修士論文

言語モデルの推論能力に関する研究

原口大地

主指導教員 白井清昭

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和６年３月

Abstract

Reasoning is a fundamental capability of human intelligence, which allows us to make inferences or predictions about unknown things based on known information. In the field of natural language processing (NLP), a subfield of artificial intelligence that deals with language, the reasoning ability of NLP models is a key factor directly linked to their generalization performance in machine learning contexts. Despite the significant progress in the language capabilities of NLP models since the advent of Transformer, numerous challenges still exist in their reasoning abilities. Shortcut reasoning refers to the irrational reasoning of NLP models, which is a critical issue for reasoning ability. Specifically, shortcut reasoning involves a type of reasoning that performs well with input that follow the same distribution as the training data, but perform poorly on input with a different distribution from the training data. Previous studies have shown that this irrational reasoning can degrade the robustness of NLP models.

Recently, Large Language Models (LLMs) have attracted much attention for their language capabilities that surpass previous NLP models. Improvements have also been observed in the reasoning performance of LLMs. Previous studies have reported that attaching specific prompts to the input or combining LLMs with external algorithms for reasoning improves the performance.

However, Hallucination, the generation of counterfactual knowledge by LLMs, is known as a significant challenge. Even if it is possible to construct the reasoning process correctly, the final answer may contain errors when the knowledge used in the process is incorrect. Previous work has shown that hallucinations can snowball, where the errors generated during reasoning successively affect downstream reasoning processes. Given these issues, various solutions are being studied. One of them is a method called Retrieval-Augmented Generation (RAG). RAG is a paradigm that retrieves documents from a knowledge base in response to an input and generates an answer while referring to the retrieved documents. Many research has shown that this approach reduces hallucination and improves performance. However, there is still room for improvement in the retrieval mechanism of RAG. A typical example is that the retrieved fixed number of documents may contain unnecessary information or may not contain the necessary information. Unnecessary information can introduce noise and amplify hallucination. In addition, there are still unresolved issues, such as requiring retrieval for each reasoning step.

LLM's reasoning ability itself still has problems. Previous studies have shown that LLMs are good at single-step reasoning, but inherently struggle with compositional reasoning problems that require synthesizing partial reasoning results, such as computation problems like 3-digit multiplication. It has been revealed that

the seemingly excellent reasoning of LLMs we observe is due to pattern matching using the reasoning processes included in the massive pre-training data.

In view of the aforementioned issues related to the reasoning of NLP models and LLMs, we aim to realize a reasoning system with self-awareness and interpretability. Self-awareness refers to the ability of the system to recognize what information it possesses and what it lacks. Interpretability, on the other hand, pertains to its ability to explain how input questions are broken down and which knowledge is consulted during the reasoning process. This system outputs a final prediction by decomposing the input question, querying a knowledge base, and synthesizing the obtained results. The proposed system comprises three major modules: a knowledge base module, a reasoning module, and a central control module. The knowledge base module employs a self-aware LLM that stores and retrieves relevant knowledge related to a given question. The reasoning module adopts a probabilistic logical programming, which decomposes the question into sub-questions and refers to the knowledge base to obtain necessary information. Finally, the central control module integrates the retrieved information to generate a final prediction of answer.

To achieve this goal, we have conducted research on reasoning ability of NLP models, followed by a discussion toward developing a self-aware knowledge base module. Specifically, we worked on the following three themes.

(i) In automatic discovery of shortcut reasoning with generality, we addressed the challenges of existing methods and proposed a method for automatically detecting shortcut reasoning. The previous methods to discover shortcut reasoning within NLP models have several issues, such as pre-definition of shortcut reasoning, not using internal states of the models, or requiring human evaluation. Even though the latest work overcame those problems, its method still suffer from several limitations. As a result of experiments with our proposed method on NLP models trained on sentiment analysis and natural language inference tasks, we succeeded in detecting shortcut reasoning without human intervention and discovering unknown shortcut reasoning not revealed in previous studies as well as known ones.

(ii) In our research on a logical rationale-based machine reading comprehension model, we conducted experiments on shortcut reasoning in Explainer for the machine reading task. Explainer is a module of Explain-then-predict architecture. Explain-then-predict is an architecture generally used for Rationalization, where models are forced to output their rationale along with the predictions. As the rationalization architecture has explainer, which extracts necessary information for the right inference, previous work hypothesize the architecture can improve robustness by excluding unnecessary noise in the input documents. However, the results

did not support their expectation. Thus, we hypothesized that the robustness improvement was not achieved since explainer was performing shortcut reasoning. The empirical experiment revealed explainer's shortcut reasoning by showing that accuracy did not drop even when the explainer's input was destructively corrupted.

(iii) In our discussion toward large language models as knowledge bases with self-awareness, we approached the relationship among confidence, accuracy and frequency in terms of LM as KB, which utilize language models as knowledge bases. In previous work, confidence, the probability that the prediction is correct, in prediction by LLMs is known to be well-calibrated. Correlations between (calibrated) confidence and accuracy, and between frequency and accuracy are revealed in several studies, but not for the correlation between frequency and confidence. Therefore, we analyzed the relation between frequency and confidence, and conducted preliminary experiments to verify our hypothesis that the confidence of LLM's knowledge reflects the frequency of its appearance in pre-training rather than its correctness. We employed PopQA, which contains triplets of knowledge (subject, relation, object) retrieved from Wikipedia and each of which is annotated with its popularity as a proxy of frequency the knowledge appeared in pre-training data. We used several LLMs for the experiments, and when using GPT-3.5-turbo, we obtained results showing that confidence and accuracy were positively correlated for knowledge with high frequency but that accuracy did not increase even when confidence increased for knowledge with low frequency. Although these results were intriguing, they did not lead to the verification of our hypothesis.

概要

推論とは未知の事柄について自らの知る情報から予想・論じることであり，人間の知的機能の基盤をなす能力である．人工知能における言語に関する研究分野である自然言語処理において，言語処理モデルの推論能力は機械学習の文脈における汎化性能に直結する重要な要素である．Transformer の登場以来，言語処理モデルの言語能力は飛躍的に向上したものの，未だ推論能力に関しては多くの課題が存在している．Shortcut reasoning は，言語処理モデルの非合理的な推論のことである．具体的には，学習データと同じ分布を持つ入力に対しては有効であるが，学習データと異なる分布を持つ入力に対して有効ではない推論のことである．この非合理的な推論により，言語処理モデルの頑健性が低下することが先行研究で明らかになっている．

昨今大きな話題となっている大規模言語モデル (LLM) は，既存の言語処理モデルを凌駕する言語能力をみせている．LLM は推論においても性能の向上が観察されている．入力に際して特定のプロンプトを与えると性能が大きく向上したという結果や，外部のアルゴリズムと組み合わせて LLM に推論させる手法なども提案され，LLM に推論を行わせることに関心が寄せられている．

しかしながら，Hallucination と呼ばれる LLM の生成する反事実的な知識が大きな課題として知られている．仮に推論プロセスを正しく構築することが可能であっても，その過程で用いられる知識に誤りがあれば最終的に出力される解答に間違いが含まれる可能性がある．先行研究では，推論に際して発生した hallucination が下流の推論プロセスに次々に影響を与え，雪だるま式にエラーが増大していくことが明らかになっている．こうした問題を受け，様々な解決策が研究されている．その一つが RAG と呼ばれる手法である．RAG は入力に対して知識ベースから取得し (Retrieval)，取得した文書を参照しながら解答を生成するパラダイムのことである．これにより，hallucination が低減され，性能が向上したという実験結果が多数の研究により示されている．しかし，RAG の retrieval 機構にもいくつかの改善の余地が存在する．その代表例が，取得した k 個の文書群の中に不必要な文書が含まれている，もしくは必要な文書が含まれていないことがある点である．不必要な文書がノイズとなり，Hallucination を増長させてしまう可能性がある．また，推論のたびに retrieval が必要である点など，解決すべき課題は未だ残っている．

LLM の推論能力自体についても，依然として問題点が見受けられる．先行研究によると，LLM はその構造上，単一ステップの推論には成功するものの，部分的な推論結果を合成しながら解く必要があるような推論問題，例えば 3 桁 $\times$3 桁のような計算問題を解くことが不得意であることが示されており，我々が観察できる LLM の優れた推論は大規模な事前学習データ内に含まれる推論プロセスを使ったパターンマッチングにより行われていることが明らかになっている．

以上のような，言語処理モデル，あるいは LLM の推論に関する諸問題を受け，我々は自己認識能力を持つ，解釈可能な推論システムの実現を目指す．自己認識

能力とは，何を知っていて，何を知らないのかを説明できることである．また，解釈可能とは，どのように質問を分解して，どの知識を参照したのかを説明することが可能なことである．このシステムは，入力された質問の分解と，知識ベースへの問い合わせ，そして得られた結果の合成によって，最終的な予測を出力する．知識ベースモジュール，推論モジュール，中央制御モジュールの 3 つのモジュールから構成され，それぞれが質問に関する知識の取得，質問の分解，そして，解答の統合・出力を行う．

この実現にあたり，言語処理モデルの推論能力に関する分析・考察を経た後，自己認識可能な知識ベースモジュールの実現に向けた研究を行った．具体的には，以下の 3 つのテーマに取り組んだ．

(i) 一般性を考慮した Shortcut reasoning の自動検出では，先行研究で提案されていた手法の課題を解決し，shortcut reasoning を自動的に検出する手法を提案した．先行研究で提案されている検出手法では，shortcut reasoning の形態を事前に定義している，内部情報を用いていない，人手による評価を必要としている等の課題が存在していた．最新の研究の提案手法によってそれらの課題は克服されたものの，依然としていくつかの制約を抱えている．感情分析タスクと自然言語推論タスクの分類問題で学習された言語処理モデルを対象とした実験を行い，人間の介入無しで shortcut reasoning を検出することに加え，先行研究で明らかになっていなかった未知の shortcut reasoning を発見することに成功した．

(ii) 論理的根拠に基づく機械読解システムに向けた研究では，機械読解タスクでの Explainer における shortcut reasoning について実験を行った．Explain-then-predict は予測とその論理的根拠を出力させるパラダイムである Rationalization において頻繁に使用されるアーキテクチャであり，explainer はその一部を構成するモジュールである．Explainer は適切な推論のために入力中の必要な情報のみを抽出する，すなわち推論に不必要なノイズを低減してくれる効果が期待できることから，ある先行研究はこのアーキテクチャは頑健性を向上できるという仮説を立て，検証したものの，頑健性の向上は限定的であった．この結果を踏まえ，我々は explainer が shortcut reasoning を行っているため，頑健性の向上が実現できなかったという仮説を立てた．経験的な実験の結果，explainer の入力に破壊的な編集を加えても精度が落ちなかったことから，モデル内の shortcut reaosning が示唆された．

(iii) 自己認識が可能な知識ベースとしての大規模言語モデルの実現に向けた議論では，LLM の知識について，予測の自信度，その正解率，その知識の事前学習データにおける頻度の 3 つの要素の関係性について分析を行った．先行研究では，既存の手法で計算された LLM の予測の自信度は，正解率を良く反映していることが明らかになっている．また，自信度と正解率，正解率と事前学習における頻度の相関関係はいくつかの研究により明らかになっているが，自信度と頻度の関係性については我々の知る限り未知である．そこで，我々は LLM の知識の自信度が，正解率ではなく，知識が事前学習に現れる頻度を反映しているという仮説を立て，

予備的な検証実験を行った．実験に用いた PopQA データセットは，Wikipedia から収集した知識と，その知識の Wikipedia ページにおける月間閲覧数を頻度として近似した人気度と呼ばれる指標が付与されている．実験の結果，GPT-3.5-turbo を用いたとき，頻度が大きい知識に関して自信度と正解率が正相関するが，頻度の低い知識に対しては自信度が上がっても正解率が上がらなかったという結果を得た．この結果は興味深いものであるものの，仮説の実証には至ることができなかった．

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

推論 (Reasoning) とは，既知の情報から未知の事柄について予想し，論じることである．自然言語処理 (NLP) において，言語処理モデルの推論能力は未知の入力に対して効果的に正しい解答を出力すること，すなわち汎化性能に直結する重要な要素である．事前学習済み Transformer [36] をベースとしたモデル (BERT [5], GPT [2]，T5 [29]) は，多様なタスクで大きく精度を向上させたものの，推論能力に関して未だ懸念すべき課題が存在している．**Shortcut reasoning** はその一つであり，推論モデルによる短絡的かつ非合理的な推論を指す．Shortcut reasoning は学習データと同じ分布を持つデータ (Independent and Identically Distributed: IID) と，異なる分布を持つデータ (Out of Distribution: OOD) の解析性能の乖離をもたらす [8]．推論は未知の事象の予測に対して使われるものであるが，shortcut reasoning はその効果を得られないことが懸念される．例えば，感情分類タスクにおいて，Positive とラベル付けされた *Spielberg* という単語を含む文 (例：*Spielberg is a great director!*) が数多くあるデータセットからモデルを訓練すると，モデルは *Spielberg* という単語を含むどんな入力文に対しても Positive を予測するように学習してしまう．この場合，映画レビュー以外の OOD の入力 (例：ニュース記事) に対しては必ずしも *Spielberg* を含む文の感情は Positive ではないため，例のような推論は有効とは言えない．この現象は，機械学習の文脈では頑健性を低下させる要因として挙げられ，その検出・軽減が議論されている．

大規模言語モデル (Large Language Model: LLM) は極めて大規模な事前学習を行った言語モデルであり，推論能力を含めた極めて自然な言語能力や自然言語による簡単な命令のみから達成される高いタスク処理能力を背景に，世間から大きな注目を浴びている．Chain of Thought (CoT) は，LLM に対し，*Let's think step by step* のように段階的な推論を明示的に出力させるようにプロンプトで指示した上で問題を解かせる手法であり，単に問題を入力したときの精度と比較して大きく向上することが分かっている [40]．また，Tree of Thought (ToT) はある推論ステップにおいて LLM にいくつかの出力を出させ，最終的に最良の推論パスが得られるように木探索アルゴリズムを応用した手法であり，CoT 以上の精度を達成した [42]．以上のようなアイデアでは，LLM が外部情報無しに事前学習で獲得したパラメータ内部に持つ知識をもとに推論問題を解くことができる能力を持つことを示しているといえる．

一方で，Hallucination と呼ばれる LLM の事実と反する内容の出力も大きな問題となっている．前述の通り，LLM の出力は非常に自然であり，人間がその出力だけを見て人間が書いたものであるのか，LLM が生成したものであるかどうかを判定するのも極めて困難であることはよく知られている．**Retrieval Augmented Generation (RAG)** は Hallucination の低減に対する有力なアプローチの一つである．RAG はモデルに入力されたクエリについて，それに関連する文書等を取得 (Retrieval) し，それを参照させて解答を生成させる手法のことである．実際に先行研究では，RAG が反事実的な出力を低減させる効果が得られることが分かっている．こうした利点から，RAG は機械読解システム等の推論タスクにおいてしばしば応用されている．具体的には，対象とするドメインや Web ページ上の文書から文書集合を作成し，与えられたクエリと文書群との類似度を計算し，スコア上位 k 個の文書をクエリと併せて読解モデルに入力する，といった手法を採ることが一般的である．しかしながら，現在の RAG はその Retrieval 機構に欠点を抱えている．前述の通り，RAG では多くの場合，推論時に定数個の文書を取得するが，(i) 推論のたびに retrieval をすることと，(ii) 取得した k 個の文書群の中に不必要な文書が含まれている，もしくは必要な文書が含まれていないことがある，といった課題が存在する．課題 (i) では，実行時の時間的コストが要求されることに加え，LLM を応用する場合に内部知識を最大限活用できない．課題 (ii) では，不必要な文書がノイズとなり，Hallucination を増長させてしまう可能性がある．また，必要な文書が取得できなかった際にも，パラメータに埋め込まれた知識から回答をする必要があるため，反事実を生成させてしまう懸念がある．

LLM の推論能力自体においても，その存在について懐疑的な意見が少なくない．その一つが，**Compositional reasoning**，すなわち部分的な推論結果を合成しながら解く必要があるような推論問題 (例：多段推論) を解く能力である．このような推論問題において，GPT-4 を始めとした LLM は多段推論における初期の単一ステップの推論には成功するものの，それらを合成して最終的な解答を導くための能力が低いことが明らかになっている [7]．また，モデルが事前学習で覚えた推論プロセスをそのままパターンマッチングを使って解いていることが想定されており，それにより学習データと類似した推論問題に対しては高い精度を示すものの，そうでない未知の入力に対しては精度が低下するということが分かっている [7]．この現象は，まさしく shortcut reasoning と合致する．

1.2 目的

前節で述べた課題を考慮し，我々は自己認識能力を持つ，解釈可能な推論システムの実現を目指す．具体的には，自己認識能力とは，何を知っていて，何を知らないのかを説明できることである．また，解釈可能とは，どのように質問を分解して，どの知識を参照したのかを説明することが可能なことである．このシステムは，入力された質問の分解と，知識ベースへの問い合わせ，そして得られた結

果の合成によって，最終的な予測を出力する．この手法を実現するに当たり，以下の 3 つのモジュールが実現に必要と考えられる．(i) 知識ベースモジュールは自己認識が可能な知識ベースである．同義なクエリに柔軟に対応できる．入力されたクエリに対する解答には確信度を付与し，知識ベースモジュールが出力した知識をどれだけ確信しているかを明示的に表す．(ii) 推論モジュールは質問の分解を行うモジュールである．知識ベースモジュールから得られた確信度が低い場合，質問を分解して解く必要がある．その処理を行うのが，このモジュールの役割である．(iii) 中央制御モジュールは入力された質問の受け取りと，他の 2 つのモジュールの制御，そして最終的な解答の統合・出力を行う．具体的には，2 つのモジュールの情報のやり取りを媒介し，解答出力時にはそれらの情報を集約し，入力された質問に対する解答を作成する．

本研究では，上記の推論システムの実現に先立ち，以下の 3 つテーマに取り組んだ．

1. Shortcut reasoning の自動検出
2. 論理的根拠に基づく機械読解システムに向けた分析
3. 自己認識が可能な知識ベースとしての大規模言語モデルの実現に向けた議論

1. Shortcut reasoning の自動検出では，先行研究で提案されていた手法にアプローチし，shortcut reasoning を自動的に検出する手法を提案した．感情分析タスクと自然言語推論タスクの分類問題において，提案手法は人間の介入無しで shortcut reasoning を検出することに加え，先行研究で明らかになっていなかった未知の shortcut reasoning を発見することに成功した．

2. 論理的根拠に基づく機械読解システムに向けた研究では，shortcut reasoning について，機械読解タスクでの実験を行った．具体的には，Explainer と呼ばれるモジュールを読解モデルの上流においた Explain-then-predict 型の機械読解システムの非合理的推論を分析した．機械読解における explainer は，入力された質問とドキュメントに基づいて，質問の解答に必要な情報を文書から抽出し，下流の読解モデル (Predictor) に質問と抽出された情報を渡すモジュールである．先行研究に基づいて立てた，既存の explain-then-predict 型の機械読解システムの explainer はパターンマッチング等で質問を参照して文書から情報を抽出するのみで，人間のような推論プロセスを行っていないという仮説について分析を行った．実験の結果，質問や文書の並びをランダムに混ぜ，意味が通らない入力を与えても精度が落ちなかったことから，explainer の shortcut reaosning を経験的に明らかにした．

3. 自己認識が可能な知識ベースとしての大規模言語モデルの実現に向けた議論では，我々が目標とする解釈可能な推論システムにおいて，自己認識可能な知識ベースの構築に向けて予備的な実験を行い，その結果を考察した．LLM の知識の自信度は，その正解率ではなく，その知識が事前学習に現れる頻度を反映してい

るという仮説に対し，その証拠ともとれる結果を得たものの，仮定の実証には至ることができなかった．

1.3 本論文の構成

本論文の構成は以下の通りである．2 章では，本研究の関連研究について述べる．3 章では，提案する shortcut reasoning の自動検出手法について解説する．4 章では，機械読解モデルにおける shortcut reasoning について分析を行う．5 章では，自己認識可能な LM as KB に向けた議論をする．

第2章　関連研究

2.1　言語モデル

本研究では，言語モデルの推論能力について考察する．ここでは言語モデルの定義や代表例を紹介する．

言語モデル (Language model: LM) とは，狭義には与えられた自然言語文に対して自己再帰的に次の単語を生成する (次単語予測，Next Token Prediction: NTP) モデルのことである．コーパス上の単語と品詞の出現頻度から予測確率を計算する統計的言語モデルを起源として，現在ではニューラルネットワークを応用したニューラルベース言語モデルが主流である．特に，GPT [2] に代表される Transformer [36] のデコーダー部分を用いて事前学習された言語モデル (Pretrained Language Model, PLM) は卓越した言語能力を実現している．

一方で，言語モデルは広義には自然言語処理を行う機械学習モデル一般を指すこともある．代表的な例としては，Transformer のエンコーダー部分を応用した BERT [5] である．BERT は GPT 等と同様に事前学習されたモデルであるが，事前学習では次単語予測ではなく，単語の穴埋め (Masked Language Modeling: MLM) と次文予測 (NextSentence Prediction: NSP) であるため，厳密には狭義の言語モデルとは異なったアーキテクチャである．したがって，本論文では GPT 等の狭義のニューラル言語モデルを「言語モデル」や「LM」，BERT 等の広義の言語モデルを「言語処理モデル」と表現する．

2.2　Shortcut reasoning

1 章で述べたように，言語処理モデルの推論能力について，その非合理性が指摘されている．本節ではこれに関する先行研究について述べる．

2.2.1　データセット内の疑似相関と非合理的推論

Spurious feature は，学習データセット内の特徴量とラベルの疑似相関であり，数多くの先行研究が自然言語処理タスク用のデータセットに spurious feature が含まれていることを明らかにしている．Poliak らは自然言語推論 (Natural Language Inference: NLI) タスクの評価用データセットである MNLI において，2 つの入力

premise と hypothesis のうち，hypothesis のみから解答が可能であることを示唆している [28]．Gururangan らは，NLI データセットにおいて入力データ中の否定表現と contradiction ラベルの疑似相関を指摘している [9]．Mccoy らは NLI における語の重複など，様々なヒューリスティックな特長からモデルが正しく解答できてしまうことを明らかにした [22]．Geirhos らは機械学習モデルにおいて spurious feature を学習してしまうことより発生する，IID の入力に対しては有効であるが，OOD に対してはそうでないような現象を Shortcut learning と定義した [8]．この現象は shortcut learning 以外にもいくつかの名称で呼ばれているが，我々の分析の対象は自然言語処理における推論 (reasoning) であることが理解しやすいよう，本研究では Shortcut reasoning と呼ぶ．

こういったモデルの振る舞いは頑健性を低下させる要因として，より詳細なデータセットの分析や改善に向けた議論がなされている [31, 39, 11]．

2.2.2 Shortcut reasoning の検出

以上のような問題を受け，shortcut reasoning の検出を試みた研究が複数存在する．Rebeiro らは，CheckList と呼ばれる言語処理モデルの言語能力を評価する包括的なマニュアルを提案した [30]．Han らは，学習データのどの事例があるテストデータの予測に影響しているかを示す手法である Influence function [19] を応用し，推論プロセスの解釈とデータセット内の交絡を発見することを試みた．Pezeshkpour らは，Influence function と，あるテストデータの入力の各単語の予測への貢献度を示す手法を組み合わせて，shortcut reasoning を発見する手法を提案した [27]．

以上のどの手法も，手動による評価が必要なことや，言語モデルの内部情報を用いた検出を行っていないことなど，課題が残っている．Wang らは，自動的に shortcut reasoning を検出する手法を提案したが，依然として制約を抱えている (詳細は後述する) [38]．

2.3 機械読解システム

機械読解タスクは質問と文書から解答を予測する NLP のタスクである．このタスクを解く機械読解モデルは複雑な推論能力を応用することが必要とされる．このことを踏まえ，本研究では機械読解モデルの推論能力を検証する．本節ではこれに関連する先行研究について述べる．

2.3.1 機械読解モデルの頑健性

Jia らは機械読解システムが，文書中の紛らわしい情報につられて誤った解答をしてしまうことを示し，頑健性の課題があることを主張した [13]．Sen らは，モ

デルが機械読解タスク用のデータセットから何を学習しているかを分析した結果，質問や文書を部分的にしか読解していないことや，特定の表現や固有表現から解答を予測していることが明らかになり，実験に用いたデータセットすべてで頑健性の懸念を抱えていることを示した [32].

2.3.2 Rationalization と頑健性

Rationalization は，論理的根拠を解答とともに出力させるパラダイムであり，タスクの精度とモデルの説明可能性の向上が期待できる手法である．Explain-then-Predict はその具体的なアーキテクチャの一つである．このアーキテクチャは説明を作成する Explainer とその説明から解答を予測する Predictor の 2 つのモジュールから成る．

Paranjape らは，Information bottleneck [35] を利用した explain-then-predict 型の機械読解モデルを提案し，精度と説明可能性の向上に取り組んだ [25]．Inoue らは，既存の文書中の文や単語を抽出して説明を生成する Extractive explainer ではなく説明文を新たに生成する Abstractive explainer を導入した explain-then-predict 機械読解モデルを提案した [12]．Chen らは rationalization が頑健性を向上させる効果を持つという仮説に対して，rationalization が可能なモデルを用いた検証に取り組んだが，限定的な向上にとどまった．

2.4 LLM

Large Language Model(LLM) は推論能力を大きく向上させた．しかしその一方で，その能力には議論の余地がある．ここでは LLM における推論能力に関する過去の議論を概観する．

2.4.1 LLM と推論

推論能力の向上

Chain-of-Thought(CoT) は，LLM に特定のプロンプトと共に質問を入力することで推論能力を向上させる手法である [40]．それに対し，Tree-of-Thought(ToT) は木構造を用いて最適な推論パスを選択しながら推論問題を解く手法である [42]．CoT では，推論パスも各推論ステップで要求される知識もすべて LLM により制御されている一方，ToT において LLM は各推論ステップにおける簡単な推論や知識の取得しか行わず，推論プロセスの決定自体は木構造とその探索により決定される．

LLM の推論能力の課題

Zhang らは，推論プロセスの初期段階で発生したエラーが後段の推論に伝播し，最終的な解答性能を下げることを明らかにした [44]．Dziri らは LLM の推論能力について，複数ステップに及ぶ多段推論をする能力は低いことを主張し，推論プロセスの決定において事前学習データに現れたものからパターンマッチしていることを示した [7].

2.4.2 自己認識可能な LM as KB

以上の課題を踏まえ，我々は自己認識可能かつ説明可能な推論システムの構築を目指す．その一部を構成するのが，自己認識可能な知識ベースとしての言語モデル (LM as KB) である．

LM as KB

Petroni らは，言語モデルが知識に関するクエリに対して対応する正しい知識を出力することが可能であること示し，LM as KB というパラダイムが注目されるきっかけとなった [26]．その一方で，Cao らは言語モデルがクエリの一部のみから知識を予測しており，異なることを問い合わせたクエリのデータセットに対して，同様の解答を出力する傾向を明らかにした [3].

言語モデルの自己認識可能性

Yin らは，LLM の自己認識について，認識している既知 (Known known)，認識している未知 (Known unknown)，認識していない既知 (Unknown known)，認識していない未知 (Unknown unknown) の 4 つに分類した上で分析を行った [43]. Kadavath らは，様々な設定で LLM が自己認識能力を持っているかを分析し，自己評価や自信度のキャリブレーションにおける LLM の一定の性能を明らかにした [16].

言語モデルの自信度

言語モデルの生成結果の信頼性の推定に関する研究は，近年盛んに行われている．各トークンの確率分布を正規化する手法 [23] や，エントロピーを計算する手法 [37] の他，LLM にプロンプトで直接自信度を出力するように指示する手法 (Verbal confidence) [34] 等，様々な手法が提案されている．

第3章　Shortcut reasoningの自動検出

近年，事前学習モデルをはじめとした自然言語処理モデルがあらゆるタスクで精度の向上を見せている．一方で，学習データ内の交絡あるいは疑似相関 (Spurious feature) [9, 28, 22] にモデルが依存することで発生する，推論プロセスにおける非合理的な推論 (**Shortcut reasoning**) が指摘されている [31, 39, 11]．

Shortcut reasoning を発見・解消させようとする試みは既に行われているが [30, 27, 10]，そこで提案された手法は，(i) shortcut reasoning の形態について事前に想定している，(ii) モデルの内部情報を考慮していない，(iii) 人手による判定を必要としている という制約を持っている．(i) は事前に想定した shortcut reasoning に対してのみ検証をするため，モデルに潜在する未知のものを明らかにできない可能性がある．(ii) については，内部情報を使わない手法の多くが何らかの編集や特徴を加えた入力を作成し，それに対する出力を分析することで，shortcut reasoning の存在を明らかにしようとしているが，この手法によって我々が知り得るのはモデルの出力のみであり，得られる情報には限度がある．(iii) に関しては，単純にコストがかかるのに加え，一見 shortcut reasoning に見えないような事例を見逃す可能性がある等の課題を抱えている．

Wang らは shortcut reasoning の事前の定義なし，かつ内部情報を使った自動検出の手法を提案しているが，依然としていくつかの課題を抱えている [38]．まず，彼らのフレームワークは発見された shortcut reasoning が OOD の入力に対してどの程度脆弱であるかについての評価に欠いている，モデルの頑健性にほとんど悪影響を与えない限り，検出された shortcut reasoning を懸念する必要性はないと言える．第二に，彼らの手法では異なるデータセットに渡って予測に有用とされるトークン (*genuine tokens* - 例：*good*，*bad*) は shortcut reasoning の原因にはならないとして，考慮していない点である．一見合理的と思えるものの，Joshi らはそのようなトークンは spurious feature でも多くの割合を占めていることを主張している [15]．その理由として，genuine tokens は確かにラベルの予測に必要なものであるが，予測の決定に十分なものではないからある．例えば，*This movie is not good*という文において，*good* という単語の情報は適切な予測に必要であるものの，*not* がそれを否定しているため，*good* 単体だけでは正しい結果を導けないことがわかる．したがって，genuine tokens は shortcut reasoning の検出には無視できないものであるということができる．

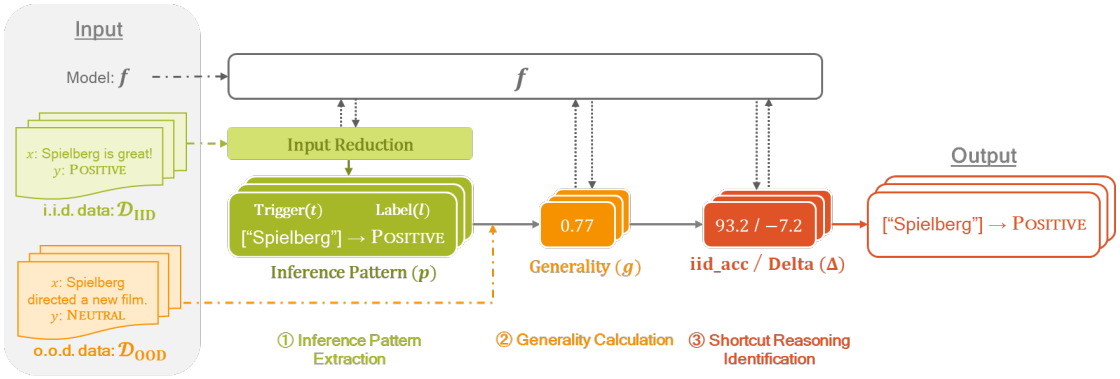

図 3.1: Shortcut reasoning 検出の手順

そこで，我々は以上のような諸問題を解決した手法を提案する．提案手法の具体的な貢献は以下の通りである．

- Shortcut reasoning の自動検出手法を提案した．
- より客観的な shortcut reasoning の基準 [8] を用いることで，検出に人手による判定や評価を必要としない．
- 我々の手法は OOD データに基づいて shortcut reasoning の深刻度を測定し，また shortcut reasoning を引き起こす因子の形態について仮定を一切置かない．
- 提案手法は先行研究で明らかになっているものに加えて，未知の shortcut reasoning を発見することに成功した．

3.1 Shortcut reasoning の検出

図 3.1 は提案手法の手順の概要である．提案手法は，言語処理モデル f が与えられたとき，IID データ $\mathcal{D}_{\text{IID}}$ と OOD データ $\mathcal{D}_{\text{IID}}$ を入力として，shortcut reasoning を出力として抽出する．そのプロセスは以下の 3 つの手順から構成される．

まず，手順 1 は特定のモデルの推論プロセスを特徴づける抽象的な表現である**推論パターン**を抽出する (§3.1.1)．推論パターンを抽出するために，自動的に推論パターンを導き出すアルゴリズムである **Input reduction** を使用する (§3.1.2)．

次に，手順 2 では手順 1 で抽出された推論パターンの一般性 (**Generality**) を計算する．Generality は推論パターンの強さを表す尺度であり，そのパターンとしての規則性を示す (§3.1.3)．

最後に，手順 3 では shortcut reasoning の判定を行う．手順 2 で得られた generality の情報に加えて，$\mathcal{D}_{\text{IID}}$ と $\mathcal{D}_{\text{OOD}}$ それぞれに対する推論パターンの有効性を shortcut reasoning の深刻度の指標とすることで，人間の介入なしに，ある推論パターンが shortcut reasoning であるかどうかを自動的に判定する (§3.1.4)．

3.1.1 推論パターン

本研究では，ある入力に対するモデル f の推論プロセスにおいて，何らかのトリガーが特定のラベルの予測をもたらす規則を推論パターンと定義する．推論パターン p は形式的に次のように定義できる．

$$p \stackrel{\text{def}}{=} t \xrightarrow{f} l \tag{3.1}$$

t はトリガー，l はトリガーによりもたらされたラベルを示す．

本章の冒頭で述べた通り，shortcut reasoning は学習データの spurious feature によりもたらされるが，それらが推論パターンにおけるトリガーとラベルであることは，その推論パターンが shortcut reasoning であることの必要条件といえる．したがって，推論パターンの定義において，spurious feature の特徴を十分に考慮する必要がある．

Pezeshkpour らは，spurious feature には *Granular feature* (語彙的素性) と *Abstract feature* (抽象的素性) の 2 種類があると分類している [27]．前者は，“Spielberg” のような予測と無関係な個別の単語である．後者は，単語の重複 (Lexical overlap) のように表層的に現れない高次なパターンのことを指す．

本論文では，shortcut reasoning の検出において，語彙的素性に注目し，抽象的素性に関しては今後の課題とする．したがって，推論パターンは以下のように再定義される．

$$p \stackrel{\text{def}}{=} \mathbf{w} \xrightarrow{f} l, \tag{3.2}$$

$\mathbf{w}$ は単語の系列 $[w_1, w_2, \cdots, w_n]$ を表す．

本研究では，語彙的素性のみを扱うものの，複数単語の組み合わせ等の多様な形態の推論パターンに対応することが可能である．例えば，感情分類モデルにおいて，[“not”, “bad”] $\rightarrow$ NEUTRAL や [“Spielberg”] $\rightarrow$ POSITIVE (shortcut reasoning) といったようなパターンが考えられる．

3.1.2 推論パターンの候補の抽出

推論パターンを抽出する手法として，新たに Input Reduction (IR) を導入する．対象モデル f と IID のデータセット $\mathcal{D}_{\text{IID}} = \{(x_i, y_i)\}_{i=1}^{N}$ が与えられたとき，各 x_i に対し IR を適用し，推論パターンの候補 C を抽出する．IR では，式 (3.2) におけるトリガー $\mathbf{w}$ について，ラベルの予測に必要となる最低限の単語の系列と考える．また，Label はそれを入力としたときの出力ラベルとする．これは，トリガーがある予測ラベルを引き起こすため**だけ**に用いられた情報であることを保証するためである．

IR の概要を図 3.2 に示す．大きな流れは以下の通りである．あるデータセット $\mathcal{D}$ を入力として受け取った後，データセットの各インスタンス (x, y) について，推論パターンの候補 $w = (\mathbf{w}, l)$ を抽出し，その集合 C を出力する．候補の抽出では，

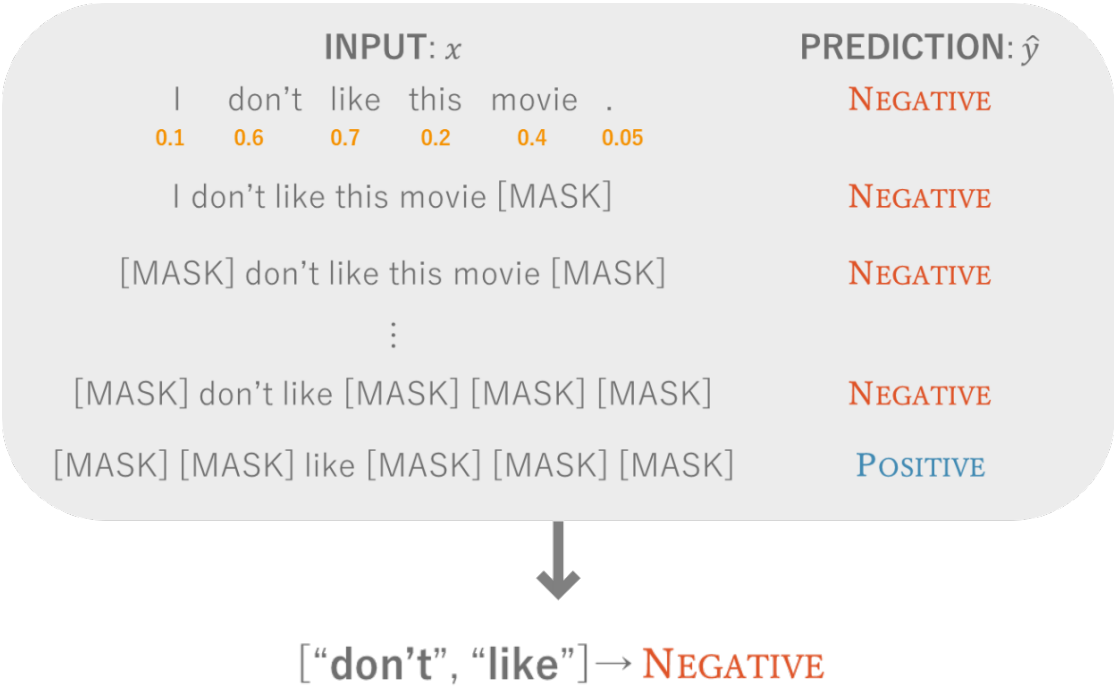

図 3.2: Input Reduction の概要

入力系列 x が与えられたとき，x に含まれる各単語に対して一つずつマスクをかける ([MASK] に置換する)．マスクの数を増やしていき，それを入力をしたときの予測が変化するまで繰り返す．予測が変われば，その直前の系列をトリガーとする推論パターンの候補を得る．

しかしながら，ナイーブな実装ではマスクの組み合わせが膨大になり計算量が負担となる．したがって，Integrated Gradient (IG) [33] を応用する．IG は予測の各特徴量の貢献度を計算する手法であり，値が高いほど予測に大きく影響を与えた特徴量であることを表す．IG により計算したスコアに基づき，重要度の低い単語順にマスクをかける．

Input Reduction の擬似コードを Algorithm 1 に示す．

3.1.3 一般性の計算

推論パターンを，推論における「パターン」と呼ぶためには，一定の規則性が認められなければならない．そこで，推論パターンの一般性を計算する．

C の i 番目の推論パターン候補 $p_i = (\mathbf{w}_i, l_i)$ の一般性 g_i を次のように計算する．トリガーの単語の系列 $\mathbf{w}_i$ を含む事例の集合 $(x'_j, y'_j) \in E_{\mathrm{OOD}}(\mathbf{w}_i)$ を $\mathcal{D}_{\mathrm{OOD}}$ から取得する．例えば，推論パターン $p_i =$ ["Spielberg"] $\rightarrow$ POSITIVE においては，$E_{\mathrm{OOD}}(\mathbf{w}_i)$ は *I grew up with Steven Spielberg's films. His films are always great!!* や *Spielberg*

Algorithm 1 Pseudo-code of Input reduction

function INPUT_REDUCTION_IG($\mathcal{D}$)
  for all $(x, y) \in \mathcal{D}$ **do**
    $\hat{y} \leftarrow f(x)$; $x' \leftarrow x$; $\hat{y}' \leftarrow \hat{y}$
    while $\hat{y} = \hat{y}'$ **do**
      $x'_{prev} \leftarrow x'$; $\hat{y}'_{prev} \leftarrow f(x'_{prev})$
      $x' \leftarrow$ IG_mask(x')
      $\hat{y}' \leftarrow f(x')$
      if all tokens in x' are mask **then**
        break
      end if
    end while
    $C \leftarrow C \cup \{p = (x'_{prev}, \hat{y}'_{prev})\}$
  end for
  return C
end function

is overrated. といったような文章を含むと考えられる．取得した $E_{\mathrm{OOD}}(\mathbf{w}_i)$ を基に，推論パターン p_i の一般性 g は以下のように計算される[1]:

$$g(p_i) \stackrel{\text{def}}{=} \frac{\sum_{x' \in E_{\mathrm{OOD}}(\mathbf{w}_i)} \mathbb{1}\left[f(x') = l_i\right]}{|E_{\mathrm{OOD}}(\mathbf{w}_i)|} \times 100. \tag{3.3}$$

この式 (3.3) は，トリガーを含む OOD の全データに対するモデルの予測のうち，どれだけの数が l と同じであるかの割合を示したものである．この割合大きいほど，モデルが OOD の入力に対しても IID から抽出された推論パターンを適用していることを表し，推論パターンの一般性が確かであることになる．

3.1.4 Shortcut reasoning の判定

Geirhos によると shortcut reasoning は以下の条件：(i) IID の入力に対しては有効に機能するが，(ii) OOD の入力に対しては有効ではないこと の**両方**を満たす [8]. これらの条件を抽出した推論パターンの候補に適用する．

IR を用いて $\mathcal{D}_{\mathrm{IID}}$ より抽出された推論パターン $p_i = \mathbf{w}_i \rightarrow l_i$ について，条件 (i) は，p_i が $\mathcal{D}_{\mathrm{IID}}$ で有効であることがその条件を満たすことになる．つまり，モデルが $\mathbf{w}_i$ を含むような IID データに対してうまく予測できるときである．したがって，$E_{\mathrm{IID}}(\mathbf{w}_i)$ における各推論パターン候補の性能を評価する．具体的には，以下のように定義される評価指標 $\mathrm{iid\_acc}_i$ を用いる．

[1] $\mathbb{1}[a = b]$ は，$a = b$ のとき 1，$a \neq b$ のとき 0 を返す関数.

$$\text{iid\_acc}_i \stackrel{\text{def}}{=} \frac{\sum_{x \in E_{\text{IID}}(\mathbf{w}_i)} \mathbb{1}\left[f(x) = l_i \wedge l_i = y_i\right]}{\sum_{x \in E_{\text{IID}}(\mathbf{w}_i)} \mathbb{1}\left[f(x) = l_i\right]} \times 100. \tag{3.4}$$

この評価指標はトリガー ($\mathbf{w}_i$) が含まれる入力に対してどれだけ正しく予測できているかを示す.

条件 (ii) は，p_i が OOD のデータセット $\mathcal{D}_{\text{OOD}}$ に対して有効でない場合，つまりモデルがトリガーを含む OOD データ ($E_{\text{OOD}}(\mathbf{w}_i)$) に対して性能が低いときである．そこで，$E_{\text{OOD}}(\mathbf{w}_i)$ と $\mathcal{D}_{\text{OOD}}$ における予測の F1 スコアの差 Δ を計算し (式 (3.5))，その差を推論パターンがどれだけ OOD での予測に失敗しているかの指標とする．

$$\Delta_i \stackrel{\text{def}}{=} \text{F1}(E_{\text{OOD}}(\mathbf{w}_i), f) - \text{F1}(\mathcal{D}_{\text{OOD}}, f). \tag{3.5}$$

したがって，ある推論パターン $\tilde{p}_i = \mathbf{w}_i \rightarrow l_i$ が shortcut reasoning であるとは，$g(p_i)$ と iid_acc が十分に高く，Δ_i が十分に低い，ということである．形式的には，shortcut reasoning の集合 $\tilde{P}$ は式 (3.6) のように書ける．

$$\tilde{P} \stackrel{\text{def}}{=} \{p_i \in C \mid g(p_i) > \lambda_1, \text{iid\_acc}_i > \lambda_2, \Delta_i < \lambda_3\}. \tag{3.6}$$

ここで，λ_1，λ_2，λ_3 は事前に定義する閾値である．λ_2 はチャンスレート，λ_3 は 0 以上でなければならないことに注意する必要がある．以上の定量的な shortcut reasoning の定義が，先行研究 [27, 38] とは異なり，OOD で大きな影響を与える shortcut reasoning の自動検出を可能にしている．

3.2 実験

3.2.1 設定

提案手法の評価に関して，言語処理モデルが行っている shotcut reasoning の ground truth を使用することが最もシンプルなアプローチである．しかしながら，昨今のニューラルネットに基づく言語処理モデルはブラックボックスであり，ground truth を得て評価用データセットを構築することは非常に困難である．そこで，我々は先行研究で spurious feature を含んでいることが認められている自然言語推論タスク (Natural Language Inference: NLI) や感情分類タスク (Sentiment Analysis: SA) の評価用データセットを使用して，推論パターンによって spurioues correlations の特徴を捉え，shortcut reasoning を検出することができるかを確認する．

データセット・モデル

SA **TweetEval** [1] は Twitter に投稿されたツイートにいくつかの情報がアノテーションされた英語のデータセットである．sentiment はそのサブセットであり，

表 3.1: データセットの詳細

データセット	train	validation	test
SA			
Tweeteval (sentiment)	45,615	2,000	12,284
MARC (en)	200,000	5,000	5,000
NLI			
MNLI (matched)	392,702	9,815	9,796
ANLI (round3)	100,459	1,200	1,200

表 3.2: モデルの詳細

タスク	モデル
SA	cardiffnlp/twitter-roberta-base-sentiment
NLI	roberta-large-mnli

positive/neutral/negative の 3 つの極性クラスがラベル付けされている．**MARC** [17] は Amazon における多言語の商品レビューと 5 段階の星の数による評価がアノテーションされたデータセットである．今回は英語レビューのデータセットを使う．前処理として，5 段階の評価に関して，星の数が 4 以上のレビューを positive，3 を neutral，2 以下を negative とした．

NLI　**MNLI** [41] は，premise と hypothesis の 2 つの文に対し entailment / neutral / contradiction の 3 つのラベルがアノテーションされた NLI のデータセットである．contradiction がラベル付けされている事例の hypothesis の多くに negation が含まれるという spurious feature が明らかになっている [9]．**ANLI** [24] は MNLI 同様に 3 種類のラベルがアノテーションされており，MNLI よりも複雑で難易度の高い NLI のデータセットである．

本実験では，IID として TweetEval と MNLI，OOD として，MARC と ANLI を使用する．予測モデルは，Huggingface で公開されている RoBERTa [20] ベースの fine-tuned モデルを使用する．詳細を 3.2.1 に示す．

Shortcut reasoning 検出に関する設定

Input reduction の入力 (i.e., $\mathcal{D}_{\text{IID}}$) として，train (学習データ) と test (テストデータ) の 2 通りを用いる．前者はモデルの傾向 (内部状態) を捉えた推論パターンが得られることを期待する．後者は，推論パターンを得るための最も直観的な手法である．本実験では，ランダムに選んだ 1,000 件のデータに IR を適用する．Shortcut reasoning 判定に係るハイパーパラメータについて，今回の実験では $\lambda_1 =$

50, $\lambda_2 = 70$，そして $\lambda_3 = -0.05$ と設定する．また，一般性の低い推論パターンをより正確に排除するために，$|E_{\mathrm{IID}}| \geq 100$ である推論パターンのみを分析の対象とする．

3.2.2 結果

検出された shortcut reasoning に該当する推論パターンを表 3.3 から表 3.6 に示す．

NLI

OOD に対する性能 $\mathrm{F1}(\mathcal{D}_{\mathrm{OOD}}, f)$ は 77.8 を示した．“/s” は入力の premise と hypothesis を隔てる特殊トークンである．得られた推論パターンを概観してみると，hypothesis に含まれる単語がトリガーの大半を占め，premise の中の単語を含む推論パターンは数件しかなかった．この結果はモデルが hypothesis に依存して文間関係を予測しているということを意味し，同様の結果が Poliak 等によって報告されている [28]．さらに，否定表現が hypothesis に含まれている推論パターンが多く shortcut reasoning として判定されたが，これについても先行研究 [9] で指摘されている shortcut reasoning である．以上から，提案手法が多様な shortcut reasoning を適切に検出できていることがわかった．

SA

OOD における性能 $\mathrm{F1}(\mathcal{D}_{\mathrm{OOD}}, f)$ は 60.3 であった．得られた推論パターン全体を見てみると，その多くが [“love”] → Positive や [“awful”] → Negative 等の感情語に関連したものであった．このことから，SA においてモデルは感情語を予測の重要な手がかりとしていることがわかる．shortcut reasoning であるものについても，感情語を含む推論パターンが多かった．したがって，shortcut reasoning は必ずしも先行研究で報告されている [“Spielberg”] → Positive のような予測と無関係な単語のみに反応しているのではないことがわかる．一方，そのような予測と無関係な単語がトリガーとなる推論パターンは得られなかったが，IID と OOD を逆にした設定では異なる結果が得られる可能性がある．

MARC に関して，neutral とラベル付けされたレビュー (星 3 つ) においては positive な表現と negative な表現が混ざったものが多く見られた．Δ の絶対値が大きい推論パターンを観察してみると，総じて neutral を誤って予測していることから，本来であればレビューを総合的に評価しなければならないにも関わらず，レビュー中のどちらか一方の極性の感情語だけに依存して推論していることがわかった．

表 3.3: 検出された shortcut reasoning (タスク：NLI / $\mathcal{D}_{\text{IID}}$：test)

$\tilde{p}$	g	iid_acc	Δ	$\lvert E_{\text{IID}}(\mathbf{w}_i)\rvert$
[“/s”, “has”, “no”]→ Contradiction	88.3	97.1	-6.4	428
[“/s”, “never”]→ Contradiction	80.3	99.3	-9.0	1515
[“/s”, “no”]→ Contradiction	69.0	94.1	-6.7	2038
[“/s”, “soon”]→ Neutral	64.1	100.0	-6.5	142
[“/s”, “ ’t”]→ Contradiction	56.0	92.8	-14.6	2832
[“/s”, “not”]→ Contradiction	55.0	94.5	-51.6	8708
[“,”, “too”, “/s”, “not”]→ Contradiction	54.9	100.0	-8.3	164
[“is”, “/s”, “not”, “.”]→ Contradiction	54.5	93.5	-14.6	2495
[“Power”, “/s”]→ Entailment	52.1	100.0	-21.7	163
[“/s”, “ ’t”, “.”]→ Contradiction	51.9	92.9	-6.9	1868

表 3.4: 検出された shortcut reasoning (タスク：NLI / $\mathcal{D}_{\text{IID}}$：train)

$\tilde{p}$	g	iid_acc	Δ	$\lvert E_{\text{IID}}(\mathbf{w}_i)\rvert$
[“/s”, “never”]→ Contradiction	80.3	99.5	-9.0	1515
[“/s”, “been”]→ Neutral	77.6	96.9	-9.7	2496
[“/s”, “no”]→ Contradiction	69.0	99.1	-6.7	2038
[“might”, “/s”, “not”]→ Contradiction	65.1	100.0	-6.1	126
[“/s”, “soon”]→ Neutral	64.1	97.8	-6.5	142
[“/s”, “did”, “not”]→ Contradiction	61.3	96.8	-10.4	1329
[“/s”, “is”, “always”]→ Neutral	60.8	96.6	-5.2	102
[“/s”, “is”, “not”]→ Contradiction	59.9	97.4	-16.4	2381
[“/s”, “ ’t”]→ Contradiction	56.0	98.5	-14.6	2832
[“/s”, “not”]→ Contradiction	55.0	97.9	-51.6	8708
[“/s”, “wasn”, “ ’t”]→ Contradiction	53.7	96.7	-5.1	164

表 3.5: 検出された shortcut reasoning (タスク：SA / $\mathcal{D}_{\mathrm{IID}}$：test)

$\tilde{p}$	g	iid_acc	Δ	$\lvert E_{\mathrm{IID}}(\mathbf{w}_i)\rvert$
[“Excellent”]→ Positive	96.2	100.0	-7.2	184
[“Not”, “worth”]→ Negative	96.2	100.0	-7.0	220
[“awful”]→ Negative	90.1	80.0	-18.9	161
[“enjoyed”]→ Positive	90.0	100.0	-8.3	251
[“Love”]→ Positive	88.5	83.7	-10.6	925
[“Great”]→ Positive	87.8	80.0	-10.5	1627
[“fantastic”]→ Positive	87.5	92.9	-8.2	128
[“amazing”]→ Positive	85.3	91.7	-6.5	381
[“love”]→ Positive	83.5	87.7	-11.6	2418
[“Beut”]→ Positive	82.5	84.5	-11.9	143
[“awesome”]→ Positive	80.9	96.3	-7.2	340
[“frustrating”]→ Negative	80.7	100.0	-11.3	171
[“worse”]→ Positive	78.2	93.1	-13.8	147
[“beautiful”]→ Positive	78.0	83.3	-8.4	487
[“favorite”]→ Positive	76.4	87.5	-5.4	267
[“Good”]→ Positive	76.0	80.8	-9.5	1050
[“great”]→ Positive	75.3	82.4	-9.2	5092
[“Nice”]→ Positive	71.8	90.9	-10.6	602
[“cute”]→ Positive	69.3	100	-8.4	1146
[“hate”]→ Negative	69.2	83.0	-16.9	133
[“nice”]→ Positive	62.9	71.4	-5.2	2426
[“are”, “bad”]→ Negative	58.2	80.0	-5.2	239
[“excited”]→ Positive	57.0	86.7	-16.6	309
[“liked”]→ Positive	56.3	92.7	-9.2	632
[“always”]→ Positive	51.6	76.2	-5.3	579

表 3.6: 検出された shortcut reasoning (タスク：SA / $\mathcal{D}_{\text{IID}}$：train)

$\tilde{p}$	g	iid_acc	Δ	$\lvert E_{\text{IID}}(\mathbf{w}_i)\rvert$
[“worst”]→ Negative	97.45	81.7	-25.4	158
[“Perfect”]→ Positive	96.0	88.6	-12.9	324
[“enjoyed”]→ Positive	90.0	93.9	-8.3	251
[“Great”]→ Positive	87.8	94.7	-10.5	1627
[“poor”]→ Negative	87.1	76.0	-12.9	458
[“disapointed”]→ Negative	86.5	100.0	-12.6	1480
[“amazing”]→ Positive	85.3	98.4	-6.5	381
[“love”]→ Positive	83.5	94.3	-11.6	2418
[“perfect”]→ Positive	83.5	94.1	-7.9	1219
[“awesome”]→ Positive	80.9	96.2	-7.3	340
[“favorite”]→ Positive	76.4	90.4	-5.4	267
[“fun”]→ Positive	76.3	92.2	-11.1	396
[“loved”]→ Positive	76.1	95.7	-7.4	725
[“Good”]→ Positive	76.0	90.4	-9.5	1050
[“great”]→ Positive	75.3	92.1	-9.2	5092
[“Nice”]→ Positive	71.8	87.7	-10.6	602
[“cute”]→ Positive	69.3	95.1	-8.4	1146
[“hate”]→ Negative	69.2	75.8	-16.9	133
[“interesting”]→ Positive	68.2	91.7	-11.0	195
[“sweet”]→ Positive	65.0	93.8	-6.2	143
[“nice”]→ Positive	62.9	92.5	-5.2	2426
[“best”]→ Positive	62.2	91.1	-5.1	792
[“is”, “bad”]→ Negative	60.5	77.2	-8.9	521
[“excited”]→ Positive	57.0	97.8	-16.6	309

train or test

Input reduction の適用先について，train と test から得られた推論パターンには大きな差は見られなかった．具体的なパターンは異なるが，特徴はおおむね同じであった．

未知の shortcut reasoning と考えられるもの

NLI では hypothesis に含まれる [“popular”] → NEUTRAL や，[“as”, “well”] → NEUTRAL が新しい shortcut reasoning として得られた．どちらも一般性，スコア差ともに十分に shortcut reasoning と判断できる水準にある．一方，SA については特に見つけることができなかった．

第4章　論理的根拠に基づく機械読解システム

機械読解 (Machine Reading Comprehension: MRC) とは，質問と対応する文書を受け取り，読解問題を解く自然言語処理における応用タスクのことである．読解のプロセスでは，質問の内容を正しく理解し，文書から解答に必要な情報を取り出し，最終的な解答を推論することが要求される複雑なタスクである．Transformer の登場以来 MRC の性能は向上したものの，機械読解モデルの頑健性に関する懸念が依然として課題となっている．機械読解モデルは質問と文書を受け取って読解問題を解くが，文書中に解答に必要な情報とよく似た紛らわしい情報について書かれた文章が含まれていると，その文章から解答を作成してしまうといった問題が，数多くの先行研究によって明らかになっている．こういった，入力中のノイズに対する脆弱性が，機械読解モデルの頑健性の問題として議論されている．加えて，説明可能性 (Explainability) も課題となっている．通常の MRC では，モデルは解答のみを出力するため，ユーザーはなぜその解答に至ったのかを知ることができない．特に，ニューラルネットワークをベースとする読解モデルは内部の推論プロセスを解釈することが極めて困難であるため，説明可能性の向上が盛んに議論されている．以上のような問題点を背景に，様々な先行研究が頑健性・説明可能性の向上に取り組んだ．

Jia らや Jiang らは自動生成した敵対的事例 (Adversarial examples) を用いて頑健性の向上にアプローチした．敵対的事例とは，モデルが間違いやすい紛らわしい情報を加えた入力であり，それを用いて訓練することでノイズ対して頑健なモデルを構築できることが期待される [13, 14].

いくつかの先行研究は，**Rationalization** と呼ばれるアプローチを試みている [25, 12]．Rationalization とは，あるタスクにおいて，解答とその論理的根拠 (Rationale) を言語処理モデルに出力させるパラダイムである．推論プロセスの上で論理的根拠を作らせ，解答と合わせて出力させることで，精度の向上に加え，言語処理モデル，特に BERT や GPT に代表されるブラックボックスな事前学習済み言語モデルの説明可能性を向上させることを目的として研究されている．Explain-then-Predict は，rationalization が可能なアーキテクチャである．具体的には，入力されたデータに対し，それらを用いて予測を行うモジュール (Predictor) の前に，Explainer と呼ばれる入力の解答に必要となる情報が記載された部分のみを入力データから抽出して predictor に渡す機能を持ったモジュールを置いた構造のことを指す．このアー

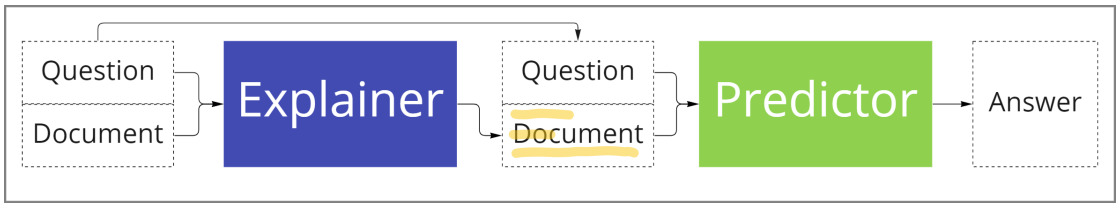

図 4.1: 機械読解における Explain-then-Predict アーキテクチャの概要

キテクチャの概要を図 4.1 に示す．Explainer の抽出した解答に必要な情報がそのまま論理的根拠となり，またそれを基に predictor は読解を行っているため，アーキテクチャ自体が説明可能であり，説明が推論プロセスをいかに表現しているかを表す尺度である忠実性 (Faithfulness) が高いことがこのアーキテクチャの特長である．Chen らは，explain-then-predict 型の rationalization について，explainer が入力に含まれる余計な情報 (ノイズ) を排除してくれる役割を持っているため，学習データと異なる分布を持つ OOD データへの処理能力，すなわち頑健性が向上するのではないかという仮説を検証し，一部の実験結果では頑健性の向上が見られたものの，その効果は限定的であった [4].

本節では，explain-then-predict 型の機械読解システムを再検討する．具体的には，explainer が前節で取り上げた Shortcut reaosning を行っているという仮説について検証を行う．検証実験の結果，explainer は質問や文書に欠損を加えた上でそれらを入力しても，説明抽出の精度が変化しなかったことが明らかとなった．この結果は，explainer がパターンマッチ質問と文書を読解していないことが示唆された．

4.1 Explainer における Shortcut reasoning

4.1.1 Shortcut reasoning の分類

我々は,「Explain-then-predict 型の機械読解システムにおいて，explainer が shortcut reasoning を行っている」という仮説の検証を行う．この仮説を検証するに先立ち，機械読解タスクにおける explainer の shortcut reasoning の分類を行う．Shortcut reasoning は，モデルによる短絡的かつ非合理的な推論であり，正しい予測を行うためには不十分な情報や，予測に不必要な情報から解答を推論することが多い．機械読解タスクにおいて，入力は質問と文書の 2 つが与えられるため，そのどちらにおいて shortcut reasoning が発生しているかに基づいて分類する．

Type 1：質問における shortcut reasoning

1 つ目の分類は，質問において発生している shortcut reasoning である．想定される形態としては，特定の疑問詞が含まれている場合に文書中の決まった部分を

Edited Question:
- First half *"What is the name of the quarterback"*
- First word *"What"*
- No Q *NaN*
- Q shuffle *"Where was the largest single capital investment IBM made have been built?"*
- Q word shuffle *"is ? who of XXXIII Super What name in 38 the the quarterback Bowl"*

Edited Document :
- NE shuffle
"39 became the first quarterback ever to lead two different teams to multiple Champ Bowl XXXIV. He is also the oldest quarterback ever to play in a 38 at age Super Bowl. ..."
- D word shuffle
"lead to oldest also 39 became in quarterback Champ He the Super Bowl to. quarterback teams to 38 Bowl multiple first XXXIV. ever is at the different ever play a age two..."

図 4.2: 入力に加える編集の例

常に抽出するというものである．例えば，質問 "Who is a current president of the USA?" に対して，文書中の個人名が含まれている文を論理的根拠として出力する，というようなケースである．

Type 2：文書における shortcut reasoning

2 つ目は，文書内のある特定の情報に依存して，出力を行う shortcut reasoning である．入力された質問に関わらず，固有表現の含まれる文を出力するといった形態が想定される．

4.1.2 検証方法

Shortcut reasoning の分類に基づき，explainer への入力である質問と文書を編集し，出力の変化を検証する手法を採用する．本研究で採用した編集の種類とその例を図 4.2 に示す．

質問に対する編集では，質問の一部を欠損させる手法を用いる．質問の前半分のみ (First half)，最初の単語 (First word)，質問なし (No question) を設定する．加えて，他の文書に対する質問と入れ替える編集 (Q shuffle) と，質問文内の単語をシャッフルする (Q word shuffle) 編集も行う．

文書に対する編集では，文書中に含まれる固有表現の位置をシャッフルする設定 (NE shuffle) と，単語をシャッフルする設定 (D word shuffle) を採用する．

4.1.3 評価指標

前述のとおり，この実験では explainer の出力の変化を観察する．それに伴い，結果がどのように変化したかを図る指標を導入する．

Gold rationale F1 (GR) は，Chen らが定義した評価指標であり，モデルの予測した rationale と人間がアノテーションした rationale の間の F1 スコアである．本実験では，文単位での GR を計算する．

Matched rationale F1 (MR) は，編集前と編集後の入力に対する説明の一致度を示す尺度である．具体的には，編集の前後での予測された説明の F1 スコアを計算し，それを MR とする．

GR は予測した rationale が正解かどうであるかを評価する指標であるが，我々の関心は予測精度ではなく，出力の変化であるため，その変化を評価する指標 MR も導入する．

4.2 実験

4.2.1 設定

機械読解モデルとして，Paranjape らの提案した Information bottleneck を応用した VIB を用いる [25]．VIB は，explainer と predictor ともに BERT を採用しており，rationalization と推論能力を評価するためのデータセットである ERASER [6] で高い精度を達成している．データセットは，ERASER のサブセットである，様々なジャンルの読解問題が収録された MultiRC [18] を用いる．MultiRC には複数の文からなる文書が各質問に付与されており，explainer は解答に必要な文をそこから選択する形式で出力する．

4.2.2 結果

実験の結果を図 4.3 に示す．GR は緑色，MR は青色のバーで示されている．図左端の Original と Random は，それぞれ編集前の入力とランダムに文を選択したとき (チャンスレート) の GR である．

質問に対する編集

グラフ中央の質問に対する編集に対する変化を観察すると，質問を別の質問に変えたり，大きく欠損させると，MR は 40 付近の低い値を示し，GR もチャンスレート付近まで低下していることが確認できた．したがって，質問に対する著しい編集は explainer の精度を下げる，つまり予測に際してある程度は質問を考慮できていることが推測できる．

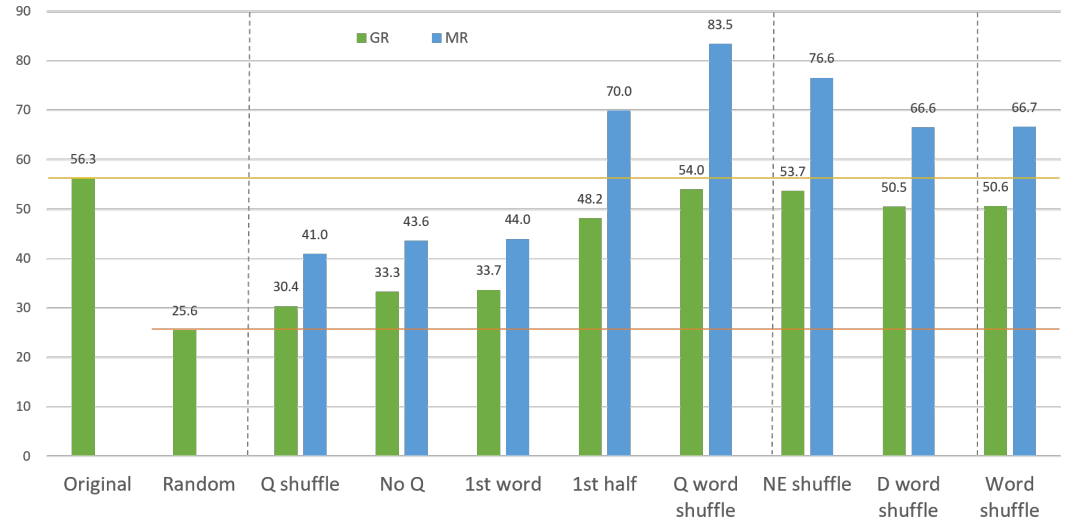

図 4.3: Explainer への入力に編集を加えたときの出力の変化を観察した実験結果

その一方で，質問の半分のみや単語の位置をシャッフルした設定での GR は Original と比較して大きく変化しなかった．MR をみても，それぞれ 70.0，83.5 と概ね高い水準であることがわかる．特に，全単語の位置をシャッフルした Q word shuffle における 83.5 という MR は非常に高い値であり，explainer の出力の大半が編集により変化しなかったことを表している．この結果から解釈できるのは，まず質問内の前半分に依存して説明を作成しているということである．そして，explaiener が語順に対して鈍感であるということである．

文書に対する編集

グラフ右方に示した文書に対する編集の結果をみてみると，GR・MR ともに高い値を示していることがわかる．つまり，編集の前後で出力が大きく変化しなかったことを意味しており，shortcut reasoning の可能性が示唆される．特に，固有表現の位置をシャッフルした NE shuffle では，GR は Original とほぼ同水準であり，MR は 70 以上という顕著な結果を示した．この結果から，explainer は文章の意味や文脈に関わらず，文中の固有表現を抽出して説明を選択している可能性が示唆される．

第5章　自己認識が可能な知識ベースとしての大規模言語モデルの実現に向けた議論

知識ベースとしての言語モデル (Language Models as Knowledge Bases: LM as KB) は，主に Transformer をベースとした事前学習済み言語モデル (LM) を知識ベース (KB) として応用するパラダイムのことである [26]．BERT や GPT といった LM はその事前学習において，与えられた文の穴埋め (Masked Language Modeling: MLM) や文中のある地点の次の単語の予測 (Next Token Prediction: NTP) を解くタスクを大規模に行うことが一般的である．事前学習データには Wikipedia 等の Web 上から取得された膨大なテキストデータを用いるが，LM は事前学習を通してそこに含まれる事実的知識や一般常識などの多種多様な知識 (i.e., 世界知識：World Knowledge) を内部パラメータに保存し，様々なトピックに関する質問に答えたり，特定の情報を提供する能力を持つことが知られている．

言語モデルを知識ベースとして用いることの利点として，(i) 自然言語での問い合わせが可能である, (ii) 多様な入力クエリに対応が可能である，(iii) 非常に広範な知識を持っている，といった点が挙げられる．(i) は，SQL 等の知識ベースでは問い合わせたいことについて特定の形式言語等に変換する必要があるのに対し，LM as KB は直感的な自然言語で問い合わせることが可能であるといった特徴である．(ii) については，ある知識に関する問い合わせにおいて，特定の問い合わせ形式に縛られることなく，(i) で述べたように自然言語の様々な型式や文脈に対応することができる．(iii) は，前述の通り Wikipedia や論文等から大規模な事前学習を行っているため，多様なジャンルの知識を持っていることが期待できる他，一般常識のような具体的に定式化できないような抽象的な知識にも対応することができる．

以上のようなメリットの一方で，いくつかの考慮すべき課題も存在する．その一つが，Hallucination という，いわゆる言語モデルの「嘘」である．事前学習済み言語モデルは，一般的な知識ベースのように，正しい知識を体系的に保存することを目的として構築されておらず，内部構造もブラックボックスであるため，その出力は必ずしも正しいとは限らないのに加えて，その出力に至ったプロセスを理解することは困難である．

そういった課題を背景に，LLM の生成結果の信頼性 (i.e., 予測の自信度) を推定する研究が盛んに行われている．先行研究では，各トークンの確率分布を正規化

する手法 [23] や，エントロピーを計算する手法 [37] の他，LLM にプロンプトで直接自信度を出力するように指示する手法 (Verbal confidence) [34] 等が予測の正解率をよく反映していること，すなわち自信度と正解率が正相関していることが明らかになっている．ここで，生成結果の信頼性の研究で議論されている「自信度」とは，基本的にモデルが自らの予測が正しいと考える程度のことを指すことがほとんどである．

その一方で，事前学習データにおける知識の出現頻度と正解率が正の相関をしている結果が Mallen らによって示されている [21]．しかしながら，LLM は識別問題や翻訳等の具体的なタスクに対応できるよう訓練されたわけではなく，一部のモデルは Instruction tuning や RLHF(Reinforcement Learning from Human Feedback) でアライメントされているものの，基本的には文の穴埋めや入力文の次の文もしくはトークンを予測することを事前学習で解いているのみである．したがって，LLM の自信度が果たして予測の正しさを反映しているかどうかは議論の余地がある．

そこで我々は，LLM の生成結果の信頼性推定に関する研究における自信度は，予測が正しい確率ではなく，予測について記憶している程度を示している，という仮説を主張する．より詳細には，LLM が示す自信度は大規模な事前学習により結果として正解率と相関しているが，実際には事前学習においてどれだけの頻度で事例を観察したかを示しているに過ぎない，という仮説である．本章では，この仮説に対して言語モデルに関する実験を行い，その結果を考察する．

5.1 LM as KB

5.1.1 問題の定式化

本研究では，LM as KB における言語モデル LM は狭義の言語モデル，つまり文脈から次々に単語を予測していくモデル，特に GPT や PaLM 等の Transformer の Decoder をベースとした事前学習モデルを扱う．また，知識ベース KB について，知識 k とは subject (s_k), relation (r_k), object (o_k) のトリプレットのタプル (s, r, o) として表す．

LM as KB では，知識を問い合わせる際には，s_k と r_k から作成されたクエリ q_k を言語モデルに入力し，出力された $\hat{o}_k$ を知識として得る．これは以下のように定式化される．

$$\hat{o}_k = \mathcal{M}_\theta(q_k) = \underset{e \in V}{\operatorname{argmax}}\, p(e \mid t(s_k, r_k), \theta) \tag{5.1}$$

$\mathcal{M}_\theta$ は事前学習によってパラメータ θ を得た言語モデル，$t(\cdot)$ はクエリ作成に適用するテンプレート，e は言語モデルの語彙集合 V に含まれるエンティティである．

ここで，我々の関心は言語モデルの自信度であるため，ある知識に関する予測において必ずしも object エンティティを得る必要はない．また，LLM の中で比較的パラメータ数の小さいモデルでは，クエリに対して特定の object エンティティ

ではなく，ジェネリックな文章が生成されることが多々あるため，出力の評価が困難となる．したがって，LLM の中で比較的パラメータ数の小さいモデルの性能を考慮して，LM as KB を以下のように変形する．

$$\hat{b}_k = \mathcal{M}_\theta(q_k) = \underset{b \in \{\text{True,False}\}}{\operatorname{argmax}} \; p(b \mid t'(s_k, r_k, o_k), \theta) \tag{5.2}$$

これは，入力クエリを平叙文に置き換え，“{statement of `subject`, `relation`, `object`}. *True or False?*” のように，真偽をモデルに問い合わせるように変えたものである．この変形により，言語モデルの出力を安定させることができ，評価が容易になる．

以上を踏まえて，LLM の自信度 c は以下のように定義される．

$$c(k) = \gamma(\hat{b}_k) = \gamma(\mathcal{M}_\theta(q_k)) \tag{5.3}$$

γ は自信度を計算する任意の関数である．$c(k)$ は 0 から 1 までの連続値で表され，値が高ければ高いほど自信度も強いことを表す．

続いて，自信度で表されるものの正体の仮説で述べた，予測した知識が正しい確率を正解度 $a(k)$，LLM が知識を記憶している程度を記憶度 $m(k)$ と定義し，それぞれ以下のように表す．

$$a(k) = p(\hat{b}_k = y) \tag{5.4}$$

$$m(k) \approx f(k) = \text{number of } k \text{ in pre-training data} \tag{5.5}$$

y は正解の真偽値 ($\{\text{True}, \text{False}\}$) を表す．記憶度 m については，事前学習で k が現れる頻度と近似する．本研究において，モデルがある知識についてどれだけ記憶しているかが，学習データに含まれる頻度であることの厳密な考証は行わないものの，一般に機械学習において学習データ内の対象の出現頻度は，その対象の予測精度と相関することはよく知られている．よって，以後，特に断りがない限り記憶度 m は頻度 f として表す．

5.1.2 仮説の検証

前項で定義された，自信度 c，正解度 a，頻度 f を用いて仮説を改めて簡潔に定義すると，c は a ではなく f を反映している，となる．この仮説の検証のため，まず先行研究で示されている，自信度と正解度，頻度と正解度の相関に加え，我々の関心である自信度と頻度の相関を確認する．その後，自信度が正解度ではなく頻度を反映しているかどうかを検証するため，頻度・正解度・自信度の関係を分析する．

頻度と正解度の相関を明らかにした Mallen らは，Wikipedia から取得した知識のトリプレットの各データに対して subject エンティティの人気度を付与したデータセット PopQA を構築した [21]．PopQA の各知識に付与されている人気度は，

subject エンティティの Wikipedia ページの月間閲覧数を用いている．本実験では，式 (5.6) に示すように，この人気度を頻度 $f(k)$ とし，自信度や正解度との関係性を検証する．

$$f(k) \approx p(s_k) = \text{number of monthly access to the subject Wikipedia page} \quad (5.6)$$

5.2 実験

5.2.1 設定

データセット

PopQA [21] を使用した．Wikipedia から取得した subject,relation と object のトリプレットが約 14,000 事例収録されている．含まれている知識はすべて事実，すなわち True(正例) であるため，同じ relation を持つ他の事例からランダムに取得した object(o^*) を自身のオブジェクト (o) と入れ替えて負例を作成する．

データセットに含まれる relation には，occupation，capital-of，birth-place 等がある．実験では，データセット全体での評価のほか，正例のみ，負例のみ，relation ごとの評価を行う．

モデル

Llama2 の chat モデル (7B，13B)，GPT-3.5-turbo を使用する．なお，GPT-3.5-turbo に関しては，API の使用上，予測したトークンの確率分布について上位 5 つしか得られないのため，“True” もしくは “False” が上位 5 トークンに含まれていなかった場合には，検証の対象から除外する．

自信度

モデルの自信度を測定する手法 (i.e., γ) はいくつか存在するが，今回の実験では先行研究 [34] に基づき，予測した $\{\text{True}, \text{False}\}$ の確率分布 $\mathcal{B}$ の正規化エントロピー NH を使用し，以下のように定義する．

$$c = \gamma(\hat{b}) = 1 - \text{NH}(\mathcal{B}) = 1 - \left(-\frac{\sum_{b \in \{\text{True,False}\}} \mathcal{B}(b) \log \mathcal{B}(b)}{\log 2} \right) \quad (5.7)$$

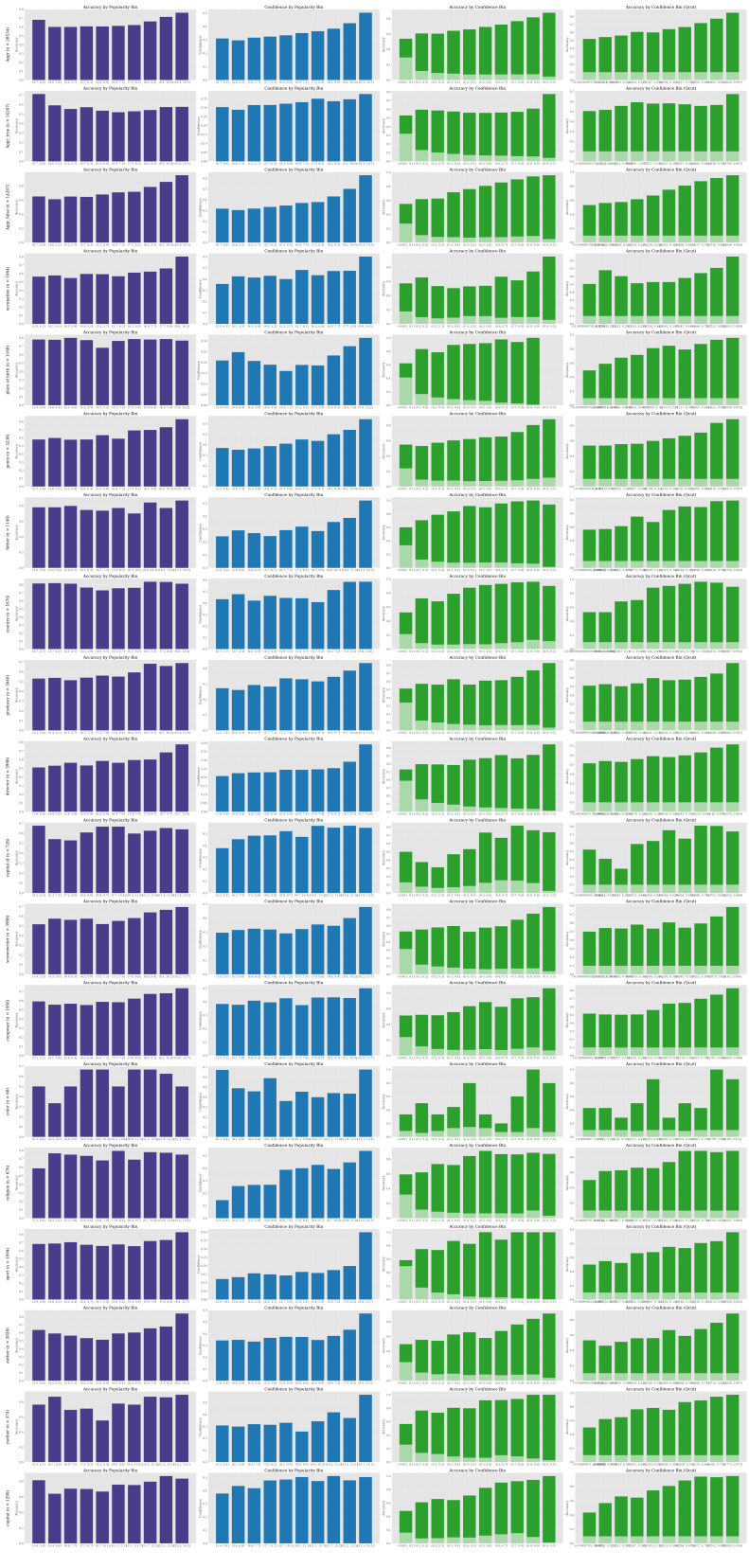

図 5.1: 自信度、正解度、頻度の相関 (Llama2-7b-chat)

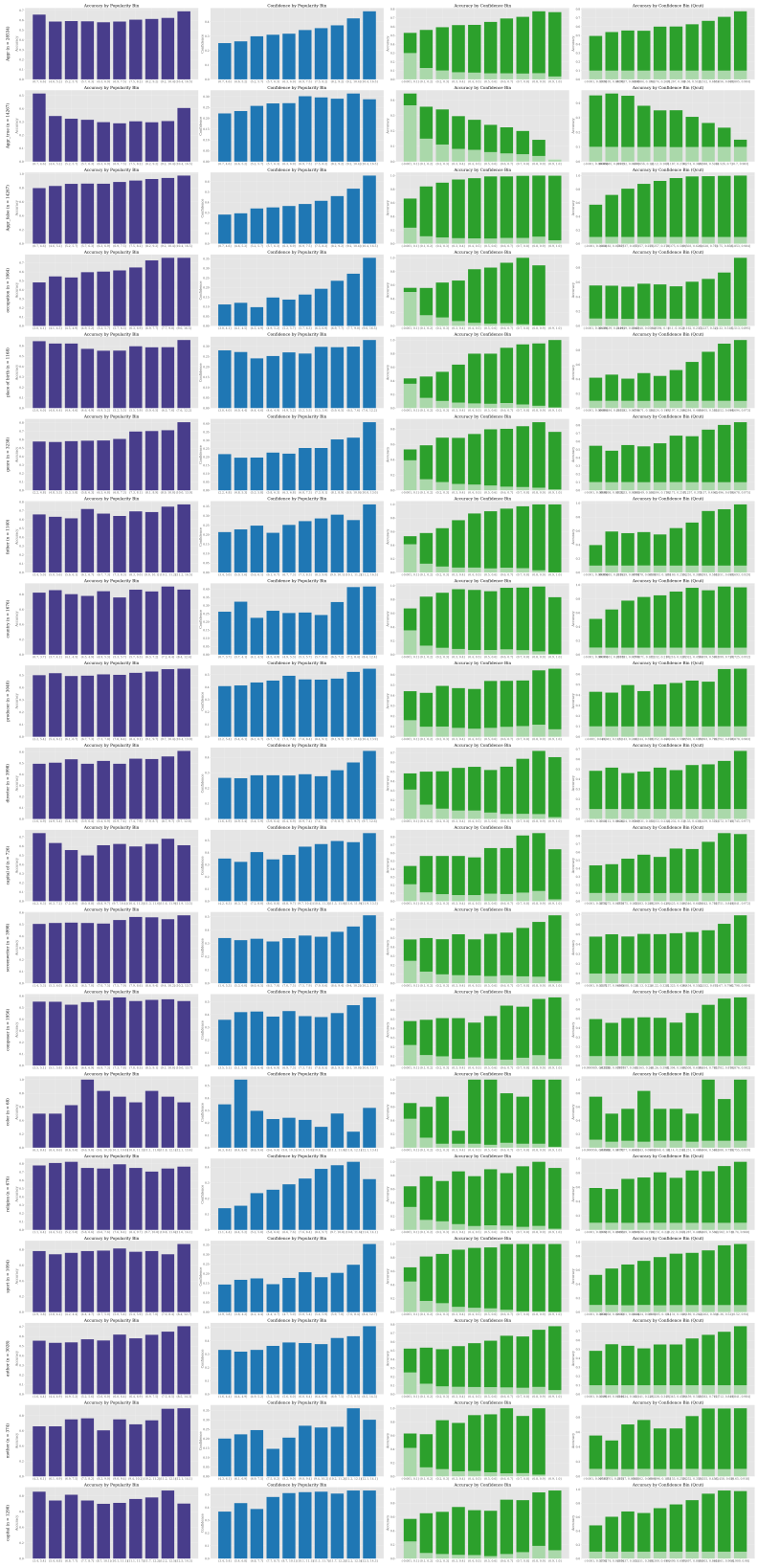

図 5.2: 自信度、正解度、頻度の相関 (Llama2-13b-chat)

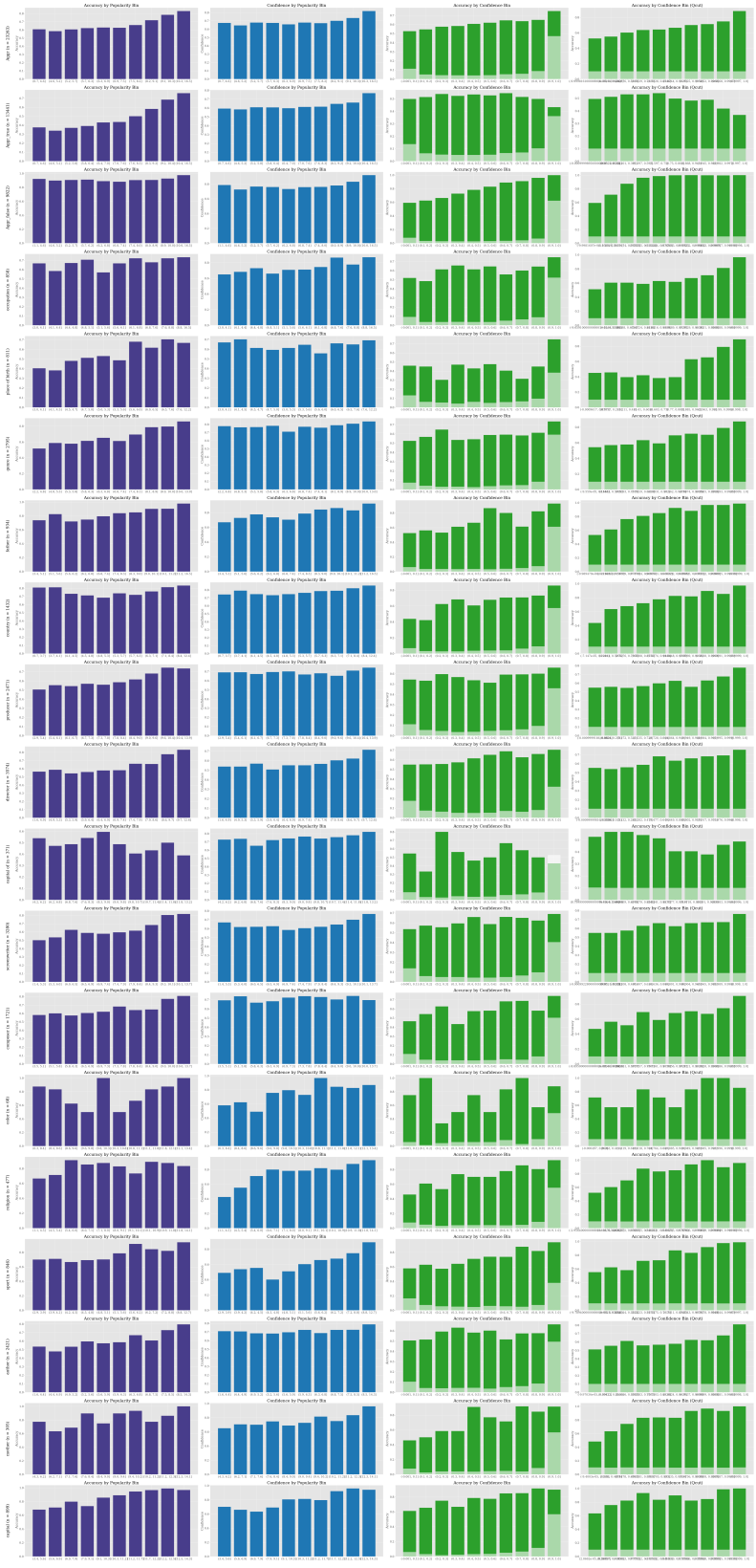

図 5.3: 自信度、正解度、頻度の相関 (GPT-3.5-turbo)

5.2.2 結果

各指標の相関

実験の結果を図 5.1〜5.3 に示す．各図の列は，左から順に，頻度 (横軸) と正解度 (縦軸)，頻度 (横軸) と自信度 (縦軸)，自信度 (横軸) と正解度 (縦軸)，一様分布になるように分割した自信度 (横軸) と正解度 (縦軸)，の相関を表している．各グラフは，横軸の指標が取り得る値の範囲を 10 個に分割し，それぞれを 1 つの bin としている．また,「一様分布になるように分割した自信度」とは，自信度の値によって 10 分割するのではなく，各 bin に割り当てられる事例の数が同じになるように自信度の範囲を 10 分割して可視化されたグラフであることを表す．正解度 a_k は一つの事例のみから計算することはできないため，サブセット (1 つの bin) に対して正解度を計算する．一方，各図の行は，1 行目はテストデータ全体の結果，2 行目は正例のみの結果，3 行目は負例のみの結果，4 行目以降は relation ごとの結果である．

全体の結果を見てみると，全モデルにおいて正解度対頻度，自信度対頻度，自信度対正解度どの結果でも相関が観察された．正解度対頻度，自信度対正解度については，概ね先行研究の主張に沿った結果となった．しかしながら，Llama2 を使用した正例のみの結果において，正解度対頻度の相関が比較的弱い傾向が見られた．また，Llama2-13b においては正解度対自信度が逆相関する結果が観察された．これは，自信度が高いほど不正解していることを示している．

人気度ごとの頻度・正解度の関係性

続いて，頻度ごとに自信度と正解度の関係を分析する．結果を図 5.4〜5.6 に示す．この実験では，頻度順に出力の集合を 5 分割し，それぞれについて自信度と正解度を計算した結果を可視化した．グラフの各線は，最低頻度帯の水色から，頻度が高くなるにつれ紫に色調を変化させている．GPT-3.5-turbo については，自信度 c の分布が偏っていることを考慮して，正解率の計算に用いる事例の数が一様になるように自由度の範囲を分割してプロットした結果を図 5.7 に示す

Llama2 の結果は，頻度帯ごとの自信度対正解度に大きな差異はみられなかった．したがって，頻度に関わらず，自信度と精度は相関していることを示している．

一方で，GPT-3.5-turbo における結果 (図 5.6，5.7) は興味深いものとなった．頻度帯が高くなるにつれ，自信度と正解度の相関が強くなっていることがわかる．グラフ左端ではどの頻度帯も 0.55 付近の近しい位置から始まっている一方で，右端の方では，低頻度帯はグラフ中央に終端があるのに対し，高頻度帯はグラフ上方に終端がある．つまり，高頻度帯の知識に対しては自信度が上がるほど正解するのに対し，低頻度帯の知識に対しては自信度が上がっても正解率はそこまで上がらないことを意味している．この結果から，自信度は本来頻度を反映しているが，頻度の高い知識については事前学習で頻繁に出現するため，自動的に正解率も上がっている，

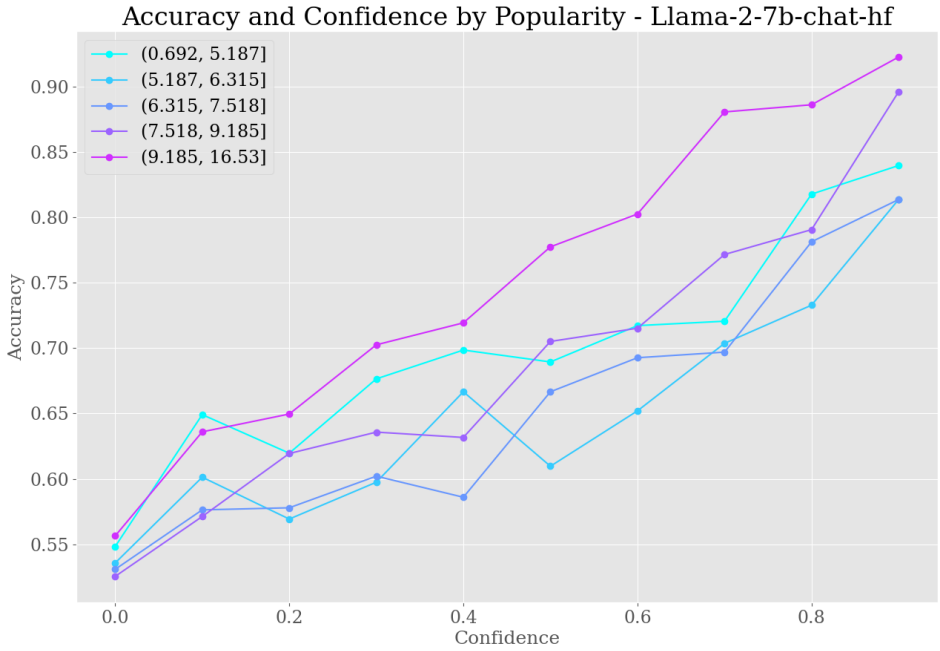

図 5.4: 頻度別に見た自信度と正解度の関係 (Llama2-7b)

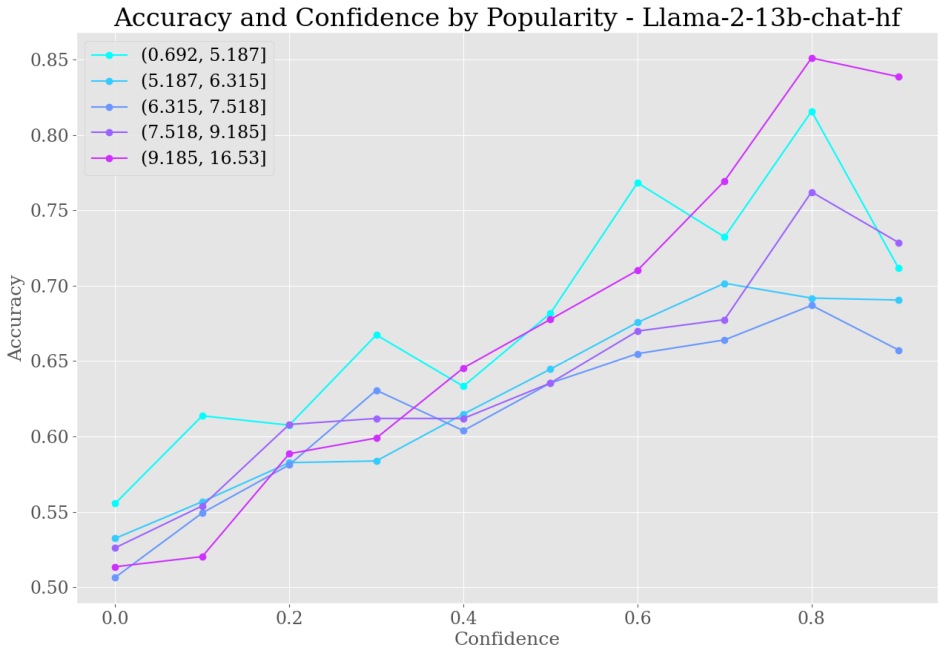

図 5.5: 頻度別に見た自信度と正解度の関係 (Llama2-13b)

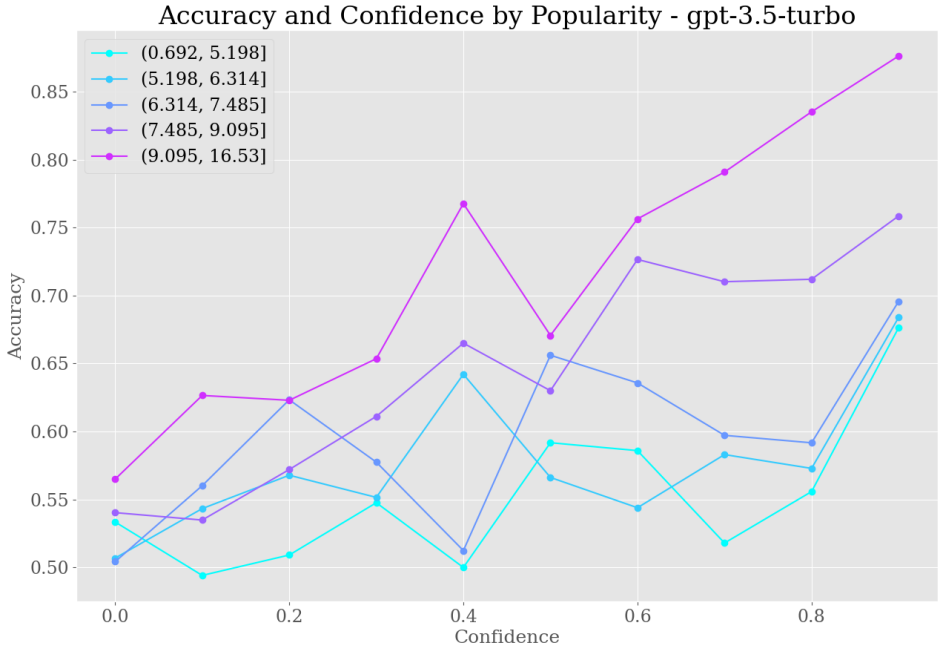

図 5.6: 頻度別に見た自信度と正解度の関係 (GPT-3.5-turbo)

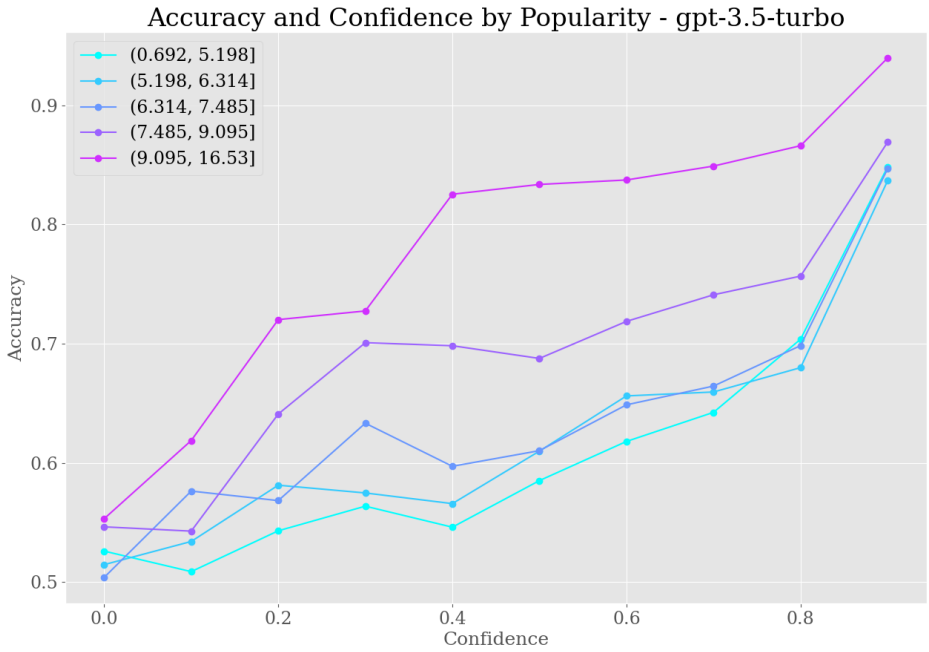

図 5.7: 頻度別に見た自信度と正解度の関係 (GPT-3.5-turbo) – 自信度を一様分布にしたとき

という現象が発生している可能性が示唆される．しかしながら，GPT-3.5-turbo のみでこの現象が発生している理由を説明することができないこともあり，この現象が我々の仮説を裏付けると直ちに主張することはできない．

第6章　おわりに

6.1　本研究のまとめ

本研究では，言語モデルの推論に関する実験や検証を行った．3 章では，言語処理モデルの shortcut reasoning を自動的に検出する手法を提案した．いくつかのタスクにおける評価の結果，提案手法は先行研究ですでに明らかになっていた shortcut reasoning に加え，未知のものについても自動的に人手による評価なしに検出することができた．4 章では，機械読解システムにおける Explain-then-Predict 型のアーキテクチャの Explainer が shortcut reasoning を行っているという仮説の検証に取り組んだ．実験の結果，入力の質問・文書にノイズを加えても出力が大きく変わらないケースを確認し，explainer が入力された情報を十分に読解してないことが考察された．5 章では，自己認識可能で説明可能な推論システムの一部の成す自己認識可能な知識ベースとしての LLM の実現に向け，LLM の自信度の本質を明らかにするための分析を行った．自信度が正解率ではなく頻度を反映しているという仮説に対して，確証は得られなかったものの，興味深い示唆を得ることができた．

6.2　今後の課題

今後の課題として，最終的な目標である自己認識可能かつ説明可能な推論システムの構築に向けた取り組みを行っていきたい．具体的には，第 5 章の分析結果を活用しつつ，事前学習データにおけるある知識の頻度を表すより適した指標の導入や，自己認識可能な知識ベースとしての LLM において実際に自信度を計算する手法の考案，LLM の持つ知識に関する包括的な考察や分析等を行う予定である．

参考文献

[1] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650, Online, November 2020. Association for Computational Linguistics.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[3] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1860–1874, Online, August 2021. Association for Computational Linguistics.

[4] Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. Can rationalization improve robustness? In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3792–3805, Seattle, United States, July 2022. Association for Computational Linguistics.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, Online, July 2020. Association for Computational Linguistics.

[7] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality, 2023.

[8] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, p. 665–673, 2020.

[9] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5553–5563, Online, July 2020. Association for Computational Linguistics.

[11] Xanh Ho, Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. A survey on measuring and mitigating reasoning shortcuts in machine reading comprehension. *Computing Research Repository*, Vol. arXiv:2209.01824, , 2022.

[12] Naoya Inoue, Harsh Trivedi, Steven Sinha, Niranjan Balasubramanian, and Kentaro Inui. Summarize-then-answer: Generating concise explanations for multi-hop reading comprehension. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6064–6080, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[13] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[14] Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2726–2736, Florence, Italy, July 2019. Association for Computational Linguistics.

[15] Nitish Joshi, Xiang Pan, and He He. Are all spurious features in natural language alike? an analysis through a causal lens. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9804–9817, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[16] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022.

[17] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual Amazon reviews corpus. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pp. 4563–4568, Online, November 2020. Association for Computational Linguistics.

[18] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[19] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions, 2020.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *Computing Research Repository*, Vol. arXiv:1907.11692, , 2019.

[21] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.

[22] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics.

[23] Kenton Murray and David Chiang. Correcting length bias in neural machine translation. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 212–223, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[24] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics.

[25] Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1938–1952, Online, November 2020. Association for Computational Linguistics.

[26] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[27] Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron Wallace. Combining feature and instance attribution to detect artifacts. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1934–1946, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[28] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In Malvina Nissim, Jonathan Berant, and Alessandro Lenci, editors, *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[29] 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. *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

[30] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In

Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics.

[31] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Beyond leaderboards: A survey of methods for revealing weaknesses in natural language inference data and models. *Computing Research Repository*, Vol. arXiv:2005.14709, , 2020.

[32] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2429–2438, Online, November 2020. Association for Computational Linguistics.

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[34] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics.

[35] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[37] Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. Efficient out-of-domain detection for sequence to sequence models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1430–1454, Toronto, Canada, July 2023. Association for Computational Linguistics.

[38] Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. Identifying and mitigating spurious correlations for improving robustness in NLP models. In

Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1719–1729, Seattle, United States, July 2022. Association for Computational Linguistics.

[39] Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and improve robustness in NLP models: A survey. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4569–4586, Seattle, United States, July 2022. Association for Computational Linguistics.

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[41] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[42] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

[43] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8653–8665, Toronto, Canada, July 2023. Association for Computational Linguistics.

[44] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball, 2023.