれぞれ予測を行う。

調理器具の予測 分割された各工程について使用する調理器具を予測する。手順が適切に分割されていれば、ここで予測された調理器具はその工程の完了まで占有されると考えることができる。

3.2 工程の計画

手順の解析結果に基づいて、分割された手順を組み替えて調理工程の順序を決定し、適切な計画を作る。工程の計画は以下のルールに従うものとする。

作業時間の制約 1 人の作業者が調理する想定のため、作業時間が重複してはいけない。ただし、待ち時間の間に他の工程の作業を行ったり、2 つ以上の工程の待ち時間が重なるのは問題ない。

調理器具の制約 同時に利用できる調理器具の個数が指定され、これを超えるような並列作業を計画してはいけない。

4 手法

LLM を利用した調理工程の計画手法の概観を図 1 に示す。入力是指示と例示を含む System Prompt と、計画対象の献立と制約を含む User Prompt からなる。出力は JSON 形式のデータで、手順の分析と大まかな計画を自然言語で出力した後、計画された調理工程を含む。

4.1 Prompt

System Prompt はタスクの概要と入出力の例示、出力におけるルールの指定からなる。例示はあるひとつの献立について、想定される入力の形式と理想的

表 1: 人手評価の結果

(a) 手順解析の評価					(b) 計画の評価			(c) 総合評価		
正解率					制約を満たす事例の割合			成功率		
分割	補完	時間	器具	欠損率	調理器具	物理	4 段階評価	手順分析	制約	全体
85.5	84.3	96.2	95.7	40.4	92.9	96.9	3.17	13.4	92.1	13.4

な計画の出力を示している。ルールには入出力の項目に関する説明と、計画を立てる上での注意事項が記載されている。

User Prompt は同時に利用できる調理器具の個数についての制約と、献立に含まれるレシピの情報(タイトル, 材料欄, 手順)を含む JSON 形式のテキストである。

実験に利用した System Prompt・User Prompt の詳細は付録 A に記す。

4.2 LLM の出力

LLM は System Prompt の指示に従い, User Prompt の内容に基づいて以下の情報を JSON 形式で出力する。

thoughts: 与えられた入力について考察を行い, 大まかな計画の案を自然言語で出力する。

operations: 分割された作業の内容を, 開始する順序に従って出力する。各工程には識別子 id, 作業時間 working_time, 待ち時間 waiting_time, 終了を待つ必要がある工程の識別子 wait_for (optional), 使用する調理器具のリスト tools (optional) が含まれる。

4.3 計画

LLM の出力に対して以下のルールに基づいて各工程の開始のタイミングを決定し, 計画を完成させる:

1. operations の順序に従って工程を並べる
2. 直前の工程に待ち時間があり, wait_for による依存関係がない場合は直前の工程の作業完了直後から次の作業を開始できる
3. wait_for によって繋がった一連の工程全体の所要時間が前の工程の待ち時間より短い場合, 一連の工程の終了を前の工程の待ち時間の終了にそろえる

5 実験

LLM を用いて複数のレシピを含む献立に対して調理計画を生成し, 人手による評価を行った。

5.1 データセット

Cookpad Recipe Dataset [5] に含まれる献立データから, 2 ~ 4 個のレシピで構成されるものをランダム

に 127 件取得して調理計画のためのデータセットを作成した。

各献立について, 4 節の手法に基づき LLM で調理工程の計画を生成する。LLM には OpenAI の gpt-4o [6] を利用した。調理器具の制約は, コンロ・オーブン・グリル・電子レンジについて献立ごとに設定した。コンロは同時に使用可能な個数を 1 ~ 3 の間でランダムに割り当て, それ以外の調理器具はすべて 1 個とした。

5.2 評価指標

手順の解析結果と生成された計画の妥当性を確かめるために, 手順の解析と工程の計画について幾つかの観点に基づくチェックリスト・評価指標を導入した。指標の詳細は付録 C に記す。

それらの評価指標に基づいて, 1 人のアノテータが各献立の計画に対して評価を行った。

6 結果と考察

6.1 手順の解析

手順の解析結果に関する評価を表 1a に示す。各指標はそれぞれの観点に合致するレシピの割合となっている。正解率においては所用時間と調理器具の予測が 95% ~ 96% ほどと比較的高い結果となった。この中では手順の補完についての正解率が最も低く, 揚げ物の調理に使う油の加熱やオーブンの余熱が必要な手順において作業開始前の加熱時間が考慮できていない例が複数確認された。

欠損率は, LLM の出力した工程からレシピに書かれた工程の一部が欠落している手順を含むレシピの割合を示している。欠損した事例を見ると, 複数の工程が記述された手順において後に登場する工程が失われる場合が多かった。今回の手法では全てのレシピの情報を同時に処理するため, LLM の処理するコンテキストが長くなり見落としが発生する確率が高くなっていると考えられる。手順の解析を個別に実行することで, 欠損率を下げられる可能性がある。

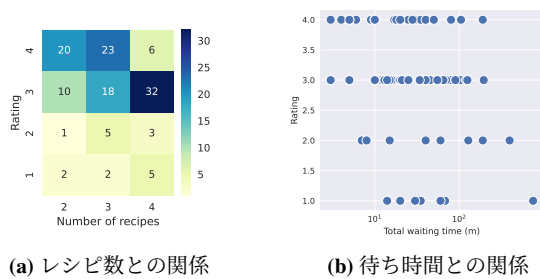


図 2: 4 段階評価の詳細

6.2 調理工程の計画

調理工程の計画について評価した結果を表 1b に示す。制約を満たす事例の割合は調理器具・物理的制約の双方で 9 割を超えている。これは LLM の出力から作業のタイミングを決定するアルゴリズムが作業の重複を抑えるルールを含んでいるため、制約を満たしやすくなっていると考えられる。

4 段階評価の平均は 3.17 となった。レシピ数と評価の関係を図 2a に示す。評価スコアが 1, 2 の事例は 18 件と他に比べて少ないことがわかる。これは多くの事例において制約が満たされるという表 1b の結果を反映したものと考えられる。評価スコアが 3, 4 の場合をみると、特にレシピ数が 4 件の事例において評価 4 を獲得した事例数が少なくなっている。また、工程全体の待ち時間と評価スコアの関係 (図 2b) をみると、待ち時間が長いほど評価スコアが低下する傾向が見られた。

各献立に対する手順の解析と計画の全体を通しての成功率を表 1c に示す。手順分析の列は献立に含まれるレシピの全ての手順の解析に成功した割合、制約の列は調理器具・物理的制約の双方を満たした計画の割合、全体は手順分析と制約の双方に問題のない計画の割合を示す。手順解析の誤りが全体の評価に大きな影響を及ぼしていることがわかる。

低い評価の事例を見ると、続けて行うべき調理工程の間に他のレシピの手順が割り込んだものや、wait_for の指定が不適切で並列に作業すべき工程が見落とされているものが多く見られた。図 2a の結果からも、レシピ数や工程が増えると一次的に並べられた工程から作業の連なりや並列性を認識することが困難になったと示唆される。

6.3 計画の自動評価

調理工程の計画に対する LLM の認識能力を確認するために、生成され調理工程に対して LLM を用

表 2: LLM による評価と人手評価の比較

(a) 手順解の評価					
一致率					
分割	補完	時間	器具	欠損	
73.9	77.2	80.2	63.5	69.5	
(b) 計画の評価					
4 段階評価	制約に関する判断の一致率				
相関	調理器具	物理			
0.28	90.6	94.5			

いた評価を行い、その結果が人間の評価とどの程度類似しているかを検証した。評価のための指示にはタスクの概要と 5.2 節で示した評価指標の説明を与え、各評価項目に対する判断の結果とスコアを生成した。

人手評価と LLM による評価の結果を表 2 に示す。手順解析の評価 (表 2a) を見ると、各観点での評価結果は 60 ~ 80% ほどであり、所用時間の予測に対する判断の一致率が最も高いことがわかる。人手による手順解析の結果では所要時間の正解率が最も高かったことから、工程にかかる時間の予測・認識においては比較的よい判断を下すことができると考えられる。欠損の検出は一致率が最も低く、生成・評価の双方において情報の欠落に対する対応が課題であることがわかった。

計画の評価に関する類似性 (表 2b) をみると、制約に関する判断は 90% ほどの一致が見られた一方、4 段階評価のスコアの相関は 0.28 であり、強い相関があるとは言えない。自動評価の結果 (表 3 を見ると、人手の評価に比べて高いスコアになる傾向がある。ここでも欠損や作業の連なり・並列性の認識に課題があることが示唆された。

7 おわりに

本研究では LLM を活用して同時調理の工程を計画する手法を検討し、生成結果の評価と分析を行った。評価の結果、使用する調理器具や所要時間は比較的高い精度で推論できることが示された一方、工程の欠損・補完、時間的な連続性や並列性の認識に大きな課題があることがわかった。欠損について手順の分析を個別のタスクに切り出すことで改善できる可能性がある。その他の課題については改善の糸口を見出すために更なる調査が必要と考えられる。

今後は手順の分析とプランニングのそれぞれについてより詳細な研究を進め、LLM の特性の理解と高精度な計画手法の開発に繋げたい。

参考文献

- [1] Jin Nakabe, Teruhiro Mizumoto, Hirohiko Suwa, and K. Yasumoto. Optimal cooking procedure presentation system for multiple recipes and investigating its effect. **Algorithms**, Vol. 14, p. 67, 2021.
- [2] Naoaki Kanazawa, Kento Kawaharazuka, Yoshiki Obinata, Kei Okada, and Masayuki Inaba. Real-world cooking robot system from recipes based on food state recognition using foundation models and pddl. **Adv. Robotics**, Vol. 38, pp. 1318–1334, 2024.
- [3] Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, and Denny Zhou. Natural plan: Benchmarking llms on natural language planning. 2024.
- [4] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In **Forty-first International Conference on Machine Learning**, 2024.
- [5] Jun Harashima, Michiaki Ariga, Kenta Murata, and Masayuki Ioki. A large-scale recipe and meal data collection as infrastructure for food research. In **Proceedings of the Tenth International Conference on Language Resources and Evaluation**, 2016.
- [6] OpenAI. Gpt-4o system card, 2024.

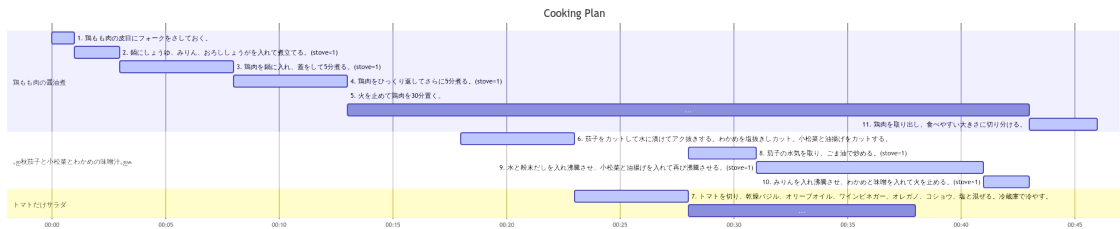


図 3: 生成された計画の例

A Prompt の詳細

図 4 に実験に使用した System Prompt を示す。入出力例の一部を省略して記載している。

```
# 調理工程の計画
複数のレシピから構成される献立を与えられた制約のもとでひとつの調理工程にまとめて下さい。

## 入出力の例
入力には以下のような形式の JSON で与えられます。

```json
{
 "constraint": { "tools": { "stove": 2, "grill": 1, ... } },
 "recipes": {
 "recipe_1": { "title": "オイマヨサラダ", "steps": ... },
 "recipe_2": { "title": "基本のひじきの煮物", "steps": ... },
 ...
 }
}
```

出力は以下のような形式の JSON で出力して下さい。

```json
{
 "thoughts": [
 "まず、各レシピの調理工程を分析し、調理器具の使用状況と調理時間を把握する。",
 ...
],
 "operations": [
 {
 "id": 1,
 "recipe_id": "recipe_2",
 "step_position": 1,
 "step_description": "ひじきは水に 3 0 分浸して",
 "instruction": "水を入れたボウルにひじきを入れ、30 分待つ。",
 "note": "浸す準備に 2 分ほどかかると想定。",
 "working_time": "2m",
 "waiting_time": "30m"
 },
 ...
]
}
```

## ルール
- 入力
  - 制約は `constraint` フィールドで与えられる
  - `constraint.tools` には調理器具の種類と同時に利用可能な数が与えられる
  - 調理工程の計画ではこの制約を満たすように効率的な調理工程を計画する
  - レシピは `recipes` フィールドで与えられる
- 出力
  - まず最初に `thoughts` フィールドで調理工程の計画に関する考察を出力する
  - 調理工程の計画を `operations` フィールドで出力する
  - 各レシピの調理工程を不可分な操作に分割し、作業の順序に従って並べる
  - 各調理工程は以下の情報を含む
    - `id`: 調理工程の ID (1 から始まる連番)
    - `recipe_id`: レシピの ID
    - `step_position`: レシピ内の工程の位置
    - `step_description`: レシピ内の工程から抽出した説明 (記述されていない工程を補足する場合は `null`)
    - `instruction`: 調理工程の実行内容
    - `note`: 作業時間や利用する調理器具に関する補足情報
    - `wait_for` (optional): 同じレシピの別の工程を待つ必要がある場合、その工程の ID を指定
    - `working_time`: 作業にかかる時間
    - `waiting_time`: 待ち時間
    - `tools` (optional): 使用する調理器具の種類と数
  - 作業者は `operations` の内容を順番に実行することで調理工程を進める
  - まず最初に `working_time` 分の作業を行い、その後 `waiting_time` 分待つ
  - 待ち時間が発生する場合、その間に次の工程をすすめる
  - ただし、`wait_for` が指定されている場合は指定された工程が完了するまで待つ
  - から作業を開始する
    - 同時に作業できない工程がある場合は適切に `wait_for` を指定する必要がある
```

図 4: System Prompt

B 生成事例

図 3 は LLM により生成された調理計画の例である。道具の制約はコンロ 2 つ・他 1 つの設定が与え

らた。人手による評価の結果は評価スコア 4 となっている。

C 評価指標の詳細

生成された計画に対して人手・自動で評価する際に利用した指標を以下に示す。

手順解析のチェックリスト:

- 分割: 工程の分割は適切か？
- 補完: 補完された工程は適切か？
- 時間: 所要時間の見積もりは適切か？
- 器具: 予測された調理器具は適切か？
- 欠損: 分割の結果、欠損した工程はないか？

計画の評価指標:

- 4 段階評価:
 1. 制約を満たしていない、実行困難な計画である
 2. 実行可能であるものの、順序が不適切である
 3. 作業自体は適切だが、明らかにより最適な計画が考えられる
 4. 概ね理想的な計画である
- 制約評価:
 - 調理器具: 制約が満たされているか？
 - 物理: 物理的に実現可能な工程か？

D 自動評価の結果

LLM による計画の評価を行った結果を表 3 に示す。

表 3: 自動評価の結果

| 4 段階評価 | 成功率 | | |
|--------|------|-------|------|
| | 手順分析 | 制約 | 全体 |
| 3.528 | 22.8 | 0.945 | 22.5 |