Rationale の自動生成による CoT データセット構築

横野 光 [1]　平岡 達也 [2,3]　関根 聡 [4]
[1] 明星大学　[2]MBZUAI
[3] 奈良先端科学技術大学院大学　[4] 株式会社いちから
hikaru.yokono@meisei-u.ac.jp　tatsuya.hiraoka@mbzuai.ac.ae
satoshi.sekine@ichikara.ai

概要

本論文では LLM の推論性能の向上に利用できる Chain-of-Thought のデータセットの構築について述べる．rationale (思考過程) を人手で作成することは困難であるため，既存のインストラクションデータに対して LLM による rationale の自動生成と，生成した rationale の LLM と人手による二段階のフィルタリングを行うことで人手の作業コストの軽減を図った．また，JEMHopQA を用いて構築したデータセットによる追加学習の影響を分析した．

1　はじめに

Chain-of-Thought (CoT) は，大規模言語モデルにおいて問題を回答する過程にその思考過程を含めることで推論タスクの性能を改善できるプロンプトエンジニアリングの一種である [1]．しかし，その効果は大規模なモデルにおいて限られており，小規模なモデルでは CoT による性能改善が現れないということも示されている [2].

モデルの CoT による推論性能を向上させるために，CoT の事例を用いてモデルの追加学習をさせるという方法が考えられる．そのためには大量の事例が必要となるが，CoT データの構築において重要となる rationale(思考過程) の作成は，推論の過程を言語化する必要がある，同じ問題であっても回答に至る過程は一意とは限らない，といった理由からアノテーションの品質の管理が難しく，人手での構築は非常にコストがかかる．これに対して，LLM に問題とその回答から rationale を生成させ，それを人手で判定するということでデータ構築の効率化を図るという方法が考えられる．

本論文では，LLM を用いた rationale 生成による CoT データ構築手法と，ベンチマークを用いた構築したデータセットの評価について述べる．

2　関連研究

モデルの推論性能の向上のために有用な CoT データの自動構築に関して，Kim らは Flan Collection データセットの事例に対して複数の CoT の種類の rationale を LLM で生成した CoT データセットを構築した [3]．また，Zelikman らは問題から rationale と回答を生成させ，回答が正解と合っている場合はその rationale を採用し，そうでない場合は正解を提示した上で再度 rationale と回答を生成させ，正解であればその rationale を採用するという機械的な CoT データの作成手法を提案した [4].

生成された rationale の妥当性の機械的な検証について，Golovneva らは生成された rationale の入力との整合性や推論ステップ間での一貫性など，複数の指標に基づいた rationale の評価手法を提案した [5]. Prasad らは各ステップが入力や前のステップに対して妥当な推論になっているかや，各ステップが回答を導くための新しい情報を提供しているか，という観点に基づいた評価手法を提案した [6].

3　CoT データセット構築

本研究では，株式会社いちからが提供している Ichilara-Instruction2[7] のデータの問題と回答に対して，LLM (`LLM-jp-3-13b-instruct3`[8]）で問題から回答へと至る rationale を生成し，その妥当性を人手で判定することで CoT データセットを構築する．

Ichikara-Instruction2 には様々な分野，タイプの事例が収録されている．この中から今回は domain と source-to-answer の組がオープン質問-知識，ブレインストーミング (ブレスト)-知識，分類-知識，選択-知識の問題を対象とした．rationale の生成に関しては表 1 に示す 5 種類の CoT 種類を採用し，1 つの事例に対して CoT 種類毎に rationale を生成する． 以前のデータセット構築において，全ての分野に対し

表 1　CoT 種類

General Knowledge (GK)
回答に必要となる知識をリストアップし，それに基づいて最終的な回答を作成する．
Least to Most (L2M)
部分的な問題に分解し，順番に解決するようにして最終的な回答を作成する．
Plan and Solve (P&S)
先に問題を解く方針を考えてから，その方針に従って最終的な回答を作成する．
Self-refine (SF)
先に回答を考えてから，その回答が正しいかを自分自身でフィードバックし，それに基づいて最終的な回答を作成する．
Step by Step (SbS)
ステップバイステップに考えて回答を作成する．

て 5 つの CoT 種類全てが有用というわけではないという結果を得ていたため，今回のデータセット構築では分野と CoT 種類の組み合わせを選択した．

3.1　rationale の生成

rationale の生成に用いたプロンプトは CoT 種類に応じた指示と全体で共通した指示と CoT 種類毎に作成した Few-shot の事例で構成され，問題と回答のペアを入力として，rationale と入力の問題に対する回答を出力させるような指示となっている (図 1)．LLM に rationale だけでなく回答も出力させている理由は，rationale の妥当性の検証において使用することを考慮したからである．

```
あなたは質問応答の思考過程を再構築するアナライザーです．[問題] と [答え] に対して，その答えに至るまでの思考過程を次の方針で生成してください．
- 答えを導く過程において必要となる具体的な知識を [理由] として箇条書きで列挙する
……
- 最終的な答えを得るまでは，複数の候補や一般的な知識を検討するプロセスを書いても良い
【入力】
[問題]
# ここに問題が入る
[答え]
# ここに回答が入る
【出力フォーマット】
[理由]
{質問から回答に至る思考過程を記述}
答え:
{回答を記述}
```

図 1　GK の rationale 生成のプロンプト (一部)

rationale は各事例*CoT 種類に対して 5 個の候補を生成し，後段の妥当性判断で妥当なものを 1 つ採用した．rationale と回答の生成は LLM の出力がプロンプトで定めるフォーマットに沿った形になるまで繰り返しており，あらかじめ決定した生成の試行の上限回数に達してもフォーマットに沿った出力が得られなかった場合，その事例は除外した．

最終的に得られた rationale の例を表 2 に示す．

表 2　CoT データの例

Input （Ichikara-Instruction2 に収録）
香水はどこに浸けると人に近づけた時にこの人いいにおいすると思われますか?今使ってるのは Chanel のものです。
Output （Ichikara-Instruction2 に収録）
香水をつける場所に決まりはありませんが、腰付近や膝の裏につけると近い距離の人にだけ香らせることができます。脇など体温が高い部分は香水がよく香る部分でもあります。 その他、アキレス腱や太ももの内側、肘の内側なども効果的です。天気や塗布する場所、体温、肌質によって香りは変わりますのでいろいろな場所につけて楽しんでみるのも良いでしょう。
Rationale(SbS) （本研究で生成）
ステップバイステップに順序立てて回答を作成します。 ステップ 1: 香水をつける目的と条件の確認 まず、香水をつける目的は、近い距離にいる人にだけ「この人いいにおい」と思わせることです。使用している香水は Chanel のものであることがわかっています。 ステップ 2: 香りが効果的に漂う場所を探す 効果的な香りを漂わせるには、香水の香りが強くなりすぎず、適度に香る場所を選ぶ必要があります。腰付近や膝の裏、脇など体温が高い部分は、香水がよく香る部位とされています。これらの部位は皮膚が柔らかく、体温により香りが温められて揮発しやすくなるからです。 ステップ 3: おすすめする具体的な部位 腰付近や膝の裏以外にも、アキレス腱や太ももの内側、肘の内側も効果的です。これらの場所は、体の中心に近くて体温が高いため、香水がよく香ります。 ステップ 4: 注意点をまとめる 香りを効果的に楽しむためには、天気や塗布する場所、体温、肌質によって香りは変わることを念頭に、いろいろな部位につけてみることもおすすめします。

3.2　生成した rationale の妥当性の判定

生成した rationale が，問題から回答を導く思考過程として適切かどうかの妥当性判定を行った．判定は LLM と人手の 2 段階で行っており，LLM の妥当性判定を通ったものについてのみ人手での判定を行うことで，人手判定のコスト削減を図っている．

LLM (`gpt-oss-20B`[9]) による妥当性判定では，問題-rationale-回答を並べたテキストが一貫しているかを判定させた．生成された rationale の数と LLM による妥当性判定を通過した rationale の数の内訳を表 3 に示す．表中の “-” はその組み合わせでの rationale の生成を行っていないことを表す．

表 3　CoT 種類ごとの rationale の内訳

	GK	L2M	P&S	SR	SbS
オープン質問-知識	427/500	447/500	420/500	430/495	441/500
ブレスト-知識	-	-	356/475	-	379/480
分類-知識	232/465	-	238/460	-	233/465
選択-知識	182/240	-	164/200	-	200/245

(LLM によるチェックを通った Rationale の数/生成された Rationale の数)

人手の判定についても同様に，問題から回答を得るための思考過程として rationale が妥当かを判定した．最初に 1 人の作業者が LLM の妥当性判定を通過した事例に対して判定を行い，その後に別の作業者 1 人 (本論文の著者) が再度判定した．人手判定の際には軽微な誤りを修正している．

なお，各事例*CoT 種類に対して最大 5 個の rationale を生成しているが，人手の判定では rationale

の出力順に判定を行い，妥当な rationale を発見した時点でその事例*CoT 種類に対する判定を終了した．そのため，最終的なデータセットでは，事例*CoT 種類に対して 1 個の rationale が付与されている．

最終的に得られたデータは 668 件であった．ドメインと CoT の種類毎の rationale を生成した事例数，LLM による判定を通った rationale が少なくとも 1 つはある事例数，人手による判定を通った事例数を表 4 に示す．表中の “-” はその組み合わせでの rationale の生成を行っていないことを表す．

表 4　CoT 種類ごとの事例の内訳

	GK	L2M	P&S	SR	SbS	集計
オープン質問-知識	66/95/100	68/98/100	63/97/100	74/96/99	63/97/100	334/483/499
ブレスト-知識	-	-	57/86/95	-	66/92/96	123/178/191
分類-知識	56/74/93	-	43/70/92	-	56/70/93	155/214/278
選択-知識	19/48/48	-	17/38/40	-	20/48/49	56/134/137
集計	141/217/241	68/98/100	180/291/327	74/96/99	205/307/338	668/1009/1105

(人手のチェックを通った事例数/LLM によるチェックを通った事例数/Rationale が生成された事例数)

LLM による妥当性判定の効果は分野によって傾向が異なる．選択-知識では LLM による妥当性判定を通った事例のうち人手の妥当性判定を通ったものは半分以下であったが，それ以外の分野では LLM による判定を通った事例のうち約 7 割が人手の判定を通っている．LLM による妥当性判定を通ったが誤りを含む rationale の例を表 5 に示す．実際の『老人と海』の発表年は 1952 年であるが，本研究での LLM による妥当性判定では事実性の確認ができていない．

表 5　誤りを含む CoT データの例 (下線は誤り箇所)

Input（Ichikara-Instruction2 に収録）
ヘミングウェイが書いたのは「白鯨」と「老人と海」どちらだったかわからなくなりました。
Output（Ichikara-Instruction2 に収録）
「老人と海」です。
Rationale(GK)（本研究で生成）
質問に関係する知識をリストアップしてから回答を作成します。まず、質問に関係する知識を列挙します。
-『白鯨』(Moby-Dick) はアメリカの作家ハーマン・メルヴィルによる小説。
-『老人と海』(The Old Man and the Sea) はアメリカの作家アーネスト・ヘミングウェイによる短編小説で、1953 年に発表された。
- 両作品ともアメリカ文学の代表作として知られ、『白鯨』はハーマン・メルヴィルにより 1851 年に発表された。
- ヘミングウェイの作品は、アメリカ文学において「簡潔な文体」と「ストーリーテリング」が特徴的である。
回答を作成します。質問の内容から、ヘミングウェイが書いた作品を正しく選択しました。

4　評価

`LLM-jp-3-13b-instruct3` に対して，CoT データでの LoRA チューニング [10] を行った．思考過程の出力を介した LLM の利用事例 [11] に倣い，表 7 のように Rational を [思考] タグ，Output を [回答] タグに続く形で記載した学習データを作成した．5 つの CoT 種類を同時に学習し，推論時の指示を切り替えることで，異なる CoT を使い分けた推論を行う．

4.1　ホールドアウトデータでの評価

人手での妥当性判定を通過した 668 件のうち，5 種類の CoT に共通する問題 10 件を除外し，モデルの学習を行った．その後，未見の 10 件に対して rationale を生成し，指示通りの CoT 種類で推論するかを人手で評価したところ，全ての CoT 種類において，大まかな推論のスタイルは指示通りであった．しかし，一部の事例（P&S と SbS は 1 件，SF は 2 件）では，やや不自然な思考の展開が見られた．

4.2　QA ベンチマークでの評価

CoT データセットを学習したモデルの推論能力を評価するために，マルチホップ推論のベンチマークである JEMHopQA [12] を使用した．CoT データセット全体，あるいはその一部や作成過程に得られたデータを用いて，`LLM-jp-3-13b-instruct3` の LoRA チューニングを行い，JEMHopQA の開発データ 120 件に対して，5 つの CoT 指示を用いて推論を行った．最終的な推論結果が正答と合致しているかを `gpt-oss-20b` で評価し，表 6 に結果をまとめた．

本研究で取り組んだフィルタリングの有効性を確認するために，以下の 4 つのモデルを比較した．

Vanilla: CoT データセットの学習を行わないモデル．

+ All Data: 各問題ごとに各 CoT 種類一件ずつを乱択して作成したデータ．フィルタリングを行わない乱雑なデータを用いた学習の有効性を調べる．

+ GPT-Filtering: GPT が妥当と判断した rationale を，各 CoT 種類一件ずつ乱択して作成したデータ．GPT でのフィルタリングの有効性を調べる．

+ GPT&Human-Filtering: GPT と人間が妥当と判断した rationale を，各 CoT 種類一件ずつ収録したデータ．人手でのフィルタリングの有効性を調べる．

与えられた 2 つの要素を比較して回答する Comparison 設定では，多くのケースで人手フィルタリングによる性能の向上が見られた．**Vanilla** と**+All Data** を比較すると，LLM で生成した rationale をそのままに使用しても，性能の向上に寄与しないことがわかる．対して，**+GPT-Filtering**，**+GPT&Human-Filtering** とフィルタリングを重ねることで，多くのケースで性能の向上が見られる．フィルタリングによって学習に用いる件数が少なくなるにもかかわらず性能が向上することから，品質の高いデータセットを用意する必要性が示唆される．

正解となるテキストを知識から引き出して生成す

表 6　CoT データセットを用いて `llm-jp-3-13b-instruct3` の LoRA チューニングを行い，JEMHopQA の開発データ（120 件）で評価した結果．データセットのうち，同じ問題に対して異なる CoT 種類の Rationale が付与されたサンプルの重複を許して数えたものが「学習に用いた件数」，重複を許さずにユニークな問題数を数えたものが「問題数」である．

	学習に用いた		Comparison					Compositional				
設定名	件数	問題数	GK	L2M	P&S	SR	SbS	GK	L2M	P&S	SR	SbS
Vanilla	0	0	49.32	50.68	52.05	53.42	53.42	**27.66**	**29.79**	**27.66**	**27.66**	**29.79**
+ All Data	1223	378	49.32	53.42	50.68	53.42	52.05	19.15	10.64	14.89	19.15	23.40
+ GPT-Filtering	1115	360	50.68	60.27	50.68	**58.90**	54.79	12.77	17.02	25.53	19.15	17.02
+ GPT&Human-Filtering	668	271	**54.79**	**61.64**	**63.01**	53.42	**60.27**	21.28	27.66	20.79	23.40	**29.79**

表 7　JEMHopQA の開発データにおける，P&S での推論結果．問題の正答は「任天堂」である．

これから与えられる質問や問題について、先に問題を解く方針を考えてから、その方針に従って最終的な回答を作成してください。
川崎重工業と任天堂、本社が京都にあるのは？

+ All Data （フィルタリングなし）

[思考]
回答の正確性を上げるために、問題を解く方針を先に考えてから回答します。

1. 最初に、問題の条件を整理し、求められている情報を明確にします。
2. 次に、川崎重工業と任天堂という企業について、本社所在地の情報を調べます。
3. それぞれの企業の本社所在地を確認し、京都にあるかどうかを特定します。
4. 最後に、特定した情報をもとに、正しい回答を作成します。

方針に従って、回答を作成します。

1. 問題の条件を整理すると、以下の点が重要であることが分かります。
- 川崎重工業と任天堂という企業について、本社所在地を比較する。
- 京都に本社があるのはどちらかを特定する。

2. 各企業の本社所在地について調べます。
- 川崎重工業 → 兵庫県神戸市に本社があります。
- 任天堂 → 京都府京都市に本社があります。

3. それぞれの企業の本社所在地を確認し、京都にあるかどうかを特定します。
- 川崎重工業 → 神戸に本社があるため、京都ではありません。
- 任天堂 → 京都に本社があるため、京都です。

4. 以上の情報を基に、正しい回答を作成します。

[回答]
川崎重工業、任天堂はどちらも本社が京都にあります。</s>

+ GPT&Human-Filtering

[思考]
回答の正確性を上げるために、問題を解く方針を先に考えてから回答します。

1. 最初に、問題の条件を整理し、求められている情報を明確にします。
2. 次に、川崎重工業と任天堂という企業名と、それぞれの本社所在地に関する情報を調査します。
3. 調査した情報をもとに、本社が京都にあるのはどちらかを特定します。
4. 最後に、特定した情報をもとに、正しい回答を作成します。

方針に従って、回答を作成します。

1. 問題の条件を整理すると、以下の点が重要であることが分かります。
- 川崎重工業と任天堂という企業名がある。
- それぞれの企業の本社所在地を確認する必要がある。

2. 各企業の本社所在地について調査します。
- 川崎重工業 → 本社は兵庫県神戸市に所在。
- 任天堂 → 本社は京都府京都市に所在。

3. 調査結果から、本社が京都にあるのは「任天堂」であることが分かります。

4. 以上の情報を基に、正しい回答を作成します。

[回答]
「任天堂」です。</s>

る必要がある Compositoinal 設定では，生成された rationale（**+All Data**）の使用が性能の悪化に繋がった．これは，表 5 に示すように `LLM-jp-3-13b-instruct3` を用いて生成した rationale に，事実とは異なる情報が含まれており，学習済みの Vanilla モデルの知識を汚染したことが原因と考えられる．この問題は GPT を用いたフィルタリング（**+GPT-Filtering**）でも防ぐことはできず，人手でのフィルタリング（**+GPT&Human-Filtering**）によって，ある程度問題が緩和されることが結果として見て取れる．

表 7 に P&S での推論結果を示す．フィルタリングなしのデータで学習した場合の典型的な誤りとして，思考過程は正しいが最終的にそれを反映した回答を出力できないという事例を多く観察した．これは，rationale と回答に不一致があるデータを，そのまま学習したためであると考えられる．GPT や人手でのフィルタリングによって，このような学習事例が減り，正しい回答の導出に貢献している．

5　おわりに

本論文では，rationale の自動生成と LLM による妥当性判定を用いた CoT データセットの構築を行い，JEMHopQA を用いた評価実験を通してデータセットの品質とモデルの推論性能の関係の分析を行った．評価実験の結果，生成した rationale を学習に用いても性能向上は見られなかったが，フィルタリングを重ねることで向上が見られた．この結果はデータの量よりも品質が重要であることを示している．

rationale の自動生成と LLM による妥当性判定は人手作業の負担軽減を目的としたものである．しかし，LLM の妥当性判定では事実性の確認ができておらず，その確認のために外部資料を参照するという作業負担の軽減には至っていない．今後は LLM による自動生成や妥当性判定の改善や，他のモデルを用いた場合でのデータセットの品質の分析を行う予定である．

参考文献

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In **Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS2022)**, 2022.

[2] Leonardo Ranaldi and Andre Freitas. Aligning large and small language models via chain-of-thought reasoning. In Yvette Graham and Matthew Purver, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1812–1827, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[3] Seungone Kim, Se Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The CoT collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 12685–12708, Singapore, December 2023. Association for Computational Linguistics.

[4] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, **Advances in Neural Information Processing Systems**, 2022.

[5] Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In **The Eleventh International Conference on Learning Representations**, 2023.

[6] Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. ReCEval: Evaluating reasoning chains via correctness and informativeness. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 10066–10086, Singapore, December 2023. Association for Computational Linguistics.

[7] 株式会社いちから. https://ichikara.ai/.

[8] Llm-jp-3-13b-instruct3. https://huggingface.co/llm-jp/llm-jp-3-13b-instruct3.

[9] gpt-oss-20b. https://huggingface.co/openai/gpt-oss-20b.

[10] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In **International Conference on Learning Representations**.

[11] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In **Conference on Robot Learning**, pp. 1769–1782. PMLR, 2023.

[12] Ai Ishii, Naoya Inoue, Hisami Suzuki, and Satoshi Sekine. JEMHopQA: Dataset for Japanese explainable multi-hop question answering. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 9515–9525, Torino, Italia, May 2024. ELRA and ICCL.