修士論文

言語モデルの信頼性向上にむけて -論理的同値性を理解できるか-

田中 健史朗

主指導教員 井之上 直也

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 9 月

Abstract

Large Language Models (LLMs) are language models trained on huge amounts of data, achieving near-human accuracy in various benchmarks such as inference and translation. However, it has been reported that LLMs may produce different outputs depending on the variations in input expressions, even if the meaning remains the same. For example, when querying an LLM about Anne Redpath's place of death, the input prompts "Anne Redpath's life ended in" and "Anne Redpath passed away in" may yield different results, with the LLM responding with "London" and "Edinburgh," respectively. This issue raises concerns about the reliability of LLMs. Despite their high performance on benchmarks, doubts remain about whether LLMs can effectively utilize this high performance, making it difficult to trust their responses. This inconsistency in responses is not limited to vocabulary-level paraphrasing, as previous research has pointed out, but also extends to logically equivalent expressions, as revealed by preliminary investigations in this study.

To address the inconsistency in LLM responses, existing research suggests that fine-tuning LLMs with data extended through paraphrasing could potentially enhance their robustness. Additionally, there is existing research on data augmentation using equivalence transformations in propositional logic. While the generation way of the logical formulas is relatively simple, LLMs using the augmented data have shown improved accuracy on benchmarks, indicating a possible enhancement in robustness. However, propositional logic represents a relatively simple form of logic found in natural language sentences, and there is a need to improve robustness concerning more expressive predicate logic.

Therefore, this study proposes the construction of a language model that is robust to logically equivalent expressions in predicate logic. To achieve this, we aim to use a data augmentation method that guarantees logical equivalence to learn that the extended sentences have the same meaning as the original ones. This will enable the language model to implicitly learn logically equivalent relationships, thereby enhancing the robustness of such expressions. Regarding the data augmentation method that guarantees logical equivalence, we propose a paraphrase method LET, which transforms transforming input sentences into logical formulas, performing equivalence transformations using first-order predicate logic, and then converting them back into natural language sentences.

LET consists of three components: T2L (Text-to-Logic translation), L2L (Logical form-to-Logical form translation), and L2T (Logical form-to-Text translation). These components are connected in a pipeline to achieve data augmentation with logical equivalence.

For T2L, translating natural language sentences into logical formulas has been a

long-standing area of research, particularly focused on first-order predicate logic. This translation enables the use of theorem provers like Coq, which can apply logically grounded approaches to a wide range of natural language processing tasks, such as RTE and calculating sentence similarity. Traditionally, Text-to-Logic translation has been explored using rule-based methods. However, due to the complexity of natural language, extending these rule-based approaches to practical applications is challenging. With the success of large language models, there has been growing interest in methods that use LLMs, which offer a balance of translation accuracy and flexibility. The emergence of powerful LLMs like GPT-3.5 and GPT-4 has led to a new paradigm where LLMs are employed to handle most of the translation tasks. Nonetheless, recently, ccg2lambda has been proposed for semantic analysis and reasoning from the perspective of mathematical logic. ccg2lambda is a system that consistently performs CCG parse tree generation (syntactic parsing) using Combinatory Categorial Grammar (CCG), logical formula generation (semantic analysis) using higher-order logic, and automated reasoning using the Coq. One advantage of ccg2lambda over LLM-based methods is that, with correct syntactic parsing, it can generate unique logical formulas that faithfully preserve the meaning of sentences. Therefore, this study prioritizes the reliability of the generated logical formulas and adopts ccg2lambda for T2L.

For L2L, while there are many possible transformations for logically equivalent expressions, many of these transformations may not result in natural expressions when converted back into the same natural language or may not produce noticeable differences. Therefore, this study focuses on equivalence transformations in first-order predicate logic and adopts rules that ensure the resulting expressions are different when translated back into natural language.

For L2T, translating logical formulas into natural language sentences (Logic-to-Text) has also been a focus of research, particularly in the context of first-order predicate logic. Since logical formulas can be viewed as a language that translates natural language sentences into a specific theoretical framework, Logic-to-Text has traditionally been considered a simpler task than Text-to-Logic and has been approached using rule-based methods. With the advent of LLMs, researchers have explored using these models for Logic-to-Text, leveraging their high text generation capabilities and flexibility, which rule-based methods lack. Although this area has not been as actively pursued as Text-to-Logic, there have been studies utilizing LLMs for Logic-to-Text translation. However, because ccg2lambda generates formulas in higher-order predicate logic, existing translation methods cannot be directly applied. Therefore, this study aims to construct an LLM-based model that takes logical formulas as input and outputs natural language sentences.

To construct a language model that is robust to logically equivalent expressions,

we will use LET to extend the RTE (Recognizing Textual Entailment) problem and fine-tune the language model with the extended data. This approach aims to enable the model to implicitly learn logically equivalent relationships and improve its robustness to such expressions.

In experiments, to effectively verify the robustness of LLMs to first-order predicate logic equivalence, we focused on FOLIO, a natural language RTE dataset for first-order predicate logic. We used LET to paraphrase the premise sentences in FOLIO and extend the RTE problem. By fine-tuning the large language model with the extended RTE problems, we aimed to construct an LLM robust to logically equivalent expressions. The results confirmed that using manually extended data, which represents the theoretical upper bound of LET, improved accuracy on the RTE problems while achieving robustness on Llama2 (7B). This demonstrates that the proposed method is effective in enhancing LLM robustness to logically equivalent expressions without compromising inference ability. However, the effectiveness of the model fine-tuned with automatically extended data using LET could not be confirmed, indicating that the paraphrasing accuracy of LET is still insufficient. Additionally, varying the parameter size of the base model did not validate the effectiveness of LET even at its theoretical upper bound, suggesting that further investigation into the effectiveness of LET is needed.

概要

大規模言語モデル (LLM) は大量のデータによって学習を行った言語モデルであり, 推論や翻訳など様々なベンチマークにおいて人間の精度に迫る高い性能を発揮している. しかし, その一方で, LLM には同じ意味の入力でも表現の違いにより出力結果を変えてしまう問題が報告されており, LLM の信頼性を揺るがしている. 入力頑健性が低い現状では, 仮に LLM がベンチマークで高い性能が実現していたとしても, LLM がその高い性能を正しく活用できるかどうかという観点で疑問が残り, LLM の回答を信頼することはできない. このような回答の非一貫性は, 既存研究で指摘されていた語彙レベルの言い換えだけに留まらず, 論理的に同値な表現についても発生していることが本研究の予備調査で明らかとなった.

LLM の回答の非一貫性に対して, 既存研究では Paraphrase により拡張したデータを用いて LLM をファインチューニングさせることが LLM の頑健性を高める可能性があることを示唆している. また命題論理の同値変形によるデータ拡張についても研究がされており, 同値変形元となる論理式の生成が簡易的ではあるものの, 拡張したデータを利用した LLM はベンチマーク上で正答率を向上させており, 頑健性についても向上している可能性が示されている. ただし, 命題論理は, 自然言語文に現れる論理の中では比較的簡素なものであり, より表現力の高い述語論理に関して頑健性を向上させる必要がある.

そこで本研究では, 述語論理上で同値な表現に対して頑強な言語モデルの構築を提案する. これを実現するために, 論理的な同値性を保証するデータ拡張手法を使い, 拡張した文が拡張元の文と同じ意味であることを学習させることで, 言語モデルが暗黙に論理的に同値な関係性を学習し, 論理的に同値な表現に対して頑健性を向上させることを目指す. また, 論理的な同値性を保証するデータ拡張手法については, 入力文を論理式へ変形し, 一階述語論理を活用した同値変形を行い自然言語文に戻すことで, 論理的な同値性を担保する Paraphrase 手法 LET を提案する.

LET は, 自然言語文を論理式に変換する T2L (Text-to-Logic translation), ある論理式を論理的に同値な論理式へと変形する L2L (Logical form-to-Logical form translation), 論理式から自然言語文へと変換する L2T (Logical form-to-Text translation) の 3 つのコンポーネントからなり, これらをパイプライン状に繋げることで論理的に同値なデータ拡張を実現する. T2L には, 論理式の信頼性を重視して, 形式統語論と形式意味論の見地から高階述語論理を活用して意味解析・推論を行う ccg2lambda を採用する. L2L には, 古典一階述語論理における同値変形に注目し, 自然言語文に戻したときに異なる表現となるような規則を採用する. L2T には, 高階述語論理に対応するため, 論理式を入力として自然言語文を出力する, LLM ベースのモデルを構築する.

そして, 論理的に同値な表現に対して頑強な言語モデルの構築では, LET を使って RTE 問題を拡張し, その拡張データをファインチューニングすることで, 言語モデルに暗黙に論理的に同値な関係性を学習させ, 論理的に同値な表現に対して頑健性を向上させる.

評価実験では, 一階述語論理の同値性に対する LLM の頑健性を効果的に確認するため, 一階述語論理の自然言語文の RTE データセットである FOLIO に着目し, LET を使って FOLIO の前提文を言い換え RTE 問題を拡張した. そして, 拡張した RTE 問題で大規模言語モデルをファインチューニングすることで, 論理的に同値な表現に対して頑健な LLM の構築を目指した. 結果として, LET の理論上限である, 手作業で拡張したデータを用いた場合には RTE 問題に対する正答率が向上させながら頑健性を獲得していることを確認し, 本手法が推論能力を損なわずに論理的に同値な表現に対する LLM の頑健性向上に有効な手法であることを示した. しかし, LET により自動的に拡張したデータを用いてファインチューニングしたモデルについては有効性が確認できず, LET 自体の言い換え精度は未だ不十分である. また, ベースモデルのパラメタサイズを変更すると LET の理論上限でも有効性が確認できず, LET の有効性についてさらなる調査が必要となる結果となった.

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

大規模言語モデル (LLM) は大量のデータによって学習を行った言語モデルであり, 推論や翻訳など様々なベンチマークにおいて人間の精度に迫る高い性能を発揮している. また, その高い応用力から, 対話チャットサービスである ChatGPT をはじめ, あらゆる用途における社会実装が進んでいる.

しかし, その一方で, LLM には同じ意味の入力でも表現の違いにより出力結果を変えてしまう問題が報告されており, LLM の信頼性を揺るがしている [11, 8]. 例えば，Anne Redpath の死没地を LLM に問い合わせる際，入力プロンプト “Anne Redpath’s life ended in” と “Anne Redpath passed away in” では，LLM の返す結果がそれぞれ London, Edinburgh と異なってしまう [8]. このように入力頑健性が低く, 論理的整合性がとれていない状況では, 仮に LLM がベンチマークで高い性能が実現していたとしても, LLM が学習した知識を使って正しく推論しているのかどうか, 実応用できるかという観点で疑問が残り, LLM の回答を信頼することはできない.

さらに, このような回答の非一貫性は, 語彙レベルの言い換え (例えば “X’s life ended in Y” と “X passed away in Y”) だけに留まらず, 論理的に同値な言い換え (例えば “All X are Y” と “If something is X, it is Y”) についても発生していることを本研究の予備調査で確認した. 図 1.1 に示すテキスト含意関係認識 (RTE) 問題 [10] における例を用いて説明する. RTE 問題とは, 前提から帰結を導くことができるかを True/False/Uncertain で判定する推論タスクである. この例では, 前提文 1, 3, 5, 6, 7, 8 より bird が kinetic かつ changing であることが読み取れるため, 正解は True となる. まず, Llama2(7B) [22] の 3-shot prompting [1]によりこの問題を解いたところ, 予測ラベルは Uncertain となった. 一方で, 7. All unstable things are kinetic. を論理的同値表現である 7’. If something is unstable, it is kinetic. へと変更すると, Llama2(7B) の予測ラベルは True へと変化した. このように論理的に同値な表現についても, LLM は入力の頑健性が低いことがわかっており, これは LLM の信頼性を損なうものである.

[1]プロンプト詳細は，4.2.3 節に記載している.

前提: 1. Everything is either big or small.
2. All big things are heavy.
3. All small things are light.
4. All heavy things are still.
5. All light things are unstable.
6. All unstable things are changing.
7. All unstable things are kinetic.
8. A bird is not both heavy and still.

帰結: A bird is kinetic or changing.

図 1.1: 一階述語論理 RTE [10] 問題の例.

1.2 目的

本研究では, 論理的に同値な表現に対して頑強な言語モデルの構築を目的とする. これを実現するために, 論理的な同値性を保証するデータ拡張手法を使い, 拡張した文が拡張元の文と同じ意味であることを学習させることで, 言語モデルが暗黙に論理的に同値な関係性を学習し, 論理的に同値な表現に対して頑健性を向上させることを目指す.

論理的な同値性を保証するデータ拡張手法については, 入力文を論理式へ変形し, 一階述語論理を活用した同値変形を行い自然言語文に戻すことで, 論理的な同値性を担保する Paraphrase 手法 LET (図 3.1) を提案する.

1.3 本論文の構成

本論文の構成は以下のとおりである. 第 2 章では, 本研究に関する既存研究と基礎事項について述べる. 第 3 章では, 本研究の提案手法を詳述する. まず, 論理的な同値性を担保したデータ拡張手法 LET について述べる. 次に LET を活用して, 論理的に同値な表現に対して頑強な言語モデルの構築手法を提案する. 第 4 章では, 評価実験について述べる. まず, データ拡張手法 LET の新規開発したコンポーネント L2T について述べる. その後, LET を活用して構築した言語モデルの, 論理的に同値な表現に対して頑強性について述べる. 第 5 章では, 本研究のまとめと今後の課題を述べる.

第2章　関連研究

2.1　言語モデルの信頼性

2.1.1　大規模言語モデルの課題

LLM の学習方法

現在主流となっている LLM は, BERT [6], GPT [3] などであるが, これらは Transformer [23] をベースとしている言語モデルである. Transformer は, 6 層の Transformer ブロックを持つエンコーダーと同じく 6 層の Transformer ブロックを持つデコーダーからなるエンコーダーデコーダーモデルであり, 並列計算を可能とした単純な機構としたことで, LSTM モデルなどの既存の言語モデルと比べるとパラメタ数が大幅に増え, 結果高い性能を持つ言語モデルとなった.

Transformer の学習は通例 2 段階で行われており, 1 段階目の学習は大量のコーパスを用いて, 与えられた単語列から次の単語を予測する学習を行う. これは事前学習と呼ばれる. 事前学習では, 知識の学習や推論能力の学習が行われている.2 段階目の学習は, 特定のタスクについてのデータセットを学習を行い, この学習はファインチューニングと呼ばれる. ファインチューニングでは, 回答様式の学習など LLM が学習した知識の使い方の学習が期待できる一方で, 新しい知識の学習はうまく機能しないことが知られている [7]. なお現在の公開されている大規模言語モデルのファインチューニングでは, より汎用的な性能を獲得できるよう, 後述する Instruction Tuning [25] が行われることが一般的である.

LLM が事前学習でパラメタ上に獲得した知識を活用する手法

前節で述べたように, LLM の知識の学習は次単語予測の学習の中で暗黙的に行われており, LLM がパラメタに学習した知識を効率的に活用する方法は自明ではなく, 手探り的に様々な手法が模索されている. まず注目されたのは, LLM への入力プロンプトを工夫する手法であった. ある問題に答えさせる前に, 問題と回答のサンプルを数個〜数十個与える Few-Shot Prompting という簡素なテクニックは, これまでタスクに適合させるためにパラメタの更新を必要としていたファインチューニングと比較しても同等の性能を発揮することを示した. また, これまで LLM が苦手としていた段階的な推論を必要とする問題に対して, Chain of Thought [26] と

いうテクニックでは, プロンプトに *Let's think step by step* という推論を段階的に行うことを促す表現を入れるだけで, LLM の多段推論能力が大幅に向上することがわかった.

ファインチューニングを工夫する手法についても革新的な手法が提案されている. Instruction Tuning [25] では, タスクをファインチューニングで学習させる際に, タスクの問題だけでなく, 指示 (Instruction) も受け取って回答できるようにファインチューニングすることで, 未知の指示に対しても適切に回答できるように学習を行っており, Few-Shot Prompting を使用しなくても同等の性能が発揮できることがわかっている.

さらに, LLM の内部状態を活用した手法が多く提案されるようになってきている [12, 1]. これは, LLM の知識の内部表現には, LLM が生成する文以上の情報が埋め込まれているらしいことがわかってきたためである. 例えば, LLM が事実に即した文と事実に反する文を生成する際の内部状態は, その文の真偽によって正しく分類できることがわかっている [12]. しかし, LLM は事実に反する文でもあっても, あたかも事実であるかのような文を生成してしまう問題 (ハルシネーションと呼ばれる) を抱えており, LLM は内部表現で認知している知識を十分に活用して文を生成できていないことがわかる.

2.1.2 言語モデルの入力に対する頑健性向上に向けて

1.1 節で述べたように, LLM には同じ意味の入力でも表現の違いにより出力結果を変えてしまうという入力の頑健性が低いという問題が報告されており, 本研究の予備調査では, 論理的に同値な言い換えにおいてもこのような頑健性の問題が発生していることを確認した. 先行研究は, ファインチューニングを工夫する手法, 具体的には, Paraphrase により拡張したデータを用いて LLM をファインチューニングさせる手法が, LLM の頑健性を改善する可能性を持つことを示唆する. 例えば, MetaMath [29] では, 数理問題の数字部分を変えずに, 問題文部分を Paraphrase しデータ拡張し, 拡張した問題をファインチューニングすることで, 性能が向上することがわかっている. これは数学的には同じ問題を様々な問題文で学習することで, 入力文に対して頑健性を獲得したと見なすことができる. また命題論理の同値変形によるデータ拡張についても研究がされており, 同値変形元となる論理式の生成が簡易的ではあるものの, 一定の成果を収めている [24]. そこで本研究は, 述語論理を対象として, 論理式を介したより精度の高い Paraphrase 手法を提案し, 提案手法により拡張したデータでファインチューニングを行い, 論理的に同値な表現に対して頑健性への影響を評価する.

2.2 言語モデルと論理学

2.2.1 論理学: 命題論理と一階述語論理

本節では, 本研究で扱う論理学の基礎的な事項を紹介する.

論理学について

推論 (inference) とは, 前提から結論を導き出す判断のことである. そして, 推論が正しいかどうかは, 妥当な数学的な証明 (proof) が与えられるかどうかで判断される. 論理学とは, このような証明を説明対象とした理論であり, 妥当な証明とは何かを予測, 説明することを目的としている [31]. 論理学の中でも数理論理学は, 証明とその構成要素を記号体系に構成するという方法論を採っており, 記号論理学や形式論理学とも呼ばれており, 論理学の主流の 1 つとなっている.

命題論理 (proposition logic)

推論で使われる前提や結論は, 真偽を問うことができる形式であると言える. 例えば, 「学生は化学が得意であるか, 国語が得意である.」という文は, 正しいか正しくないか, つまり真か偽かを明確に定めることができる. このような真偽を問うことができる形式を命題 (proposition) と呼ぶ. また, 上記の命題は, 「学生は化学が得意である.」, 「学生は国語が得意である.」という 2 つの命題に分割することができる, しかし逆に, それ以上に分割すると真偽を問う形式を保つことができない. このように, それ以上分割できない命題を原子命題 (要素命題) と呼ぶ.

前節では, 数理論理学は妥当な証明を予測, 説明するために, 記号を用いて体系化することを説明した. では証明はどのような記号で表し, どのように体系化できるだろうか. 命題論理では, 証明を命題として表現され, 命題は原子命題に分解される. そして, 原子命題の内容を問わずにその真偽のみに着目して命題の真偽を定めることで, 証明の妥当性を体系化している.

命題論理で扱う記号は, 以下の通りである.

実際に上記の命題論理の記号を用いて, 命題「学生が化学が得意であるならば, 学生は実験を喜ぶ.」を論理式にしてみると以下のように整理される.

命題 P	学生は化学が得意である.
命題 Q	学生は実験を喜ぶ.
論理式	$P \to Q$

ここで, 命題が真であることを 1, 偽であることを 0 で表し, この数字を真理値と呼ぶ. そして論理式の真理値は, 記号の定義によって決定する. 例えば, 先ほどの

表 2.1: 命題論理で扱う記号

論理式を表す文	名称	論理式
p ということはない	否定	$\neg p$
p かつ q	連言	$p \wedge q$
p または q	選言	$p \vee q$
p ならば q	含意	$p \to q$
p のとき, またそのときのみ q	同値	$p \leftrightarrow q$
真である	真	$\top$
偽である	偽	$\bot$

$P \to Q$ という命題の真理値は, 含意記号の定義により, 以下のように機械的に決定する.

表 2.2: 真理値表

P	Q	$P \to Q$
1	1	1
1	0	0
0	1	1
0	0	1

つまり, $P \to Q$ という命題の真理値は, P が真で Q が偽のときは偽となり, それ以外の場合は真となる.

一階述語論理 (first-order predicate logic)

ここで次の推論を考えてみる.

前提: 1. すべての学生は化学が得意である.
　　2. 太郎は学生である.
帰結: 太郎は化学が得意である.

この推論を命題論理の論理式にすると以下のようになる.

直感的に考えると, 命題 P, Q が真であれば, 命題 R も必ず真であるように思われる. しかし, 命題 P,Q,R は, 命題論理上では単に別々の要素命題に過ぎず, 命題論理の論理式では直感的に正しい関係性を表現することはできない.

そこで一階述語論理は, 命題論理を拡張して, 命題の内部を表現できるようにした. 一階述語論理では, 要素命題を名前 (name) と述語 (predicate) に分割する. 例えば, 要素命題「太郎は学生である」は, 名前「太郎」と述語「x は学生である」に

命題 P	すべての学生は化学が得意である.
命題 Q	太郎は学生である.
命題 R	太郎は化学が得意である.
論理式	$P, Q \to R$

分割され, 論理式は $\_student(Taro)$ となる. 述語は, 名前の代わりに特定の条件を持つ x を用いて論理式を作ることができ, 「すべての x について」という条件を全称量化と呼び, 「ある x について」という条件を存在量化と呼ぶ. 記号はそれぞれ $\forall x$, $\exists x$ である.

先程の推論を一階述語論理の論理式にすると以下のようになる.

名前 Taro	太郎
述語 $\_student(x)$	x は学生である
述語 $\_good(x)$	x は化学が得意である
論理式	$\forall x(\_student(x) \to \_good(x)), \_student(Taro) \to \_good(Taro)$

一階述語論理の論理式の真理値について説明する. 一階述語論理の論理式の真理値は, 命題論理のように機械的な手続きでは決定できず, 以下の推論規則を導入し自然演繹法と呼ばれる手法で決定する.

表 2.3: 一階述語論理で導入される推論規則

全称消去 ($\forall\_$)	$\forall x F(x) \to F(u)$	(u は任意の個体)
	$\forall x F(x) \to F(a)$	(a は特定の個体)
全称導入 ($\forall.$)	$F(u) \to \forall x F(x)$	(u は任意の個体)
存在消去 ($\exists\_$)	$\exists x F(x) \to F(a)$	(a は特定の個体)
存在導入 ($\exists.$)	$F(u) \to \exists x F(x)$	(u は任意の個体)
	$F(a) \to \exists x F(x)$	(a は特定の個体)

最後に, 先程作った一階述語論理の推論が妥当であるかを自然演繹法を使って確認する.

(1)	$\forall x(\_student(x) \rightarrow \_good(x))$	前提
(2)	$\_student(Taro)$	前提
(3)	$\_student(Taro) \rightarrow \_good(Taro)$	(1)$\forall$_
(4)	$\_good(Taro)$	(2)(3)

よって, 推論は妥当であることが確認できた.

2.2.2 自然言語文と論理式間の翻訳

自然言語文から論理式への翻訳

自然言語文から論理式への翻訳 (Text-to-Logic) は, 一階述語論理を中心に古くから盛んに研究されてきた. 論理式に変換し, Coq のような定理証明系を活用することで, 含意関係認識 [17] や文類似度計算 [27] など, 幅広い自然言語処理のタスクで論理的に裏打ちされた応用が見込めるからである.

伝統的に, Text-to-Logic はルールベースの手法 [30, 2] を使って研究されてきた. 自然言語文は複雑であるため, ルールベースの手法は実応用に拡張することが困難である. そこで大規模言語モデルの成功に伴って, 翻訳精度と柔軟性を兼ね備えた LLM を用いた手法に関心が高まっている [14, 9]. さらに, GPT-3.5 や GPT-4 のような強力な LLM が出現したことで, LLM を使って翻訳タスクの大部分を実行するという新しいパラダイムが生まれている [28].

しかし, 近年, 数理論理学の観点から意味解析・推論を行う ccg2lambda [16] が提案されている. ccg2lambda は, 組合せ範疇文法 (Combinatory Categorial Grammar, CCG) [21] による CCG 構文木生成 (構文解析) , 高階論理による論理式生成 (意味解析) , 証明支援系 Coq による自動推論までを一貫して行うシステムである. ccg2lambda が LLM ベースの手法と比較したときに優位な点として, ccg2lambda では構文解析が正しければ論理式は一意に定まるため, 文の意味を忠実に保持した論理式が生成されることが期待できる. 本研究では, 生成される論理式の信頼性を重視し, 論理式の生成に ccg2lambda を採用している.

論理式から自然言語文への翻訳

論理式から自然言語文への翻訳 (Logic-to-Text) に関しても, 一階述語論理を中心に研究が進んできた. 特に論理式自体が自然言語文を特定の理論で変換した言語であるという側面があることから, Text-to-Logic よりは簡単なタスクとして, ルールベースの手法で取り組まれていた [18, 20]. LLM の出現後には, その高い文章生成能力とルールベースにはない柔軟性を活用しようと, Text-to-Logic よりは活発では

ないものの, LLM を利用した Logic-to-Text の研究が行われてきた [15]. Logic-to-Text の研究において重要な課題の一つは, 翻訳後の自然言語文の評価指標である. Logic-to-Text の翻訳評価においては, 機械翻訳の評価で用いられてきた BLUE [19] を使うことが一般的である. しかし, 論理式の翻訳評価という観点では, BLUE は構文や意味の評価を行わないため, 不十分な指標である. そのため, 翻訳後の自然言語文を正答である自然言語文との RTE タスクとして機械的に翻訳評価する手法 [15] や, 人による翻訳評価 [4] などが提案されている.

第3章　提案手法

本研究では, 論理的な同値性を担保したデータ拡張を使いRTE問題を拡張し, 拡張したRTE問題をファインチューニングすることで, 論理的に同値な表現に対して頑強な言語モデル構築することを提案する. まず, 3.1節では, 論理的な同値性を保証するデータ拡張手法を提案する. そして, 3.2節では, 3.1節で提案したデータ拡張手法を活用して, 論理的に同値な表現に対して頑強な言語モデルの構築手法を提案する.

3.1　LET (Logically Equivalent Transformation)

本節では, 新たなデータ拡張手法であるLETを述べる. LETは自然言語文を論理式に変換し, 論理的に同値な論理式に変形したあと, これを自然言語文に戻すことで, 論理的な同値関係を担保するParaphrase手法である (図3.1) .

LETは, 自然言語文を論理式に変換するT2L (Text-to-Logic translation), ある論理式を論理的に同値な論理式へと変形するL2L (Logical form-to-Logical form translation), 論理式から自然言語文へと変換するL2T (Logical form-to-Text translation) の3つのコンポーネントからなり, これらをパイプライン状に繋げることで論理的に同値なデータ拡張を実現する.

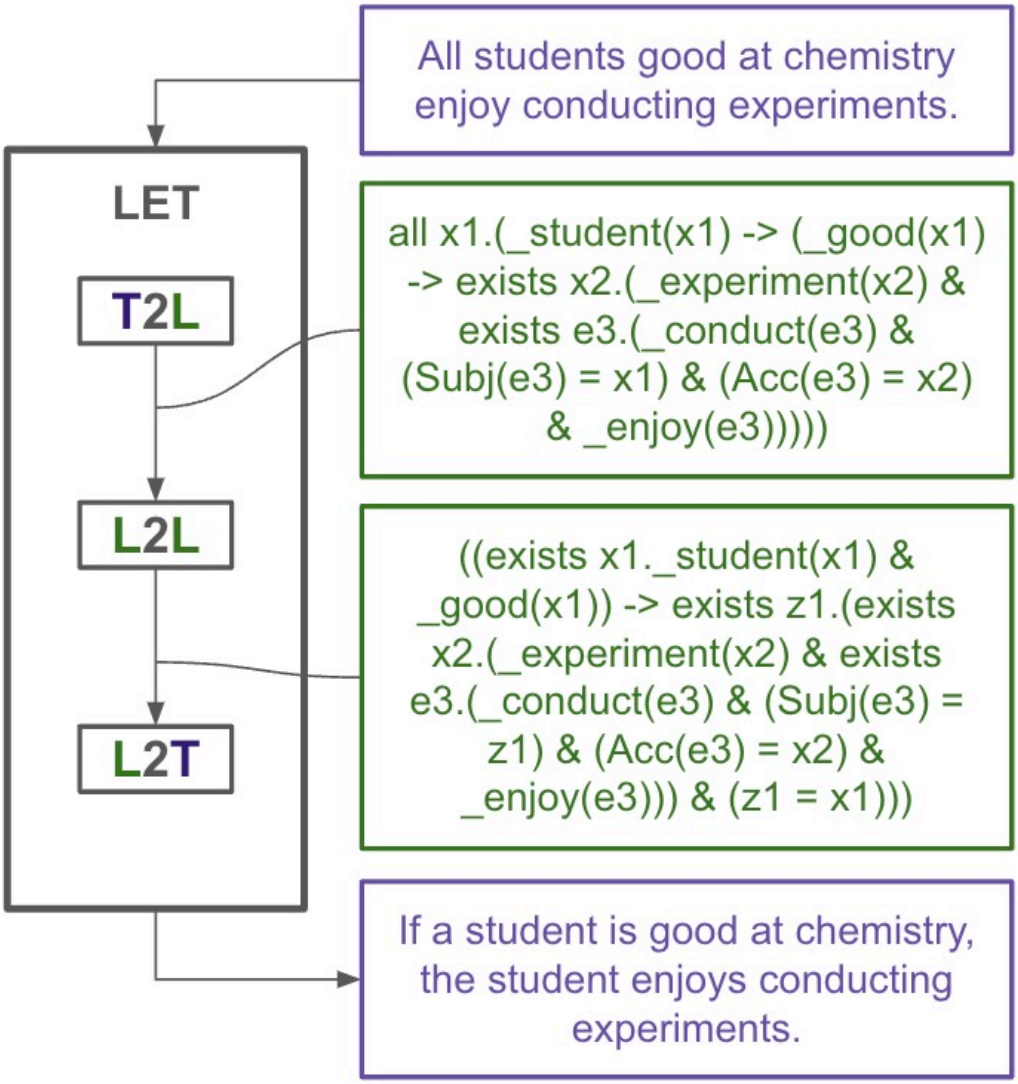

図 3.1: LET の概要．LET は入力文を論理式へ変形し, 一階述語論理を活用した同値変形を行い自然言語文に戻すことで, 論理的な同値性を担保する Paraphrase 手法である．

3.1.1 T2L: Text-to-Logical form translation

T2L では, 自然言語文を論理式表現へ変換する. 本研究では, T2L に ccg2lambda を採用する.

ccg2lambda [16] とは, 組合せ範疇文法 (Combinatory Categorial Grammar, CCG) [21] による CCG 構文木生成 (CCG 解析) , 高階述語論理による論理式生成 (意味合成) , 証明支援系 Coq による証明 (定理証明) までを一貫して行うシステムである (図 3.2). ccg2lambda は, 深い意味解析と推論を可能とするシステムとして, 含意関係認識 [17] や文類似度計算 [27], また複合語解析などの複雑な意味解析を必要とする医療分野 [33] や金融分野のテキスト分析 [32] などにも活用されている.

本研究では自然言語文から論理式を生成するために, CCG 解析と意味合成部分のみを使用する.

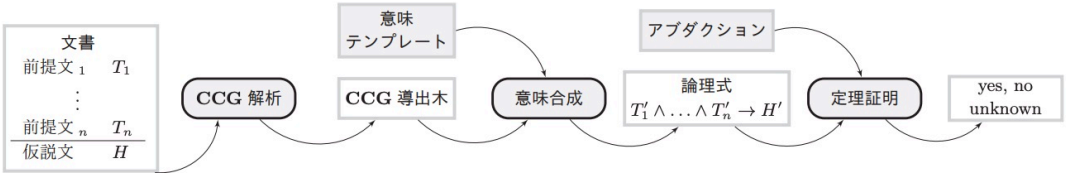

図 3.2: ccg2lambda の構成 [32].

3.1.2 L2L: Logical form-to-Logical form translation

L2L では, 表 3.1 の規則を使い, 前段の T2L で得られた論理式と論理的に同値な論理式を得る.

表 3.1: 古典一階述語論理における同値変形

type1	$\neg\forall y A \leftrightarrow \exists y \neg A$
type2	$\neg\exists y A \leftrightarrow \forall y \neg A$
type3	$(\forall y A \rightarrow B) \leftrightarrow \exists z (A[z/y] \rightarrow B)$
type4	$(\exists y A \rightarrow B) \leftrightarrow \forall z (A[z/y] \rightarrow B)$

論理的に同値な式変形は多数存在するが, その多くは自然言語文に戻したときに自然な表現ではなかったり, 差がなかったりする. 例えば, $\forall x(A \rightarrow B(x))$ と $A \rightarrow \forall x B(x)$ は同値な式であるが, 自然言語文にするとどちらも,「A であれば, どんな x でも B である」という表現となる.

今回は自然言語文の Paraphrase としての同値変形を行うため, 自然言語文に戻したときに異なる表現となることが期待できる規則として, 異なる量化子への変形となる表 3.1 の規則を採用した. なお type3 の右→左に関しては, type3 の右がデータセット内に期待できないため, また type3, 4 の左→右の変換に関しては,

ccg2lambda において未解決となっている照応解析が必要なため, 今回は拡張対象外としている.

3.1.3 L2T: Logical form-to-Text translation

L2T では, L2L で得られた変形後の論理式を自然言語文に戻す. T2L が高階述語論理に対応しているため, それに伴い L2T も高階述語論理に対応する必要があるが, 該当する既存研究は研究されていない. そのため, 論理式を入力として自然言語文を出力する, LLM ベースのモデルを構築する.

3.2 論理的に同値な表現に対して頑強な言語モデルの構築

3.1 節で提案したデータ拡張手法を使って拡張した文が拡張元の文と同じ意味であることを学習させることで, 言語モデルが暗黙に論理的に同値な関係性を学習し, 論理的に同値な表現に対して頑健性が向上することを目指す.

具体的には, RTE 問題を用いて, すべての前提文に対して LET によって論理的に同値な前提文に言い換えを行う. その後, 言い換え元の前提文と言い換えた前提文をすべての組合せで RTE 問題を拡張する. そして拡張した RTE 問題すべてを使って LLM のファインチューニングする (図 3.3).

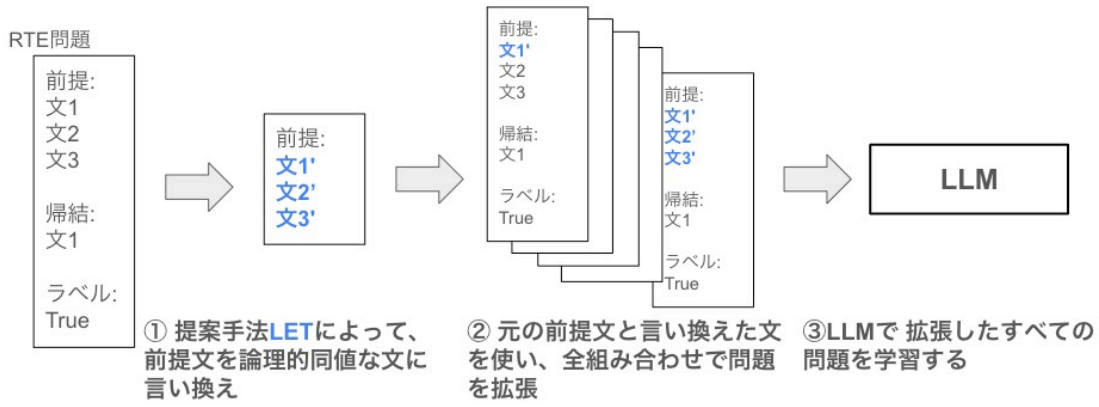

図 3.3: 論理的に同値な表現に対して頑強な言語モデル構築の概要.

第4章　実験・評価

本章では, まず 4.1 節で, LET のコンポーネントである L2T の評価を行い, 十分な精度で論理式を自然言語表現に変換できることを確認する. 次に 4.2 節で, 論理的に同値なデータ拡張の効果を分析するため, 提案手法である LET の評価を行う.

4.1　L2T 評価実験

4.1.1　データセット

L2T は, 論理式を入力として英文を出力するため, L2T のデータセットとしては, 入力となる論理式と正解データである英文が必要となる. そこで学習データとして, FOLIO を拡張した MALLS [28] の train データ全件より英文を正解データとして取得し, T2L として採用した ccg2lambda によって論理式へ変換した.

その結果, 26,940 組の (英文, 論理式) のペアが得られた.[1] 評価用のデータとしては, FOLIO の train データより正解データとして 50 文の英文を取得し, 同様に ccg2lambda により論理式へ変換した.

4.1.2　モデル

本実験では, L2T のモデルとして GPT-4 (gpt-4-1106-preview), GPT-3.5 (gpt-3.5-turbo-1106) 及び Llama2 (7B) [22] を利用する. GPT-4 及び GPT-3.5 は OpenAI API 経由で利用している (temperature=0). Llama2 (7B) については文献 [28] を参考に Supervised Fine-Tuning (SFT) [22] を行った. また計算効率をあげるため, QLoRA[5] を用いてファインチューニングしている. ファインチューニング時のプロンプトは以下の通りである.

[1] MALLS は FOLIO のデータと同じデータが含まれていないことが確認されている. [28]

L2T SFT プロンプト

```
### Instruction:
Translate the following nltk's higher-order logical form text to
English:
### higher-order logical form text:
{logical_form}
### English:
{english}
```

4.1.3 評価指標

L2T の評価手法としては, 2.2.2 節で見たように「論理式→英文」の評価手法は確立されていないため, 「論理式の作成元の英文 → 論理式 → 論理式から作成された英文」を, 論理式を介した英文からの英文への翻訳とみなし, 翻訳評価のプロンプト手法である GEMBA [13] を修正して L2T を評価した. GEMBA は, GPT3.5 以上の LLM を活用した翻訳評価の機械的な評価手法であり, 人間の評価との一致率において State-of-the-Art を達成している評価手法である. 修正内容としては, 「x1」のような論理式に特有の表現をそのままにせず, 正しく自然言語文として訳せているかどうかという観点を追加している. 修正後のプロンプトは以下の通りである.

L2T GEMBA 修正プロンプト

```
Score the following translation from English to English with
respect to  the human reference with one to five stars.  Where
one star means "Nonsense/No meaning preserved",  two stars mean
"Some meaning preserved, some variables like x1 remained, but
not understandable", three stars mean "Some meaning preserved,
few variables like x1 remained, and understandable",  four stars
mean "Most meaning preserved with possibly few grammar mistakes
and no variables like x1", and five stars mean "Perfect meaning
and grammar".
English source:  {source}
English translation:  {translated}
Stars:
```

表 4.1: L2T GEMBA 修正プロンプトによる評価 (5 点満点)

		スコア平均	スコア分散
GPT-3.5	zero-shot	2.80	0.69
	2-shot	3.58	0.75
GPT-4	zero-shot	3.10	0.83
	2-shot	3.58	0.73
Llama2 7B	SFT	**3.78**	1.05

実験結果

表 4.1 に示す通り, Llama2 (7B, SFT) が最も良い結果となった. GPT モデルの 2-shot が Llama2 (7B, SFT) を下回った要因としては, T2L で生成される論理式に現れる文法が 2-shot では十分に学習されなかったためだと考えられる. なお, GPT4 (2-shot) と Llama2 (7B, SFT) が 0.2 ポイントの差に留まっている原因としては, T2L によって誤った論理式のもとに作成された英文が低スコアになっているためであり, Llama2 (7B, SFT) のスコア分散が一番高いことからも, T2L が正しく論理式を生成した場合でも Llama2 (7B, SFT) の性能が一番高いことが期待できる. このため, 4.2 節の実験では, 最も性能の高かった Llama2 (7B, SFT) を L2T として採用する.

4.2 LET 評価実験

4.2.1 データセット

一階述語論理の同値性に対する LLM の頑健性を効果的に確認するため, 一階述語論理の自然言語文の RTE データセットである FOLIO [10] を用いる. FOLIO の RTE 問題は, 数件の前提文から帰結文が帰結されるか True/False/Uncertain で答える 3 値分類の問題であり, 前提文と帰結文に多くの量化表現を含んでいる. そのため, 他のデータセットよりもデータを拡張できることが予想され, LET の効果を評価しやすいことを期待して採用している. また FOLIO の train/validation データの件数はそれぞれ 1,004/204 件である. モデルの学習は, train データと後述の拡張データ, 評価に関しても validation データと後述の拡張データ上で行った.

4.2.2 データ拡張

本実験では, 提案手法の有効性を検証するため, 次の 3 種類のデータ拡張を行った.
LET: FOLIO の train データの前提文に使われているユニークな文 1,644 件に対して, LET による Paraphrase を行い, 拡張された前提文の全組み合わせでデータ拡

表 4.2: LET による FOLIO の拡張件数

		前提文 (増加数)	RTE 問題 (増加数)
LET	train	1,991 (+347)	4,651 (+3,647)
	valid	451 (+86)	774 (+570)
LET (理論上限)	train	1,978 (+335)	4,604 (+3,600)
	valid	446 (+81)	729 (+525)

表 4.3: LET による FOLIO のラベルごとの RTE 問題拡張件数

		True	False	Uncertain
train	FOLIO	388	289	324
	LET	1,687(+1,299)	1,523(+1,234)	1,441(+1,096)
	LET (理論上限)	1,663(+1,275)	1,510(+1,221)	1,431(+1,086)
valid	FOLIO	72	62	69
	LET (理論上限)	189(+117)	162(+90)	174(+105)

張を実施した. LET による拡張件数は表 4.2, ラベルごとの RTE 問題拡張件数は表 4.3, 同値言い換えの対象となる type ごとの件数は表 4.4 の通りである.

LET (理論上限) : LET の理論上限を測るため, 人手によって論理的な同値関係を担保した Paraphrase も行った（以下 LET (理論上限) と呼称）. 手順としては, 以下の通りである.

1. T2L で得た論理式のうち変形対象の論理式について, 否定または量化が正しく解析されているか精査する.
2. 正しく解析されている変形対象の論理式について, L2L を使い, 同値な論理式へ変形する.
3. 同値な論理式について, L2T を使わず手作業で自然言語文へ変換する.

Paraphrase (GPT-3.5, GPT-4): 述語論理の同値言い換えを学習することで LLM の論理的に同値な表現に対しての頑健性が向上することを確認するため, GPT-4 (gpt-4-1106-preview) , または GPT-3.5 (gpt-3.5-turbo-1106) を使った prompt による Paraphrase, すなわち LLM による論理的に同値とは限らない言い換えによる前提文の拡張を行った. プロンプトは以下に示す. また, LLM の言い換えによる前提文のデータ拡張は LET (理論上限) が拡張した件数分行ったため, 拡張された RTE 問題の件数も LET (理論上限) と同数である.

表 4.4: 同値変換 type ごとの FOLIO train データの拡張件数

	LET	LET (理論上限)
type1 (左→右)	2	0
type1 (右→左)	0	0
type2 (左→右)	23	12
type2 (右→左)	0	0
type4 (右→左)	322	322

Paraphrase プロンプト

```
Please read the following sentence and generate a new sentence
with the same meaning using a different expression.  Do not
provide any information other than the sentence.
{sentence}
```

4.2.3 モデル

本実験では, 計算資源の制約から Llama2 (7B, 13B) [22] を使用して, 各 Paraphrase 手法によって拡張したデータセットを使い Supervised Fine-Tuning (SFT) [22] を行った. また計算効率をあげるため, QLoRA[5] によってファインチューニングしている. プロンプトは以下の通りである.

FOLIO SFT プロンプト

```
### Instruction:
Based on the facts below, determine whether the following
statement is true, false, or uncertain.  Do not provide any
information other than the answer.:
### Facts:
{premises}
### Statement:
{conclusion}
### Answer:
{answer}
```

参考のためにLlama2 (7B) のpretrainedモデル, またGPT-4 (gpt-4-1106-preview) と GPT-3.5 (gpt-3.5-turbo-1106) を OpenAI API 経由で用い (temperature=0), 3-shot で評価している. プロンプトは以下の通りである.

3-shot プロンプト

```
### Instruction:  Based on the facts below, determine whether the
following statement is true, false, or uncertain.  Do not provide any
information other than the answer.:

### Facts:
'If something is a plant, then it is not a cute animal.
Simeng:  All plants are not cute animals.
All flowers are plants.
Every kitten is a cute animal.
If something is grown in a garden, then it is a flower.
Piper is a kitten or a cute animal.'
### Statement:
'Piper was not grown in a garden.'
### Answer:
True

### Facts:
'LanguageA is a universal language.
If a universal language exists, then for every two people if they
both know the same universal language they can communicate.
Katya cannot communicate with Danil.
Katya knows LanguageA.'
### Statement:
'Danil knows LanguageA.'
### Answer:
False

### Facts:
'Animals who need large territory travel far.
Every animal that eats a lot needs a large territory.
If something is a big animal, then it will eat a lot.
Bears are big animals.
Larry is a big animal.'
### Statement:
'Larry is a bear.'
### Answer:
Uncertain

### Facts:
{premises}
### Statement:
{conclusion}
### Answer:
```

4.2.4 評価指標

本実験では, LLM の論理的に同値な表現に対しての頑健性を向上させることを目的としている. しかし, LLM の信頼性を向上させるという観点から, 回答の頑健性が高いだけではなく, その回答が正しい回答であることが期待される. よって, LET (理論上限) で拡張した FOLIO の validation データを使って, 頑健性と正答率の 2 つの観点から評価を行う.

頑健性 LLM が, 論理的に同値に拡張された問題に対して, 拡張元の問題と同じ回答をしたかどうかを評価する. そこで, 以下に定める回答一致率を指標とする.

$$\text{回答一致率} = \frac{LLM \text{が拡張元の問題と同じ回答をした数}}{\text{拡張問題数}}$$

正答率 LLM が RTE 問題に対して正しい回答をしたかどうかを評価する.

拡張問題の正答に関して, LET (理論上限) で拡張された問題は, 論理的には拡張元と同じ問題であるため, 当然正答も同じである. また RTE 問題は 3 択問題であるため, ラベルごとの正答率も評価する.

4.2.5 実験結果

述語論理の同値変形を学習することは LLM の頑健性を向上させるか?

頑健性の評価結果について表 4.5 に示す.

Llama2 (7B) について, Paraphrase (GPT-3.5, GPT-4) による学習モデルが通常の train データを使った学習モデルよりも回答一致率を落としている一方で, LET (理論上限) はスコアを上げている. この結果は, LET (理論上限) によるデータ拡張が, 既存手法と同程度に論理的に同値な表現に対する LLM の頑健性を向上させる可能性があることを意味している.

Llama2 (13B) について, 回答一致率は一定して高く, また GPT3.5 や GPT4 の頑健性も同程度に高いことから, 頑健性は言語モデルのパラメータサイズに大きく依存する可能性がある.

述語論理の同値変形を学習することは述語論理を扱う RTE 問題の正答率を向上させるか?

正答率の詳細評価結果について FOLIO validation データによる評価を表 4.6, FOLIO/validation データを LET (理論上限) で拡張したデータによる評価を表 4.7 に示す.

Llama2 (7B) について, Paraphrase (GPT-3.5, GPT-4) による学習モデルが通常の train データを使った学習モデルよりもスコアを落としている一方で, LET (理

表 4.5: LET (理論上限) で拡張された FOLIO/validation データによる頑健性評価

		回答一致率	正解率		
			All	FOLIO	LET 拡張
Llama2 7B	3-shot	-	-	0.36	-
	SFT	0.74	0.43	0.44	0.43
	+ Paraphrase (GPT-3.5)	**0.86**	0.33	0.37	0.30
	+ Paraphrase (GPT-4)	**0.86**	0.44	0.45	0.43
	+ **LET**	0.75	0.40	0.38	0.40
	+ **LET (理論上限)**	**0.86**	**0.52**	**0.55**	**0.52**
Llama2 13B	3-shot	**0.97**	0.39	0.43	0.37
	SFT	0.95	**0.57**	**0.62**	**0.55**
	+ **LET (理論上限)**	0.92	0.40	0.42	0.39
GPT-3.5	3-shot	0.91	0.41	0.47	0.38
GPT-4	3-shot	0.90	0.51	0.64	0.45

論上限) は最も良いスコアとなっており, 8 ポイント向上している. また, 回答ラベルごとの性能を確認すると, 通常の train データや Paraphrase による学習モデルが True に偏って回答しているが, LET (理論上限) ではバランスよく回答しており, 述語論理の同値性について効果的な学習をしていることが期待できる. この結果は, LET (理論上限) によるデータ拡張を学習した LLM は, 述語論理を扱う RTE 問題への推論能力を維持/強化していることを意味している.

Llama2 (13B) について, LET (理論上限) は通常の train データを使った学習モデルよりもスコアを落としている. この原因は True と回答しやすいモデルとなっているためであり, LET の有効性は言語モデルのパラメータサイズに大きく依存する可能性がある.

LET と LET (理論上限) の性能差の要因は何か?

Llama2 (7B) について, LET の正答率は, LET (理論上限) だけでなく, SFT の手法の中でも最低の正答率となっている. これは, T2L に起因している可能性がある. T2L の結果をサンプルして手動で評価したところ誤った論理式が生成されているのを確認している. T2L は LET の最初のコンポーネントで, かつ L2T の学習データを生成するコンポーネントでもあるため, T2L の誤りは LET の正答率に大きく影響している.

表 4.6: FOLIO validation データによる正答率評価

		All	True			False			Uncertain		
		Pre.	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama2 7B	3-shot	0.36	0.36	0.92	0.52	0.43	0.05	0.09	0.2	0.01	0.03
	SFT	0.51	0.49	0.93	**0.64**	0.55	0.39	0.45	0.62	0.19	0.29
	+ Paraphrase (GPT-3.5)	0.37	0.35	0.75	0.48	0.27	0.09	0.14	0.53	0.23	0.32
	+ Paraphrase (GPT-4)	0.45	0.44	0.86	0.58	0.47	0.15	0.23	0.48	0.28	0.36
	+ **LET**	0.38	0.36	0.12	0.18	0.35	0.38	0.36	0.40	0.65	0.49
	+ **LET (理論上限)**	**0.55**	0.64	0.40	0.49	0.47	0.79	**0.59**	0.62	0.49	**0.55**
Llama2 13B	3-shot	0.43	0.41	0.93	0.57	0.56	0.33	0.42	0.0	0.0	0.0
	SFT	**0.62**	0.63	0.86	**0.73**	0.55	0.74	**0.63**	0.82	0.27	**0.41**
	+ **LET (理論上限)**	0.42	0.39	0.91	0.54	0.59	0.30	0.40	0.66	0.02	0.05
GPT-3.5	3-shot	0.47	0.54	0.62	0.58	0.42	0.80	0.55	0.0	0.0	0.0
GPT-4	3-shot	0.64	0.77	0.70	0.73	0.64	0.66	0.65	0.53	0.56	0.54

表 4.7: FOLIO/validation データを LET (理論上限) で拡張したデータによる正答率評価

		All	True			False			Uncertain		
		Pre.	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama2 7B	SFT	0.51	0.49	0.93	**0.64**	0.55	0.39	0.45	0.62	0.19	0.29
	+ Paraphrase (GPT-3.5)	0.31	0.30	0.66	0.42	0.21	0.09	0.13	0.55	0.13	0.22
	+ Paraphrase (GPT-4)	0.43	0.40	0.82	0.54	0.45	0.08	0.14	0.54	0.35	0.42
	+ **LET**	0.40	0.21	0.05	0.09	0.35	0.63	0.45	0.54	0.57	0.56
	+ **LET (理論上限)**	**0.52**	0.53	0.32	0.40	0.44	0.76	**0.56**	0.67	0.50	**0.57**
Llama2 13B	3-shot	0.37	0.37	0.96	0.54	0.45	0.09	0.15	0.0	0.0	0.0
	SFT	**0.55**	0.59	0.82	**0.69**	0.46	0.68	**0.55**	0.96	0.13	0.24
	+ **LET (理論上限)**	0.39	0.36	0.88	0.52	0.42	0.06	0.11	0.63	0.16	**0.26**
GPT-3.5	3-shot	0.38	0.42	0.40	0.41	0.36	0.77	0.49	0.01	0.01	0.03
GPT-4	3-shot	0.45	0.58	0.38	0.46	0.45	0.44	0.45	0.39	0.58	0.46

第5章　おわりに

5.1　本論文のまとめ

本研究では, 論理的な同値性を保証した言い換え手法LETを提案し, LETを使ってFOLIOの前提文を拡張し大規模言語モデルをファインチューニングすることで, 論理的に同値な表現に対して頑健なLLMの構築を目指した. 結果として, LETの理論上限である, 手作業で拡張したデータを用いた場合にはRTE問題に対する正答率が向上することを確認し, 本手法が推論能力を損なわずに論理的に同値な表現に対するLLMの頑健性向上に有効な手法であることを示した. しかし, LETにより自動的に拡張したデータを用いてファインチューニングしたモデルについては有効性が確認できず, LET自体の言い換え精度は未だ不十分である. また, ベースモデルのパラメタサイズを変更するとLETの理論上限でも有効性が確認できず, LETの有効性についてさらなる調査が必要となる結果となった.

本研究の概要は, 次の通りである.

- 自然言語文を論理式に変換し, 述語論理の同値関係を活用することで, 論理的な同値性が担保することを目指した新たなParaphrase手法 LET (図 3.1) を提案した（3章）.
- 一階述語論理のRTEデータセットであるFOLIO [10] において，LETによって拡張されたRTE問題をLlama2 (7B) で学習すると正答率が低下したが, 人手で論理的同値性が担保されたLETによって拡張されたRTE問題を学習すると正答率が向上することから, LETによるデータ拡張が, 推論能力を損なわずに論理的同値表現に対して頑健性なモデルにつながる可能性を示した（4）.

5.2 今後の課題

本研究の今後の課題を以下にまとめる.

LET の性能向上

本論文では, LET と合わせて, LET の理論値として LET (理論上限) を構築した. 結果, LET は LET (理論上限) に大きく劣ることがわかった. 先述した通り, この原因は, LET の最初のコンポーネントである T2L による変換ミスが大きく影響している. そのため, 本研究では T2L として採用を見送った, LLM ベースの自然言語文から一階述語論理式へ変換する LOGICLLAMA [28] などを使い, T2L の変換精度の向上を図りたい. また, T2L が変換した論理式は L2T の学習データとして使用しているため, T2L の変換精度向上は, L2T の変換精度の向上にもつながることが期待でき, LET の性能向上が期待できる.

LET 評価データセットの拡充

本論文では, LET の性能が理論値よりも大幅に劣っており, 頑健性の評価実験としては FOLIO による評価のみで留まってしまった. しかし, LET によるデータ拡張が真に LLM の論理的な同値性への頑健性を高めるのであれば, 感情分析データセットなどの様々な NLP データセットでデータ拡張した場合でも, 正答率を低下させずに頑健性は向上することが期待される. そのため, 先述の LET の性能向上が達成された後に, 他のデータセットでの頑健性評価を行いたい.

言語モデルごとの LET がもたらす頑健性に関するさらなる評価実験

本論文では, Llama2 の 7B モデルをベースモデルとして扱い, LET で拡張したデータを学習したモデルは, 理論値上は RTE 問題の推論能力を維持したまま頑健性を向上させるという結果を得た. しかし, Llama2 の 13B モデルをベースモデルとして LET で拡張したデータを学習したモデルは, 通常の SFT モデルに大きく劣る結果となり, LET の効果が言語モデルのパラメタサイズに影響される可能性が示された. そのため, 様々なパラメタサイズの大規模言語モデルをベースモデルとして LET の評価実験を行う必要がある.

謝辞

本研究を進めるにあたり多大なるご指導をいただきました井之上直也准教授に深厚なる謝意を表します. 数理論理学の知見や ccg2lambda の活用方法につきまして, お茶の水女子大学所属の戸次大介教授と同研究室所属の高橋優太特任助教に有益な助言をいただきました．皆様に感謝いたします．最後に, 研究/生活全般にわたりお世話になりました井之上研究室 (RebelsNLU) の皆様に感謝の意を示します.

参考文献

[1] A. Azaria and T. Mitchell. The Internal State of an LLM Knows When It's Lying. arXiv.

[2] J. Bos and K. Markert. Recognising textual entailment with logical inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pp. 628–635. Association for Computational Linguistics.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. teusz Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[4] E. Calò, E. Van Der Werf, A. Gatt, and K. Van Deemter. Enhancing and Evaluating the Grammatical Framework Approach to Logic-to-Text Generation. pp. 148–171.

[5] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[7] Z. Gekhman, G. Yona, R. Aharoni, M. Eyal, A. Feder, R. Reichart, and J. Herzig. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?

[8] L. Hagström, D. Saynova, T. Norlund, M. Johansson, and R. Johansson. The Effect of Scaling, Retrieval Augmentation and Form on the Factual Consistency of Language Models. arXiv.

[9] C. Hahn, F. Schmitt, J. J. Tillman, N. Metzger, J. Siber, and B. Finkbeiner. Formal Specifications from Natural Language.

[10] S. Han, H. Schoelkopf, Y. Zhao, Z. Qi, M. Riddell, L. Benson, L. Sun, E. Zubova, Y. Qiao, M. Burtell, D. Peng, J. Fan, Y. Liu, B. Wong, M. Sailor, A. Ni, L. Nan, J. Kasai, T. Yu, R. Zhang, S. Joty, A. R. Fabbri, W. Kryscinski, X. V. Lin, C. Xiong, and D. Radev. FOLIO: Natural Language Reasoning with First-Order Logic.

[11] Z. Jiang, J. Araki, H. Ding, and G. Neubig. How Can We Know *When* Language Models Know? On the Calibration of Language Models for Question Answering. 9:962–977.

[12] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, D. Ganguli, D. Hernandez, J. Jacobson, J. Kernion, S. Kravec, L. Lovitt, K. Ndousse, C. Olsson, S. Ringer, D. Amodei, T. Brown, J. Clark, N. Joseph, B. Mann, S. McCandlish, C. Olah, and J. Kaplan. Language Models (Mostly) Know What They Know.

[13] T. Kocmi and C. Federmann. Large Language Models Are State-of-the-Art Evaluators of Translation Quality. arXiv.

[14] X. Lu, J. Liu, Z. Gu, H. Tong, C. Xie, J. Huang, Y. Xiao, and W. Wang. Parsing Natural Language into Propositional and First-Order Logic with Dual Reinforcement Learning.

[15] K. Manome, M. Yoshikawa, H. Yanaka, P. Martínez-Gómez, K. Mineshima, and D. Bekki. Neural sentence generation from formal semantics. In *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 408–414. Association for Computational Linguistics.

[16] P. Martínez-Gómez, K. Mineshima, Y. Miyao, and D. Bekki. Ccg2lambda: A Compositional Semantics System. In S. Pradhan and M. Apidianaki eds., *Proceedings of ACL-2016 System Demonstrations*, pp. 85–90. Association for Computational Linguistics.

[17] K. Mineshima, P. Martínez-Gómez, Y. Miyao, and D. Bekki. Higher-order logical inference with compositional semantics. In L. Màrquez, C. Callison-Burch, and J. Su eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2055–2061. Association for Computational Linguistics.

[18] A. Mpagouli and I. Hatzilygeroudis. A Rule-Based System Implementing a Method for Translating FOL Formulas into NL Sentences. In G. Governatori,

J. Hall, and A. Paschke eds., *Rule Interchange and Applications*, Vol. 5858, pp. 167–181. Springer Berlin Heidelberg.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, p. 311. Association for Computational Linguistics.

[20] A. Ranta. Translating between Language and Logic: What Is Easy and What Is Difficult. In N. Bjørner and V. Sofronie-Stokkermans eds., *Automated Deduction – CADE-23*, Vol. 6803, pp. 5–25. Springer Berlin Heidelberg.

[21] M. Steedman. *The syntactic process*. MIT press, 2000.

[22] H. Touvron, L. Martin, K. R. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. M. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. M. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models.

[23] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need.

[24] S. Wang, W. Zhong, D. Tang, Z. Wei, Z. Fan, D. Jiang, M. Zhou, and N. Duan. Logic-Driven Context Extension and Data Augmentation for Logical Reasoning of Text.

[25] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned Language Models Are Zero-Shot Learners.

[26] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, F. Xia, Q. Le, and D. Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models.

[27] H. Yanaka, K. Mineshima, P. Martínez-Gómez, and D. Bekki. Determining Semantic Textual Similarity using Natural Deduction Proofs. In M. Palmer,

R. Hwa, and S. Riedel eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 681–691. Association for Computational Linguistics.

[28] Y. Yang, S. Xiong, A. Payani, E. Shareghi, and F. Fekri. Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation.

[29] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models.

[30] L. Zettlemoyer and M. Collins. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars.

[31] 戸次大介. 数理論理学 = Mathematical logic. 東京大学出版会, 2012.

[32] 外園, 長谷川, 渡邉, 馬目, 簗, 谷中, 田中, M.-G. Pascual, 峯島, 戸次. 意味解析システム ccg2lambda による金融ドキュメント処理. 人工知能学会全国大会論文集, JSAI2018:3G105–3G105, 2018.

[33] 石田, 谷中, 戸次. 日本語症例テキストの複合語解析・推論システム medc2l. 自然言語処理, 30(3):935–958, 2023.