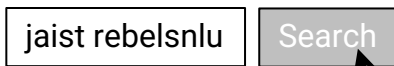
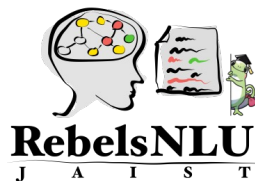


Large Language Models (LLM)

大規模言語モデルは ”考える”ことができるのか

井之上 直也



講演者紹介



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 准教授
言語推論研究室 (RebelsNLU) 主宰者
理化学研究所 革新知能統合センター客員研究員
博士 (情報科学)

略歴

- 2010年 奈良先端科学技術大学院大学 情報科学研究科 修士課程修了
- 2013年 東北大学 情報科学研究科 博士課程修了
- 株式会社デンソー、東北大学助教、米 Stony Brook University ポスドクを経て2022年より現職
- 自然言語処理 (特に知識・推論) の研究に14年間従事

ChatGPT4 に講演のアドバイスをお願いしてみた



You

JAIST産学官共創フォーラムにて、「大規模言語モデルは”考える”ことができるのか」という題目で講演をすることになりました。聴衆想定は、AI技術に興味のある一般の方々も含まれます。どのような流れでトークをするのがよいでしょうか。



ChatGPT

JAIST産学官共創フォーラムでの「大規模言語モデルは”考える”ことができるのか」というテーマに基づいた講演の流れを以下のように提案します。聴衆がAI技術に興味のある一般の方々も含むことを考慮に入れて、専門的過ぎず、かつ内容が理解しやすいように構成します。

1. 導入

- **背景の説明:** AIと大規模言語モデルの基本概念の紹介。
- **講演の目的:** 「考える」とは何か、そして大規模言語モデルがそれにどう関わるかを探る。

2. 大規模言語モデルとは

- **定義と概要:** GPT-4や他の大規模言語モデルの基本的な仕組みと機能。
- **開発の歴史:** 簡単な歴史的背景とその進化。

⋮

本日のメニュー

講演1: LLM は "考える" ことができるのか (本講演)

- Part 1: LLM の仕組み (25分)
 - 自然言語処理の基礎
 - LLM の基礎
- Part 2: LLM の推論能力 (25分)
 - ベンチマークデータの構築: 日本語マルチホップQA、画像に対する推論
 - 無知の知は LLM の信頼性を向上させるか: Back-off LMKB

講演2: オープンなLLMの発展と、それを活かした日本語LLMの開発について (株式会社ELYZA 佐々木彬氏)

言語 (Language) とは

- 言語: 文法と語彙からなる何らかのシステム
- 形式言語 (Formal Language)
 - ある目的のために人工的に設計された言語
 - プログラミング言語、エスペラント語、楽譜、化学式…
- 自然言語 (Natural Language)
 - 人間が日常生活で自然に使う言語
 - 日本語、英語、中国語…

自然言語処理 (Natural Language Processing; NLP)

- 自然言語を処理する**計算モデル**を研究する学問
- 計算モデルとは
 - 数式 $y = f(x)$, x は解析対象の文
 - アルゴリズム (手続き; 処理の手順)
- NLP の研究は**例外**との闘い
 - `def chatbot(x):`
 - `if x == "Hello": print("Hi, how are you?")`
 - `chatbot("Hello")`
 - `chatbot("Hello!") chatbot("hello") chatbot("Hi!!!")`

NLP の応用例

- 情報検索: Google, Bing, Semantic Scholar, Google Scholar
- 機械翻訳: Google Translation, DeepL
- 言語学習: Grammary, Duolingo
- 自動採点: ETS e-rater
- バーチャルアシスタント: Siri, Alexa, Cortana
- 推薦システム: Amazon, Rakuten, e-bay, Mercari
- テキスト分類: SmartNews, Google News, Yahoo! News
- 対話システム: ChatGPT

何が難しいか

- **言語特徴量問題**

- 言語をうまく扱える、機械専用のデータ形式をいかに設計するか
- 同犠牲、多義性、曖昧性

- **知識推論問題**

- 人間の持つ知識をどのように機械に教え込み、使いこなせるようにするか
- 世界知識、推論

ここが難しい (1/3): 同義性

異なる表現、同じ意味

私は石川生まれだ

私は石川県の出身だ

私の人生は石川から始まった

...

=

「私は石川で生まれた」

ここが難しい (2/3): 多義性、曖昧性

同じ表現、異なる意味

The man went to the bank.

私は犬をかっている。

(単語の多義性)

私は友達とごはんを食べた。

私はりんごとみかんを買った。

(統語構造の曖昧性)

ここが難しい (3/3): 世界知識

自然言語の理解には世界知識が必要

1 a) 太郎は、次郎を怒鳴った。
彼はイライラしていたからだ。

1 b) 太郎は、次郎を怒鳴った。
彼はふざけていたからだ。

アプローチの歴史

- 1950-2013年: **パイプライン方式**

- 機能別にモジュールをつくり組み合わせる
- 言語は様々な単位の塊からなる: 単語、句、文、文書…
- 解析対象 → **形態素解析** → **構文解析** → **意味解析** → **文脈解析** → 出力
- 各モジュールの解析結果から特徴量を抽出する

- 2014年-: **End-to-end 方式**

- 言語の意味は分散表現 (ベクトル) で表現できる
- 解析対象 → **ニューラルネットワーク** → 出力
- 昨今の LLM もこの流れを汲む

パイプライン方式の概要

- **形態素解析**

- 単語区切り、品詞を決める

- **構文解析**

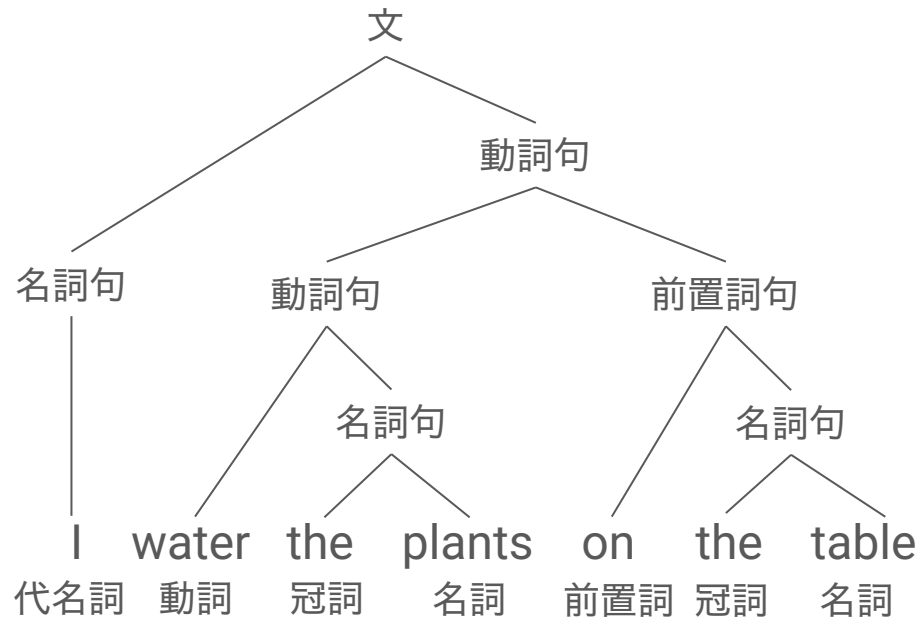
- 文の構文的な構造を決める
- 単語と単語の関係

- **意味解析**

- 文の意味解釈を決める

- **文脈解析**

- 文脈や状況から文の解釈を決める



- 工場 ✓
- 植物

End-to-end 方式の概要

- 言語を分散表現で表現
 - 単語をベクトル化
 - これらのベクトルを合成して句や文、文書の意味を表現
- 分散表現を用いて NLP タスクを解く
- 単語のベクトルや、合成の仕方はどうやって学習するか？
→ **深層学習**の出番

深層学習とは？

- 深層構造により抽象度が高い**特徴量を自動学習**する方法
 - 特徴量 = 分散表現
- 深層学習時代以前は…
 - 基本的な枠組み: 入力 → 特徴量抽出 → 分類/生成
 - 特徴量関数を**人手により**設計
- ニューラルネットワークに基づく手法が主流
- AI 系研究への適用事例多数

LLM のアイデア

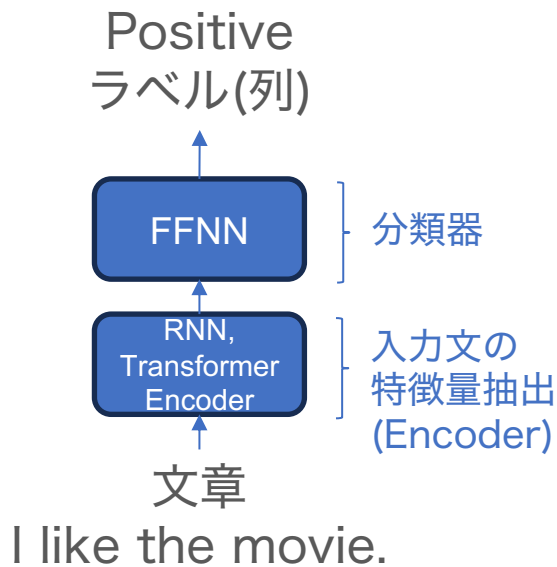
- **事前学習 (Pretraining)**: 様々な NLP タスクに有用な、**汎用的な** 特徴量を学習する (深層学習を使う)
 - 汎用的なモデルを設計する
 - 汎用的な特徴量を学習するためのタスクを設計する
 - 大規模なウェブコーパスを使用
- 学習したモデルを、他のタスクに転用する
 - 比較的少量の注釈付きコーパスを使用
 - **ファインチューニング (Finetuning)** (2018-)
 - **コンテキスト内学習 (In-context Learning)** (2023-)

NLP でよく使われる深層学習モデルの構成素

- 順伝播型ニューラルネットワーク
(Feedforward Neural Network; FFNN, Multi-layer Perceptron; MLP)
- 可変 (Variable-length) 系列データのモデル化
 - 再帰型ニューラルネットワーク (Recurrent Neural Network; RNN)
 - Gated Recurrent Unit (GRU)
 - Long Short-Term Memory (LSTM)
 - Transformer
- 系列変換のモデル化
 - Encoder-Decoder モデル

代表的なモデルの組み合わせ方 (模式図)

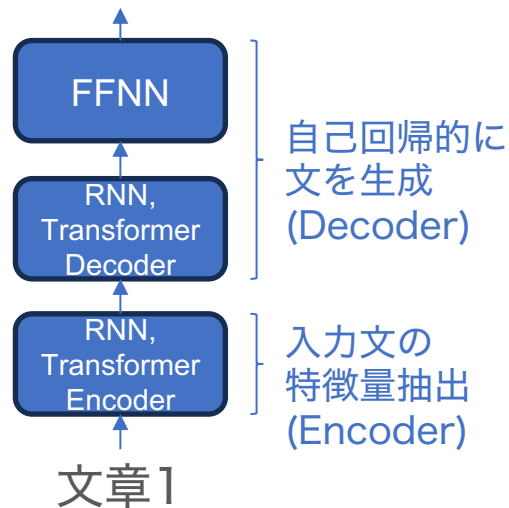
ラベリングタスク



系列変換タスク

私は石川県在住です。

文章2

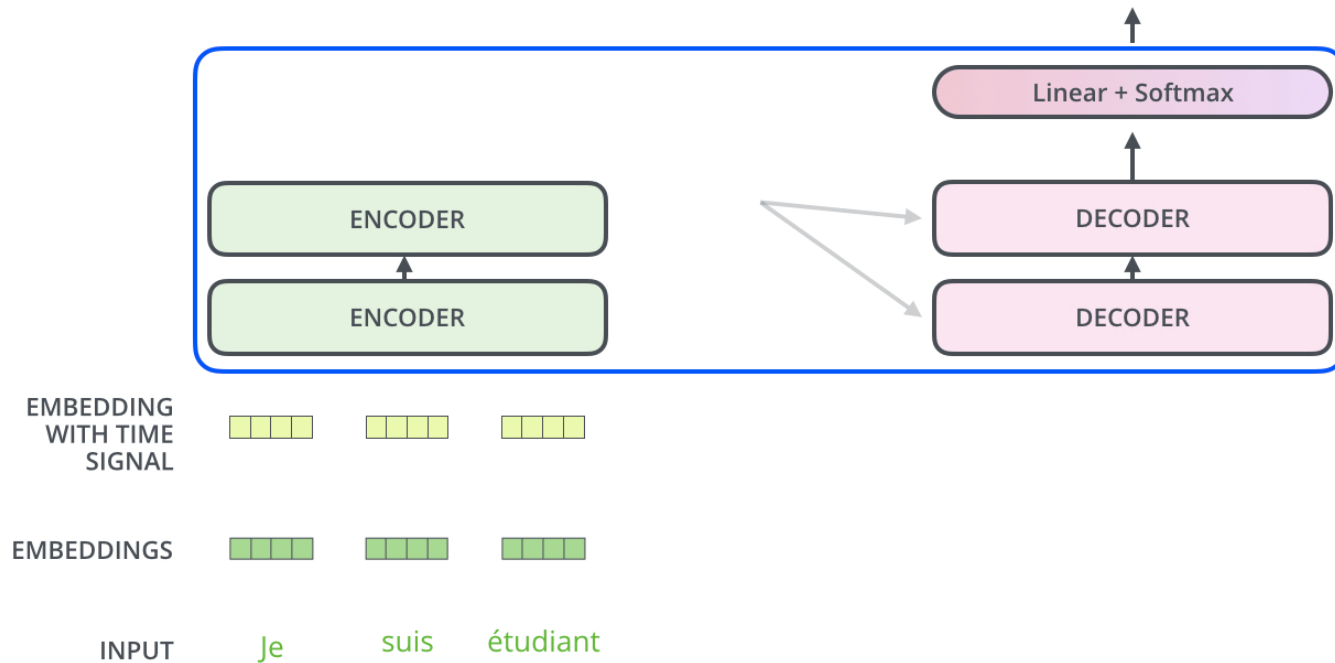


I live in Ishikawa.

Transformer (Vaswani+2017)

Decoding time step: ① 2 3 4 5 6

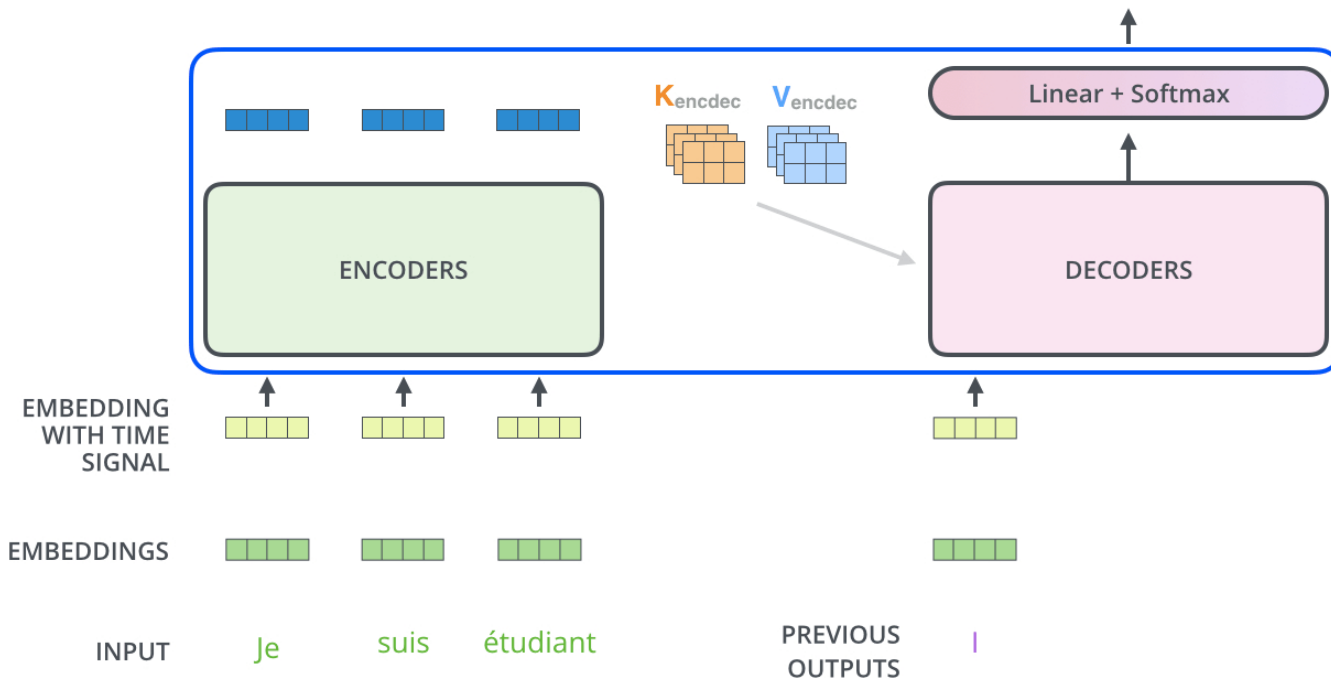
OUTPUT



Transformer (Vaswani+2017)

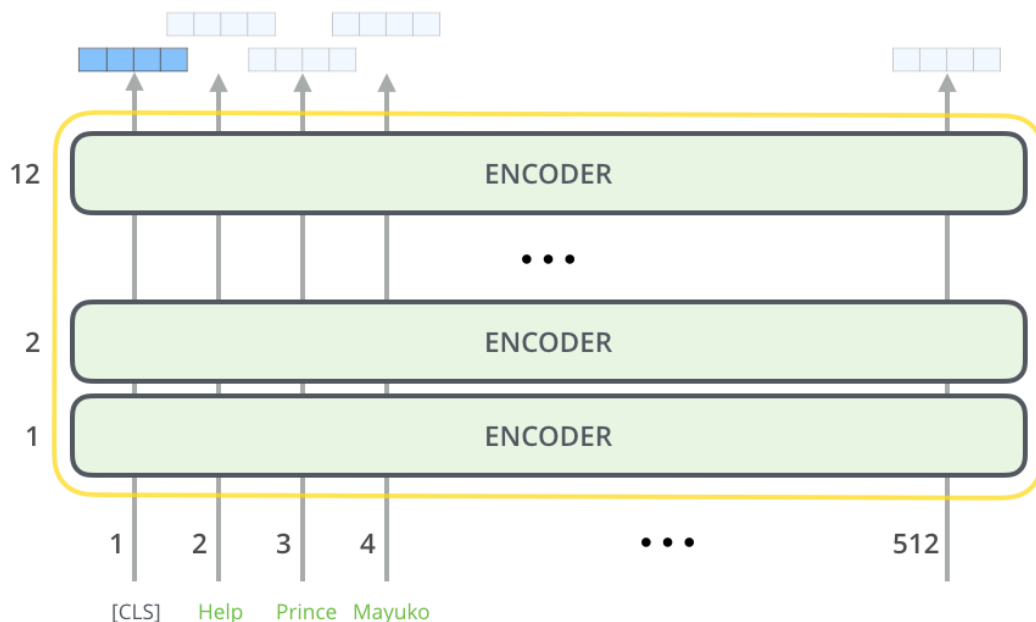
Decoding time step: 1 2 3 4 5 6

OUTPUT |



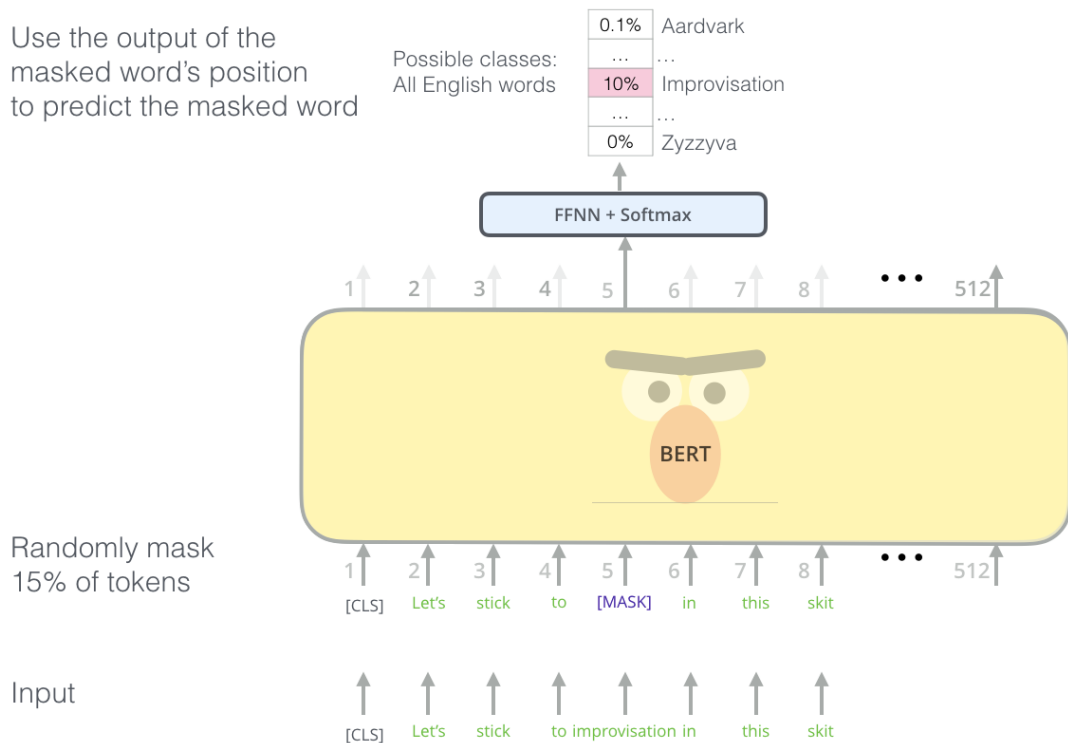
BERT: Bidirectional Encoder Representations from Transformers (Devlin et al. 2019)

- モデル: Transformer エンコーダ (base: 12層 or large: 24層)
- 文全体の埋め込みを得るため文頭に [CLS] を追加



BERT: 事前学習

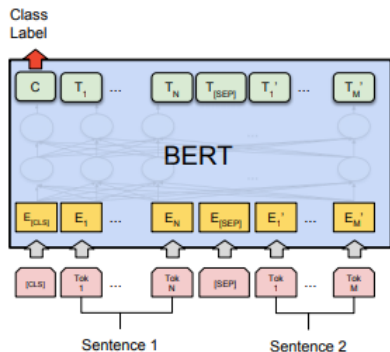
Use the output of the masked word's position to predict the masked word



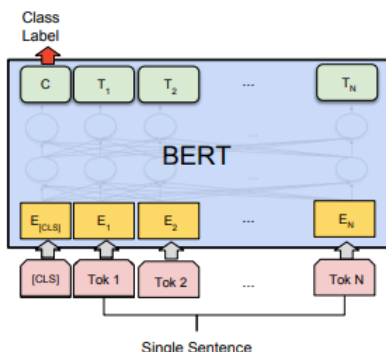
事前学習タスク

- マスク言語モデリング
- 文隣接性判定
- 生コーパスから自動的に訓練データを構築可能

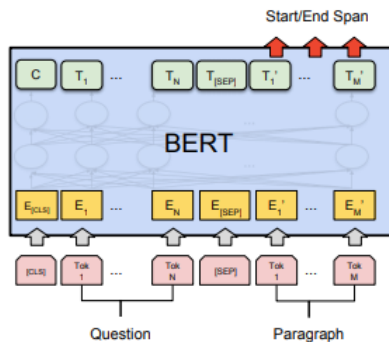
BERT: 自然言語処理タスクへの適応方法



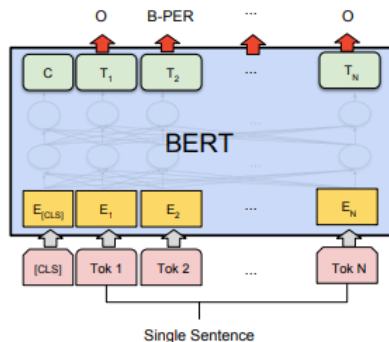
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



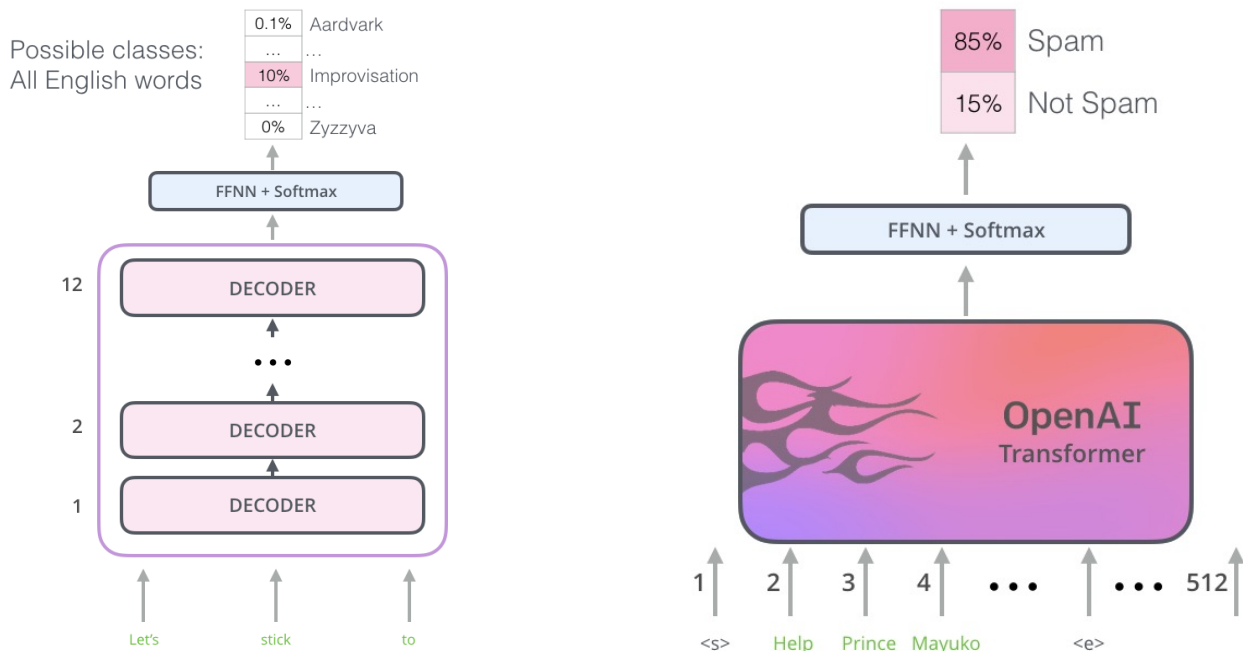
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

ファインチューニング

- 比較的少量の注釈付きデータセットで追加学習をする
- モデルの重みパラメータを変更する
- 様々なNLPタスクで最高性能を達成し当時は衝撃が走った

GPT: Generative Pretraining Transf. (Radford et al. 2018)

- モデル: Transformer デコーダ (12層)
- 事前学習タスク: 次単語予測



OpenAI の大規模言語モデル

- GPT-3 (Brown+2020): Transformer デコーダ (Vaswani+2017)
 - 1,750億個の重みパラメタ
 - 570GB のコーパスで事前学習 (事前学習タスク: 次単語予測)
- InstructGPT (Ouyang+2022): GPT-3 をインストラクションデータにて訓練
 - 完全教師あり学習 (約1万事例)、強化学習 (約6万事例)
- ChatGPT: GPT-3.5 [2] を対話向けに調整、技術詳細非公開
- GPT-4 [3]: Transformer、画像も入力可、技術詳細非公開
- ChatGPT-4: GPT-4 + 各種拡張モジュール (検索, 計算, etc.)

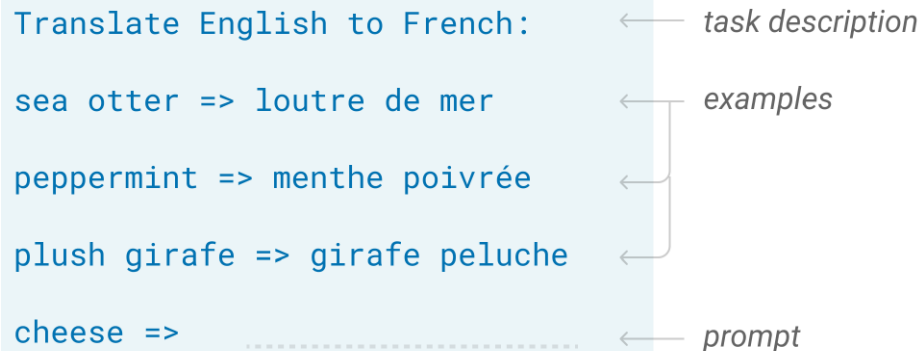
[1] <https://openai.com/blog/chatgpt>

[2] <https://platform.openai.com/docs/model-index-for-researchers/models-referred-to-as-gpt-3-5>

[3] <https://openai.com/research/gpt-4>

コンテキスト内学習 (In-Context Learning)

- モデルの重みパラメタを**変更せず**タスクの解法を学習
- 自然言語によりタスクを指示
- モデルへの入力
 - タスクの説明 (Task description)
 - 少量 (10~100個) の入出力の例 (Examples)
 - 解きたい問題 (Prompt)
- 最先端のシステムと近い、またはそれ以上の性能を達成



The diagram shows a light blue rectangular area containing text. To the right of this area, four horizontal arrows point left towards the text, each with a label. The first arrow points to the first line and is labeled 'task description'. The second, third, and fourth arrows are grouped by a vertical bracket on their right side and are collectively labeled 'examples'. The fourth arrow points to the last line and is labeled 'prompt'.

```
Translate English to French: ← task description
sea otter => loutre de mer ← examples
peppermint => menthe poivrée ← examples
plush girafe => girafe peluche ← examples
cheese => ..... ← prompt
```

(Brown+2020)

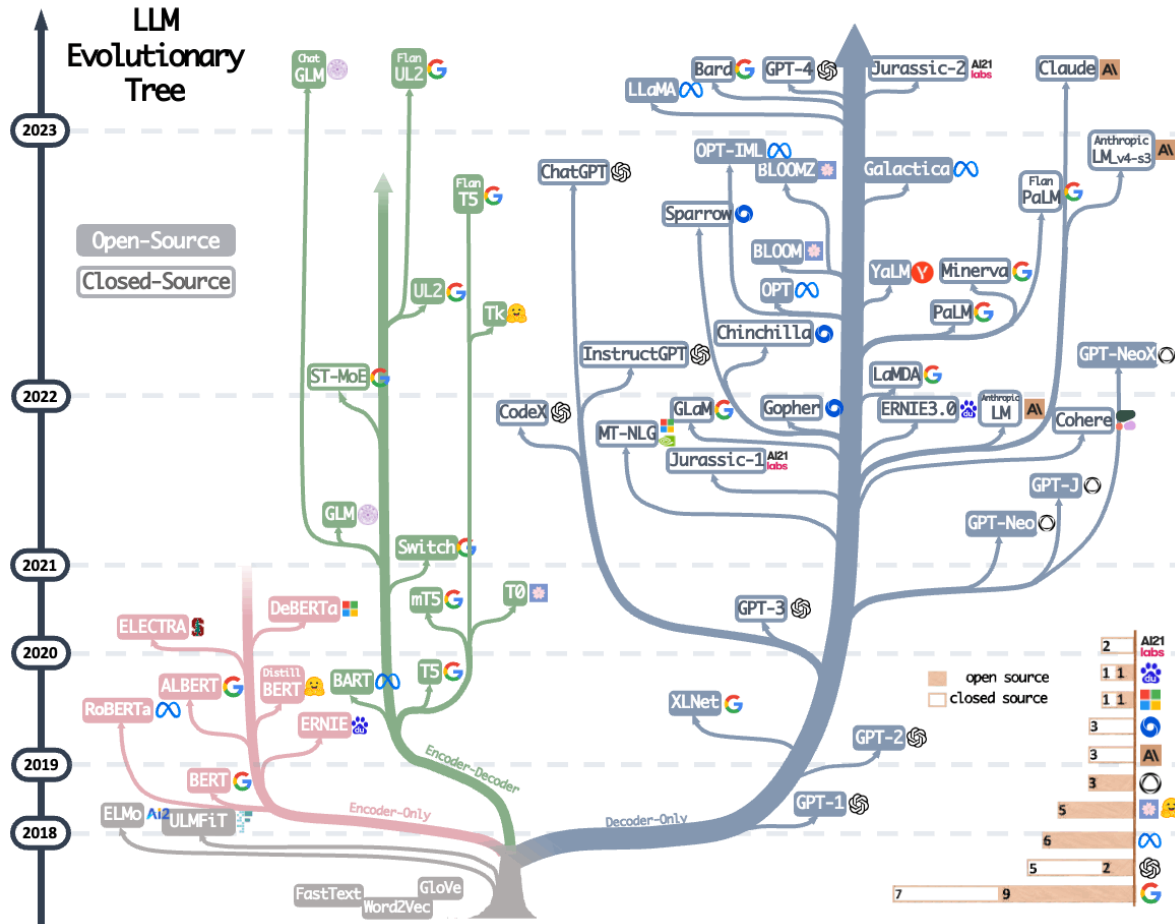
LLM は自然言語処理の難題にどのように挑戦したか

● 言語特徴量問題

- 言語をうまく扱えるような機械の形式をいかに設計するか
 - 同犠牲、多義性、曖昧性
- 単語の穴埋め問題に基づく事前学習

● 知識推論問題

- 人間の持つ知識をどのように機械に教え込み、使いこなせるようにするか
 - 世界知識、推論
- 大規模な訓練データから人間の常識を学習し、
大規模なモデルパラメタに知識を蓄える



LLM ですべてが解決するわけではない

- データ汚染
 - 訓練事例に既存のNLP評価データが含まれており、公平な性能評価ができていない可能性がある
- 幻覚 (hallucination): 事実と異なることを言う
 - 訓練データ/入力文に書いていない事実を作り出してしまう (単語予測エラー)
... Confabulation (作話)
 - そもそも訓練データに嘘が書いてあり、忠実にそれを再現した (LLMに非はない)
- 安全性
 - 幻覚により事実と異なる解答を出してしまい、人々を混乱させる
 - 人類に害のある解答を出してしまう (爆弾の作り方など)
 - 言語モデルの解釈性、技術の詳細が非公開

ChatGPT4 に JAIST の場所を聞いてみた



You

JAISTはどこにありますか？



ChatGPT

日本先端科学技術大学院大学（JAIST）は、日本の石川県野々市市にあります。この大学は、1990年に設立され、特に科学技術の分野で高度な研究と教育を行っています。石川県は日本の中部地方に位置しており、歴史的な建造物や文化、美しい自然環境で知られています。JAISTは、この地域の静かな環境の中で、学術的な優れた環境を提供しています。

!?

本日のメニュー

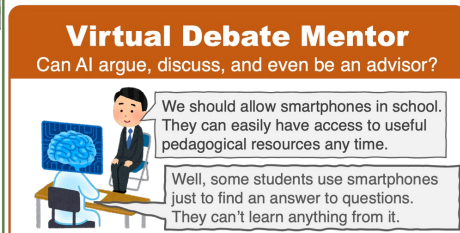
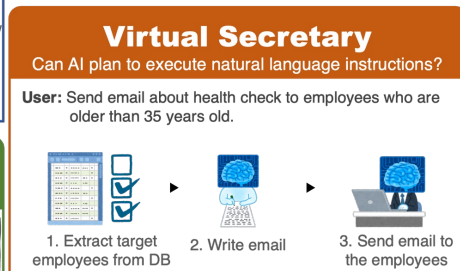
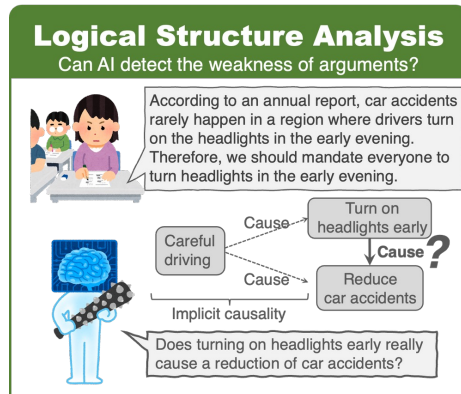
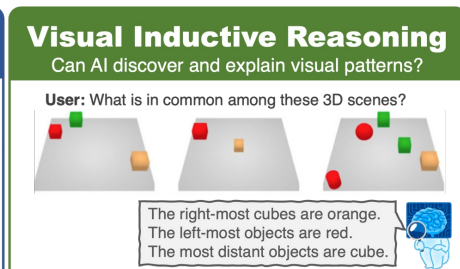
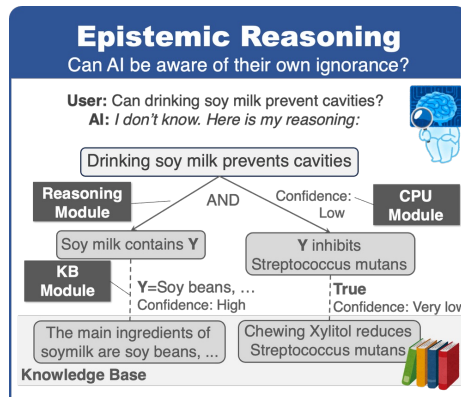
講演1: LLM は ”考える” ことができるのか (本講演)

- Part 1: LLM の仕組み (25分)
 - 自然言語処理の基礎
 - LLM の基礎
- Part 2: LLM の推論能力 (25分)
 - ベンチマークデータの構築: 日本語マルチホップQA、画像に対する推論
 - 無知の知はLLMの信頼性を向上させるか: Back-off LMKB

講演2: オープンなLLMの発展と、それを活かした日本語LLMの開発について (株式会社ELYZA 佐々木彬氏)

言語推論研究室 (RebelsNLU@JAIST) の紹介

- 推論系 NLP を徹底的に研究する
インターナショナルラボ
- 2022年4月設立、現在:
 - 修士課程: 10 (+4) 人
 - 博士課程: 3人
 - (うち8人が留学生)



JEMHopQA: 概要



- Japanese Explainable Multi-hop Question Answering Dataset
 - “マルチホップ” QA: 複数の知識断片を組み合わせる高度な推論を要する質問応答
 - 知識源: Wikipedia (2021 dump)
 - 1,179 件のデータセットを構築 (**日本語で世界初**)
 - **導出** (解答に必要となる知識断片の集合) も付与

- 問題の例

質問: ルーヴル美術館が所在する都市の市長の名前は?

解答: アンヌ・イダルゴ

導出: (ルーヴル美術館, 所在地, パリ)
(パリ, 市長, アンヌ・イダルゴ)

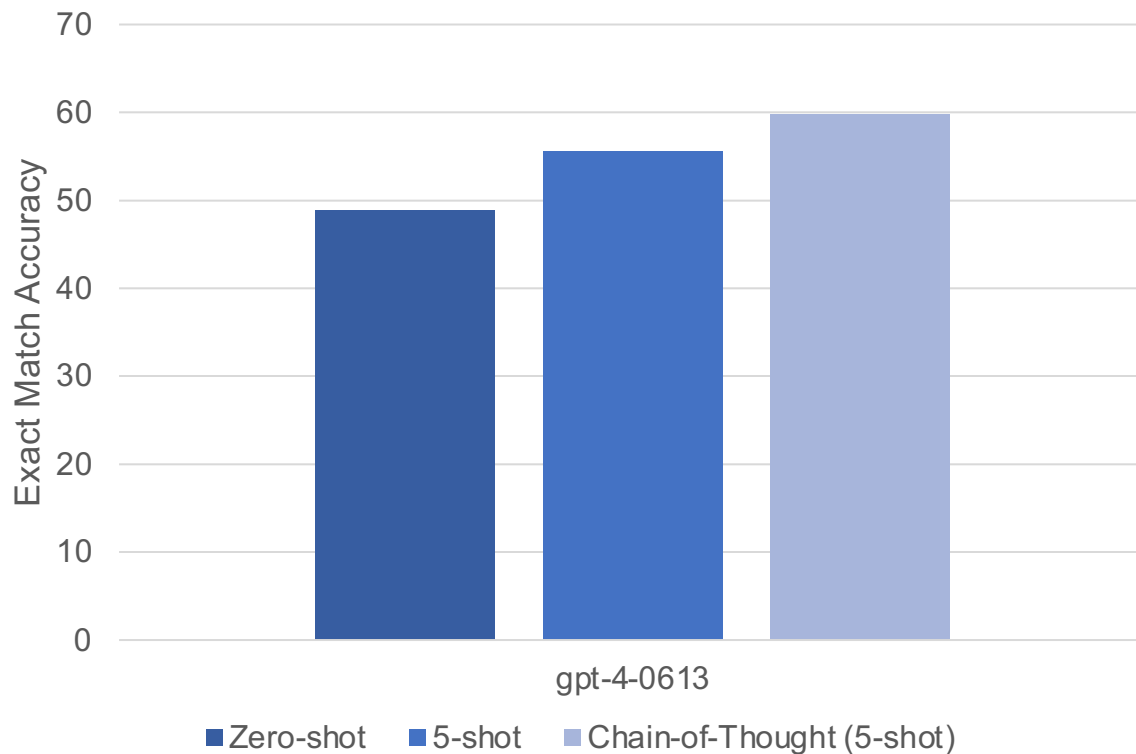
質問: 『天空の城ラピュタ』と『となりのトトロ』の公開日が早いのは、『となりのトトロ』ですか?

解答: NO

導出: (天空の城ラピュタ, 公開年, 1986年)
(となりのトトロ, 公開年, 1988年)

<https://github.com/aiishii/JEMHopQA>

GPT-4 はどれぐらい解けるのか？



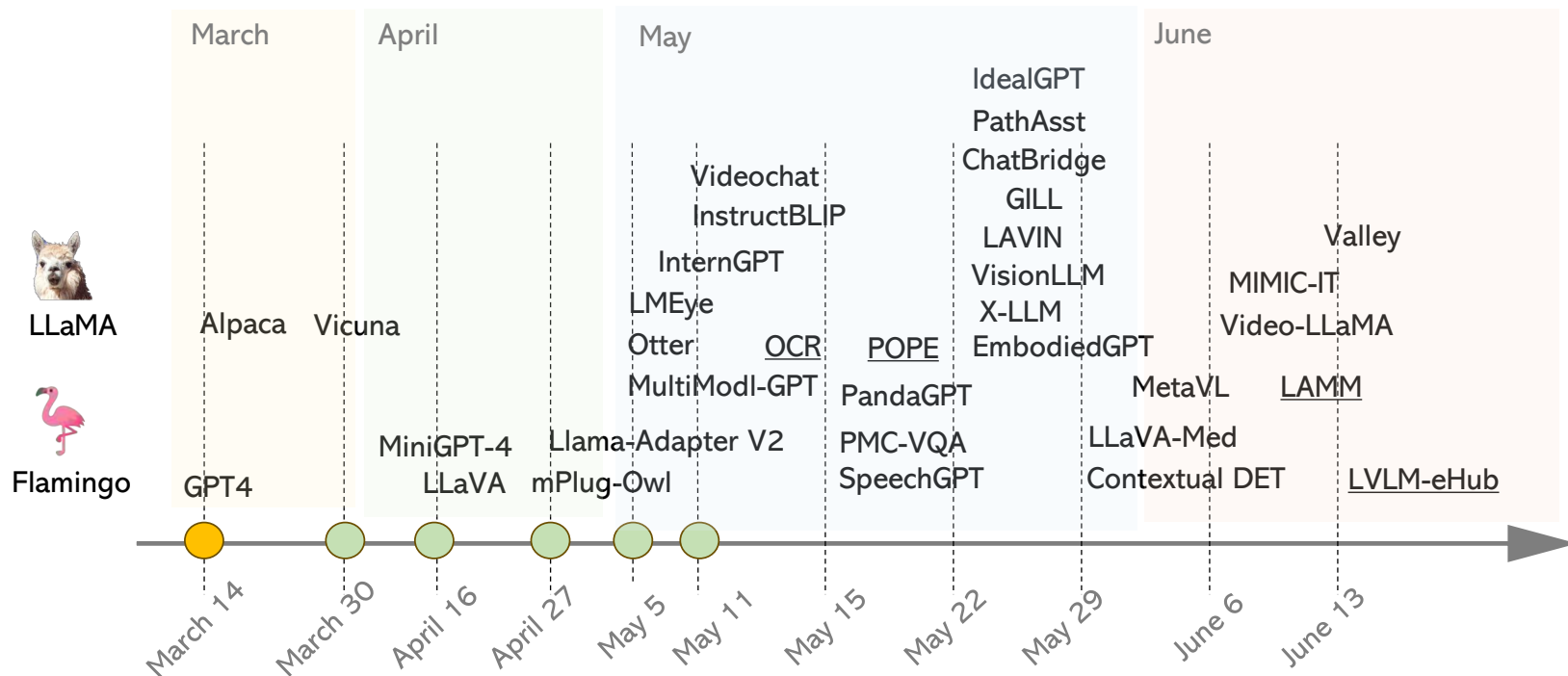
分析: GPT-4 はなぜ間違えるのか？

- 作話の導出に依拠した推論 (47/48事例)
 - 質問: 東名阪自動車道と伊勢自動車道、高速道路ナンバリングによる路線番号は違いますか？
 - GPT-4 解答: NO / 正解: YES
 - GPT-4 導出:
 - (東名阪自動車道, 高速道路ナンバリング, **E1**) ※正しくはどちらも E23
 - (伊勢自動車道, 高速道路ナンバリング, **E42**)
- **作話が主たるエラーの要因**
- 正しい導出を与えて解答を試みると、95.6% の正解率！
- **GPT-4 の推論能力は十分に高く、作話の解消が大きな課題**

分析: GPT-4 は本当に正解しているのか？

- 解答は正解しているが誤った知識に依拠 (23/72事例)
 - 質問: 永山瑛太の実弟の職業は？
 - GPT-4 解答: 俳優 / 正解: 俳優
 - GPT-4 導出:
 - (永山瑛太, 実弟, **永山俊輔**) ※実在しない人
 - (**永山俊輔**, 職業, 俳優)
- 答えありきで辻褃合わせの作話をしたのかも？

マルチモーダル (画像、音声への拡張) LLM の進展



GPT-4 Vision (GPT-4V)

- 画像も入力できる
マルチモーダル言語モデル
- GPT-4 はもともとマルチ
モーダル言語モデル
- 10月以降、画像入力も受付ける
形となった [1]



ここはどこでしょう？



これは「JAIST (Japan Advanced Institute of Science and Technology、日本先端科学技術大学院大学)」の看板の写真のようです。JAISTは日本の石川県に位置する大学院専門の大学です。



何本の木が見えますか？

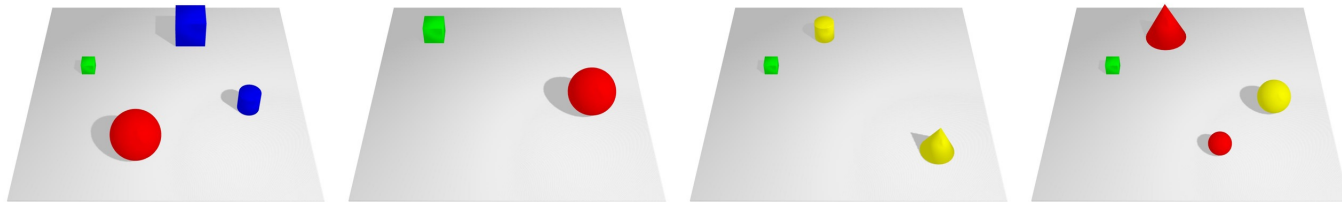


写真には、明確に区別できる木が5本見えます。しかし、背景や遠くにも木があるかもしれませんが、その詳細な数はこの写真からは判断できません。

[1] <https://platform.openai.com/docs/guides/vision>

Find-the-Common: 概要

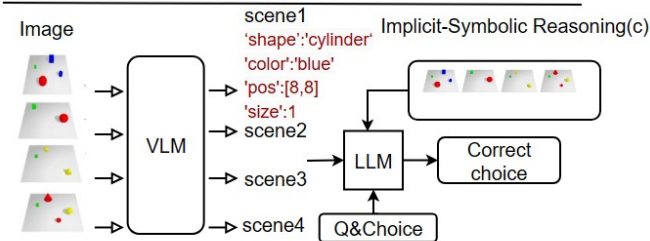
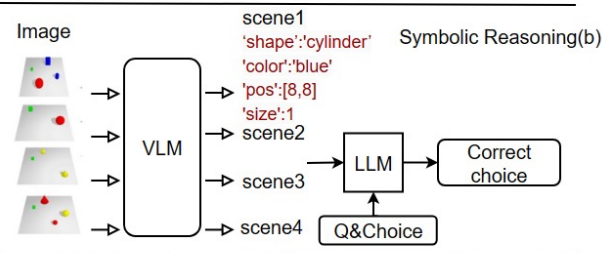
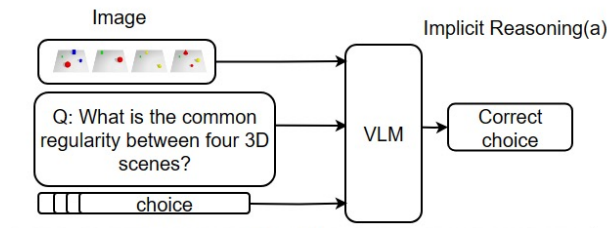
- 画像に対する帰納的推論のベンチマークデータ 500問 (**世界初**)
- 入力: 複数の3Dシーン画像



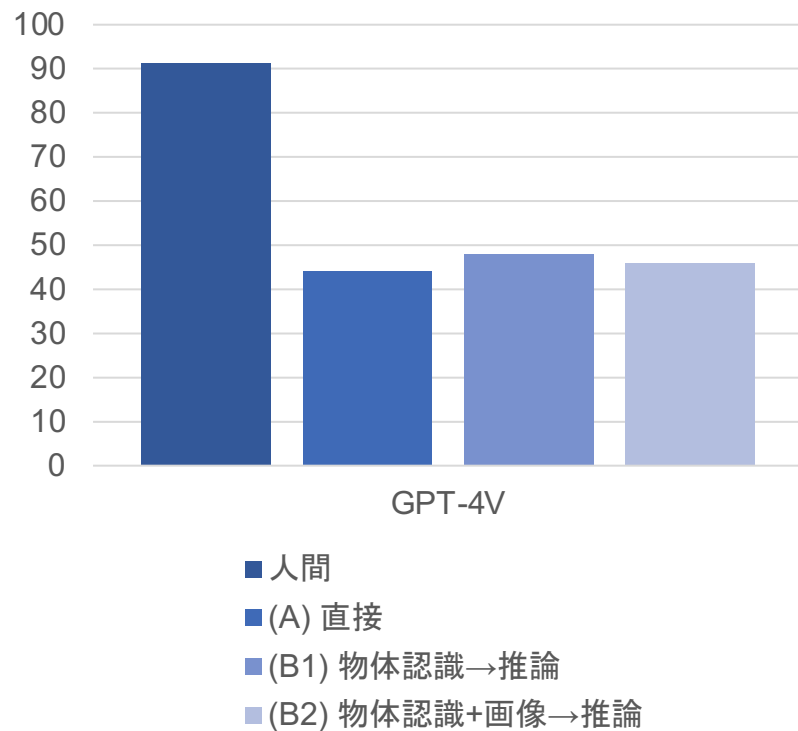
- タスク: 画像に共通するパターンを見つける
 - (a) 黄色い物体はすべて筒状である
 - (b) 赤い球体は手前にある
 - (c) 最も左にある直方体は緑色である ✓
 - (d) 最も遠い物体は紫色である

解き方のバリエーション

- 直接 Vision & Language Model (VLM) に聞く … (A)
- 段階を踏んで聞く
 1. シーンにどのような物体があるかを答えさせる
 2. 物体認識結果をもとに推論をする
 - 物体認識結果を与え、完全に**言語ベース**で推論する … (B1)
 - 物体認識結果 + 画像情報を与え、**画像と言語の両方**に基づいて推論する … (B2)

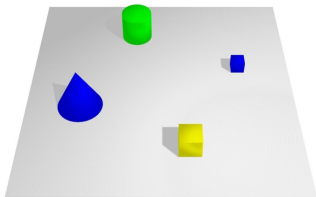


GPT-4V はどれぐらい解けるのか？

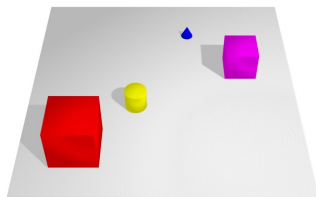


分析: GPT-4V はなぜ間違えるのか？

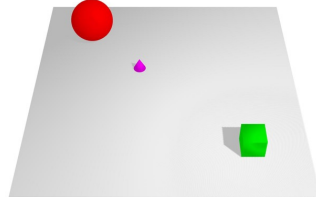
- 実際にはない物体を誤認識してしまう (約80%の事例で確認)



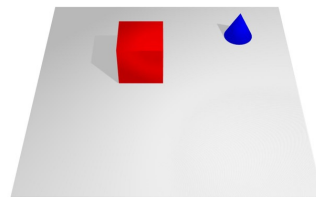
A green cylinder
A blue teardrop or droplet shape
A blue cube
A yellow cube



A purple cube
A blue teardrop or droplet shape
A red cube
A yellow sphere*



A large red sphere
A small purple teardrop or droplet shape
A green cube
A yellow cube*



A red cube
A blue teardrop or droplet shape
A blue cube*

- 正しい物体認識結果を与えた場合の正解率

- B1法 74.0% / B2法 92.5% !

→ **推論能力は十分に高い、作話が大きな課題 (again)**

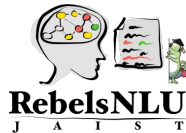
ここまでのまとめ

- LLM は NLP の難題に有望な解決策を与えた
- LLM の振る舞い分析からの知見
 - 推論能力は十分にあるように見える
 - 作話してしまうことに大きな課題あり
 - テキスト入力、画像入力に共通の課題
- 今後のビジョン
 - **作話にどう対処していけば良いか**が一つの鍵になる



Back-off LMKB: スタンス

- LLM は超超超巨大記憶装置を持った柔軟な**パタンマッチャー**
 - 訓練データに直接的に書かれていないことを正確に答えるのは難しい
 - そもそも書かれていることを全て正確に覚えているわけではない
- 無理矢理答えるから作話が起こる
 - その場合には「分からない」という解答が欲しい
- **問い:** LLM は「自分が何を知っているか」を自己認知できるか？
- **重要キーワード:** **無知の知**



Back-off LMKB: アイデア

- LLM に問い合わせ、解答を試みる
- 自己認知機構により、解答が分からない (無知) と判断した場合、質問を言い換えて検討し直す (=バックオフ; **世界初**)
- バックオフを繰り返しても分からない状態が続く場合には、「分からない」と解答させる

p: 文在寅は、大都の区域外で生まれた。



LLMで分解: $q_1 \& q_2 \rightarrow p$

q₁: 文在寅は、巨済市で生まれた。

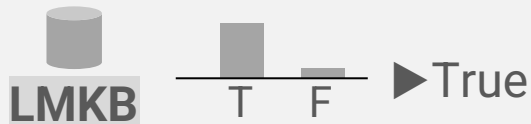


q₂: 巨済市は大都の区域外にある。



LLMで分解: $r_1 \& r_2 \rightarrow q_2$

r₁: 巨済市は韓国にある。

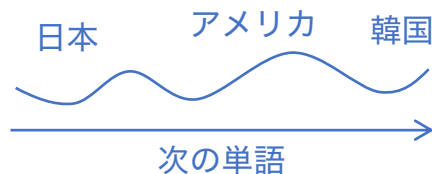
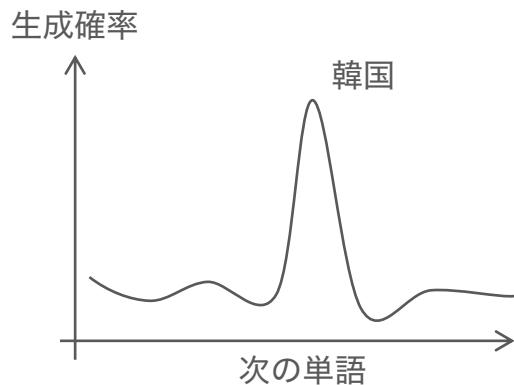


r₂: 大都は、現在の中国の北京に存在した。



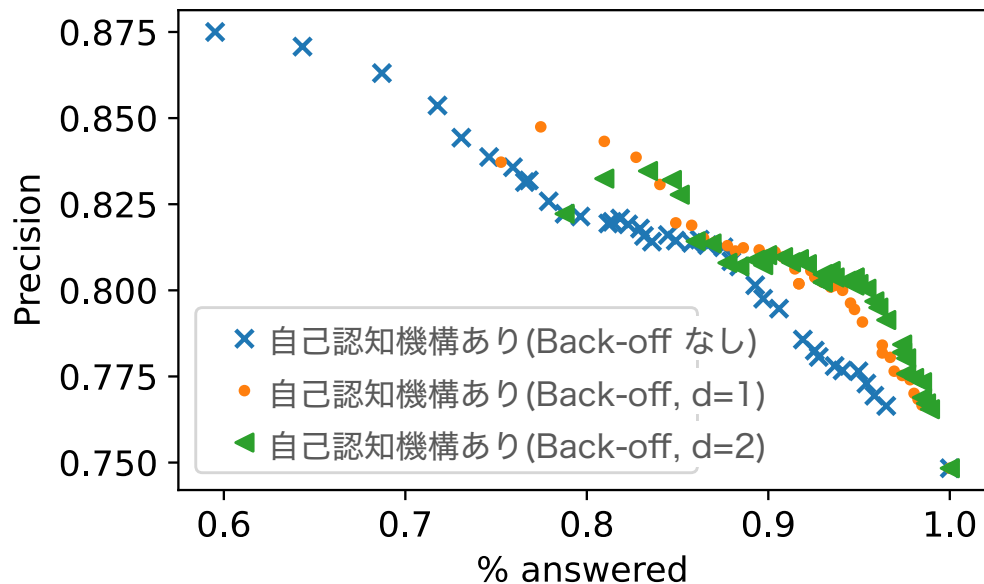
無知の知: 単語の生成確率分布の情報量から推定

- 単語の生成確率分布の情報量 (エントロピー) をヒントとする
- 文在寅の出身国は、_____



→ **情報量が低い、無知の状態
(作話が起こりやすい状況)**

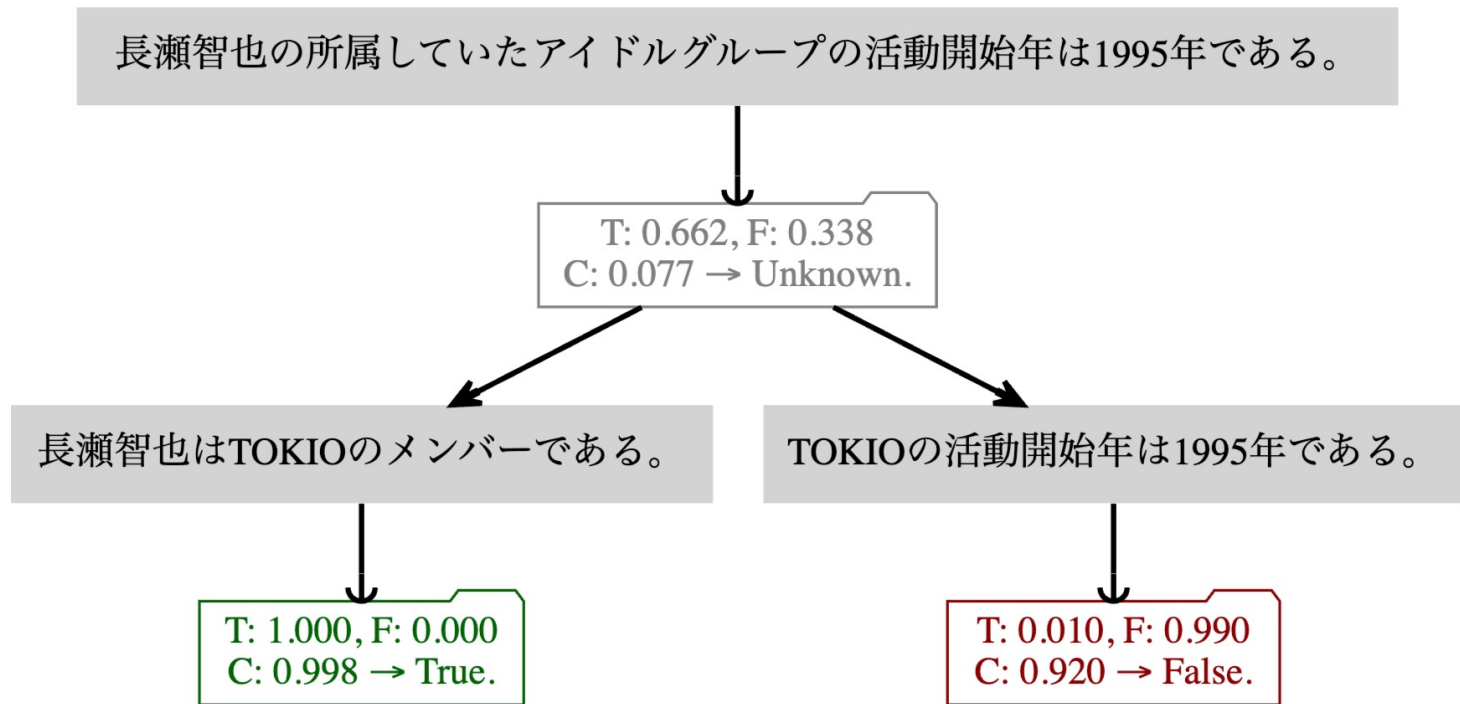
結果: GPT-4 の作話を減少させることに成功



テストセット:
Wikipedia に関するマルチホップ質問応答
(Geva et al. 2021)

- 全ての問題に無理矢理答えさせる: 正解率 74.8%
- 単語の生成確率分布は、作話判定の良い指標になる
- バックオフにより、解答率を上げてても、高い精度を維持

例: バックオフにより GPT-4 の作話を防げたケース



※TOKIOの活動開始年は1990年 (訓練データには入っているはず) なので、「偽」が正解。
素のGPT-4は「真」と作話しようとしたが、**自己認知機構により「不明」と解答できた。**

Five Orders of Ignorance (Armour+ 2000)

- 0th: I provably know something
 - 金沢駅は何県にありますか？
- 1st: I know that I don't know something ← **Back-off LMKB**
 - 火星の今日の最低気温は何度ですか？
- 2nd: I don't know that I don't know something
 - コーヒー豆は豆の一種ですか？
- 3rd: I don't know a suitably efficient way to find out
I don't know that I don't know something ← **素の LLM**
- 4th: I don't know about the Five Orders of Ignorance.

まとめ

- LLM は NLP の難題に有望な解決策を与えた
- LLM の振る舞い分析からの知見
 - 推論能力は十分にあるように見えるが、作話に課題あり
 - バックオフ (無知の自覚+分解による再検証) により、作話を防いで解答精度が向上することを実験的に確認
- LLM の未来: **自己認知 (無知の知を含む) でもっと賢くなれる**
 - 自己認知の応用先: 作話防止、発言内容の自主改良、好奇心に基づく自己学習
- まだまだ残る課題
 - LLM 単体で全てをこなすのは難しい、外部からの制御やツールとの連携が重要
 - 作話、安全性、技術のオープン性
 - 各種言語への適応