

# 自動生成されたサブタスクに対するユーザフィードバックの特性分析

堀川 達平<sup>†</sup> 北山 大輔<sup>†</sup>

<sup>†</sup> 工学院大学 情報学部 システム数理学科 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [tj321401@ns.kogakuin.ac.jp](mailto:tj321401@ns.kogakuin.ac.jp), [tkitayama@cc.kogakuin.ac.jp](mailto:tkitayama@cc.kogakuin.ac.jp)

**あらまし** 我々が抱えるタスクの中には、分解することでメリットが得られる複雑なタスクが存在する。そのため、複雑なタスクから分解されたサブタスクを自動的に生成する研究が行われており、様々な手法が提案されている。しかし、これらの手法は個人への最適化を狙っておらず、利用の際にはユーザに合わせた修正を行う必要がある。その際、サブタスク生成結果にはユーザによる修正に利益となるような情報が含まれていることが望ましいと我々は考えた。そこで本研究では、どのような生成結果がこのような情報を含んでいるかをサブタスク編集履歴の特性分析によって調査し、修正のための気付きを与えるサブタスク生成について考察する。

**キーワード** タスク管理, ユーザフィードバック, 行動解析

## 1 はじめに

我々は、不足した食料品の購入や、仕事、学校における提出物の準備等、様々なタスクを抱えて生活している。これら多くのタスクを抜けなく円滑に遂行するには、タスク管理が重要となる。このようなタスクを管理する需要から、近年は Trello<sup>1</sup> や Backlog<sup>2</sup> 等、様々なタスク管理サービスが運営され、利用されている。

タスクは時に完了のために複数のタスクの完了が求められるような複雑性を持つ場合がある。このような複雑なタスクは、具体的な行動計画の用意やサブタスクへの分解を行うことにより、完了までの時間が短縮できたり結果の質が高まるといったメリットが生じることが示されている [1] [2]。

タスク管理サービスにはチェックリスト、ガントチャート、リマインダーなどといった、作業手順を明確にしタスクの複雑性を低下させる様々な支援が存在する。しかし、これらの支援を受けるためには、ユーザ自身がタスクの詳細な内容を理解し、自ら計画の用意や分解を行う必要がある。このようなタスクの詳細化は煩雑で難しさがああり、タスク管理を進める上で大きな負担となっている。Yamamoto ら [3] が詳細化に関するテストコレクションを作成した際、3 人の評価者が検索クエリの分解に取り組んだが、その結果は適切な分解ができる人とそうでない人がいることを示しており、個人差はあれどタスクの詳細化が負担となっていることを示していた。

上記のような理由から、複雑なタスクを分解する研究が行われている。Zhang ら [4] は、複雑なタスクから分解されたサブタスク集合を自動的に生成するための手法として MSBART (Multi-Source Bidirectional Auto-Regressive Transformer) を提案し、BART (Bidirectional Auto-Regressive Transformer) [5]、T5 (Text-to-Text Transfer Transformer) [6] といった既存の言語モデルを上回る性能を示した。しかし、この手法は個人への

最適化を狙った手法ではないため、利用の際にはユーザに合わせた修正が必要となる場合がある。この時、修正によってより適切に分解されたタスク集合を得るために、生成結果に「不足しているタスクや複雑なタスクの具体的な内容についてユーザが発見することが可能となる情報」が含まれていることが望ましいと我々は考えた。

そこで本研究では、どのような生成結果がこのような情報を含んでいるかに関してサブタスク編集履歴の特性を分析することによって、修正のための気付きを与えるサブタスク生成について考察する。

本稿では、2 節でサブタスクによるタスク管理の効率化に関する研究、およびサブタスク生成に関する研究を概説し、本研究を位置付ける。そのうえで、3 節において、実験にて利用する自動生成手法についての概説を行う、4 節ではユーザの編集履歴取得の実験について解説し、どのような情報を集め、どのような内容が考察可能かを議論した上で、その実験結果を記載する。その後、5 節においてユーザに気付きを与える情報が何であるかを考察し、最後、6 節に今後の展望を記載する。

## 2 関連研究

複雑なタスクを分解しタスク管理を効率化する研究はこれまでに複数行われている。

森川ら [7] はタスク管理の際、遂行の見通しを立てるには、タスクの細分化により得られるサブタスクの詳細を把握する必要があることに着目し、サブタスクを期日のみを持つ「Todo 形式」と開始日時や終了日時を持つ「Event 形式」の 2 種類に分類して管理する手法を提案した。この手法はサブタスクの形式を上記 2 種類に仮定して情報量を削減することで、サブタスク想起の負担を軽減している。

荒井ら [8] は大きすぎるタスクは何をしたらいいかが具体的にイメージしにくい点に着目し、タスクの遂行時間の見積もりができるかという問いを投げかけることで適切な粒度への分割を促す手法を提案した。サブタスクの適切な粒度を「タスクの遂

1 : <https://trello.com/>

2 : <https://backlog.com/>

行時間を具体的に見積もることのできる大きさ」と定義することで、ユーザにサブタスク分割の基準を与えることにつながり、タスクの全体量や必要手順がつかみやすくなるといった効果が得られた。

これらの研究は複雑なタスクに分解する際の基準として時間を用いており、ユーザは時間を基準としてゼロからサブタスクを想起する必要がある点で共通している。我々の研究ではユーザは自動的に生成されたサブタスクを元に自分に合ったサブタスクを想起するため、想起がゼロからではないという点で違いが見られる。

ユーザの想起に頼らないサブタスク取得手法としては、加藤ら [9] の web からのタスク検索がある。この手法では、あるタスクを達成するために必要なサブタスクを web 検索の結果から求める。彼らは web 上での言及頻度が高いタスクは達成度が高いという仮説を立て、タスクの汎化、特化関係を考慮することにより、より正確なサブタスク集合が得られるようにした。しかしその実験結果を見ると、得られたサブタスク集合をすべて遂行しても元のタスクの達成率が 80% を超えないと感じられており、課題が残されている。

サブタスクを web 検索の結果から求める手法には、峯ら [10] のように検索先のサイトを限定し精度向上を図るという提案もある。こちらの手法は NTCIR11 Search Task Mining [11] という、テストセットとして与えられる検索クエリに対し、ランク付けされた「クエリを達成するための行動を表すタスク文字列のリスト」を返す、という課題を解決するために考案されたものである。その手法の内容としては、Q&A サービスがユーザの問題を解決するための回答を収集していることに着目し、Yahoo!知恵袋<sup>3</sup>を検索対象とすることで、より精度の高いタスク検索を可能とした、というものである。

また、これらと異なるサブタスク取得手法として、Zhang ら [4] の Learning to Decompose and Organize Complex Tasks がある。この研究では、大規模言語モデルである BART を複数ソースの参照が可能となるように拡張した MSBART を提案し、複雑なタスクの web 検索結果からサブタスク集合を生成できるようにした。さらに MSBART の中間表現を元にタスク間の時間依存関係を予測し有向グラフを出力できるようにした。この手法は前述の web からのタスク検索と比較すると、言語モデルを利用し web ページの要約を行っている点と、サブタスク間の時間依存関係を出力している点が大きく異なる。特に時間的依存性に関しては適切なデータセットが不足しており研究のハードルとなっていたが、この論文において新たにデータセットを収集、公開することにより、ハードルの解消に取り組んでいる。

これらのタスク検索に関する研究は、システムがユーザに依存せず自動的にサブタスクを生成している点で共通しており、我々の考える修正のための気付きを与えるサブタスク生成はこれらを発展させたものであると捉えることができる。ユーザが気付きを得て修正を行うことにより、web からのタスク検索における達成度が低いという課題を解決できる可能性があり、

ユーザが実際に利用する際の利便性向上につながれると考える。

### 3 複雑なタスクの自動的な分解

本節では、本稿におけるタスクの定義と Zhang ら [4] の提案手法である、MSBART による自動的なタスクの分解についての解説、並びに本研究で用いる自動的な分解手法についての解説を行う。また、より正確性の高い分解結果を用いるために、2022 年 12 月現在、対話形式において高い自然言語処理能力を持っているとして話題となった ChatGPT [12] を利用した分解手法についての解説も行う。

#### 3.1 タスクの定義

本稿では、タスクを「クリスマスプレゼントを買う」、「タイヤを交換する」、あるいは「より健康的な食事をする」といった、人々が追跡したい、思い出したい、または方法を学びたいという「目標を表す短文」として定義する。また、複雑なタスクとは、「完了するために 2 つ以上のタスクを実行する必要があるタスク」として定義する。複雑なタスクの完了に必要な個々のタスクはサブタスクと呼ぶこととする。

これらの定義をもとに、タスクの分解とは、複雑なタスクを個々のサブタスクに分解しサブタスク集合を生成することであると定義する。

#### 3.2 MSBART による自動的なタスクの分解

Zhang ら [4] の手法では、web 上にある非構造化テキストから複雑なタスクに関連する情報を見つけ出し、サブタスクに分解することを提案している。具体的には、複雑なタスク  $t$  が与えられた時、まず、検索エンジンで「Hot to  $t$ 」という用語を検索し、最も関連性の高い  $k$  個の結果を関連記事集合  $D_k(t)$  に保存する。そして、複雑なタスク  $t$  と関連記事集合  $D_k(t)$  をもとに、サブタスク集合  $ST(t)$  の生成を行う。ここで、生成される各サブタスクは次の 3 つの要件を満たす必要があると主張している。

- (1) **関連性**：生成されたサブタスクが複雑なタスク  $t$  に直接関連付けられるようにする
- (2) **抽象性**：「How To」記事はサブタスクについて説明、および発展させることが多いため
- (3) **合意性**：複数の情報源で引用されているサブタスクは重要である可能性が高いため

Zhang らは、SoTA を達成したテキスト生成用の Seq2Seq モデルである BART をもとに、上記 3 つの要件を注入する改良を加え、サブタスク集合生成モデルとして MSBART を作成した。具体的には、BART がマルチソース入力を処理できるようにし、関連性認識が行えるようカスタマイズされた cross-attention 層を設計し、生成プロセスを補助するクラスタエンコーディング手法を実装した。

#### 3.3 BART による自動的な分解の実装

Zhang ら [4] が提案した MSBART は、実装手法こそ論文に

3 : <https://chiebukuro.yahoo.co.jp/>

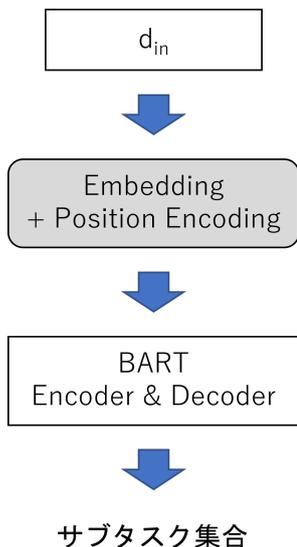


図1 サブタスク生成手法の概要図

記載されているものの、その学習済みモデルやソースコードは公開されていない。また、本稿の目的である生成されたサブタスクとそれに対するユーザフィードバックの特性分析は、生成手法には依存しない。そのため、本研究ではMSBARTを使用せず、MSBARTの比較手法としても言及されていた、一般公開されているBARTを用いて分解を行うこととする。特に本研究では日本語を対象言語とし、事前学習済みBARTは京都大学が公開している「BART日本語Pretrainedモデル」<sup>4</sup>を利用する。

ここで、BARTはマルチソース入力に対応できないため、学習・推論時は検索結果上位1件のソースを入力することでこれを代替する。また、Zhangら[4]の比較時の条件に倣い、関連性認識クロスアテンションとクラスターエンコーディングは利用しないこととする。

図1に、本研究で用いるサブタスク生成手法の概要図を示す。図中の $d_{in}$ は入力されるソーステキストを示している。この図からわかる通り、BARTのEncoder, Decoderを一切変更せずに実装できるため、簡便かつ堅実な動作結果が期待できる。

学習の際のデータセットは、サブタスク集合の生成についてはZhangら[4]に倣い日本語wikiHowデータセット<sup>5</sup>を元に作成したWKH-Rデータセットを使用した。時間依存関係の推論については本研究では実施しないこととした。また、学習時のパラメータはBART日本語Pretrainedのgithubリポジトリ内の使用法<sup>6</sup>に記載されていたFinetuneのパラメータを元に、入力文字数を増加させるため表1に従う変更を加えた。

### 3.4 ChatGPTによる複雑なタスクの自動的分解

OpenAIの公開したChatGPT[12]は、名前の通りチャットを利用して、ユーザからの様々な質問に対して応答を行う言語

表1 変更を加えた学習パラメータ

引数名	変更前	変更後
total-num-update	40000	1000
max-tokens	1024	4096
max-source-positions	無し	4096
update-freq	2	5
max-update	5000	削除
max-epoch	無し	1000
batch-size	無し	1

モデルである。ChatGPTは高い言語理解能力を備えており、例えば「〇〇を行うPythonの関数を出力して」と入力することで動作可能なPythonコードを出力することができたり、「下記を要約して」という文言と共に長文を入力するとその要約結果を出力したりすることが可能である。このような期待する動作の内容を表す文言はプロンプトと呼ばれており、複雑なタスクを自動的に分解するには、これを適切に設計する必要がある。本研究では、第3.2節で用いた「How to  $t$ 」をもとに、日本語のタスク分解プロンプトを「 $t$ 方法」をタスク化して」という文言に定めることとした。

ChatGPTでは、サービス運用上の制約からか、「また」や「そして」といったような文の途中で出力が止まってしまうことがある。しかし、ChatGPTはそのチャットセッション中の文脈を記憶しており、「続きを生成」と入力することで、ChatGPT自身の発言から続きを生成することが可能である。

これら「 $t$ 方法」をタスク化して」と「続きを生成」の2つの入力を用いて自動的なタスクの分解を実施し、より正確性の高い結果としてこれを利用する。

## 4 実験

本節では、ユーザに気付きを与える情報が何であるかを考察するための、被験者の編集履歴取得実験について解説する。この実験はサブタスク集合の生成に処理時間を要するため、主に3段階に分けて行われる。1段階目では被験者から複雑なタスクを募る。2段階目では募った複雑なタスクをもとにサブタスク集合を生成し、それらを被験者に編集させ編集履歴を取得する。最後に編集終了後に3段階目としてアンケートを実施し、修正意図などの編集履歴からは取得しきれない主観的な情報を取得することを狙う。被験者の募集は日本のクラウドソーシングサービスであるクラウドワークス<sup>7</sup>を利用して行われた。

### 4.1 複雑なタスクの収集

この段階では、募集した被験者に対し、日常に存在する複雑なタスクを入力してもらう。入力する際に複雑なタスクがイメージしやすいよう添えた文面は、同一のワーカーが複数回答を行う際に似たタスクを入力しないようにさせるため、下記のような文言とした。

4: [https://nlp.ist.i.kyoto-u.ac.jp/?BART 日本語Pretrained モデル](https://nlp.ist.i.kyoto-u.ac.jp/?BART%20%E6%9C%A9%E8%AA%8C%E3%81%97%E3%81%99%E3%81%87%E3%81%82%E3%81%84%E3%81%86%E3%81%88%E3%81%8A%E3%81%8C%E3%81%8E%E3%81%90%E3%81%92%E3%81%94%E3%81%96%E3%81%98%E3%81%9A%E3%81%9C%E3%81%9E%E3%81%A0%E3%81%A2%E3%81%A4%E3%81%A6%E3%81%A8%E3%81%AA%E3%81%AC%E3%81%AE%E3%81%B0%E3%81%B2%E3%81%B4%E3%81%B6%E3%81%B8%E3%81%BA%E3%81%BC%E3%81%BE%E3%81%B0%E3%81%B2%E3%81%B4%E3%81%B6%E3%81%B8%E3%81%BA%E3%81%BC%E3%81%BE)

5: <https://github.com/Katsumata420/wikihow-japanese>

6: [https://github.com/utanaka2000/fairseq/blob/japanese\\_bart\\_pretrained\\_model/JAPANESE\\_BART\\_README.md](https://github.com/utanaka2000/fairseq/blob/japanese_bart_pretrained_model/JAPANESE_BART_README.md)

7: <https://crowdworks.jp/>

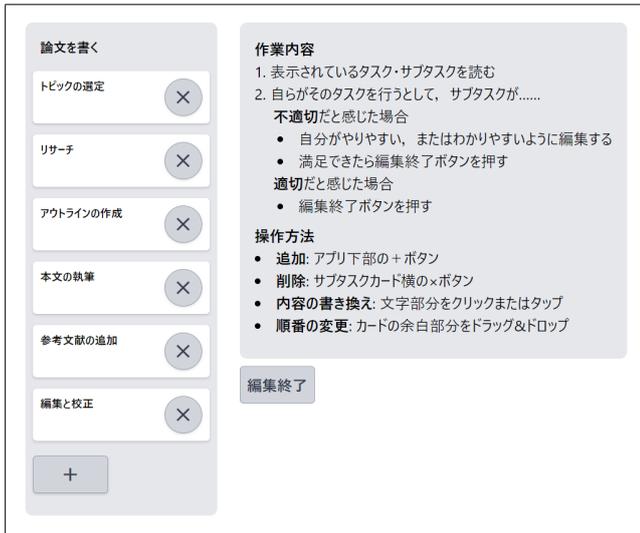


図 2 実装した編集インターフェース

最近行ったこと、もしくは近日中に行いたいことや長期的な達成目標を、可能な限り具体的に入力してください。内容は、学業や仕事、趣味、日常生活に関することや、家族や友人との約束事など、なんでも構いません。なお、複数件の回答を行う際は、各回答の大局的な目的・目標は独立するようにしてください。

#### 4.2 サブタスク編集履歴の取得

この段階では、改めて募集した被験者にサブタスク集合を編集させその履歴を取得する。

編集されるサブタスク集合は、事前に収集された複雑なタスクをもとに生成しておき、「複雑なタスク、生成されたサブタスク集合」のセットで被験者に提示する。ここで、サブタスク集合は、BARTにより生成されたより多くの修正が必要となる集合か、ChatGPTにより生成された過度に一般的な集合のどちらか一方をランダムで提示する。被験者は複雑なタスクを実際に行うとして、それが実行しやすいようにサブタスク集合の編集を行うよう依頼される。この編集内容には、例として、「用語の書き換え」「不足したサブタスクの追加」「細かすぎるサブタスクの削除・統合」などが考えられる。

上記のような編集内容の履歴取得のため、図2のような専用の編集インターフェースを実装した。編集操作は各サブタスクについて「追加」「削除」「中身の変更」「順番の入れ替え」が実行可能である。

各編集操作には被験者の様々な意思や目的が背後にあると仮定している。編集操作ごとに仮定した意思・目的を以下に述べる。

**追加** 要素が不足している場合に行う。追加の中にも、背景情報の差により次の2つのパターンがあると考えている。

- 生成サブタスク集合から着想を得て追加内容を思いついた場合

これが実行されるものが“良い”サブタスク集合である

- 元々思い描いていたサブタスク集合に対し生成サブタスク集合の内容が不足している場合

これが実行されるものが“悪い”サブタスク集合である

**削除** 不要なサブタスクが多い場合に実行される。生成サブタスク集合にタスクと関係のないものが含まれている、あるいはサブタスクが細分化されすぎている状況が考えられる。

**中身の変更** ユーザが内容を理解した上で、適切な用語ではないと感じたり、理解に時間のかかるものであると感じた場合に、自分の理解しやすい内容に変更する操作が考えられる。

**順番の入れ替え** 生成結果が適切な時間的依存関係を生成できていない場合にこれを修正する状況が考えられる。また、ユーザの経験や好みから、順番を入れ替える可能性が考えられる(簡単なものからこなすか、重いものからこなすか、など)。

#### 4.3 タスク編集者の主観の取得

編集に対する主観は編集履歴のみでは判断することができないため、これを取得するため4.2節の被験者に対し、アンケートを実施する。実施するアンケート内容と、その実施意図について以下に述べる。

- 生成サブタスク集合の適切さ

生成サブタスク集合の良し悪しを測る指標として、タスク達成への適切さが考えられた。生成サブタスク集合がタスク達成に不適切であるという主観は、被験者が思い描く理想的なサブタスク集合に対し、生成内容が不足していることにより生じると考えられる。このことから、適切、不適切という指標をサブタスク集合の良し悪しと同等に扱うことが可能となる。この設問の回答結果と操作量を突き合わせ、予想される操作量でサブタスク集合の良し悪しを決めることができるのではないかと考えた。本設問では「適切」から「不適切」までを5段階評価で調査する。

- サブタスク編集の際、生成集合から発想を受けたか

サブタスクの編集内容が元々思いついていたのか、情報を得た結果なのかを確かめるために利用する。本設問では「はい」、「いいえ」の二択で調査する。

#### 4.4 結果

##### 4.4.1 複雑なタスクの収集結果

59名が参加し、100件のタスクが収集された。収集されたタスクの例を表2に示す。これらの内容を確認したところ、複数の複雑なタスクが記載されているものや、理解の難しいタスクが存在していた。それらは全て手作業で修正を行った。修正後のタスクの例を表3に示す。

この修正済みタスクと予備実験で収集されたタスクを合わせ、合計172件となったタスクのうち20件をタスク編集に利用することとした。この時、利用するタスクが似すぎないようにグルーピングを行った上で、同一グループから利用するのは最大で5件までとして、出現頻度が多い順に選択した。

##### 4.4.2 編集履歴の取得結果

332名が参加し、合計で1838件の有効な編集履歴が取得された。まず、全体的な結果として、生成されたタスクの編集結



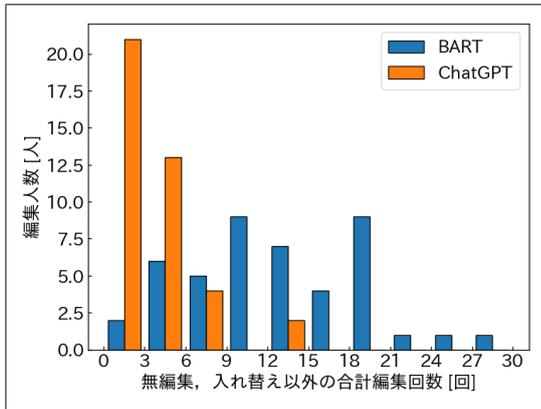


図 8 特定の編集操作における合計編集回数の頻度分布

るだけなど、その変更処理の多くを既存のテキストに頼っている編集結果が散見された。また、生成集合のほぼ全てを書き換えている結果も存在した上、その生成集合の出来の悪さからか、見るからに達成可能性の低い集合にも関わらず編集を終了している結果も見受けられた。全体的な傾向として、ChatGPTでは一つ、二つ程度の軽微な変更が多く、BARTは一気に全て変更する人とそうではない人の二極化が起きているように観測された。

この傾向を確認するため、編集量の分布をみることにする。図 7 の結果から、比較の際により分かりやすい差が出ると考えたため、特にタスク「編み物の作品を作る」についての傾向を確認する。タスク「編み物の作品を作る」の編集において、操作なしの件数と入れ替え操作の回数を除いた編集回数を編集者ごとに合計し、その合計回数を元に、bin 幅を 3、bin 数を 11 として作成したヒストグラムを図 8 に示す。この図からは、BARTにおいて被験者ごとに編集量がまばらなこと、ChatGPTにおいて数回の編集操作でそれを終える被験者が多いことが読み取れる。実際に確認された編集結果の一部を例として表 4、表 5 にそれぞれ示す。編集結果-A は文体が影響を受けている点や生成集合をそのまま利用している点など、既存の生成集合に頼った編集結果であることがわかる。編集結果-B は A とは大きく異なっており、生成集合のほぼ全てを書き換えた結果といえる。編集結果-C、D は生成集合に対して軽微な変更を行った例といえる。

上記の確認の結果、ChatGPT の生成集合では不足している要素を発見し追加させることに成功していると考えられる。一方で、BART の生成集合を元に達成可能なサブタスク集合へ編集できている人は、おそらく事前知識を持っている、生成集合をあてにしない人であると考えられる。

このような結果から、具体的な手順を表す生成集合は、経験がある被験者に対して有効な情報を提供することができていると考察する。被験者は生成集合に含まれる情報から記憶を想起し、最小の追加処理でサブタスク集合を完成させていると考えられる。一方で経験を持たない被験者は想起する記憶がないため不足している情報に気付くことができない。このような理由から、具体的な手順を示す ChatGPT の出力であっても、編集せずに終了する被験者と細かな編集を加える被験者の二極化が

起きているのではないかと考えられる。

考察をまとめると、次のようになる。

- ドメイン知識のあるユーザは具体的なタスクの集合を読み解くことで、自分の作業との不足点に気付けると考えられる
- ドメイン知識のないユーザは提示された情報を受け入れる他なく、不足しているタスクをそもそも知識として持っていないため気付くことができない

以上のことから、ユーザに修正のための気付きを与える情報とは、おそらく具体性の高い集合のことであり、その内容と自身の想定する行動との差から、必要なタスクを追加できるのではないかと考える。また、このことから、修正のための気付きを与えるサブタスク生成には、サブタスク集合の具体性を高めることが必要であると考えられる。

## 6 おわりに

本稿では、サブタスク集合の自動生成手法において、その生成結果に「不足しているタスクや複雑なタスクの具体的内容についてユーザ発見することが可能となる情報」が含まれていることが望ましいと考え、どのような情報がこれにあたるかを考察した。生成手法の実装にあたっては、高い性能を示した先行研究の手法が簡単に利用できなかったため、その先行研究を参考に簡易的な実装を行うこととした。実験については、生成元である複雑なタスクの収集、生成結果であるサブタスク集合を用いたサブタスク編集履歴の取得、さらに編集終了後のアンケート調査の 3 段階の構成で実施した。アンケート調査の実施により、編集履歴からは得ることができない被験者の意図などの取得を行った。

実験結果を精査した結果、ユーザに気付きを与える情報とは、具体性の高い情報であることが示唆された。ユーザはその内容と自身の想定する行動との差から、不足している必要なタスクに気付くことができるのではないかと考えられる。

具体性の高いサブタスク集合を自動生成する手法の開発は、今後の課題である。

## 謝 辞

本研究の一部は、2022 年度科研費基盤研究 (C)(課題番号：21K12147) によるものです。ここに記して謝意を表すものとします。

## 文 献

- [1] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moon-tae Lee, Florian Schulze, and Scott R. Klemmer. Taskgenies: Automatically providing action plans helps people complete tasks. *ACM Transactions on Computer-Human Interaction*, Vol. 20, No. 5, pp. 1–25, 2013.
- [2] Justin Cheng, Jaime Teevan, Shamsi T. Iqbal, and Michael S. Bernstein. Break it down: A comparison of macro- and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 4061–4064, 2015.
- [3] Takehiro Yamamoto, Tetsuya Sakai, Mayu Iwata, Chen Yu,

表 4 BART により生成されたサブタスク集合の編集結果の例

生成集合	編集結果-A	編集結果-B
編み物を始めよう 編み物の基本とレシピ集 38 yarn ニット教室自宅で過ごす時間に飽きてきた」という人も多いのではないかな 編み物は奥が深い おすすめの編み方編み物初心者さんが知っておきたい、おすすめレシピ集【かぎ針編み】おすすめレシピを作成します 編み物を始めたい人は、自分の好みで編み物を作ります おすすめの編み物を作りたい人は、おすすめのレシピを作成しましょう おすすめのアイテムを作きましょう おすすめのアイテムについて、おすすめのおすすめアイテムを作成しましょう 「おすすめのアイテム」を作きましょう 「編み物」を作きましょう	編み物を始めよう おすすめのアイテムを作きましょう おすすめの編み物を作りたい人は、おすすめのレシピを作成しましょう おすすめの編み方編み物初心者さんが知っておきたい、おすすめレシピ集【かぎ針編み】おすすめレシピを作成します	編み棒・編み針について理解する糸や針を用意する 作りやすいアイテムを選ぶ 小物から挑戦してみる

表 5 ChatGPT により生成されたサブタスク集合の編集結果の例

生成集合	編集結果-C	編集結果-D
編み物のパターンを選ぶ 編み物に必要な糸や針を用意する 編み方を学ぶ（本や動画などを参考に） 編み物を始める 完成したら、洗う、整える お楽しみください	編み物でどんなものを作りたいか考える 編み物のパターンを選ぶ 編み物に必要な糸や針を用意する 編み方を学ぶ（本や動画などを参考に） 編み物を始める 完成したら、洗う、整える	編み物のパターンを選ぶ 編み物に必要な糸や針を用意する 編み方を学ぶ（本や動画などを参考に） 編み物を始める 完成したら、水通しを行い整える

- Ji-Rong Wen, and Katsumi Tanaka. The wisdom of advertisers: Mining subgoals via query clustering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pp. 505–514, 2012.
- [4] Yi Zhang, Sujay Kumar Jauhar, Julia Kiseleva, Ryan White, and Dan Roth. Learning to decompose and organize complex tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2726–2735, 2021.
- [5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.
- [6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, Vol. 21, No. 1, pp. 5485–5551, 2022.
- [7] 森川佑希, 菅野裕基, 高田秀志. タスクの見直しに着目した細分化を特徴とするタスク管理手法. マルチメディア, 分散協調とモバイルシンポジウム 論文集, Vol. 2020, pp. 456–466, 2020.
- [8] 荒井健太郎, 小林稔. タスクの遂行時間に着目した適切なタスク分割支援手法. 研究報告デジタルコンテンツクリエーション (DCC), No. 39, pp. 1–6, 2018.
- [9] 加藤龍, 大島裕明, 山本岳洋, 加藤誠, 田中克己. タスクの汎化と特化に着目した web からのタスク検索. 電子情報通信学会, 第 6 回データ工学と情報マネジメントに関するフォーラム (DEIM Forum 2014), pp. C1–6, 2014.
- [10] Mine Shohei, Matsumoto Takuma, Yoshida Tomofumi, Shinohara Takuya, and Daisuke Kitayama. InteractiveMedi-aMINE at the NTCIR-11 Imine Search Task. in *Proc. of The 11th NTCIR Conference*, 2014.
- [11] NTCIR-11 IMine. <http://research.nii.ac.jp/ntcir/permission/ntcir-11/perm-ja-IMine.html>, 2015. Accessed: 2022-08-02.
- [12] Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt/>, 2022. Accessed: 2022-12-30.