



# LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# Agenda

LLMのマグニチュード論 1

## Part 1

BradleyのLLMモデル論概要

## Part 2

LLMの確率計算

## Part 3

Enrichedカテゴリー論とLLMモデルの拡大

## Appendix

# Part 1

BradleyのLLMモデル論概要

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論



BradleyのLLMのマグニチュード論

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# BradleyのLLMのマグニチュード論

今回のセミナーでは、LLMの理論研究で、もっとも新しく最も先進的な業績である Tai-Danae Bradleyの論文

“[The Magnitude of Categories of Texts Enriched by Language Models](https://arxiv.org/pdf/2501.06662)” <https://arxiv.org/pdf/2501.06662> の紹介をしようと思います。

ただし、今回のセミナー「LLMのマグニチュード論 1」は、そうした Bradleyの論文紹介を目的とした連続セミナーの第一回目です。

今回のセミナーがカバーする内容は、次に見る「この論文が扱っている二つの課題」の前半部分の「LLMのモデルの拡大」にフォーカスしたものです。

# この論文が扱っている二つの課題

Bradleyの論文は、二つの問題を扱っています。

- 一つは、LMの意味論に カテゴリー論的基礎を与えた 2022年のBradleyらの論文“[An enriched category theory of language: From syntax to semantics.](#)” のモデルを拡大するという課題です。たとえば、「プロンプトの入力」「プロンプトへの回答の出力」というような、LLMの現実の振る舞いを解釈しうるようにLLMのモデルを拡大することです。
- もう一つの課題は、こうして現実のLLMの振る舞いをシミュレートしうる拡大されたLLMのカテゴリー論的理論を構築し、それを、この間マルレクでも取り上げてきたマグニチュードの理論を結びつける課題です。

# マルレク「マグニチュードとは何か」



マグニチュードとは何か

ショートムービー再生リスト

# マルレク「マグニチュード論の展開」



ショートムービー再生リスト

# 論文の構成(前半)

Bradley論文の全体の構成についても、連続セミナーの第一回目として、ここで触れておこうと思います。

「LLMのモデルの拡大」を主な対象とする論文の前半部分は、次のような構成をしています。この部分の紹介が今回のセミナーの目的になります。

1. はじめに
2. LLMからenrichedカテゴリーを取得する
  1. トークンとテキスト
  2. LLMとテキストから誘導される確率
  3. enrichedカテゴリー論の予備知識
  4. すべてのLLMは $[0, 1]$ -カテゴリーを定義する
  5. すべてのLLMは $[0, \infty]$ -カテゴリーを定義する

# LLMモデルの拡大

第一の課題である 2022年論文のモデルの拡大については、この論文の第二セクション “2. Obtaining Enriched Categories from an LLM” で展開されています。

2022年の論文は、LLM意味論にco-presheaf意味論というフレームを導入し、さらにそのカテゴリーのオブジェクト間の射に、enrichedカテゴリーの手法で、LLMと同じように $[0, 1]$ の値を持つ確率値を付与するというものでした。それは、LLMの振る舞いと意味の数学的モデルとしては、画期的なものでした。

ただ、テキスト $x$ が与えられた時のテキスト $y$  の条件付き確率 $\pi(y|x)$ を直接与えるものではありませんでした。

今回の論文では、以前の $[0, 1]$ の値を取る確率モデルを  $[0, \infty]$  の値を取るLawvereの一般化された距離空間のモデルに変えています。ここでもenrichedカテゴリー論の手法が利用されます。

さらに、文頭を表すトークン'⊥'と文末を表すトークン'⊤'を導入して、'⊥'で始まり'⊤'でおわるプロンプト $x$ に対して、テキスト生成の終了条件を明示的に示しつつ、co-presheafに対応する  $\pi(- | x)$  の計算ができるようになりました。

論文のモデルは、現実のLLMの具体的な振る舞いを解釈することを目指したものです。

# 論文の構成(後半)

今回のセミナーでは紹介できませんが、論文の後半では、マグニチュードの定義と計算について展開しています。

## 3. マグニチュード

1. 定義
2. 半順序集合のマグニチュード
3. 有向グラフの予備知識
4. 主要な結果
5. マグニチュードホモロジー

## 4. 最終的な考察

# LLMとマグニチュード論

拡大されたLLMの理論とmagnitudeの理論を結びつけるという二つ目の課題は、この論文の第三セクション“**3. Magnitude**”で展開されています。

論文では、要素が文字列  $x, y, \dots$  であり、二点間の距離が  $d(x, y) = -\ln n(y | x)$  で与えられる一般化された距離空間  $M$  のmagnitudeが研究されています。こでの $M$ は、LLMの生成する距離空間に他なりません。

論文のタイトル“**The Magnitude of Categories of Texts Enriched by Language Models**”は、このパートが、論文の主要な内容であることを示しています。

# この論文が明らかにしたこと

この論文は、次のような問題で新しい知見が語られています。

- LLMの距離空間  $M$  のマグニチュードをどう定義するのか
- マグニチュードとシャノン・エントロピーとの関係
- magnitude Homology論の展開

それらについては、次回のセミナーで詳しく紹介したいと思います。

## セミナーの構成(論文の補足)

セミナーは基本的には、論文の構成に沿って進むのですが、Bradleyが利用した、有限カテゴリーのマグニチュードについてのVigneauxの新しいアプローチは、別途紹介しようと思っています。

次の論文です。

Juan Pablo Vigneaux

“A combinatorial approach to categorical Möbius inversion and pseudoinversion”

<https://arxiv.org/abs/2407.14647v1>



Bradleyの理論を学ぶ

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# Bradleyの理論を学ぶ

次回のセッションから、2025年のBradley論文(今回はその前半部分)の丁寧な解説を始めます。必要な情報を補いながら、論文のセクションを順番に説明していこうと思います。

前回のセッション「BradleyのLLMのマグニチュード論」は、論文のフォーマルな側面にフォーカスしたものです。

ここでは、前のセッションとは、少し別の角度から、改めてこの論文を読むことの意味について述べてみたいと思います。

大事なことは、LLMやマグニチュード論の理解にとって、Bradleyの論文を学ぶことがどういう意味を持つかということだと思います。

# LLMの働きとして、皆が知っていること

現在のLLMの振る舞いを支えている最もプリミティブな働きは、例えば、Transformerがあるトークンの次に来ることができるトークンの確率を計算し、そのトークンを選び出すことです。

このことは、多くの人がすでに知っていることです。

ただ、こうした理解にとどまる限りでは、現在のLLMを「確率論的オーム」とみなす少し飛躍した議論を、有効に批判することはできません

中心的な問題は、確率的なサンプリングの連続が、なぜ、言語の意味の理解として現れるのかを、説明するのは簡単なことではないということです。

そうした背景から生まれたLLMの能力についての説明の困難や欠如は、「LLMはブラックボックスだ」という批判(残念ながら、この批判には当たっているところもあります)と結びついています。

こうしたLLMをめぐる議論の状況も、多くの人には知っていると思います。

# Bradleyが明らかにしたこと

Bradleyの2022年の論文は、言語  $L$  の中でトークン  $x$  の後に出現できるすべてのトークンを、 $L(x, -)$  で表現して、「これが、 $x$  の意味だ」と考えることによって、LLMの意味理解に大きな突破口を開きました。

その論文は、カテゴリー論の言葉で語られていて、カテゴリー  $L$  の中のオブジェクト  $x$  から発するすべての射の全体を考えること(それが  $L(x, -)$  が表していることです)が、「 $x$  の意味」だとするものでした。それを「co-presheaf意味論」と言います。

同時に、Bradleyの2022年の論文は、このフレームワークを **enriched カテゴリー論**の手法を使って拡大すれば、 $L(x, y)$ を満たすすべての $y$ に対して、 $x$ が与えられたという条件のもとで  $y$  が出現する確率  $\pi(y|x)$  を計算することのできるフレームワークが構成可能であることを理論的に示しました。

# なぜ、Bradleyの2025年の論文が重要なのか

Bradleyの2022年の論文は、co-presheaf / enriched カテゴリーという数学的な言葉で語られていて少し難解でした。また、この論文では、このフレームワークのもとでの $p(y|x)$ の具体的な計算方法は明示的には与えられていませんでした。

僕が、Bradleyの2025年の論文に注目するのは、それが「確率論的サンプリング・マシン」という、広く流布しているLLMの振る舞いのプリミティブな理解と、Bradleyらの抽象的だが斬新な「co-presheaf意味論」を具体的に結びつける方向がわかりやすく示されているからです。

この論文では、あるプロンプトが入力に与えられたときに、LLMがどのようなテキストを返すのかのステップごとの定式化とその分析が行われています。

その分析は、この主題に限って言えば、カテゴリー論に依拠したのではなく初等的なもので、理解しやすいものだと思います。

こうしたLLMの振る舞いの具体的な分析は、今後の「LLMの新しい理論研究」のシリーズで取り上げる予定のVlassopoulosらのLLMの内部動作の数学的分析とともに、LLMを「ブラック・ボックス」ではないものとして捉えようとする重要な研究の流れを形成しています。

# マグニチュード論の新しい展開

Bradleyの2025年の論文には、もう一つ重要な意味があります。前回は、その結論部分を三つほど紹介したのですが、それは、LLMを舞台としたマグニチュード論の新しい展開です。

LLMを母胎とするAI技術は、技術史的にも経済史的にも、歴史を画するものだと思います。ただ、そのインパクトは、技術的・経済的・社会的なものだけにとどまらないと思います。

産業革命を牽引した蒸気機関の性能向上の技術的取り組みが熱力学を生み出し、それが19世紀末には統計力学として整備され、20世紀の量子力学を準備しました。

また、統計力学が理論的な定式化に成功したエントロピーの概念は、形を変えて20世紀の情報科学の中心概念となりました。

今また、LLMの理論の中心に、エントロピーと深いつながりを持つマグニチュードの理論が登場しつつあるのは示唆的です。

# 新しい未来の科学の基礎

Bradleyの理論は、カテゴリー論の応用というスタイルをとっているのですが、そうした数学的形式の新しさも刺激的なものだと思います。

現在進行中のAIの技術的進化とそれを取り巻くさまざまな研究の発展は、新しい未来の科学の基礎を準備していくと僕は考えています。

Bradleyの論文を読むことは、そうしたビジョンを共有する上では、またとないものだと思います。





# Part 2

## LLMの確率計算

### LLMのマグニチュード論 1

#### LLMの確率計算とenrichedカテゴリー論

# LLMの確率計算

LLMは、内部ではどのような確率計算を行なっているのでしょうか？

Bradleyの論文に従って、その挙動を詳しく見ていく このPart 2 は、今回のセミナーの中心的内容の一つです。

次のような問題を取り上げます。

- トークンとテキスト
- LLMとテキスト上の誘導確率
- LLMの確率計算の基本
- 論文の基本定理の証明前半



トークンとテキスト

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# 新しいLLMモデル

「[BTV22]では $p(y|x)$ の明示的な定義は与えられていなかったが、本稿ではこれらの値がLLMによって生成される次トークン確率から実際に生じ得ることを示す。

類似の構成は[GV24]にも見られるが、我々の手法は有限コンテキストサイズに加え、文頭トークン( $\perp$ で表記)と文末トークン( $\dagger$ で表記)も考慮に入れる。

これにより、 $p(-|x)$  は入力  $x$  に対する LLM の終端状態集合  $T(x)$ 、あるいは同値的に可能な出力集合上の確率質量関数と見なせることが証明できる。」

# トークンとテキスト、順序の定義

有限アルファベット  $A$  (例えばトークンの集合) を考え、自由半群  $A^*$  からなる全ての有限文字列  $a, b, \dots$  を考える。

LLM では、トークンは単語、単語の一部、または特殊文字に対応するため、より長いテキスト (例えば文、段落) は  $A^*$  の要素となる。

$A$  の要素をトークンと呼び、 $A^*$  の要素を文字列またはテキストと呼ぶ。

$a$  が  $b$  の接頭辞であるとき、すなわち  $b = aa' \in A^*$  となる  $a'$  が存在するとき、 $a \leq b$  と表記する。

$\leq$  は反射的 ( $A^*$  は空文字  $\epsilon$  を含む)、推移的、反対称的であるため、これは部分順序を定義する。

文字列  $x \in A^*$  の長さを  $|x|$  で表す。

## 二つの特殊なトークンの導入

A の要素に加え、二つの特殊なトークン  $\perp$  と  $\dagger$  を導入する。これらはそれぞれ「文頭」トークンと「文末」トークンを表す。A の要素とは異なり、トークン  $\perp$  と  $\dagger$  は任意の位置に現れない

部分順序集合  $(P, \leq)$  を、オブジェクトの集合  $P$  を持つカテゴリーと見なす。この場合、 $x \rightarrow y$  は  $x \leq y$  である場合にのみ成立する。

$N \in \mathbb{N}$  に対して、 $L := L^{\leq N}$  を  $((A \cup \{\perp, \dagger\})^*, \leq)$  の部分カテゴリーとして定義する。

これは  $\perp$  で始まり、その後  $N - 1$  個以下の記号が続く文字列で構成される。これらの記号は全て  $A$  に属するが、最後の記号は  $\dagger$  である場合もある。

# カテゴリー $L$ のオブジェクトと射

カテゴリー  $L$  のオブジェクトは次のように定義される。

$$\begin{aligned} \text{ob}(L) = & \{ \perp a : a \in A^* \text{ and } |a| \leq N - 1 \} \\ & \cup \{ \perp a \dagger : a \in A^* \text{ and } |a| < N - 1 \} \end{aligned}$$

$\perp a$  の形のオブジェクトを**未完成テキスト**と呼び、  
 $\perp a \dagger$  の形のオブジェクトを**完成テキスト**と呼ぶ。  
 $L$  は Initial Object  $1$  (すなわち  $\perp$ ) を持つ。

$(A \cup \{\perp, \dagger\})^*$  の要素である  $L$  のオブジェクト  $x, y$  が与えられたとき、 $x$  が  $y$  の接頭辞である限り、 $x \rightarrow y$  という射が存在する。  
任意の非恒等射  $x \rightarrow y$  について、接頭辞  $x$  は必ず**未完成テキスト**であり、 $y$  は**完成済み**でも**未完成**でもあり得る。

# Lのオブジェクトの階層構造

L は自然な「階層構造」を持つことに注意。これは文字列  $x$  の長さ  $|x|$  によって与えられる高さ関数を持つという意味である。

集合  $\text{ob}(L)$  を、互いに交わらない和集合で書くのが便利である。ここで  $L^{(i)}$  は長さ  $1+i$  の文字列からなる。

$$\text{ob}(L) = \bigsqcup_{i=0}^{N-1} L^{(i)}$$

恒等写像を除けば、L の射は  $L^{(i)}$  の要素から  $j > i$  となる  $L^{(j)}$  の要素へしか向かわない。

## 部分カテゴリー $L_x$

同様に、 $L$  の文字列  $x$  が与えられたとき、 $L$  の完全部分カテゴリー  $L_x$  が存在する。そのオブジェクトは  $x \rightarrow y$  を満たす  $y \in L$  である。この場合、

$$\text{ob}(L_x) = \coprod_{i=0}^{N-|x|} L_x^{(i)}$$

ここで  $L_x^{(j)}$  は長さ  $|x| + j$  の文字列のみからなり、すなわち  $x$  を右に  $j$  トークン分拡張した文字列である。したがって、

$$\begin{aligned} L_x^{(0)} &= \{x\}, \quad L_x^{(1)} = \{xa_1 : a_1 \in A \cup \{+\}\}, \\ L_x^{(2)} &= \{xa_1a_2 : a_1 \in A, a_2 \in A \cup \{+\}\} \text{となる。} \end{aligned}$$

なお、 $L_{\perp}^{(j)} = L^{(j)}$  である。これは  $j \geq 0$  に対して成り立つ。

また、 $x$  が終端テキストである場合、 $L_x$  は単一の要素  $x$  を持つ。



LLMとテキスト上の誘導確率

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# LLMの振る舞い

確率的大規模言語モデル(例: GPT、Llama)が次トークン予測で訓練された際の挙動を以下のように特徴付けることができます。

- 任意のユーザー入力  $a \in A^*$  に対し、モデルはトークン化されたプロンプト  $x = \perp a$  を入力として受け取り、確率分布  $p_x := p(-|x) : A \cup \{+\} \rightarrow [0, 1]$  を生成する。  
$$p_{\perp a}(-|\perp a)$$
- $p_x$  を生成した後、モデルは  $p_x$  に従ってトークン  $a_1$  をサンプリングする。  
$$p_{\perp a}(a_1|\perp a)$$
- $a_1 = +$  の場合、処理は終了しモデルは  $a$  を出力する。そうでない場合、トークン  $a_1$  は  $a$  に追加される。これらの手順は繰り返される。

もう少し続けてみましょう。

モデルは  $A \cup \{+\}$  上の確率分布  $p_{\perp aa_1}$  を生成し、それに従ってトークン  $a_2$  をサンプリングします。 $a_2 = +$  の場合 ( $aa_1$  を出力して) 実行を終了するか、 $a_2$  を  $aa_1$  に追加して確率分布  $p_{\perp aa_1 a_2}$  を生成して、この処理を反復します。

$$\begin{aligned} & p_{\perp a}(a_1 | \perp a) \\ & p_{\perp aa_1}(a_2 | \perp aa_1) \\ & p_{\perp aa_1 a_2}(a_3 | \perp aa_1 a_2) \\ & p_{\perp aa_1 a_2}(a_4 | \perp aa_1 a_2 a_3) \\ & \dots \dots \end{aligned}$$

# LLMのコンテキスト・サイズ

†がサンプリングされた時、あるいは拡張プロンプト  $y = xa_1 \dots \cdot a_{N-|x|}$  が最大長  $N$  に達した時点で終了すると仮定します。

$N$  をカットオフと呼びます。 $N - 1$  は言語モデルのコンテキスト・サイズ、すなわち分布  $px$  を生成するために使用できる文字列  $x$  の最大長と見なせます。

## 終端状態集合 $T(x)$

したがって、入力  $x = \perp a$  に対する終端状態の集合を以下のように定義することができます。

定義 1. 言語  $L$  における未完成テキスト  $x$  の終端状態の集合  $T(x)$  は次のように定義される。

$$T(x) = \{ y \in ob(L_x) : y \text{ は長さ } N \text{ の未完成テキスト} \\ \text{あるいは、} y \text{ は } |y| \leq N \text{ である完成テキスト} \}$$

# モデルの出力と終端状態は一対一に対応する

第一の場合 (yが未完成テキストで、カットオフが発生した場合)

終端状態  $y = xa'$  となるような  $a' \in A^*$  が存在し、  
 $|a'| = N - |x|$  です。このような終端状態に対して、モデルは  
ユーザーに  $xa'$  を出力します。

第二の場合 (yが完成テキストの場合)

終端状態  $y = xa''\dagger$  となるような  $a'' \in A^*$  が存在し、  
 $|a''| \leq N - |x| - 1$  を満たします。このような終端状態に対して、  
モデルは  $xa''$  を出力します。

いずれの場合も、モデルの出力と終端状態の間には一対一対応  
が存在します。

ここで、ユーザーの入力  $a$  に対するモデルの出力  $b$  の確率  $\pi(b|a)$  を、その生成に関与するトークンの中間確率の積として定義したくなる かもしれません。

例えば、 $N > 5$  の場合、ユーザーのプロンプト  $a = a_1 a_2$  を与えられたときに出力  $b = a_1 a_2 a_3 a_4 a_5 \in A^*$  を生成する確率を

$$\pi(a_1 a_2 a_3 a_4 a_5 | a = a_1 a_2) = p(a_3 | \perp a) \cdot p(a_4 | \perp a a_3) \cdot p(a_5 | \perp a a_3 a_4) \cdot p(\dagger | \perp a a_3 a_4 a_5)$$

のように。

## 関数 $\pi(y|x)$ の定義

しかし、この規則が確かに何らかの集合上の確率質量関数を定義しているとは明らかではありません。まず最初に、これらの積を一般に定義しておきましょう。

定義 2.  $L$ の任意のオブジェクト  $x, y$ について

$$\pi(y|x) := \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \not\rightarrow y \\ \prod_{i=1}^k p(a_{t+i}|y_{<t+i}) & \text{if } x \rightarrow y \end{cases} .$$

$$p(a_3 | \perp a) \cdot p(a_4 | \perp aa_3) \cdot p(a_5 | \perp aa_3a_4) \cdot p(\dagger | \perp aa_3a_4a_5)$$

第三の場合、 $x = \perp a_1 \cdots a_t$  かつ  $y = xa_{t+1} \cdots a_{t+k}$  とする。

ここで  $k \geq 1$ 、 $(a_i)_{i=1}^{t+k-1} \subset A$ 、かつ  $a_{t+k} \in A \cup \{+\}$  である。

記号  $y_{<t+i}$  は文字列  $\perp a_1 \cdots a_{t+i-1}$  を表し、 $y_{<t+1} = x$  である。

$\mathcal{L}$  に属する与えられた文字列  $x$  に対して、関数  $\pi(-|x)$  はその支持集合全体上では確率質量関数ではない。(そもそも  $\pi(x|x) = 1$  である。)

しかしながら、 $T(x)$  に制限された場合、それは確率質量関数となる。

# 確率質量関数とは

確率質量関数は、離散確率変数(数えられる値しかとらない変数)が特定の値をとる確率を割り当てる関数です。

具体的には、以下の2つの条件を満たす関数を指します。

- 各事象の確率が非負であること:

$$P(X = x_i) \geq 0$$

- すべての可能な事象の確率の合計が1であること:

$$\sum_i P(X = x_i) = 1$$

終端状態集合 $T(x)$ 上で、関数  $\pi(-|x)$  が確率質量関数であることを言うためには、確率質量関数であるための先の二つの条件から、次のことを示す必要があります。

- 各事象の確率が非負であることから、  
ある特定の終了状態  $x_i$  が発生する確率は0以上である
- すべての可能な事象の確率の合計が1であることから、  
すべての可能な終了状態の確率を合計すると1になる

# T(x) に制限された 関数 $\pi(-|x)$ は確率質量関数である

命題1. 言語Lにおける未完成テキストxが与えられたとき、関数  $\pi(-|x)|_{T(x)}$  は入力xの終端状態上の確率質量関数である。

これは重要な定理です。Bradley論文の前半の基本定理と呼んでいいものです。目標は、次の式が成り立つことを示すことです。

$$\sum_{y \in T(x)} \pi(y|x) = 1$$

でも証明は少し複雑です。

次回のセッションで、まず、LLMの確率計算の基本を確認しておきましょう。**Appendix A**で、証明の詳細を示すことにします。



# LLMの確率計算の基本

## LLMのマグニチュード論 1

### LLMの確率計算とenrichedカテゴリー論

# LLMの確率計算の基本

このセッションでは、「命題 1」の証明に必要なLLMの確率計算の基本を確認したいと思います。証明の詳細は、**Appendix A**で行います。

「命題 1」は、次のことを主張しています。

命題1. 言語 $L$ における未完成テキスト $x$ が与えられたとき、関数  $\pi(-|x)|_{T(x)}$  は入力 $x$ の終端状態上の確率質量関数である。

# 基本的な用語の確認

「命題 1」に出てくる基本的な言葉の意味を確認しておきましょう。

## 未完成テキスト

$\perp a$  の形のオブジェクトを「未完成テキスト」と呼び、  
 $\perp a^\dagger$  の形のオブジェクトを「完成テキスト」と呼ぶ。

## 終端状態集合 $T(x)$

定義 1. 言語  $L$  における未完成テキスト  $x$  の  
終端状態の集合  $T(x)$  は次のように定義される。

$$T(x) = \{ y \in ob(L_x) : y \text{ は長さ } N \text{ の未完成テキスト} \\ \text{あるいは、} y \text{ は } |y| \leq N \text{ である完成テキスト} \}$$

## 確率質量関数

確率質量関数は、離散確率変数(数えられる値しかとらない変数)が特定の値をとる確率分布を割り当てる関数です。

具体的には、以下の2つの条件を満たす関数を指します。

- 各事象の確率が非負であること:

$$P(X = x_i) \geq 0$$

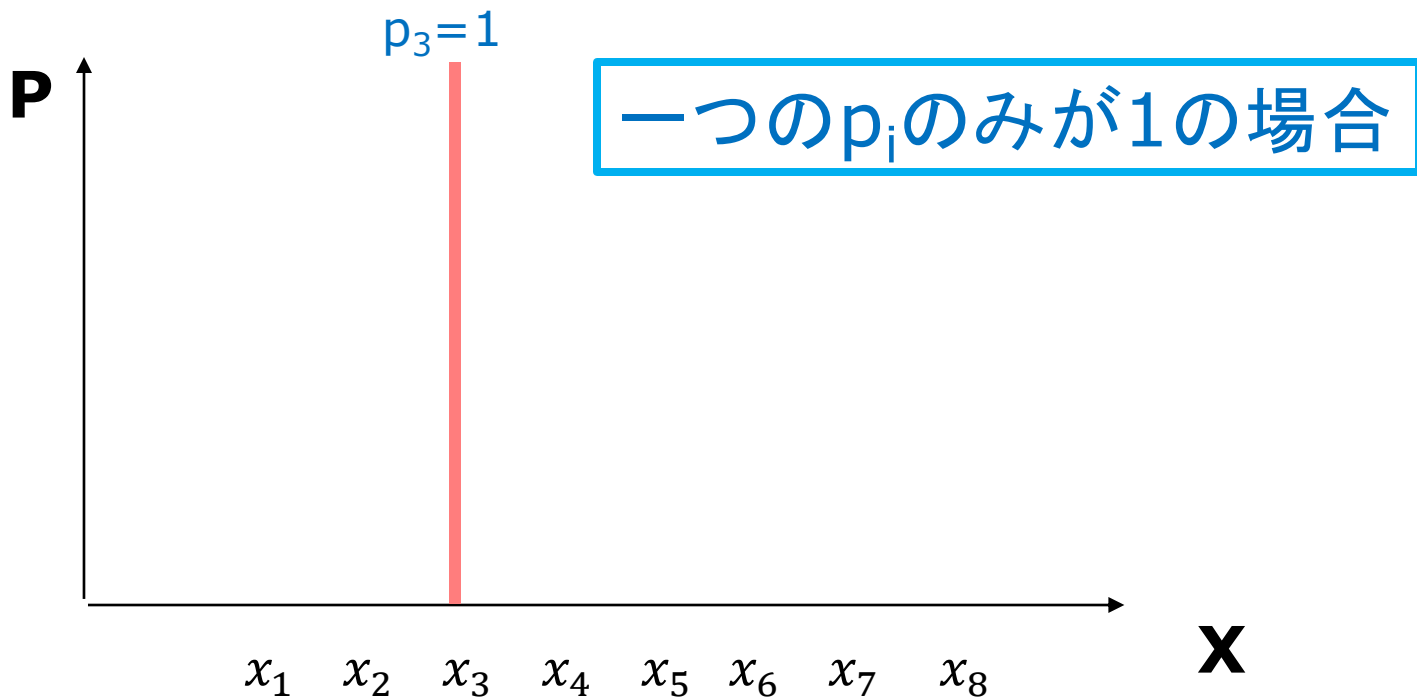
- すべての可能な事象の確率の合計が1であること:

$$\sum_i P(X = x_i) = 1$$

この時、この確率分布の、エントロピーは、

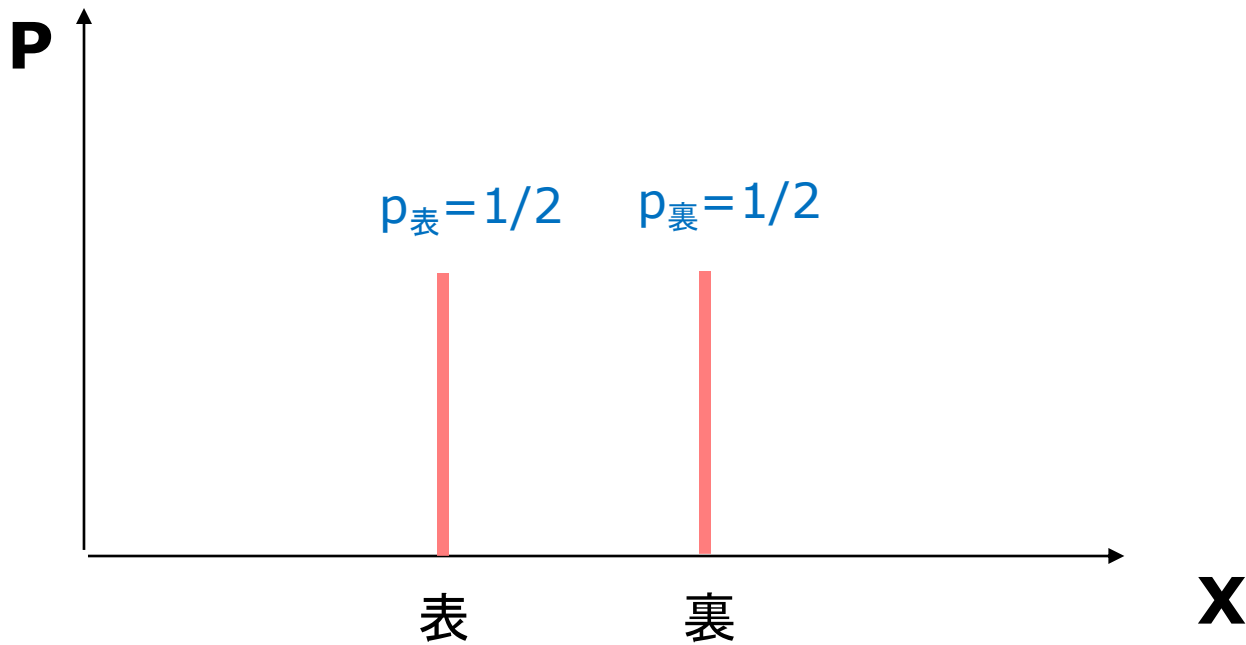
$$-\sum_i p_i \log p_i$$

## 確率質量関数の例



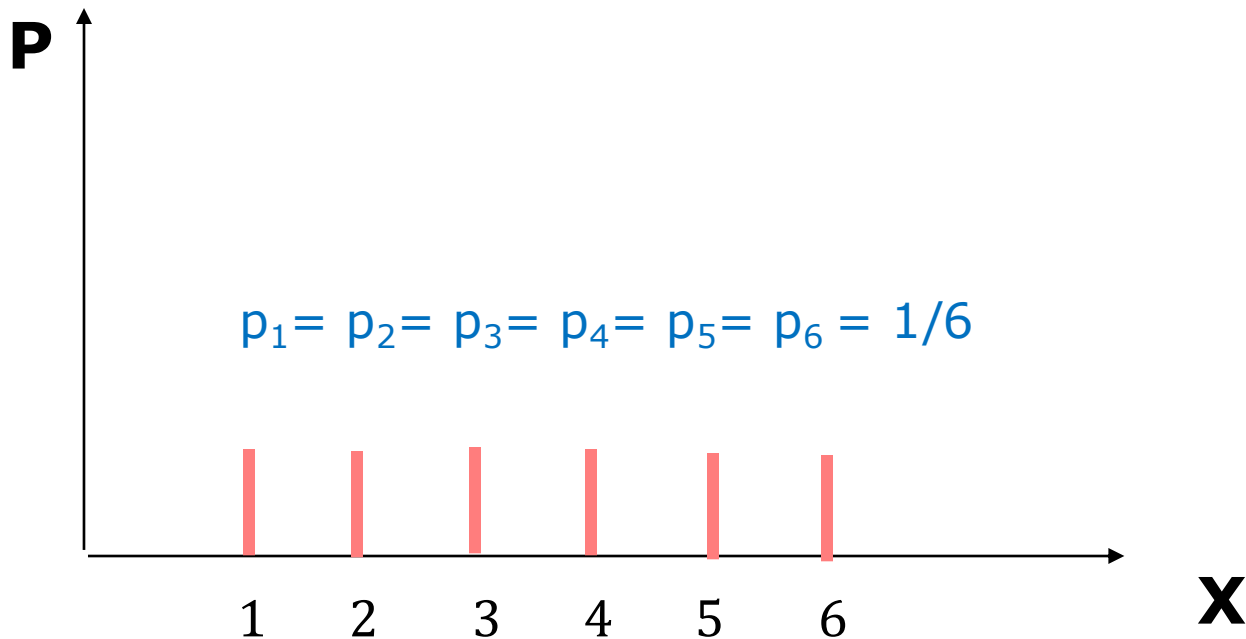
この例では、 $X$ が $x_3$ の値を取る場合のみ、 $p_3=1$ である。  
この時、 $H(X)=1 \cdot \log(1/1)=1 \cdot \log 1=1 \cdot 0=0$   
(ただし、 $0 \cdot \log(1/0)=0$ とした)

# コイン・トスの確率分布



コインを投げた時、表が出る確率は $1/2$ 、裏が出る確率は $1/2$ 。  
 $H(X) = 1/2 \cdot \log 2 + 1/2 \cdot \log 2 = \log 2 = 1$   
コイン・トスの確率分布のエントロピーは、1 bitである。

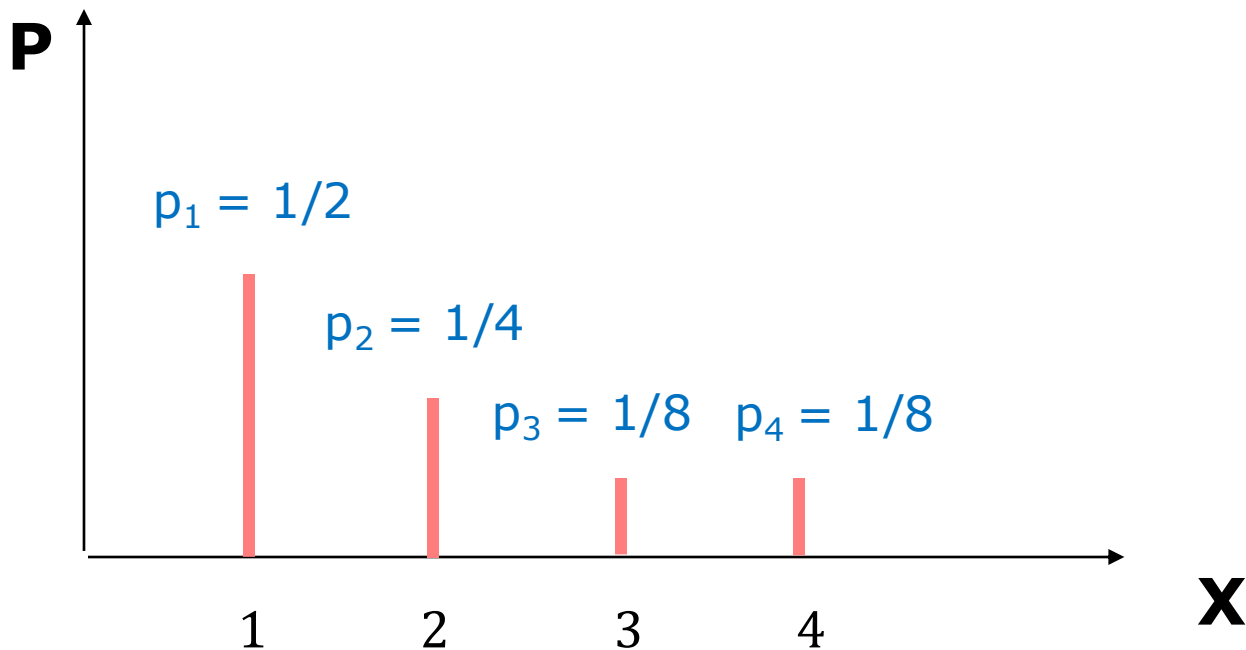
# サイコロの出る目の確率分布



$p_1 = p_2 = p_3 = p_4 = p_5 = p_6 = 1/6$  だから

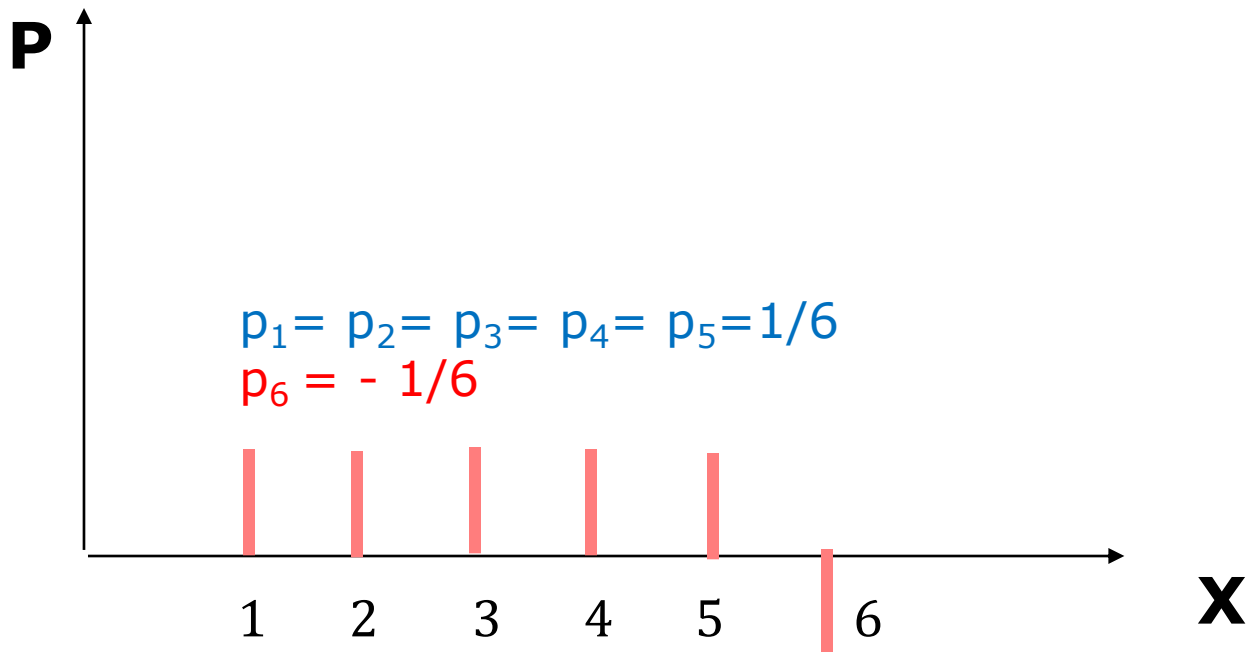
$$H(X) = 1/6 \cdot (\log 6 + \log 6 + \log 6 + \log 6 + \log 6 + \log 6) = \log 6 \\ = 1 + \log 3 \text{ bit}$$

次のような確率分布が与えられた時の  
エントロピーは 1.75 bit



$$\begin{aligned} H(X) &= 1/2 \cdot \log 2 + 1/4 \cdot \log 4 + 1/8 \cdot \log 8 + 1/8 \cdot \log 8 \\ &= 1/2 \cdot \log 2 + 1/4 \cdot \log 2^2 + 1/8 \cdot \log 2^3 + 1/8 \cdot \log 2^3 \\ &= 1/2 \cdot 1 + 1/4 \cdot 2 + 1/8 \cdot 3 + 1/8 \cdot 3 = 1/2 + 1/2 + 2 \cdot 3/8 \\ &= 1.75 \text{ bit} \end{aligned}$$

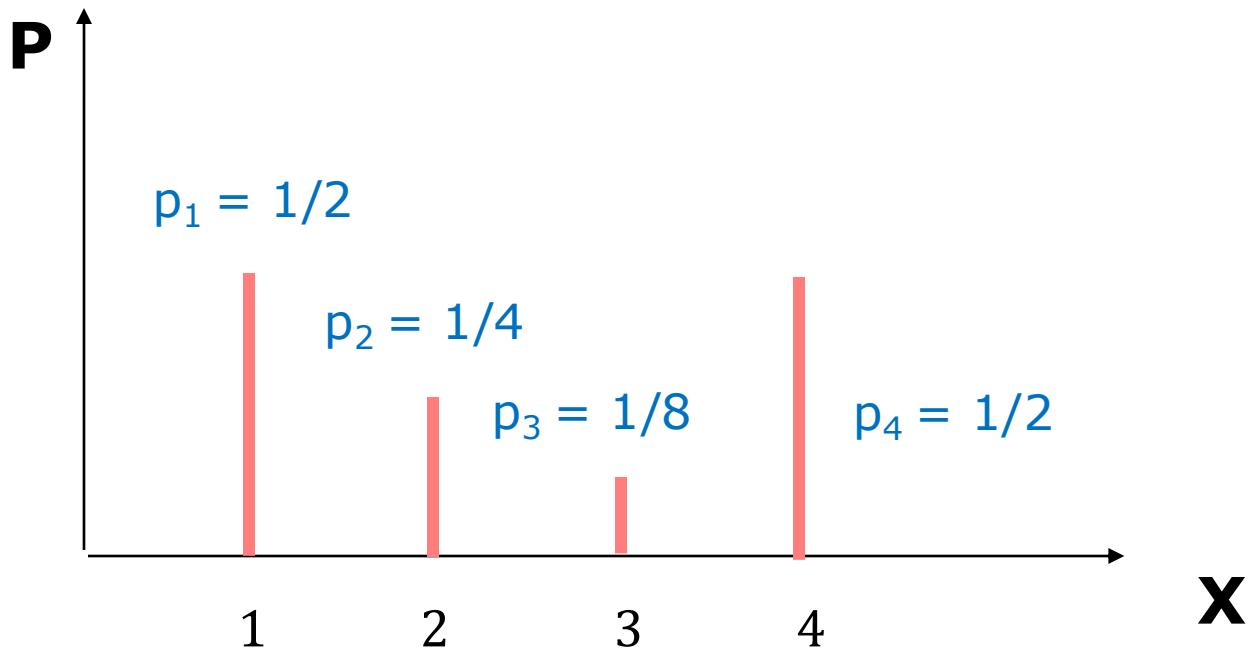
# 確率分布ではない分布



各事象の確率が非負であること:

$$P(X = x_i) \geq 0$$

# 確率分布ではない分布



すべての可能な事象の確率の合計が1であること:

$$\sum_i P(X = x_i) = 1$$

# 基本的定義の振り返り

- カテゴリーLのオブジェクト  $ob(L)$

$$ob(L) = \{\perp a : a \in A^* \text{ and } |a| \leq N - 1\} \\ \sqcup \{\perp a \dagger : a \in A^* \text{ and } |a| < N - 1\}$$

$\perp a$  の形のオブジェクトを**未完成テキスト**と呼び、  
 $\perp a \dagger$  の形のオブジェクトを**完成テキスト**と呼ぶ。

カテゴリーLのオブジェクトは  
未完成テキストと完成テキストからなる

## ● カテゴリーLの射 $x \rightarrow y$

$(A \cup \{\perp, \top\})^*$ の要素であるLのオブジェクト  $x, y$  が与えられたとき、 $x$  が  $y$  の接頭辞である限り、 $x \rightarrow y$  という射が存在する。

任意の非恒等射  $x \rightarrow y$  について、接頭辞  $x$  は必ず未完成テキストであり、 $y$  は完成済みでも未完成でもあり得る。

## ● 部分カテゴリー $L_x$

$L$  の文字列  $x$  が与えられたとき、 $L$  の完全部分カテゴリー  $L_x$  が存在する。そのオブジェクトは  $x \rightarrow y$  を満たす  $y \in L$  である

$$\text{ob}(L_x) = \prod_{i=0}^{N-|x|} L_x^{(i)}$$

ここで  $L_x^{(j)}$  は長さ  $|x| + j$  の文字列のみからなり、すなわち  $x$  を右に  $j$  トークン分拡張した文字列である。

$$L_x^{(0)} = \{x\},$$

$$L_x^{(1)} = \{xa_1 : a_1 \in A \cup \{\dagger\}\},$$

$$L_x^{(2)} = \{xa_1a_2 : a_1 \in A, a_2 \in A \cup \{\dagger\}\}$$

$$L_x^{(3)} = \{xa_1a_2a_3 : a_1 \in A, a_2 \in A, a_3 \in A \cup \{\dagger\}\}$$

## 関数 $\pi(y|x)$ の定義

重要なのは、次の関数  $\pi(y|x)$  の定義です。

定義 2.  $L$ の任意のオブジェクト  $x, y$ について

$$\pi(y|x) := \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \not\rightarrow y \\ \prod_{i=1}^k p(a_{t+i}|y_{<t+i}) & \text{if } x \rightarrow y \end{cases} .$$

$$p(a_3 | \perp a) \cdot p(a_4 | \perp aa_3) \cdot p(a_5 | \perp aa_3a_4) \cdot p(\dagger | \perp aa_3a_4a_5)$$

## 関数 $\pi(y|x)$ と値 $p(a_{t+i}|y_{<t+i})$

「定義 2」は、 $x \rightarrow y$  のとき。関数  $\pi(y|x)$  を  $k$  個の値  $p(a_{t+i}|y_{<t+i})$  の積として定義しています。次のように

$$\begin{array}{c} x \rightarrow y \text{ のとき} \\ \pi(y|x) = \prod_{i=1}^k p(a_{t+i}|y_{<t+i}) \end{array}$$

論文の前半の中心的な定理である「命題 1」は、こうして定義された関数  $\pi(y|x)$  が、 $T(x)$  上に制限すると、「確率質量関数」であること、すなわち関数  $\pi(-|x)$  が確率分布を、 $\pi(y|x)$  が確率を与えることを主張しています。

# 確率分布 $p_x(-|x)$ の生成と その分布の下でのサンプリング

LLMの働きを振り返ってみましょう。

- LLMは、テキスト  $x$  が与えられた時、次に出現するトークンを予測する確率分布  $p_x(-|x)$  を生成します。
- LLMは、一つのトークン  $a$  を選んで  $x$  に追加して、テキストを一つ分延長して  $xa$  にします。
- このとき、 $x$  の後ろに一つのトークン  $a$  が追加される確率は、 $p_x(a|x)$  になります。

## $p(a_{t+i}|y_{<t+i})$ の値は、どう与えられるか

$\pi(y|x)$ の定義に出てくる  $p(a_{t+i}|y_{<t+i})$ の値はどう計算されるのでしょうか？見やすいように、 $a = a_{t+i}$ ,  $z = y_{<t+i}$  としましょう。

- LLMは、テキスト  $z$  が与えられた時、次に出現するトークンを予測する確率分布  $p_z(-|z)$  を生成します。
- LLMは、トークン  $a$  をサンプリングします。
- $z$ の次のトークンが $a$ になる確率は、 $p_z(a|z)$ になります。

$p(a_{t+i}|y_{<t+i})$ の値は、テキスト $y_{<t+i}$ が与えられた時、その次のトークンが $a_{t+i}$ になる確率です。

# Next Token 確率 $p_x(a|x)$

テキスト  $x$  が与えられた時、次に出現するトークンを予測する確率分布  $p_x(-|x)$  を、「Next Token 確率分布」といいます。

この分布の下で  $a$  をサンプリングして、テキスト  $x$  の次のトークンが  $a$  となることを表す確率  $p_x(a|x)$  を「Next Token 確率」と呼びます。

定義 2 の  $\pi(y|x)$  の定義は、もし、 $x \rightarrow y$  である  $y$  が、 $x$  に  $k$  個のトークンを追加したものなら、その値は、 $k$  個の Next Token 確率の積で定義されるということです。

$$\pi(y|x) = \prod_{i=1}^k \boxed{p(a_{t+i}|y_{<t+i})}$$

Next Token 確率

ただし、 $p_x(a|x)$  の  $p$  の添え字の  $x$  は省略されている

# テキストへの1-トークンの追加と Next Token 確率

$$x = \perp x \rightarrow y_1 = \perp x a_1$$

確率  $p(a_1|x)$

$$y_1 = \perp x a_1 \rightarrow y_2 = \perp x a_1 a_2$$

確率  $p(a_2|y_1)$

$$y_2 = \perp x a_1 a_2 \rightarrow y_3 = \perp x a_1 a_2 a_3$$

確率  $p(a_3|y_2)$

$$y_3 = \perp x a_1 a_2 a_3 \rightarrow y_4 = \perp x a_1 a_2 a_3 a_4$$

確率  $p(a_4|y_3)$

$$y_4 = \perp x a_1 a_2 a_3 a_4 \rightarrow y_5 = \perp x a_1 a_2 a_3 a_4 a_5$$

確率  $p(a_5|y_4)$

$$y_5 = \perp x a_1 a_2 a_3 a_4 a_5 \rightarrow y_6 = \perp x a_1 a_2 a_3 a_4 a_5 a_6$$

確率  $p(a_6|y_5)$

# Next Token 確率の $p(a_{t+i} | y_{<t+i})$ という表記

$t$  は、トークン追加の起点を表しています。最初は  $t = 0$

$$x = \perp x \rightarrow y_1 = \perp x a_1 ; y_{<1} = x$$

$$\text{確率 } p(a_1 | x) = p(a_1 | y_{<1})$$

$$y_1 = \perp x a_1 \rightarrow y_2 = \perp x a_1 a_2 ; y_{<2} = y_1 = \perp x a_1$$

$$\text{確率 } p(a_2 | y_1) = p(a_2 | y_{<2})$$

$$y_2 = \perp x a_1 a_2 \rightarrow y_3 = \perp x a_1 a_2 a_3 ; y_{<3} = y_2 = \perp x a_1 a_2$$

$$\text{確率 } p(a_3 | y_2) = p(a_3 | y_{<3})$$

$$y_3 = \perp x a_1 a_2 a_3 \rightarrow y_4 = \perp x a_1 a_2 a_3 a_4 ; y_{<4} = y_3 = \perp x a_1 a_2 a_3$$

$$\text{確率 } p(a_4 | y_3) = p(a_4 | y_{<4})$$

同様に  $y_4 = \perp x a_1 a_2 a_3 a_4 \rightarrow y_5 = \perp x a_1 a_2 a_3 a_4 a_5$

$$\text{確率 } p(a_5 | y_4) = p(a_5 | y_{<5})$$

$$y_5 = \perp x a_1 a_2 a_3 a_4 a_5 \rightarrow y_6 = \perp x a_1 a_2 a_3 a_4 a_5 a_6$$

$$\text{確率 } p(a_6 | y_5) = p(a_6 | y_{<6})$$

# 関数 $\pi(y|x)$ と Next Token 確率 $p(a|x)$

$x, y, z$  が  $x \rightarrow y \rightarrow z$  を満たす3つの文字列であるとしましょう。  
この時、 $(a_i)_{i=1}^{t+k+k'-1} \subset A$ 、かつ  $a_{t+k+k'} \in A \cup \{+\}$  として、次のように記述ができます。

$$\begin{aligned} x &= \perp \overbrace{a_1 \cdots a_t}^t \\ y &= \perp a_1 \cdots a_t \overbrace{a_{t+1} \cdots a_{t+k}}^k \\ z &= \perp a_1 \cdots a_t a_{t+1} \cdots a_{t+k} \overbrace{a_{t+k+1} \cdots a_{t+k+k'}}^{k'} \end{aligned}$$

文字列  $x$  は、 $t$ 個のトークンからなり、  
文字列  $y$  は、文字列  $x$  の後ろに $k$ 個のトークンを追加したもので、  
文字列  $z$  は、文字列  $y$  の後ろに $k'$ 個のトークンを追加したものです。

関数  $\pi(y|x)$ でも  $\pi(z|x)$ でも、関数 $\pi$ は、 $x \rightarrow y \rightarrow z$  を満たす任意の長さの文字列間の関係を表現できます。

ただ、Next Token 確率を表す  $p(a|x)$ は、 $x$ の長さは、任意ですが、 $a$ は必ず一つのトークンでなければなりません。

実は、「命題 1」の証明ができれば、関数  $\pi(y|x)$  は、「**プロンプト文字列  $x$  が与えられた時、LLMの出力が $y$ となる確率**」として解釈できるのですが、それはこれからの課題です。

「命題 1」の証明後には、 $\pi(y|x)$ で表される確率を、「**パス確率**」と呼ぶことがあります。

## 文字列の分解とパス確率 $\pi(xa|x)$ の展開

「パス確率」という考えを使うと、next token 確率と合わせて、次のような導出ができるようになります。

いま、文字列  $a$  が、 $m$  個のトークンからできているとします。この文字列の最後のトークンを  $a''$  とし、その前の  $m - 1$  個のトークンからできている文字列を  $a'$  として文字列  $a$  を  $a'a''$  に分解します。

$$\begin{aligned} a &= a'a'' \\ a \in A^m &= A^{m-1} \times A, \\ a' \in A^{m-1}, a'' &\in A \end{aligned}$$

この時、

$$\pi(xa|x) = \pi(xa'a''|x) = \pi(xa'|x) \cdot p(a''|xa')$$

が成り立ちます。

$$\pi(xa|x) = \pi(xa'a''|x) = \pi(xa'|x) \cdot p(a''|xa')$$

$\pi(xa'|x)$  は、プロンプト  $x$  から中間テキスト  $xa'$  までの**パス確率**です。

$p(a''|xa')$  は、テキスト  $xa'$  が与えられたときの、次のトークンとして  $a''$  が生成される**next token確率**です。

ここでは、**パス確率**ということばを使っていますが、 $\pi(xa|x)$  が確率質量関数であるという証明されるべき事実を利用しているわけではありません。

定義 2 の **m-1** 個の next token 確率の積を、 $\pi(xa'|x)$  とみなしているだけです。



# 論文前半の基本定理の証明の概要

## LLMのマグニチュード論 1

## LLMの確率計算とenrichedカテゴリー論

# 論文前半の基本定理の証明の概要

先のセッションでは、Bradleyの2025年の論文の前半部分の中心的な内容である「命題 1」の証明に必要なLLMの確率計算の基本を見てきました。

このセッションでは、「命題 1」の証明の概要を見ていきたいと思っています。

証明の詳細は、**Appendix A** を参照ください。

# 「命題 1」の証明の概略

命題1. 言語Lにおける未完成テキストxが与えられたとき、関数 $\pi(-|x)|_{T(x)}$ は入力xの終端状態上の確率質量関数である。

目標は、次の式が成り立つことを示すことです。

$$\sum_{y \in T(x)} \pi(y|x) = 1$$

基本的には、この主張を、xの後にモデルが終了するまで続く最大記号数  $m = N - |x|$  についての帰納法で証明します。

## 帰納法の仮定 $m=0, 1$ の場合

$m = 0$  の場合、 $T(x) = \{x\}$  であり、定義より  $\pi(x|x) = 1$  である。

$m = 1$  の場合、 $T(x) = \{xa \mid a \in A \cup \{\dagger\}\}$  であり、

$$\sum_{y \in T(x)} \pi(y|x) = \sum_{a \in A \cup \{\dagger\}} p_x(a|x) = 1$$

よって、 $m=0, 1$  の場合には、命題1が成り立っていることがわかります。

## 帰納法のステップ

論文では、まず、一般の  $m \geq 1$  に対して、

$$\sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^m} \pi(xa|x)$$

と表されることを示します、

最終的に、この式の右辺が次の式に変形できることを示します。

$$= \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^{m-1}} \pi(xa'|x)$$

帰納法の仮定から、この式は 1 に等しくなります。よって、

$$\sum_{y \in T(x)} \pi(y|x) = 1$$

$x$  の後にモデルが終了するまで続く最大記号数  $m = N - |x|$  についての帰納法

# 論文での 命題 1 の証明の四つのヒント

論文では、証明の詳細は省略されています。

次の四つのステップがヒントとして示されています。

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^m} \pi(xa|x)$$

$$(4) \quad = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{\substack{a=a'a'' \\ a' \in A^{m-1}, a'' \in A}} p(a''|xa') \pi(xa'|x)$$

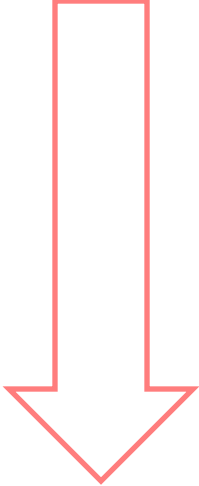
$$(5) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a' \in A^{m-1}} \pi(xa'|x) \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa')$$

$$(6) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a' \in A^{m-1}} \pi(xa'|x)$$

# 命題 1 の証明の帰納法のステップ

先の証明の概略で述べたのは、式 3 から式 6 の導出が出来ればいいということです。

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a \in A^m} \pi(xa|x)$$


$$(4) \quad = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{\substack{a=a'a'' \\ a' \in A^{m-1}, a'' \in A}} p(a''|xa') \pi(xa'|x)$$

$$(5) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x) \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa')$$

$$(6) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x)$$

# 命題 1 の証明の四つのステップ

先の証明の概略で述べたのは、式 3 から式 6 の導出が出来ればいいということです。

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a \in A^m} \pi(xa|x)$$

$$(4) \quad = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{\substack{a=a'a'' \\ a' \in A^{m-1}, a'' \in A}} p(a''|xa') \pi(xa'|x)$$

$$(5) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x) \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa')$$

$$(6) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x)$$

# 証明の詳細について

この続きは、**Appendix A** を参照ください。





# Part 3

Enrichedカテゴリー論とLLMモデルの拡大

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論



LLMへのenrichedカテゴリー論の応用

LLMのマグニチュード論 1

LLMの確率計算とenrichedカテゴリー論

# LLMへのenrichedカテゴリー論の応用

このセッションでは、前回導いた終端状態集合  $T(x)$  上で定義された関数  $\pi(y|x)$  を用いて、カテゴリー  $\mathcal{L}$  をenrich化されたカテゴリーに拡張します。

具体的には、LLMによって生成される確率  $\pi$  から、次の二つのenrichedカテゴリーの導出を行います。

- $[0, 1]$ -カテゴリー  $\mathcal{L}$
- $[0, \infty]$ -カテゴリー  $\mathcal{M}$

二つのカテゴリーのオブジェクトは、これまで見てきたカテゴリー  $\mathcal{L}$  と同じく、ある有限トークン集合からの文字列で構成されています。

# enrichedカテゴリー論の基礎を確認する

enriched カテゴリー論の基礎については、

前回のセミナー「マグニチュード論の展開」の

<https://www.marulabo.net/docs/llm1bradley/>

「Part 2 enriched カテゴリー論とマグニチュード」

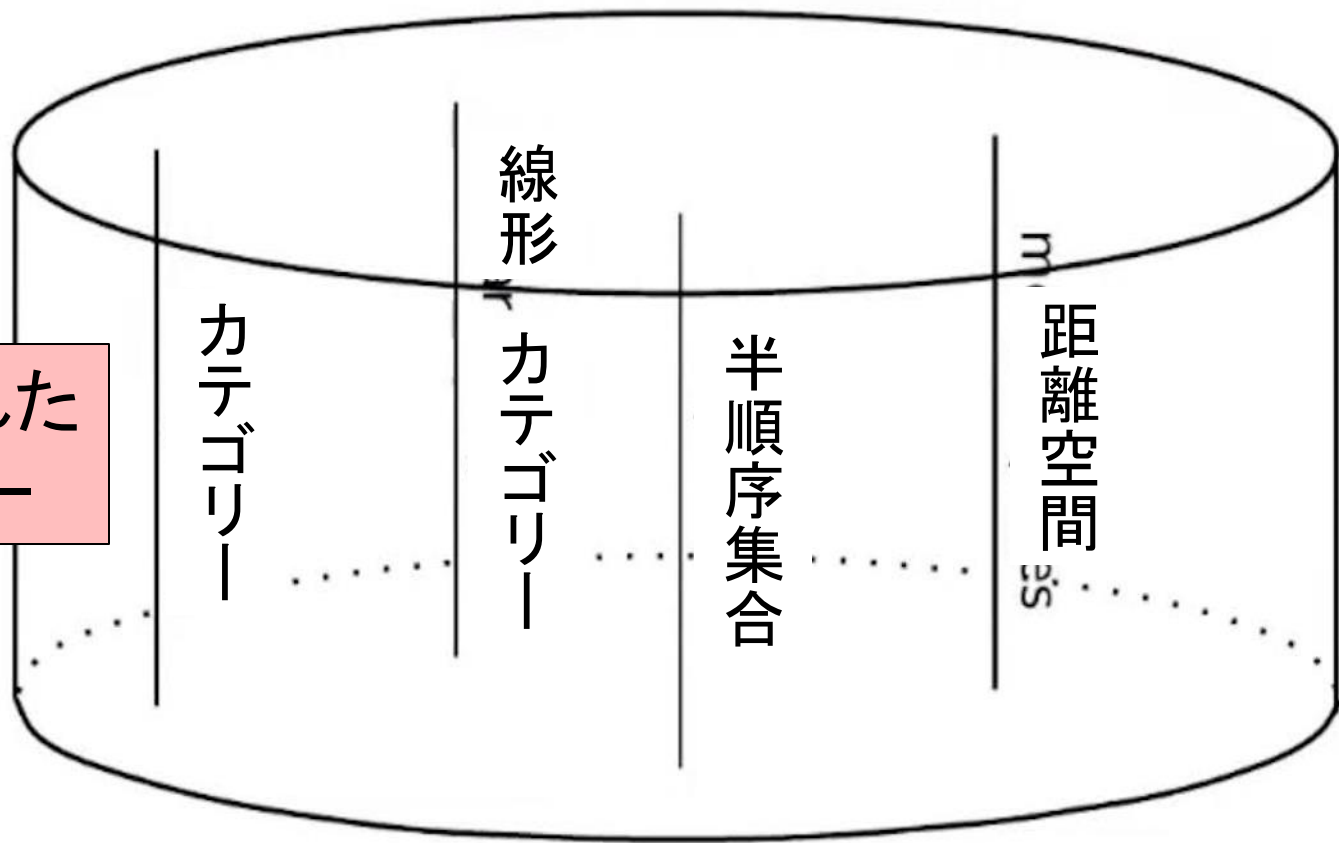
<https://www.marulabo.net/docs/llm1bradley/> -

[Part 2 enriched カテゴリー論とマグニチュード](#)

の資料pdfとビデオを参照ください。

# monoidal カテゴリーVとenrich化されたV-カテゴリー

enrich化された  
V-カテゴリー



monoidal  
カテゴリーV

- (Set, ×)
- (Vect, ⊗)
- ([0, ∞], +)
- ((0 → 1), ∧)

# Leinsterのダイヤル



# monoidalカテゴリー 前順序V上で enrich化されたカテゴリーの定義とサンプル

本論に入る前に、もう一度 Bradleyがこの論文で示している monoidalカテゴリー  $V$ と $V$ 上で定義されたenrich化されたカテゴリーの定義とサンプルを見ておきましょう。

ここでは、 $V$ が前順序の場合のみが例示されています。LLMへの適用には、これで十分です。

なお、このセッション資料は、Bradley論文を、ほとんど逐条翻訳しています。定義や命題、数式のナンバリングには、論文の参照がしやすいように、Bradleyの論文と同じ番号をふってあります。

**定義 3.** 可換(commutative) monoidal カテゴリー前順序  $(V, \leq, \otimes, 1)$  とは、前順序集合  $(V, \leq)$  と可換monoidalカテゴリー  $(V, \otimes, 1)$  からなり、 $x \leq x'$  かつ  $y \leq y'$  のとき  $x \otimes y \leq x' \otimes y'$  を満たすものである。

**定義 4.**  $(V, \leq, \otimes, 1)$  を、可換monoidal前順序とする。  
 $V$  上にenrich化されたカテゴリー、あるいは単に  $V$ -カテゴリー  $C$  は、オブジェクトの集合  $\text{ob}(C)$  と、オブジェクト  $x$  と  $y$  のすべての組に対して、 $V$ -オブジェクトと呼ばれる  $V$  のオブジェクト  $C(x, y)$  からなる。 $C(x, y)$  は以下の条件を満たす：

すべてのオブジェクト  $x, y, z \in \text{ob}(C)$  に対して、

$$1 \leq C(x, x)$$

$$C(y, z) \otimes C(x, y) \leq C(x, z)$$

## monoidal カテゴリーVの二つの例

**例1. 単位区間**  $([0, 1], \leq, \cdot, 1)$  は、通常の順序関係  $\leq$  を持つ可換 monoidal 前順序である。実数の乗法は monoid の積であり、これを並置で表す。すなわち、 $a, b \in [0, 1]$  に対して  $ab := a \cdot b$  とし、monoid の単位元は 1 である。

**例 2. 拡張非負実数**  $([0, \infty], \geq, +, 0)$  は可換 monoidal 順序構造をなす。ここで順序関係は実数上の通常の順序関係の反対順序である。monoid 積は加法であり、 $a + \infty := \infty$  および  $\infty + a := \infty$  ( $a \in [0, \infty]$ ) を満たす。monoid 単位元は 0 である。

## 関数 $\pi(y|x)$ の定義の振り返り

定義 2.  $L$ の任意のオブジェクト  $x, y$ について

$$\pi(y|x) := \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \not\rightarrow y \\ \prod_{i=1}^k p(a_{t+i}|y_{<t+i}) & \text{if } x \rightarrow y \end{cases}$$

第三の場合、 $x = \perp a_1 \cdots a_t$  かつ  $y = x a_{t+1} \cdots a_{t+k}$

記号  $y_{<t+i}$  は文字列  $\perp a_1 \cdots a_{t+i-1}$  を表し、 $y_{<t+1} = x$  である。

$$\begin{aligned} & \pi(a_1 a_2 a_3 a_4 a_5 | a = a_1 a_2) = \\ & p(a_3 | \perp a) \cdot p(a_4 | \perp a a_3) \cdot p(a_5 | \perp a a_3 a_4) \cdot p(\dagger | \perp a a_3 a_4 a_5) \end{aligned}$$

## すべてのLLMは、 [0, 1]区間上のカテゴリーを定義する

$\pi$ が与えられた時、単位区間  $[0, 1]$ 上でenrich化にされたカテゴリー  $\mathcal{L}$ を次のように定義できます。

$\mathcal{L}$ のオブジェクトはLのオブジェクトと一致し、オブジェクトxとyの組に対して、その $[0, 1]$ 区間上のhom オブジェクトを

$$\mathcal{L}(x, y) := \pi(y|x) \quad \text{と定義します。}$$

定義4に挙げられた、 $[0, 1]$ 上でenrich化されたカテゴリーの恒等性及び合成性の次の二つの要件を $\mathcal{L}$ が満たすことを検証します。

$$1 \leq \mathcal{L}(x, x) \implies 1 \leq \pi(x|x)$$
$$\mathcal{L}(y, z) \otimes \mathcal{L}(x, y) \leq \mathcal{L}(x, z) \implies \pi(z|y)\pi(y|x) \leq \pi(z|x)$$

$x, y, z$ が  $x \rightarrow y \rightarrow z$  を満たす3つの文字列であるとしましょう。  
 この時、 $(a_i)_{i=1}^{t+k+k'-1} \subset A$ 、かつ  $a_{t+k+k'} \in A \cup \{t\}$  として、次のように記述ができます。

$$\begin{array}{l}
 x = \perp \overbrace{a_1 \cdots a_t}^t \\
 y = \perp a_1 \cdots a_t \overbrace{a_{t+1} \cdots a_{t+k}}^k \\
 z = \perp a_1 \cdots a_t a_{t+1} \cdots a_{t+k} \overbrace{a_{t+k+1} \cdots a_{t+k+k'}}^{k'}
 \end{array}$$

この  $x, y, z$  に、定義 2 を使うと、次の式 (7) が得られます。

$$\begin{aligned}
 (7) \quad \pi(y|x)\pi(z|y) &= \prod_{i=1}^k p(a_{t+i} | y_{<t+i}) \prod_{j=1}^{k'} p(a_{t+k+j} | z_{<t+k+j}) \\
 &= \prod_{i=1}^{k+k'} p(a_{t+i} | z_{<t+i}) = \pi(z|x)
 \end{aligned}$$

この式 (7) の証明の詳細は、**Appendix B** にあります。

もし  $x = y$  ならば、 $\pi(y|x) = 1$  であり、明らかに  $\pi(z|y) = \pi(z|x)$  となります。 $z = y$  の場合も同様に扱われます。

より一般に、 $x, y, z$  が  $L$  の任意の要素である場合、

$$\pi(y|x)\pi(z|y) \leq \pi(z|x)$$

となります。

例えば  $x = \perp green$ 、 $y = \perp green\ lantern$ 、 $z = \perp green\ salad$  の場合、 $\pi(y|x)$  と  $\pi(z|x)$  が 0 でなくとも  $\pi(z|y) = 0$  となり得るからです。

# Bradleyのコメント

「この $[0, 1]$ -カテゴリのより一般的な形は、もともと[BTV22, 定義4]で定義された。そこでは、 $x$ が $y$ の任意の部分文字列(厳密な接頭辞ではない)である場合に $L(x, y)$ を非零とみなした。しかし、値 $\pi(y|x)$ はLLMから明示的に構築されていなかった。

一方、上記の**定義2**は、LLMが生成する確率から $\pi(y|x)$ の値が具体的に記述されているので、言語における表現の $[0, 1]$ -カテゴリを与える。

類似の定義は[GV24]にも見られるが、彼らの記述も[BTV22]の記述も、特殊文字 $\perp$ と $\dagger$ やモデルのカットオフ値を考慮していない。これらの側面を取り入れた結果、我々は新しい**命題1**を導き出した。これは $\pi$ を確率と呼ぶことを正当化するものである。」

# 命題 1 振り返り

命題 1. 言語Lにおける未完成テキストxが与えられたとき、関数 $\pi(-|x)|_{T(x)}$ は入力xの終端状態上の確率質量関数である。

この、 $T(x)$  に制限された関数  $\pi(-|x)$  は確率質量関数であることを示す命題 1は重要な定理です。

次の 式が成り立ちます。

$$\sum_{y \in T(x)} \pi(y|x) = 1$$

# Bradleyのコメント

「最も標準的なLLMを念頭に、ここでは右側への拡張のみを考慮している点に注意されたい。双方向拡張も考え得るが、右側拡張への制限によって示唆される木構造は、後述するマグニチュードの計算において重要な役割を果たす。

最後に、 $[0, 1]$ -カテゴリー  $\mathcal{L}$  を定義した後、[BTV22] の著者らは  $\mathcal{L}$  上のenrich化された co-presheafをさらに検討し、それが意味論的情報を含むことを示した。

しかし我々の現在の目的は、 $\mathcal{L}$  からそのより幾何学的なバージョンへと注意を向けることである。」

# 全てのLLMは [0, ∞]-カテゴリを定義する

[BTV22, Section 5]で論じられているように、関数  $-l n : [0,1] \rightarrow [0, \infty]$  はカテゴリの同型写像であり、確率論からより幾何学的な枠組みへの移行を提供します。

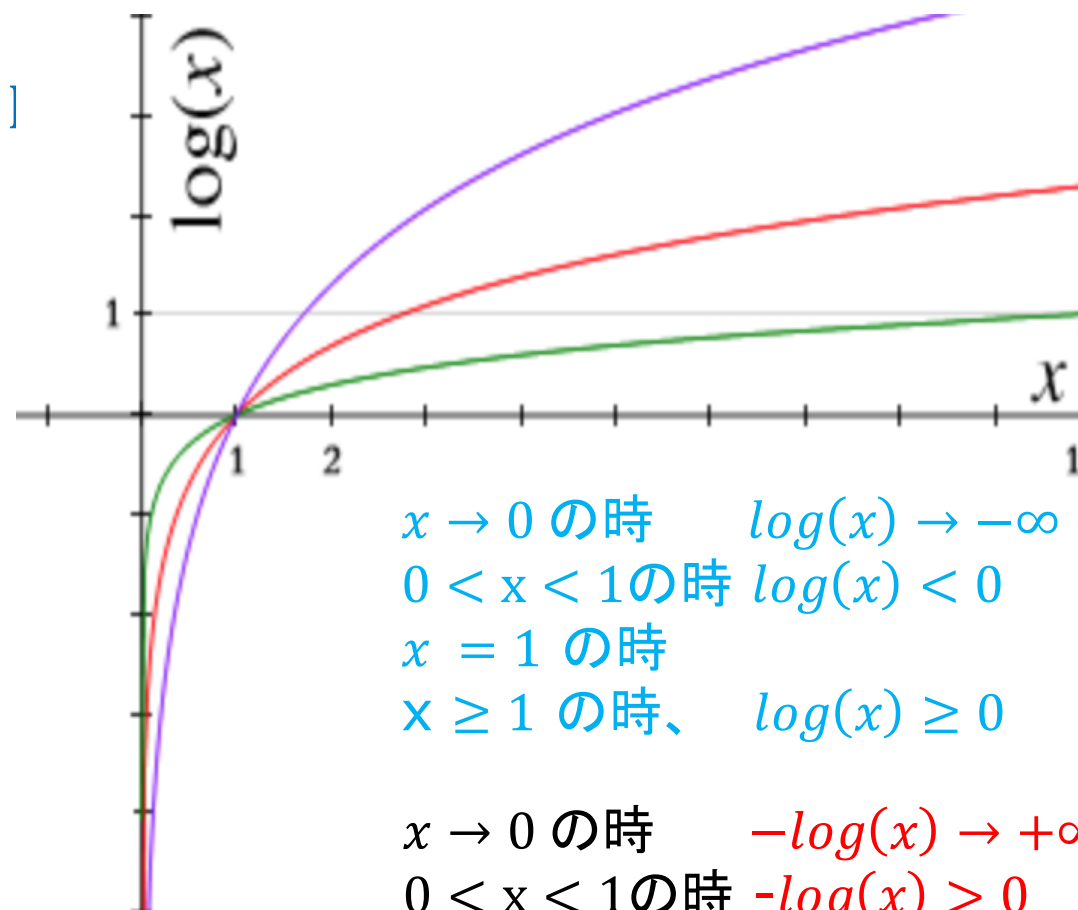
つまり、接頭辞  $x$  から文字列  $y$  を生成する確率に基づくenrich化を考える代わりに、

$$d(x, y) := -l n \pi(y|x)$$

と定義することで、 $x$  から  $y$  までの「距離」 $d(x, y)$  を捉えることができます。

# 対数 $\log$ , $-\log$ の性質

対数関数  $\log x$  のグラフの底を変えたときの様子



$x \rightarrow 0$  の時  $\log(x) \rightarrow -\infty$

$0 < x < 1$  の時  $\log(x) < 0$

$x = 1$  の時

$x \geq 1$  の時、  $\log(x) \geq 0$

$x \rightarrow 0$  の時  $-\log(x) \rightarrow +\infty$

$0 < x < 1$  の時  $-\log(x) > 0$

$x = 1$  の時

$x \geq 1$  の時、  $-\log(x) \leq 0$

三角不等式が満たされることは容易に確認できます。  
すなわち、

$$d(x, y) + d(y, z) \geq d(x, z)$$

です。

さらに全ての文字列  $x, y, z$  に対して

$$d(x, x) = 0$$

です。

(これらは式(7)と、すべての $x$ について  $\pi(x|x) = 1$ である事実から導かれます。)

このようにして、 $[0, \infty]$  上にenrich化されたカテゴリー  $\mathcal{M}$  を得ます。

# Lawvereの「一般化された距離空間」

この  $[0, \infty]$  上にenrich化されたカテゴリー  $\mathcal{M}$  は Lawvere [Law73] の「一般化された距離空間」とも呼ばれます。

それは、文字列  $x, y$  のペア間の  $[0, \infty]$ -hom オブジェクトを  
距離  $\mathcal{M}(x, y) := d(x, y)$

と定義することで得られます。

$x$  と  $y$  の「距離」を導入することで、プロンプト  $x$  を拡張する可能性が高いテキストは  $x$  に近く、 $x$  の拡張ではないテキストは無限に遠いと言うことができます。

# 「一般化された距離空間」の基礎を確認する

一般化された距離空間の基礎については、

前回のセミナー「[マグニチュード論の展開](#)」の

<https://www.marulabo.net/docs/llm1bradley/>

「[Part 3 Lawvereのenriched カテゴリー論](#)」

<https://www.marulabo.net/docs/llm1bradley>

[#Part 3 Lawvereのenriched カテゴリー論](#)

の資料pdfとビデオを参照ください。

## 論文後半へ

一般化された距離空間を得たので、そのマグニチュードを問うことができます。一般化された距離空間のマグニチュードさに関する理論はよく知られています[Lei13]。それはenrich化されたカテゴリーのマグニチュードの特殊な場合です。[LS21, LM17]。

重要なのは、この理論は有限なenrich化されたカテゴリーを扱うことを要求するため、有限のカットオフ $N$ を持つカテゴリー $L^{\leq N}$ を考慮する追加の理由となるということです。

論文の後半を扱う次のセミナーでは、マグニチュードの基礎をいくつか示し、 $[0, \infty]$ -カテゴリー $\mathcal{M}$ のマグニチュードを計算します。







# Appendix

**Appendix A:** 「命題 1」の証明

**Appendix B:** 「命題 7」の証明

# Appendix A

## 「命題 1」の証明詳細

Part 2 の「論文前半の基本定理の証明の概要」からの続きになっています。

# 命題 1 の証明の四つのステップ

先の証明の概略で述べたのは、式 3 から式 6 の導出が出来ればいいということです。

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a \in A^m} \pi(xa|x)$$

$$(4) \quad = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{\substack{a=a'a'' \\ a' \in A^{m-1}, a'' \in A}} p(a''|xa') \pi(xa'|x)$$

$$(5) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x) \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa')$$

$$(6) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa'|x)$$

## 式(3)の導出

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^m} \pi(xa|x)$$

左辺では、 $x$ は $\perp a$ の形の未完成テキストで、 $\pi(y|x)$ の和は $y \in T(x)$ についてとられています。

右辺の最初の和は**完成したテキスト**(すなわち $\dagger$ で終わるもの)を、二番目の和は長さ $N$ の**未完成テキスト**をそれぞれ表しています。右辺の二つの和は、 $T(x)$ と $\pi(y|x)$ の定義から導かれます。

$x$ の後にモデルが終了するまで続く最大記号数  $m = N - |x|$  についての帰納法

## 式(3) から 式(4) の導出

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^m} \pi(xa|x)$$

$$(4) \quad = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{\substack{a=a'a'' \\ a' \in A^{m-1} \\ a'' \in A}} p(a''|xa') \pi(xa'|x)$$

式(3)から式(4)への導出は、右辺の第2項の変形にのみ関わっています。第1項はそのままです。

$$(3) \quad \sum_{y \in T(x)} \pi(y|x) = \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a \in A^m} \pi(xa|x)$$

式(3)の右辺第2項は、長さ  $N=|x|+m$  の cutoff に達したため  $\dagger$  なしで終端したテキスト (Definition 1の第1のケース) に関する総和です。

$$\text{式(3)の第2項: } \sum_{a \in A^m} \pi(xa|x)$$

ここで、 $a$  はアルファベット  $A$  の要素からなる長さ  $m$  の文字列  $a = a_1 a_2 \cdots a_m$  を表します。

文字列  $a \in A^m$  の  $a = a'a'' \in A^{m-1} \times A$  への分解

この項  $a$  は、以下のように分解して考えることができます。

長さ  $m$  の文字列  $a \in A^m$  を、最初の  $m-1$  個のトークンからなる文字列  $a' \in A^{m-1}$  と、最後の1個のトークン  $a'' \in A$  に分解します。

$$a = a'a''$$

$$\sum_{a \in A^m} \pi(xa|x) = \sum_{a' \in A^{m-1}} \sum_{a'' \in A} \pi(xa'a''|x)$$

## パス確率 $\pi(xa|x)$ の展開

パス確率  $\pi(xa|x) = \pi(xa'a''|x)$  は、定義2により、

$$\pi(xa'a''|x) = \pi(xa'|x) \cdot p(a''|xa')$$

と分解できます。

$\pi(xa'|x)$  は、プロンプト  $x$  から中間テキスト  $xa'$  までのパス確率です。

$p(a''|xa')$  は、テキスト  $xa'$  が与えられたときの、次のトークンとして  $a''$  が生成されるネクストトークン確率です。

よって、

$$\begin{aligned} \sum_{a \in A^m} \pi(xa|x) &= \sum_{a' \in A^{m-1}} \sum_{a'' \in A} \pi(xa'a''|x) \\ &= \sum_{a' \in A^{m-1}} \sum_{a'' \in A} \pi(xa'|x) \cdot p(a''|xa') \end{aligned}$$

これで (4)式が導かれました。

## 式(4) から 式(5) の導出

$$(4) \quad \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \sum_{a'' \in A} p(a'' | xa') \pi(xa' | x)$$

$$(5) = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in A \cup \{\dagger\}} p(a'' | xa')$$

を示すことが課題です。

ここで  $m = N - |x|$  は残りの最大トークン数、 $A$  はアルファベット (トークン集合)、 $\dagger$  は終端トークン、 $p(\cdot | z)$  はプロンプト  $z$  の後のネクストトークン確率、 $\pi(y | x)$  は  $x$  から  $y$  へのパス確率です。

## ステップ 1: 第1項の分割 ( $i=m-1$ の項の分離)

式(4)の右辺第1項は、長さ  $|x|+i+1$  で  $\dagger$  で終わるテキスト (finished texts) に関する総和です。

ここで、第1項を  $i=0$  から  $m-2$  までの総和と、最後の項  $i=m-1$  の総和に分けます。

$$\begin{aligned} & \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger |x) \\ &= \left( \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger |x) \right) + \left( \sum_{a \in A^{m-1}} \pi(xa \dagger |x) \right) \end{aligned}$$

$$(5) \quad \boxed{\sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x)} + \sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in A \cup \{\dagger\}} p(a'' | xa')$$

式(5)の右辺の第1項は、この  $i=0$  から  $m-2$  までの総和です。

$$\sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x)$$

分離された  $i=m-1$  の項は、式(5)の導出過程で第2項に統合されます。

分離された項 (A):  $\sum_{a \in A^{m-1}} \pi(xa \dagger | x)$



第2項に統合

ステップ 2: 式(4)の第2項の変形と + 項の統合

$$(4) \quad \sum_{i=0}^{m-1} \sum_{a \in A^i} \pi(xa \dagger | x) + \boxed{\sum_{a' \in A^{m-1}} \sum_{a'' \in A} p(a'' | xa') \pi(xa' | x)}$$

式(4)の第2項:  $\sum_{a' \in A^{m-1}} \sum_{a'' \in A} p(a'' | xa') \pi(xa' | x)$

式(4)の右辺第2項は、長さ  $N = |x| + m$  に達したため + なしで  
終端したテキスト (unfinished texts of length  $N$ ) に関する総  
和です。

式(4)の第2項: 
$$\sum_{a' \in A^{m-1}} \sum_{a'' \in A} p(a'' | xa') \pi(xa' | x)$$

この総和は、次のように  $\pi(xa' | x)$  を外に出し、内側の和の順番を入れ替えることができます。

$$= \sum_{a' \in A^{m-1}} \pi(xa' | x) \left( \sum_{a'' \in A} p(a'' | xa') \right)$$

次に、ステップ1で分離した「†で終わる長さNのテキスト」に関する項 (A) をこの項に統合します。

項 (A) の  $\pi(xa \dagger | x)$  は、定義 (2) により、

$$\pi(xa \dagger | x) = \pi(xa | x)p(\dagger | xa)$$

と書くことができます(ここで  $a \in A^{m-1}$ )。

したがって、項 (A) は

$$\text{項 (A): } \sum_{a \in A^{m-1}} \pi(xa \dagger | x) = \sum_{a' \in A^{m-1}} \pi(xa' | x)p(\dagger | xa')$$

項(A)を式(4)の第2項に加えると、

$$\begin{aligned} \text{式(4)の第2項: } & \left( \sum_{a' \in A^{m-1}} \pi(xa'|x) \sum_{a'' \in A} p(a''|xa') \right) \\ \text{項 (A): } & + \left( \sum_{a' \in A^{m-1}} \pi(xa'|x) p(\dagger |xa') \right) \\ & \pi(xa'|x) \text{をくりだすと} \\ = & \sum_{a' \in A^{m-1}} \pi(xa'|x) \left( \sum_{a'' \in A} p(a''|xa') + p(\dagger |xa') \right) \end{aligned}$$

内側の括弧内の総和は、トークン A と終端トークン † の両方について p の総和を取ることに等しくなります。

$$\sum_{a'' \in A} p(a''|xa') + p(\dagger |xa') = \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa')$$

これにより、式(5)の第2項が導かれます。

$$(5) \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \boxed{\sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in AU\{\dagger\}} p(a'' | xa')}$$

$$\sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in AU\{\dagger\}} p(a'' | xa')$$

式(4)から式(5)への導出は、第1項から  $i=m-1$  の項を分離し、その項を  $\pi(xa \dagger | x) = \pi(xa | x)p(\dagger | xa)$  と書き直すことで、もともと第2項であった長さ  $N$  の未完了テキストの項と統合し、 $\dagger$  を含む全てのネクストトークンについての確率の和として表現し直す操作です。

## 式(5) から 式(6) の導出

$$(5) \quad \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \boxed{\sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in AU\{\dagger\}} p(a'' | xa')}$$

$$(6) \quad = \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger | x) + \sum_{a' \in A^{m-1}} \pi(xa' | x)$$

式(5)から式(6)への導出は、式(5)の右辺の第2項に着目することで説明できます。

式(5)の右辺第2項は、以下の部分です。

$$\sum_{a' \in A^{m-1}} \pi(xa' | x) \sum_{a'' \in AU\{\dagger\}} p(a'' | xa')$$

この項には、ネクストトークン確率分布  $p$  に関する内側の総和があります。

$$\sum_{a'' \in A \cup \{\dagger\}} p(a'' | x a')$$

$p$  はLLMによって生成される確率分布であり、任意のプロンプト  $z$  に対して  $p_z := p(\cdot | z): A \cup \{\dagger\} \rightarrow [0,1]$  は確率分布です。

これは、トークンの集合  $A \cup \{\dagger\}$  上の確率質量関数であることを意味します。したがって、そのサポート(ここでは  $A \cup \{\dagger\}$ )全体での確率の総和は 1 になります。

$$\sum_{a'' \in A \cup \{\dagger\}} p(a'' | x a') = 1$$

これを式(5)の第2項に適用すると、以下ようになります。

$$\begin{aligned} \sum_{a' \in A^{m-1}} \pi(xa'|x) \left( \sum_{a'' \in A \cup \{\dagger\}} p(a''|xa') \right) &= \sum_{a' \in A^{m-1}} \pi(xa'|x) \cdot \mathbf{1} \\ &= \sum_{a' \in A^{m-1}} \pi(xa'|x) \end{aligned}$$

(5)の右辺の第1項はそのまま残り、第2項が上記の通り簡略化されるため、式(6)が導かれます。

$$(6) \quad \sum_{i=0}^{m-2} \sum_{a \in A^i} \pi(xa \dagger |x) + \sum_{a' \in A^{m-1}} \pi(xa'|x)$$



# Appendix B

## 「命題 7」の証明

## (7)の導出の補足

定義2より、 $\pi(y|x)$ は  $x$  に続く  $k$  個のトークンを生成する確率の積として定義されます。同様に、 $\pi(z|y)$ は  $y$  に続く  $k'$ 個のトークンを生成する確率の積です。

$\pi(y|x)$  の項:

$x$  から  $y$  への拡張は  $a_{t+1} \cdots a_{t+k}$  であり、これは  $k$ 個のトークンから成ります。

$$\pi(y|x) = \prod_{i=1}^k p(a_{t+i} | y_{<t+i})$$

ここで、 $y_{<t+i}$ は  $\perp a_1 \cdots a_{t+i-1}$ を表します。

特に $y_{<t+1} = \perp a_1 \cdots a_t = x$  です。

## $\pi(z|y)$ の項

$y$  から  $z$  への拡張は  $a_{t+k+1} \cdots a_{t+k+k'}$  であり、これは  $k'$  個のトークンから成ります。

$$\pi(z|y) = \prod_{j=1}^k p(a_{t+k+j} | y_{<t+k+j})$$

ここで、 $y_{<t+k+j}$  は  $\perp a_1 \cdots a_{t+k+j-1}$  を表し、最初の項  $j=1$  のとき  $z_{<t+k+1} = \perp a_1 \cdots a_t a_{t+1} \cdots a_{t+k} = y$  です。

## $\pi(y|x)\pi(z|y)$ の積:

これらの積を組み合わせると、式(7)の最初の行が得られます。

$$\pi(y|x)\pi(z|y) = \left( \prod_{i=1}^k p(a_{t+i}|y_{<t+i}) \right) \left( \prod_{j=1}^{k'} p(a_{t+k+j}|z_{<t+k+j}) \right)$$

## 積の結合

次に、この積を1つの連続した積にまとめます。

$y_{<t+i} = \perp a_1 \cdots a_{t+i-1}$  および  $z_{<t+k+j} = \perp a_1 \cdots a_{t+k+j-1}$  です。

$a_{t+1}, \dots, a_{t+k}$  のインデックス  $i$  は 1 から  $k$  まで動きます。

$a_{t+k+1}, \dots, a_{t+k+k'}$  のインデックス  $j$  は 1 から  $k'$  まで動きます。

インデックスを  $I'$  で統一します。

最初の積の項は、 $p(a_{t+i} | y_{<t+i})$  で、 $i' = t + i$  は  $t+1$  から  $t+k$  まで動きます。

2番目の積の項は、 $p(a_{t+k+j} | z_{<t+k+j})$  で、 $i' = t + k + j$  は  $t+k+1$  から  $t+k+k'$  まで動きます。

ここで、 $y_{<t+i} = z_{<t+i}$  であり、 $z_{<t+k+j} = z_{<t+(k+j)}$  です。

$i'$  を  $t+i$  とすると、全体のインデックスは  $i=1$  から  $k+k'$  まで動きます。

$$\pi(y|x)\pi(z|y) = \prod_{i=1}^{k+k'} p(a_{t+i}|z_{<t+i})$$

これが式(7)の2行目です。

## 先の式が $\pi(z|x)$ と等しいこと

$z$  は  $x$  から  $k+k'$  個のトークンで拡張された文字列です。定義2により、 $\pi(z|x)$  は  $x$  に続く  $k+k'$  個のトークン  $a_{t+1} \cdots a_{t+k+k'}$  を生成する確率の積です。

$$\pi(z|x) = \prod_{i=1}^{k+k'} p(a_{t+i}|z_{<t+i})$$

ここで、 $z_{<t+1} = x$  であり、 $z_{<t+i}$  は  $\perp a_1 \cdots a_{t+i-1}$  です。

したがって、

$$\pi(y|x)\pi(z|y) = \pi(z|x)$$

が、成り立ちます。



