LLMと意味の理論モデル概説

LLMと意味の理論モデル概説
Agenda

- Part 1　LLMの理論モデルの新しい展開
- Part 2　新しい展開の背景を探る
- Part 3　Bradleyの理論の発展をたどる
- Part 4　Bradleyのcopresheaf意味論

Part 1

LLMの理論モデルの新しい展開

LLMと意味の理論モデルの新しい展開

LLMの言語の意味理解能力の獲得と驚異的な言語運用能力の実現という現実の進行を目の当たりにして、それを説明しようとする問題意識と理論が次々に生まれています。

MaruLaboでは、この夏以降しばらくの間、「LLMの理論モデルの新しい展開」という共通テーマで、複数のセミナーを連続して開催しようと思っています。

今回のセミナーは、今後予定している一連の「LLMの理論モデルの新しい展開」セミナーの全体の概要を紹介するものです。


注目すべき三つの研究

注目すべき三つの研究

現在、活発に研究されているLLMと意味の理論では、僕は次の三つの流れに注目しています。今回の連続セミナーで取り上げようと思っているのは、こうした流れです。

1. Bradleyらの意味の距離空間モデルとMagnitude概念の導入
2. Coeckeらの量子コンピュータ上でのQNLP実装の進展
3. VlassopoulosらのLLMの内部動作の解析

ここでは、こうした流れの特徴を簡単に紹介しようと思います。あわせて、それらを代表する論文を上げておこうと思います。

1. Bradleyらの意味の距離空間モデルとMagnitude概念の導入

Bradleyは、現在のLLMの働きを反映しているテキストの確率論的モデルに、再び enriched categoryの手法を適用して、新しいMagnitude概念をベースとした意味の距離空間モデルを構築し、そこが意味解釈の舞台として望ましい性質を持っていることを示しました。

The Magnitude of Categories of Texts Enriched by Language Models

https://arxiv.org/pdf/2501.06662

2. Coeckeらの量子コンピュータ上でのQNLP実装の進展

Coeckeらは、限られた実験環境のもとでも、着実に量子コンピュータ上での自然言語処理の実装を進めています。特に近年のsurface codeを中心とする量子コンピュータ実装技術の発展は、大きな追い風になっています。

Scalable and interpretable quantum natural language processing: an implementation on trapped ions

https://arxiv.org/pdf/2409.08777

3. VlassopoulosらのLLMの内部動作の解析

上の二つの流れとは、少し異なる取り組みです。LLMに外部から入力として与えられている学習データやLLMの外部への出力の特徴を分析するのではなく、現在のLLMの内部の振る舞いを具体的に数学的に解析しようとしたものです。驚くべきことに、その解析結果は、Bradleyらの意味モデルを支持するものになっています。

Directed Metric Structures arising in Large Language Models

https://arxiv.org/pdf/2405.12264

新しいモデルの数学的ツール

この注目すべき三つの流れは、それぞれ異なる数学的なツールを中心的に利用しています。

それぞれのグループごとに、その概略を紹介します。

1. magnitude論とmagnitude homology論

第一のBradleyらのグループの理論展開では、magnitude論、magnitude homology論の利用が目を惹きます。magnitude論を用いて「意味の近さ」だけでなく「意味の形」「意味の空間の幾何学（例えば、「意味空間の穴」）」が縦横に語られます。

magnitude論は、エントロピー論と深い関係を持っているのですが、Bradleyの論文は、基本的には言語理論の論文なのですが、統計力学の分配関数に遡って、シャノン・エントロピーとmagnitudeとの関係が論じられています。

Magnitudeという指標の発見

以前の彼女の論文での「意味はテキストデータに内在する複数の形式的次元と不可分である」という指摘は、非常に示唆的です。

今回の連続セミナー「LLMの理論モデルの新しい展開」の主要な原動力は、まさにこの「新しい次元」の発見にあります。それを Magnitude と言います。

LLMについていえば、「意味のMagnitude」を考えることで、さまざまな展開が生まれています。それについては、後続のセミナでその概要を述べようと思います。

興味のある方は、次の資料を参照ください。

The many faces of magnitude

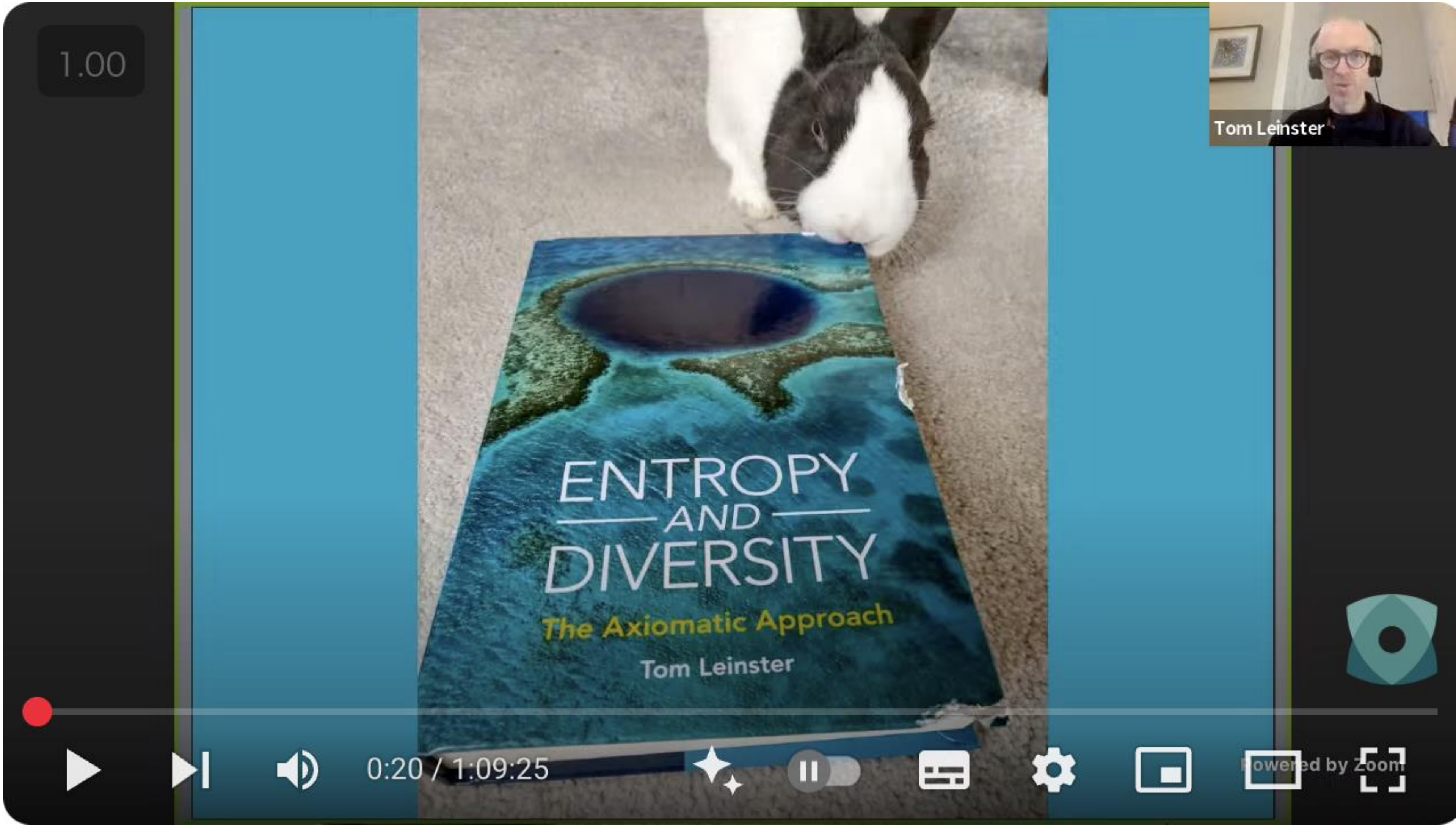
Tom Leinster

School of Mathematics
University of Edinburgh

Boyd Orr Centre
for Population and Ecosystem Health
University of Glasgow

https://webhomes.maths.ed.ac.uk/~tl/osaka_colloq/osaka_colloq_talk.pdf

Tom Leinster: "Entropy and diversity: the axiomatic approach"

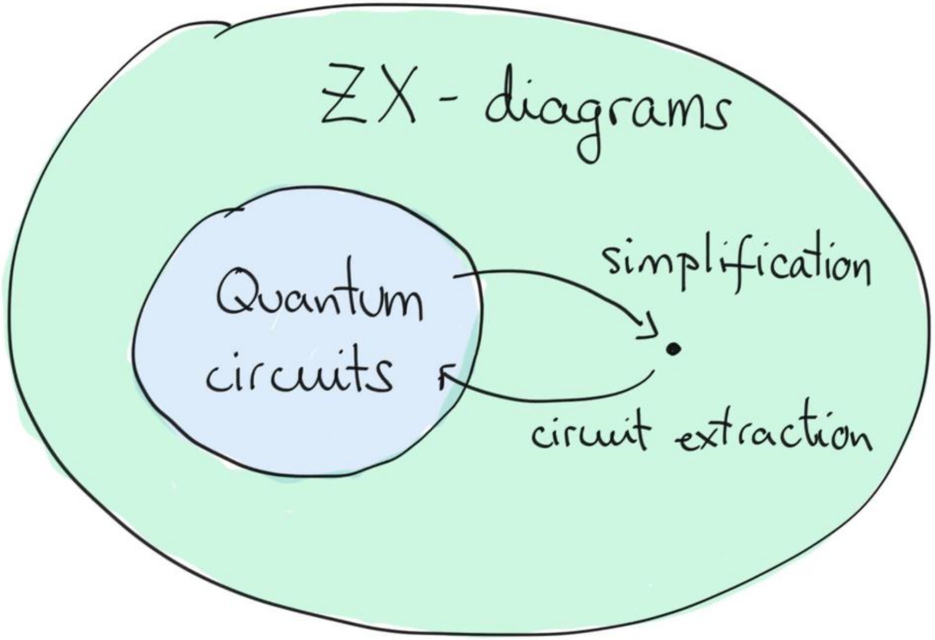


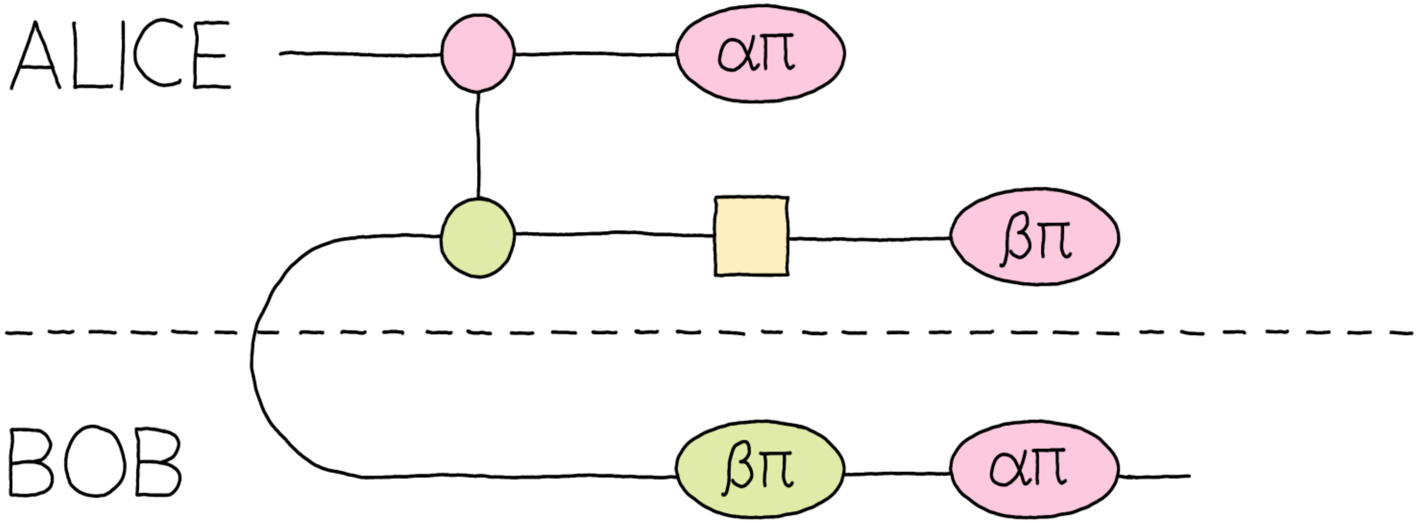
https://youtu.be/y_FfWMJMH0Q

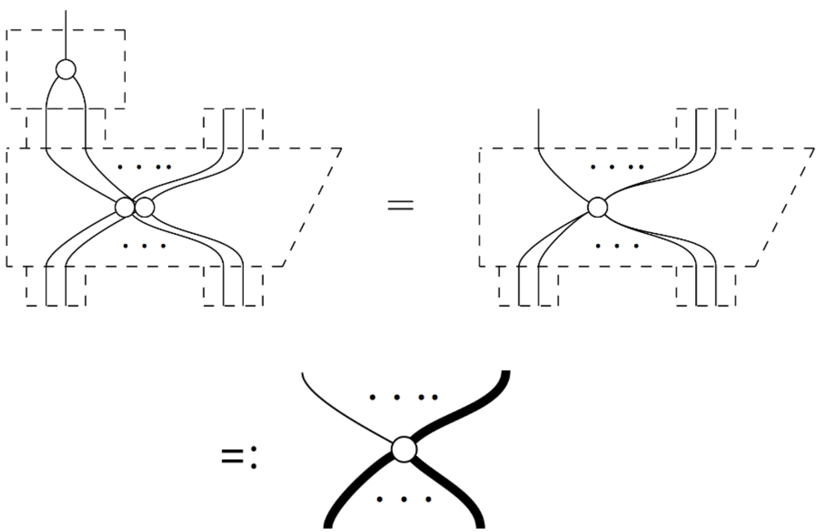
2. ZX−calculus

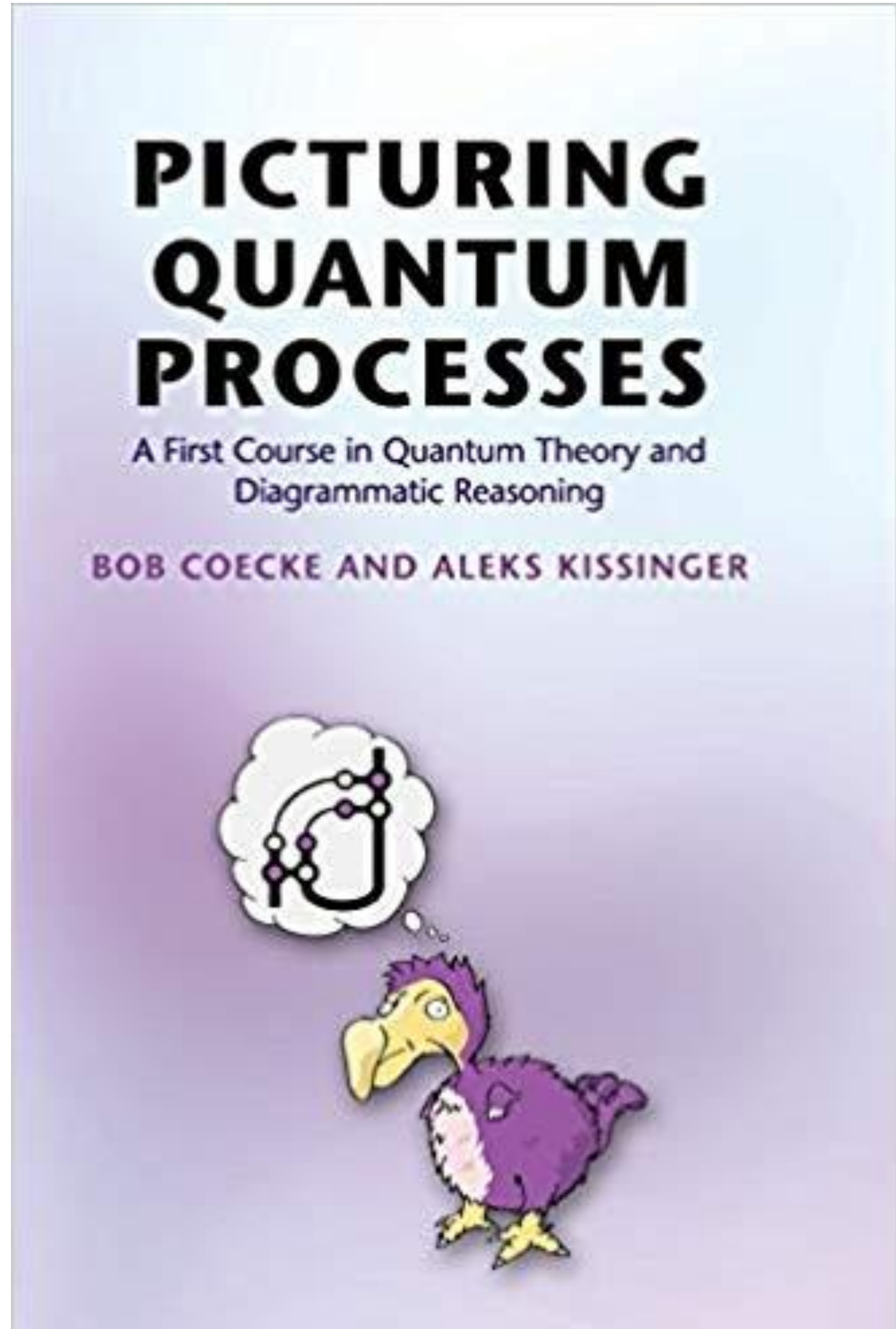
第二のCoeckeらのグループの研究が利用する数学的ツールで興味深いのは、CoeckeがQNLPとは独立に（おそらく）量子力学のツールとして展開してきた string diagramを利用するZX-calculusの技術が、ぴたりと彼のQNLPの理論と実験の中に組み込まれようとしていることです。

ZX−calculusは、現在の量子コンピュータ技術の革新の中心であるsurface codeの基礎理論として、熱い期待を集めている技術です。


ZX - diagrams
Quantum circuits
simplification
circuit extraction


ALICE
απ
βπ
BOB
βπ
απ




PICTURING QUANTUM PROCESSES
A First Course in Quantum Theory and Diagrammatic Reasoning
BOB COECKE AND ALEKS KISSINGER

3. Tropical代数

第三のVslassopoulosらの用いる数学的ツールは、少しなじみがないものかもしれません。それは、Tropical代数と言われるツールです。

Tropical代数については、マルレクで取り上げたことがあります。次の資料を参照ください。

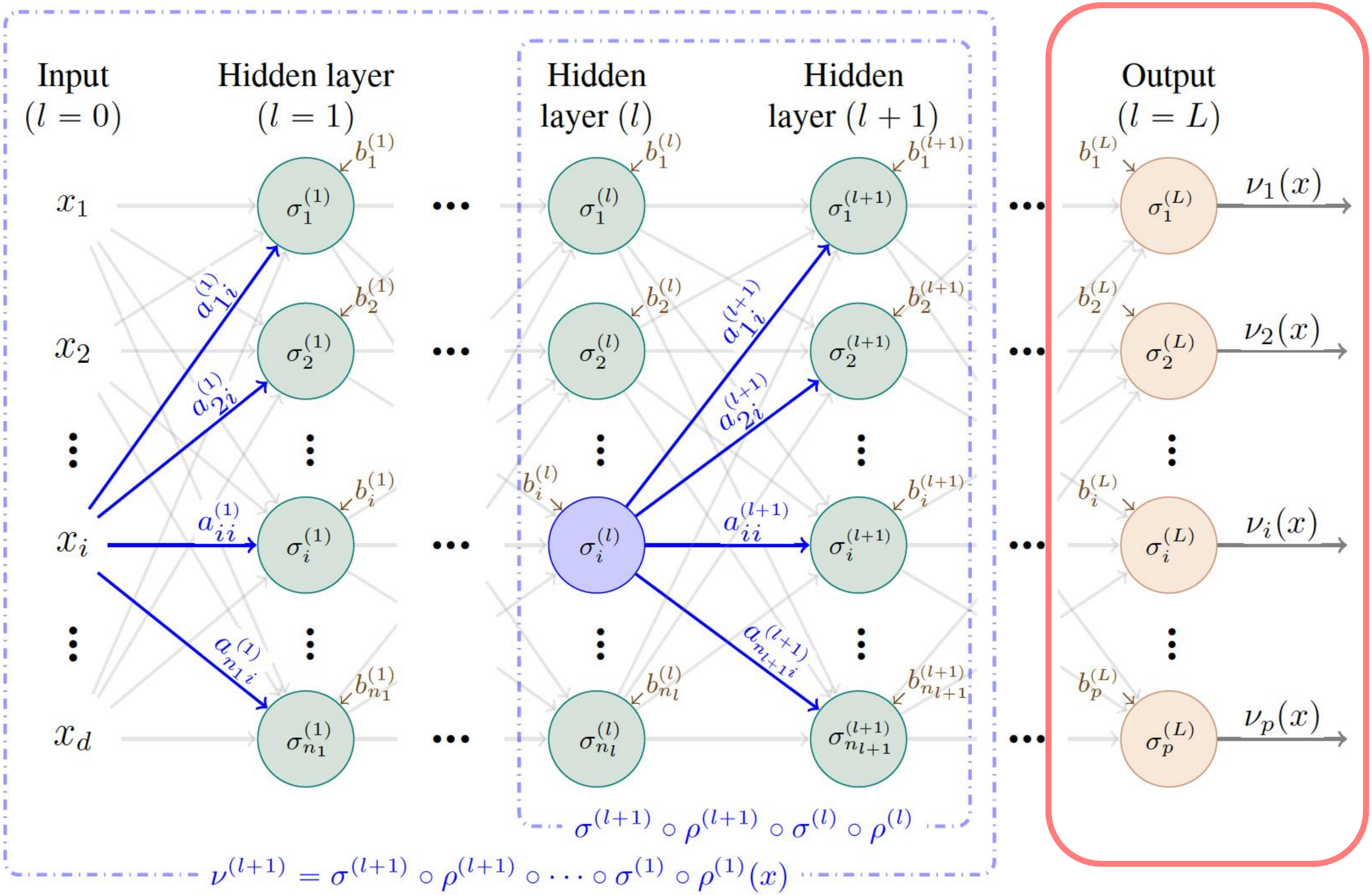
ニューラル・ネットワークの数理 –– Tropical代数入門
https://www.marulabo.net/docs/tropical/

$$
\mathrm{L}(\text{blue}, -) = \begin{bmatrix} 0 \\ .68 \\ .01 \\ 0 \\ .17 \\ 0 \\ \vdots \end{bmatrix}
\begin{array}{l} \text{deep red Bing cherries} \\ \text{small blue marble} \\ \text{beautiful blue ocean} \\ \text{did you put the kettle on} \\ \text{red and blue fireworks} \\ \text{Sencha green tea} \\ \vdots \end{array}
$$

(67)

$$
C(a_i, a_j) = \begin{cases} -\log \Pr(a_j | a_i) & \text{if } a_i \leqslant a_j \text{ and } a_j \text{ extends } a_i \text{ by a single word,} \\ \infty & \text{if } a_i \leqslant a_j \text{ and } a_j \text{ extends } a_i \text{ by more than a single word,} \\ \infty & \text{if } a_i \text{ and } a_j \text{ are not comparable.} \end{cases}
$$

出力層までのグラフ


Input
$(l = 0)$
Hidden layer
$(l = 1)$
Hidden
layer (l)
Hidden
layer $(l+1)$
Output
$(l = L)$
x_1
x_2
x_i
x_d
$\nu_1(x)$
$\nu_2(x)$
$\nu_i(x)$
$\nu_p(x)$
$\sigma^{(l+1)} \circ \rho^{(l+1)} \circ \sigma^{(l)} \circ \rho^{(l)}$
$\nu^{(l+1)} = \sigma^{(l+1)} \circ \rho^{(l+1)} \circ \cdots \circ \sigma^{(1)} \circ \rho^{(1)}(x)$

$\nu^{(l+1)}(x)$はTropical有理写像

$$\nu^{(l+1)}(x) = \max(\nu^{(l+1)}(x), t)$$

これは、ReLUの適用である。

ここで、先に見た、$\rho^{(l+1)}(x) = H^{(l+1)}(x) - G^{(l+1)}(x)$を使う。

$$= \max(H^{(l+1)}(x) - G^{(l+1)}(x), t)$$

$$= \max(H^{(l+1)}(x) - G^{(l+1)}(x) + G^{(l+1)}(x), \mathrm{t} + G^{(l+1)}(x)) - G^{(l+1)}(x)$$

$$= \max\left(H^{(l+1)}(x), G^{(l+1)}(x) + t\right) - G^{(l+1)}(x)$$

$$\max\left(H^{(l+1)}(x), G^{(l+1)}(x) + t\right) = F^{(l+1)}(x)$$

とおけば、

$$\nu^{(l+1)}(x) = F^{(l+1)}(x) - G^{(l+1)}(x)$$

$\nu^{(l+1)}(x)$はTropical有理写像である。


理論モデルの新しい展開に
ついての音声による概要まとめ

理論モデルの新しい展開については、音声概要をご利用ください

今回のセミナーでは、ここれらの理論モデルの新しい展開については個別には深入りしません。今後、トピックごとに個別のセミナーを開催する予定です。（ただ、のんびり個別のセミナーを準備していこうと思っているので、トピックによっては、何ヶ月か先になると思います。）

現在の状況を手短に理解してもらうために、理論モデルの新しい展開について音声による概要を用意しました。ぜひ、こちらもご利用ください。

次のようなコンテンツを用意しています。基本的には、現在の研究フレームと先にあげた三つの注目すべき三つの研究動向に対応しています。

音声概要のURLはこちらです

- 「LLMの謎にカテゴリー論で迫る」
  https://www.marulabo.net/wp-content/uploads/2025/08/LLMandCategory.mp3
- 「LLMのテキスト･データに隠された構造を探る」
  https://www.marulabo.net/wp-content/uploads/2025/08/TextDataandStructure.mp3
- 「自然言語理解と量子論」
  https://www.marulabo.net/wp-content/uploads/2025/08/QNLP.mp3
- 「進むLLMの内部構造の理解」
  https://www.marulabo.net/wp-content/uploads/2025/08/LLMInnerStructure.mp3
- 「QNLPの最新動向」
  https://www.marulabo.net/wp-content/uploads/2025/08/NewQNLP.mp3

次のページからも音声概要にアクセスできます

Deep Dive Audio 20250816

https://www.marulabo.net/docs/dda20250816/

Part 2

新しい展開の背景を探る

新しい展開の背景を探る

LLMの数学的構造について新しい知見が続々登場しているのですが、それには背景があります。

ここでは、理論と実践との両面から、その背景を探ってみたいと思います。


理論的背景
先行した二つの研究

先行した二つの研究 -- BradleyとCoecke

LLMの数学的構造の新しい知見の現在の展開には、理論的背景があります。

現在の展開を理論的に支えた代表的な先行研究が二つあります。

一つは、Tai-Danae Bradleyらの「米田埋め込み」にインスパイアされた enriched copresheaf 意味論で、もう一つは、Bob Coeckeらの「言語はそもそも量子論的だ」というQNLPです。

これらについては、Part 3 と Part 4で、少し詳しく説明したいと思います。

An enriched category theory of language: from syntax to semantics

Tai-Danae Bradley, John Terilla, Yiannis Vlassopoulos
2021年11/18
https://arxiv.org/abs/2106.07890

集合に値を持つ表現可能なfunctorである
意味のcategory Set^Lを、
L上のcopresheafといいます。

意味の$category$ Set^L

copresheaf

$$Set^L$$

$$\boldsymbol{hom}(x,-)$$

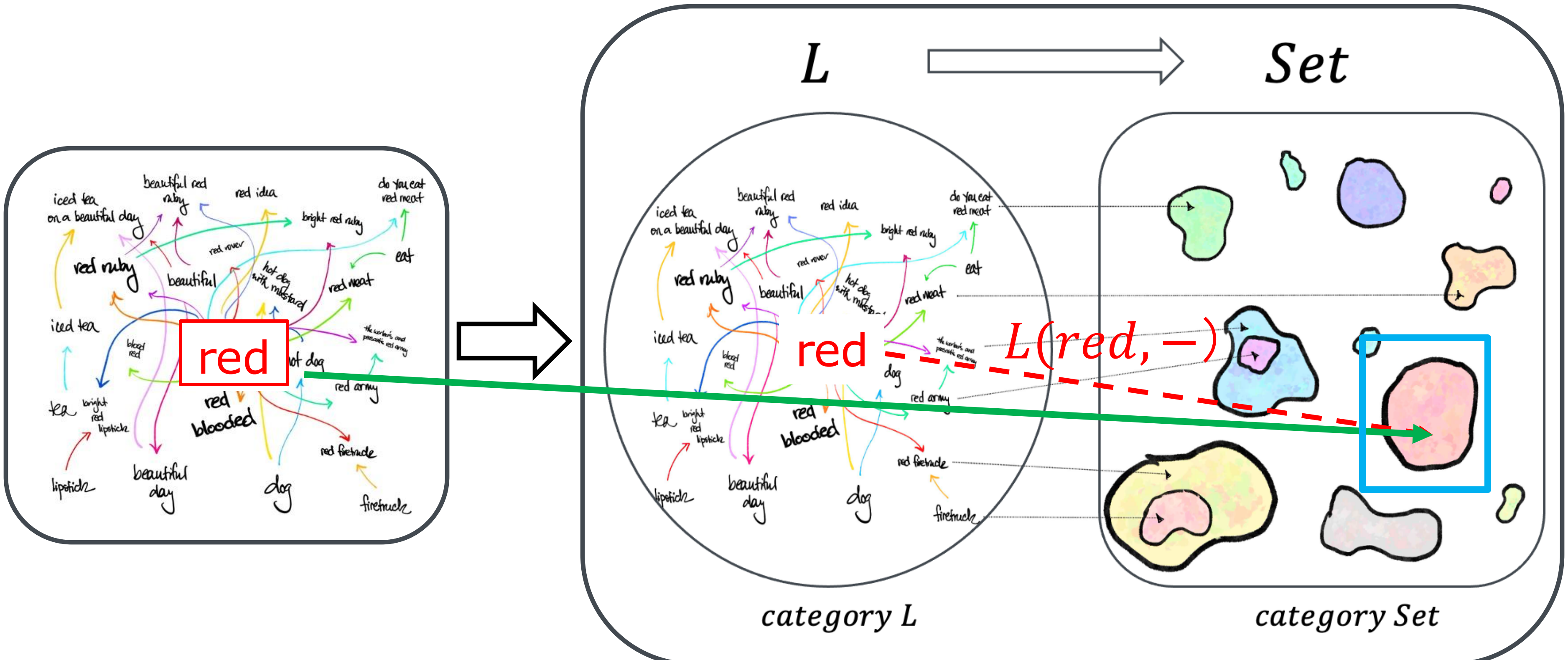
$$\boldsymbol{hom}(z,-) \qquad \boldsymbol{hom}(a,-)$$

$$\boldsymbol{hom}(y,-)$$

すべてのLのオブジェクトxについて
$\boldsymbol{hom}(x,-)$なるfunctorの集まり

この対応は、語 red を
語 red の意味を表す集合に対応づけます。

$$L \Longrightarrow Set^L$$



言語の
category L $\Longrightarrow$ 意味の
category Set^L

Foundations for Near-Term Quantum Natural Language Processing

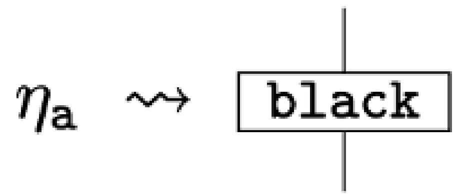
Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, Alexis Toumi

https://arxiv.org/abs/2012.03755v1

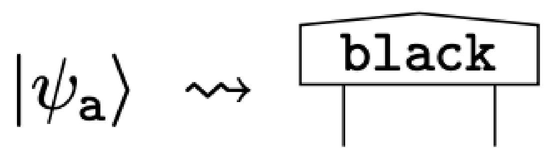
2020年

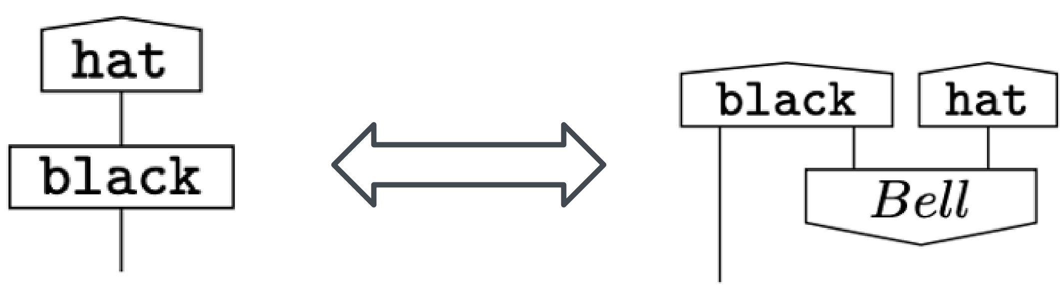
形容詞の二つの表現

形容詞aを1-qubitの写像
$\eta_a: \mathbb{C}^2 \to \mathbb{C}^2$ と考える
$|\psi_{a\cdot n} > = \eta_a(|\psi_n >)$



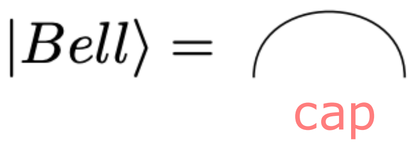
形容詞を 2-qubitの状態
$|\psi_a > \in \mathbb{C}^2 \otimes \mathbb{C}^2$と考える

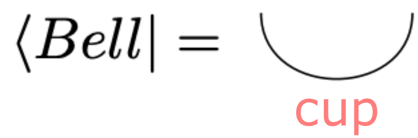




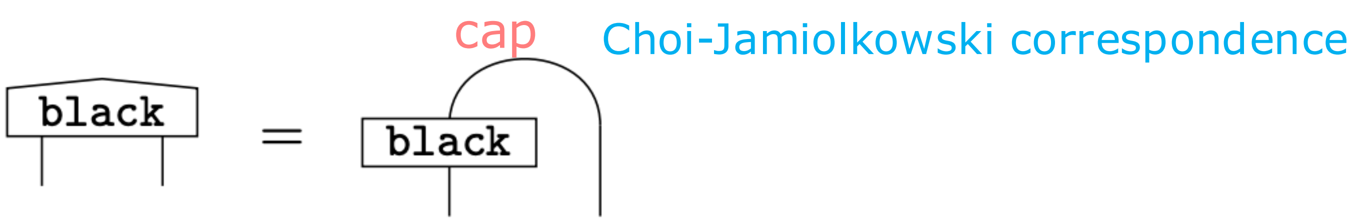
$|Bell>$ と $<Bell|$
cap と cup

$|\ Bell>$と$<Bell\ |$ は、図式的には、次のように表現される。

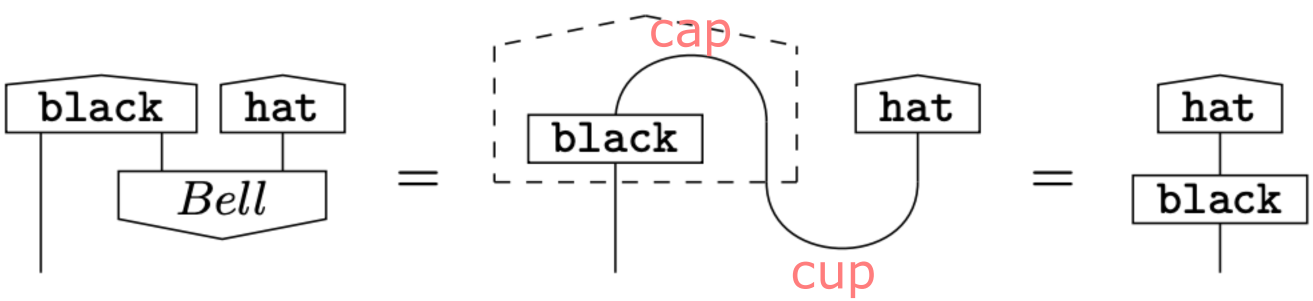




先に見た、2-qubitの状態を2×2の行列と対応する対応は、次のようになる。



この時、



これは、量子テレポーテーションの図式と同じ形をしている。

先行した二つの研究の共通性
「構成性」の重視

BradleyとCoeckeのアプローチでには大きな違いもあるのですが、言語と意味の理解についてその「構成性」が重要であるという認識では、両者は一致しています。

両者は、何の構造も持たないように見える言語データ自身に、すでに構造が先在的に内在していると考えます。こうした構造を、「構造的プライア(structural prior)」と呼びます。

こうした、Bradley とCoeckeのアプローチは、次に見る現在主流のTransformer意味論への共通の批判的立場を提供します。

先行した二つの研究の共通性
カテゴリー論の利用

両者はまた、カテゴリー論（圏論）を主要な数学的なツールとして用いています。これも大きな特徴です。

BradleyのenrichedcategiryのBradleyのenriched categiryの利用、Coeckeのcompact closed category の利用は、特徴的なものです。また、両者は文法的＝構文論的なカテゴリーから意味のカテゴリーの導出に、二つのカテゴリーの構造を保存する functor(関手)を利用します。

こうしたカテゴリー論的フレームワークは、LLMと意味の理論をめぐる現在進行中の様々な取り組みに大きな影響を与えています。

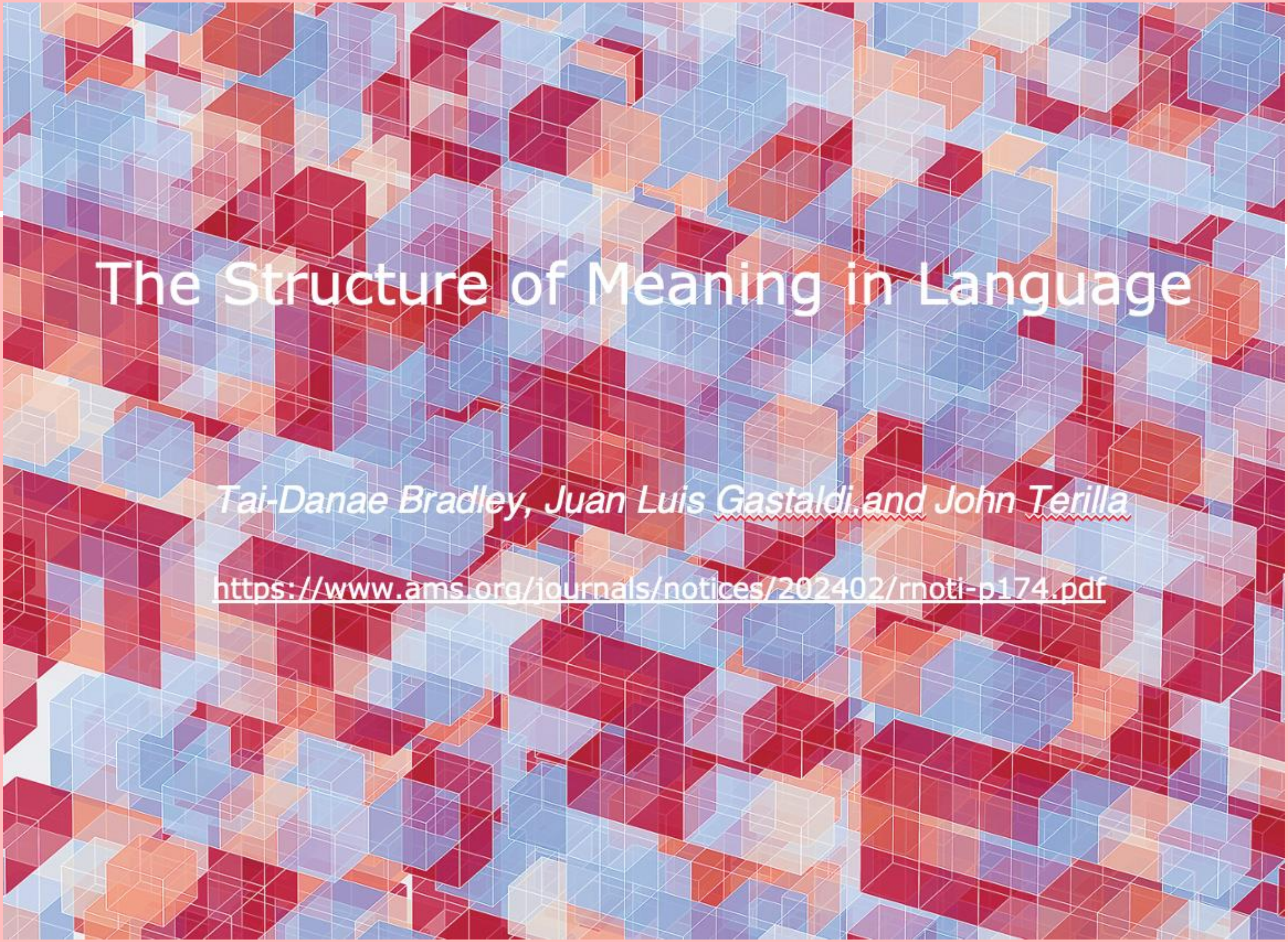
先行した二つの研究の共通性
言語理論と量子論の接点を探る

この二つの理論は、「言語データと意味にひそむ構造を探る」という共通の関心のもと、カテゴリー論を持ちいて、言語理論と量子論の接点を模索するという点で共通の性格を持っています。

残念ながら、この論点については今回のセミナーでは触れることができませんでした。興味がある方は、次の資料を参照ください。

https://www.marulabo.net/docs/density2/

https://www.marulabo.net/docs/embedding-dnn/



https://www.ams.org/journals/notices/202402/rnoti-p174.pdf

Tai-Danae Bradleyの「ニューラル言語モデル」批判

Bradleyは、この2024年のアメリカ数学会への投稿論文 The Structure of Meaning in Languageで、「ニューラル言語モデル」を、次のように批判しています。

「意味と形式は切り離せないという考え方は新しいものではないが、現在のAIをめぐる哲学的な議論には浸透していない。」

「構文対象の分析において意味が問題になるとすれば、それはすべて言語の形式に反映された構造的特徴に起因するということである。」

「しかし、現在のニューラル言語モデルが不十分なのは、まさにこの点である。というのも、ニューラル言語モデルは、そのタスクを実行する際に必然的に働く構造的特徴を明らかにしていないからである。」

ここで彼女が行おうとしていることは、「何の構造もない」ように見える言語データの中に、豊かな構造があることを見つけ出そうとしているのです。

それは、AI研究者は見過ごしているが、大規模言語モデル自身が、本当は何かを密かに知っていて、我々の知らないところでそれを利用している何かがあるのだという問題意識だと考えていいと思います。

それが、先に述べた「構造的プライア」だと思います。


実践的な背景
「Transformer意味論」の抱える問題

「Transformer意味論」の抱える問題 1
説明可能性の欠如

Bradley とCoeckeのアプローチは異なるのですが、LLMの働きについての、現在の一般的な理解に対する批判という点では両者は一致しています。

両者は、現在のLLMの意味解釈のプロセスの不透明性を批判します。それはLLMを「ブラックボックス」と見る批判と共通しています。

Transformerは、その驚異的な性能にもかかわらず、学習された表現の内部の論理を解読することが極めて困難です。その決定プロセスは、人間が理解できるシンボリックな推論ではなく、膨大な数のパラメータ間の複雑な相互作用に埋没しています。

これに対し、CoeckeのString diagramやBradleyのenriched カテゴリー構造は、意味について解釈可能なモデルを提供します。

意味の合成ステップは、隠された重みの中ではなく、数学的に透明な形で明示されることになります。この透明性は、モデルの検証、デバッグ、そして数学的証明など厳密な論理が求められる分野での応用において、決定的に重要になります。

連続セミナーの中では、「ニューロ・シンボリック」という新しいアプローチも紹介したいと思います。

Bradley とCoecke 両者のアプローチは、単に性能を追求するだけでなく、なぜその結論に至ったのかを説明できる、原理に基づいたAIを目指していると考えることができます。

「Transformer意味論」の抱える問題 2
意味の文脈依存性をめぐって

Transformerは、自己アテンション機構（self-attention mechanism）を用いて、文脈を考慮した単語埋め込みを生成します。このメカニズムにより、シーケンス内の全てのトークンが他の全てのトークンと相互作用し、その結果で全体論的な表現を構築することが可能になります。

しかし、この合成プロセスは暗黙的（implicit）かつソフト（soft）なものです。アテンションの重みは情報の混合比を決定するのですが、その合成を支配する明示的で厳密な文法的・論理的構造は存在しません。それは構造化された導出ではなく、重み付け和に過ぎないものです。このため、Transformerは訓練データにはない新しい組み合わせに対して体系的に汎化する能力（合成的一般化）に課題があると指摘されています。

対照的に、BradleyとCoeckeとのフレームワークは、いずれも明示的(explicit)な合成の数学的計算論に基づいています。

一方は enriched categoryの論理であり、他方は pregroupの代数です。規則は形式的で厳密(hard)であり、それは、学習によって獲得されるソフトなものではありません。この構造的な厳密さが、特定のベンチマークにおいて優れた合成的一般化能力をもたらすと考えられています。

同じ文脈依存の意味の埋め込みとして、Bradleyの「米田の視点」に導かれたco-presheafの構成とTransformerのアテンション機構を比較するのは、示唆的です。

Bradleyのフレームワークでは、トークンsの意味は、copresheaf L(s, -)であり、これはsと他のすべての文脈s`との関係を表すマップの可能的な全体です。s’が、学習データに含まれている必要はありません。

それに対して、Transformerの自己アテンションでは、トークンtの意味は、他のすべてのトークンt`の値の重み付き和によって形成されます。この重みはtのクエリとt′のキーの関数です。これは、Transformerでは、トークンtの意味は、その時点でTransformerが参照できる文脈の範囲内で決定されることを意味します。

Transformerの意味形成は情報の伝播において局所的かつ動的であるのに対し、Bradleyの意味は大域的かつ静的です。

この視点からは、アテンション機構がそのクエリ･キー･値システムで計算しようとしているものは、表現可能なco-presheafという原理的な数学的対象の近似である、と考えることも可能かもしれません。

「Transformer意味論」の抱える問題 3
学習データの巨大化と力まかせの処理

Transformerアーキテクチャは、その成功のために膨大な量のデータと計算リソースを必要としています。その性能は、モデルとデータの規模に大きく依存しています。それは、LLMが学習に利用する言語データが、「何の構造も持たない」と前提しているからです。

これに対し、BradleyとCoeckeのカテゴリー論的アプローチでは、何の構造も持たないように見える言語データ自身に、すでに構造が先在的に内在していると考えます。こうした構造を、「構造的プライア（structural prior）」と呼びます。

Bradleyのenriched co-presheaf は統計-圏論的なものであり、Coeckeのpregroup grammarは文法的なもので、両者の性質は異なっていますが、これらは彼らの理論の「構造的プライア」です。

両者は、言語モデルに、構造的プライアを組み込むことで、より少ないデータからより効率的に学習し、より良く汎化できると提案しています。

Bradleyのフレームワークは、テンソルネットワーク近似が高次元の作用素を効率的に扱える可能性を示唆しています。Coeckeの「量子ネイティブ」アプローチ は、空間リソースの指数関数的な節約を明確に約束しています。

Part 3

Bradleyの理論の発展をたどる

意味の理論モデルの発展をたどる

当初は、今回のセミナーの目的を、今後予定している一連の「LLMの理論モデルの新しい展開」セミナーの全体の展望を紹介することだと述べてきました。ただ、個別の具体的なセミナーの内容を十分語らずに、その展望を語ることは難しいのです。

そこで、今回のセミナーの目的を、今後の個別のセミナーを理解するのに必要な基本的な準備をすることに切り替えました。

その点でも、いろいろ語るべきことは多いのですが、基本的な情報は、意味の理論モデルの発展が辿ってきた道だと思います。

Bradleyの理論の発展をたどる

意味の理論モデルの発展をたどった道を、基本的な情報として共有しようということで、今回のセミナーがとったやり方は、少しかたよったものに見えるかもしれません。

このセミナーの Part 3, Part 4では、意味の理論モデルの発展一般ではなく、Bradleyの理論の発展にフォーカスしています。

Bradleyの理論の発展

Bradleyの理論の発展を追う

Part3とPart 4では、現在のLLMと意味の理論モデル中心人物一人であるのTai-Danae Bradley の理論の発展を追ってみようと思います。

Part3とPart 4では、彼女の次の4つの論文を取り上げます。ただ、LLMと意味の理論モデルの構築で、彼女が依拠した理論的枠組みは、これら4つの論文で、それぞれ異なっています。

こうした理論の変化とその意味を追うことを通じて、意味の理論モデルの発展をたどることが、今回のセミナーのPart3とPart 4の目的です。

Part3とPart 4で取り上げる論文

1. 2018年 [What is Applied Category Theory?]
「応用カテゴリー論とは何か？」
2. 2020年 [Language Modeling with Reduced Densities]
「Reduced Densityでの言語モデル」
3. 2021年 [An enriched category theory of language: from syntax to semantics]
「言語のenrichedカテゴリー論：構文論から意味論へ」
4. 2025年 [The Magnitude of Categories of Texts Enriched by Language Models]
「言語モデルでenrich化されたテキストのカテゴリーのマグニチュード」

それぞれの論文が依拠した理論的枠組み

1. 2018年 What is Applied Category Theory?
   DisCoCat
2. 2020年 Language Modeling with Reduced Densities
   Reduced Density
3. 2021年 An enriched category theory of language: from syntax to semantics
   Enriched Category
4. 2025年 The Magnitude of Categories of Texts Enriched by Language Models
   Magnitude

研究は発展するものですので、論文ごとに依拠する理論的枠組みが変わるのは、不思議なことではありません。ただ、彼女の研究の発展を振り返ったとき、重要なことは、1番目の論文と2番目の論文の間に、LLMと意味についてのアプローチに大きな変化があることです。

彼女は一番目の論文で依拠したDisCoCatモデルを離れ、その批判の上に彼女独自のLLMと意味の理論モデルの構築を開始します。

Part 3のセッションの構成

Part 3のセッシヨンは、次のような構成になっています。

まず、Bradleyの研究が生まれた時代背景を、「意味の分散表現論の系譜 – 大規模言語モデルへ」という流れの中で考えます。

次に、BradleyのDisCoCatモデルについての2018年の論文”What is Applied Category Theory?” の内容を少し詳しく紹介します。

最後に、Bradleyの2020年の論文 “Language Modeling with Reduced Densities” を紹介します。


Bradleyの研究の時代背景
意味の分散表現論の系譜

Bradleyの研究の時代背景

Bradleyが研究を始めた時代を振り返っておきましょう。

21世紀の初めに、Bengioらによって始められた「意味の分散表現論」は、10数年の時間をかけて「大規模言語モデル」に発展しました。

その流れを、年表「意味の分散表現論の系譜 – 大規模言語モデルへ」にまとめておきました。

意味の分散表現論の系譜 – 大規模言語モデルへ

- 2003年 Bengioの「次元の呪い」と語の特徴の分散表現
- 2006年 HintonのAuto Encoder 意味的ハッシング
- 2011年 RNNによる文の生成
- 2013年 Word2Vec 語の意味ベクトル
- 2014年 Sequence to Sequence 文の意味ベクトル
- 2015年 RNNの不思議な力 RNNは文法を理解している
- 2016年 Attention Mechanism
- 2016年 Google ニューラル機械翻訳
- 2017年 Transformer
- 2019年 BERT
- 2020年 GPT-3
- 2022年 ChatGPT

この年表については、もう少し詳細な資料を作っています。後で紹介します。

意味の分散表現論の系譜 – CoeckeとBradley

- 2003年 Bengioの「次元の呪い」と語の特徴の分散表現
- 2006年 HintonのAuto Encoder 意味的ハッシング
- 2011年 RNNによる文の生成
- 2013年 Word2Vec 語の意味ベクトル
- 2014年 Sequence to Sequence 文の意味ベクトル
- 2015年 RNNの不思議な力 RNNは文法を理解している
- 2016年 Attention Mechanism
- 2016年 Google ニューラル機械翻訳
- 2017年 Transformer
- 2019年 BERT
- 2020年 GPT-3
- 2022年 ChatGPT

2010年 Coecke DisCoCat論文

2018年 Bradley DisCoCat解説論文

2020年 Bradley DisCoCat批判

2021年 Bradley copresheaf 論文

BradleyのDisCoCatモデルからの離脱が、大規模言語モデルの成立期に起きていることがわかると思います。

それは偶然ではありません。後で見るように、Bradley自身が語っているように、彼女は、大規模言語モデルの成功に強い印象を受けます。同時に、その成功がDisCoCatモデルに依拠したものでないことにも気づきます。

何が、大規模言語モデルの成功を支えているのか？　彼女の理論的な関心のフォーカスは、そこに向けられていきます。

この年表を見て面白いことが、もう一つあります。

それは、AI研究の側での「意味の分散表現論」に大きな影響を与えた「Word2Vec」論文（今読むと、とてもプリミティブなものですが）が発表される3年近く前に、Coeckeらの「意味の分散表現論」DisCoCat論文が、ほとんど完成した形で登場していることです。ある意味、驚くべきことです。

「意味の分散表現論の系譜 − 大規模言語モデルへ」のpdf資料

https://drive.google.com/file/d/1OOE7uXBScMi2eoWNKUTZLC4xwXu_nIGZ/view?usp=sharing

YouTube 「意味の分散表現論の系譜 − 大規模言語モデルへ」 再生リスト Geneology of LLM

https://www.youtube.com/playlist?list=PLQIrJ0f9gMcPZfOVwH0wdu_PoELeAPqon


DisCoCatは、なぜ、重要なのか？

DisCotCat

DisCoCatは、Distributional Compositional Categorical Semantics の略称です。日本語にすれば、「カテゴリー論的構成的分散意味論」ということになります。

DisCoCatを提唱した最初の論文は、Bob Coeckeらが2010年に発表した次の論文です。

“Mathematical Foundations for a Compositional Distributional Model of Meaning”

https://arxiv.org/abs/1003.4394

DisCoCatでCoeckeが考えたこと

DisCoCatでのCoeckeの問題意識は、明確でかつ意欲的なものでした。

「意味の記号論と分散意味論は、ある意味直交する競合する理論である。それぞれに長所と短所がある、前者は構成的だが定性的で、後者は非構成的だが定量的である。」

認知科学の分野でも、当時、こころのコネクショニスト的モデルと記号論的モデルの間に、同じような立場の違いが存在しました。

Coeckeが考えたことは、この二つの陣営の対立を乗り越える道はないのかということです。

ポイントは、二つの陣営の対立を乗り越えるのに、カテゴリー論を利用しようということです。

「文法としてのPregroupが特に興味深く、我々がそれを使用する動機となったのは、カテゴリー論的な観点に立つと、それがベクトル空間やテンソル積と共通の構造を持つということである。」

このあたりは、実は、先日紹介した、Functor Semanticsについてのlawvereの考えと同じものなのです。

彼は、対立する陣営が、対立はしているものの、それぞれがその「市民」である「第三のカテゴリー」を具体的に発見します。

「ベクトル空間、線形写像、テンソル積のカテゴリも、Pregroupも、いわゆるcompact closedカテゴリと呼ばれるものの例である。具体的には、Pregroupでの型の並置は、monoidalカテゴリーのmonoidalテンソルに対応する。」

「文の意味を計算する数学的構造は、意味と型の二つを組み合わせたcompact closedカテゴリーとなる。」

素晴らしい！

こうした視点は、CockeがDisCoCatの言語理論をQNLP(量子論的自然言語処理)に拡張する際の指導原理になります。というのも、量子論もcompact closedカテゴリーに属するからです。

DisCoCatは、なぜ、重要なのか？
言語と意味の「構成性」

なぜ、15年前のDisCoCat 論文が重要なのでしょう？

第一に、現在のLLMと意味の理論構築の中心的関心が、成功を収めたTransformer的分散意味論の上に、言語と意味の「構成性」を回復することにあるからです。

分散意味論と構成的意味論の統一という課題に、最初に明確に取り組んだのがDisCoCat論文です。

DisCoCatは、なぜ、重要なのか？
カテゴリー論の利用

第二に、分散意味論と構成的意味論の統一という課題に、数学的ツールとして、カテゴリー論を利用することを最初に提起して、成功裡にそれを成し遂げたのは、DisCoCat論文が初めてだったからです。

その 理論的影響は、今日も続いています。

LLMと意味の理論の現代的深化は、カテゴリー論の利用なしには、考えることができません。

DisCoCatは、なぜ、重要なのか？
CoeckeとBradley

第三に、理論的というより属人的なことと思われるかもしれませんが、現代のLLMと意味の理論の代表的担い手は、Bob CoeckeとTai-Danae Bradleyです。

CoeckeとBradleyは、共に、DisCoCatのメンバーとして問題意識を共有していました。

二人のアプローチは、現在では少し異なるのですが、現代のLLMと意味の理論の源流は、DisCoCatにあると僕は考えています。


BradleyとDisCoCat

DisCoCatでBradleyが考えたこと

僕が知る限り、DisCoCat論文の最もわかりやすい解説は、Tai-Danae Bradleの次の論文です。

What is applied category theory
3.2 Natural Language Processing

https://arxiv.org/pdf/1809.05923.pdf

彼女は、DisCoCatの枠組みに基づいて、"banana is fruit"という意味の分散ベクトルを計算してみせたのです。

彼女のDisCoCat解説は、見事なものです。文 "banana is fruit"の意味は、19741 と表示されるそうです。

2018年の論文の概要

まず、DisCoCatモデルに基づくBradleyの2018年の論文 ”What is Applied Category Theory?” の概要を見ていきたいと思います。

ここでは、次のような問題意識が述べられています。

# Syntax = 文法
Semantics = 意味

「Syntax = 文法」と「Semantics = 意味」の対応づけに、Functorial Semanticsのフレームを使うには、どのような準備が必要だろうか？　次の三つが考えられる。

1. 文法のCategory: 文法を数学的に扱うにはどうすればよいか？　文法を数学的なCategoryとして捉える。
2. 意味のCategory: 意味を数学的に扱うにはどうすればよいか？　意味を数学的なCategoryとして捉える。
3. Functor:　文法 → 意味 のFunctorをどのように構成するか？　このFunctorは、文全体の意味を捉えることができるか？

文法のCategory
PregX

文法のCategoryでは、LambekのPregroup Grammar を使う。（ただし、その簡略版である。）
このカテゴリーをPregXで表そう。

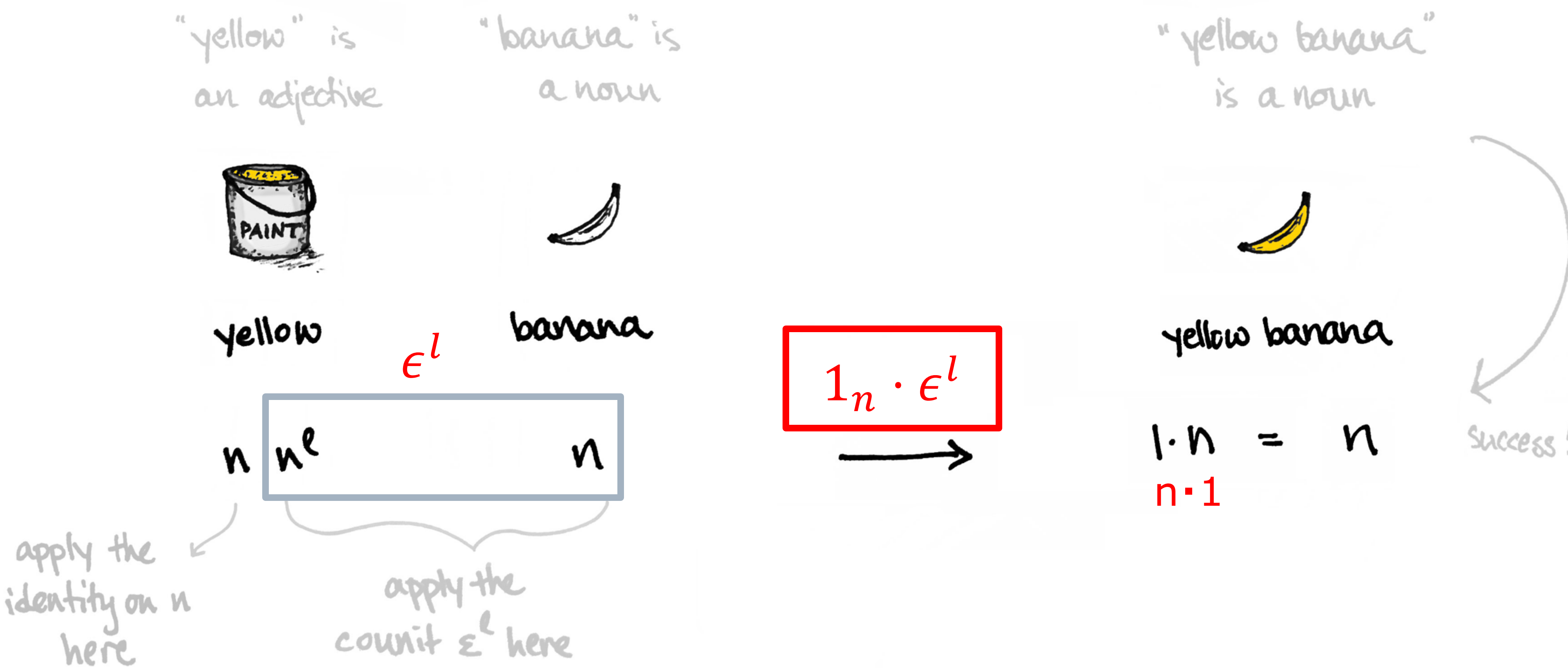
基本的な文法の型は、名詞の型 n と文の型 s である。これから、基本的な品詞の型が定義されていく。

n ＝ 名詞
s ＝ 文
nn^l ＝ 形容詞
$n^r s n^l$ ＝ 他動詞

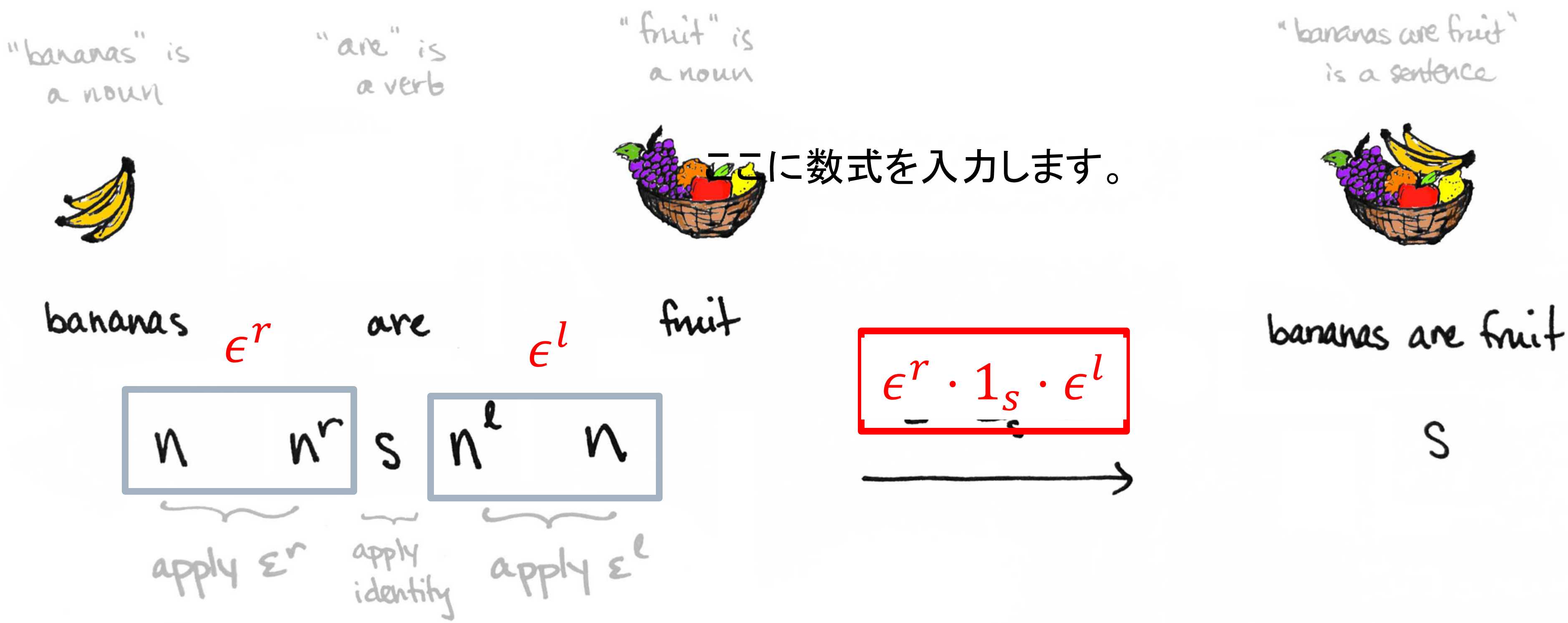
語が連接する時、対応する語の型で、次のような計算を行う。

$w^l\, w = \varepsilon^l$ （消える）
$w w^r = \varepsilon^r$ （消える）

“yellow banana” は名詞として扱われる



“bananas are fruit” は文である



$$nn^r s n^l n = nn^r \cdot s \cdot n^l n \xrightarrow{\epsilon^r \cdot 1_s \cdot 1_n} 1 \cdot s \cdot n^l n = s \cdot n^l n \xrightarrow{1_s \cdot \epsilon^l} s \cdot 1 = s$$

意味のカテゴリー ＝ ベクトル空間(Fvec) の例

{sweet, green, furry} という語を含むある本があったとしよう。この三つの語をこのコーパスのcontext word に選んだら、これを基底として、次のようにベクトルで表す。

$$\textbf{sweet} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \textbf{green} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \textbf{furry} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

この時、この本の中の語 banana, puppy, fruit は、次のように表される。

$$\textbf{banana} = \begin{bmatrix} 21 \\ 9 \\ 0 \end{bmatrix} \qquad \textbf{puppy} = \begin{bmatrix} 8 \\ 1 \\ 32 \end{bmatrix} \qquad \textbf{fruit} = \begin{bmatrix} 43 \\ 19 \\ 0 \end{bmatrix}$$

meanings of words = FVect

二つの語のベクトルの内積を計算できれば、それらのベクトルが近さがわかり、もしもそれらが同じ意味を持っていたとすれば、それを正確に表現できる。これは、NLPで仕事をしている人にはよく知られていることだ。

Coeckeらの意味論的カテゴリーは、こうして与えられる。それは、$\mathbb{R}$上の有限次元のベクトル空間である。

meanings of words = FVect

語の分散モデルは、文には適用できない！
カテゴリー論が必要になる

残念ながら、先の語の分散モデルは、文には適用できない。文書内で同じ文が現れることはほとんどないので、先の考えではうまくいかない。

ここで、カテゴリー論が助けの手を差し出してくれる。

構成性の原理に照らせば、文の意味は、文を構成する個々の語の意味とそれらを結びつける文法のルールで計算されるべきである。

我々は、「語の意味」とその語の「文法的な型」のペアを、syntax (文法）からsemantics(語の意味)への写像を通じて作ることができる。すなわち、次のようなfunctor F を通じて。

$$F : PregX \rightarrow FVect$$

The Functor: Syntax → Semantics の明示的な記述

この節では、functor F : syntax → semantics の明示的な記述を与える。より具体的には、F : PregX → FVect の。

オブジェクトと射についてのfunctorの働きを定義すればいい。

The Functor: Syntax → Semantics on objects

On objects

- **F** assigns to the noun type n
  a vector space N : Fn,
  which we'll call a **noun space**
- **F** assigns to the sentence type s
  a vector space S : Fs,
  which we'll call a **sentence space**

The Functor: Syntax → Semantics
on objects

On morphisms

F assigns to a type reduction $a \xrightarrow{r} b$

a linear map $Fa \xrightarrow{Fr} Fb$

that sends the vector corresponding to a word or phrase of type a in Fa to the vector corresponding to a word or phrase of type b in Fb.

Functor F: Preg → FVect
Fは、compact closed structure を保存する

- Fは、$Preg\{n,s\}$の**unit** $\eta_n^r: 1 \to n^r n,\ \eta_n^l: 1 \to n n^l$を、
  $FVect$の**unit** $\eta_N: \mathbb{R} \to N \otimes N$にうつす。

$$F(\eta_n^r) = F(\eta_n^l) = \eta_N \quad (s\text{についても同様})$$

- Fは、$Preg\{n,s\}$の**counit** $\epsilon_n^r: n^r n \to 1,\ \epsilon_n^l: n n^l \to 1$を、
  $FVect$の**counit** $\epsilon_N: N \otimes N \to \mathbb{R}$ にうつす。

$$F(\epsilon_n^r) = F(\epsilon_n^l) = \epsilon_N \quad (s\text{についても同様})$$

Functor F: Preg → FVect
Fは、compact closed structure を保存する

- Fは、$Preg\{n,s\}$の**dual**を、$FVect$の**dual**にうつす。
  $F(n^r) = F(n^l) = N^*$ ただし$FVect$は有限次元なので$N^* \cong N$

$\boxed{\mathrm{F}(n^r) = F(n^l) = N}$ (sについても同様)

- Fは、$Preg\{n,s\}$の複合型を、$FVect$のテンソル積にうつす。

e.g. $F(n^r s n^l) \cong F(n^r) \otimes F(s) \otimes F(n^l) = N \otimes S \otimes N$

Step 4: Preg(n, s)で、型の還元を行う

banana is fruit

$$n\ n^r s n^l\ n \xrightarrow{\epsilon_n^r \cdot 1_s \cdot \epsilon_n^l} 1s1 \longrightarrow s$$

$$nn^r \xrightarrow{\epsilon_n^r} 1 \qquad s \xrightarrow{1_s} s \qquad n^l n \xrightarrow{\epsilon_n^l} 1$$

Step 5: Functor F を適用する！
Syntax → Semantics

banana is fruit

$$n\ n^r s n^l\ n \xrightarrow{\epsilon_n^r \cdot 1_s \cdot \epsilon_n^l} s$$

Syntax

$$Fn = N,$$
$$Fn^r = N, Fs = S, Fn^l = N,$$
$$Fn = N$$

$$F(\epsilon_n^r : nn^r \to 1) = \epsilon_N$$
$$F(1_s) = 1_S$$
$$F(\epsilon_n^l : nn^l \to 1) = \epsilon_N$$

$$Fs = S$$

Functor F

$$N \otimes N \otimes S \otimes N \otimes N \xrightarrow{\epsilon_N \otimes 1_S \otimes \epsilon_N} S$$

Semantics


BradleyのDisCoCatモデルからの離脱

BradleyのDisCoCatからの離脱

Bradleyの2020年の論文 ” Language Modeling with Reduced Densities” の概要を見ていきたいと思います。BradleyのDisCoCatからの離脱をテーマとするこのセッションでは、そのことを宣言したこの論文は、大きな意味を持っています。

この論文の直接のテーマは、タイトルにあるように、意味の表現を有限次元のベクトルから密度行列に変更しようとするものです。現在の大規模言語モデルはもちろんのこと、それにはるかに先行したDisCoCatも、意味の表現には有限次元のベクトルを利用しています。

両者は、意味を空間上の一点として表現しているのです。その問題に切り込みたかったのだと思います。

大規模言語モデルの登場とBradleyが考えたこと

ただ、今回のセッションの焦点はそこではありません。大規模言語モデル の成功に対する鋭い観察は注目すべきものです。

「この研究は、今日の最先端の統計的言語モデルが、その性能において印象的であるだけでなく、より本質的に重要なことは、それが構造化されていないテキストデータの相関関係から完全に構築されているという観察から生まれたものです。」

彼女の関心は、現在の「大規模言語モデル」の「印象的」な成功に向けられています。彼女はそれがDisCoCatモデルとは少し異なる言語モデルであることも知っています。その上で、その背後にあるものを探り出そうとしているのです。

大規模言語モデルが行なっていること
彼女は何に強い印象を受けたのか?

彼女が大規模言語モデルのどんな性能に印象を受けたのかを、改めてみておこうと思います。

「(大規模言語モデルは、)対話的に、最初の文を入力すると、次の単語分布から繰り返しサンプリングすることで、オリジナルの高品質なテキストを生成することができる。」

確かにそう言われています。ただ、それだけでしょうか?
彼女は続けます。

「直感的に言えば、物語を続ける能力は、非常に高度なことを意味している。

文法的に正しい文を継続するためには、文法を習得し、注意深く代名詞のマッチングを行い、品詞の認識を持ち、時制の感覚も持たなければならない。その他多くのことを必要とする。

可能な継続の確率分布を効果的に学習する言語モデルは、明らかに意味的知識も学習しているはずだ。

物語の続きが合理的で内部的に一貫しているためには、世界に関する知識が必要である。内部的に一貫しているためには、犬とは吠える動物であり、ゴルフは日中屋外でプレーするものである、火曜日は月曜日の翌日である、などといった世界の知識が必要である。」

この驚きの感覚に、僕は共感します。

より驚くべき大規模言語モデルの能力

問題は、それだけではありません。

「驚くべきは、これらのLLMが、ラベルのないテキストサンプルを使って、次の単語を事前に指示するように学習できることである。文法的、意味的な入力は提供されないが、それにもかかわらず、複雑な構文構造、意味情報、世界知識が学習され、実証される。」

それは DisCoCatの理論家だったTai-Danaeにとっては深刻な問題だったと思います。なぜなら、前回のセッションでも述べたように、DisCoCatモデルは、入力が文法的に解析されていることを前提としているからです。

問題の立て方が重要である

僕にとって印象的だったのは、彼女が次々と問題を立てることでした。答えの前には、もちろん、問題があります。ただ、答えを見つける条件が成熟するというのは、正しく問題をたてることができるということです。

- 自然言語における表現の意味をとらえる数学的構造は何か？
- この構造は、テキスト・コーパスを用いてどの程度まで十分に検出できるのか？
- 抽象的な概念とその相互関係を自然に掘りだす方法はあるのか？
- 論理と命題の連関はどのようにして生まれるのか？

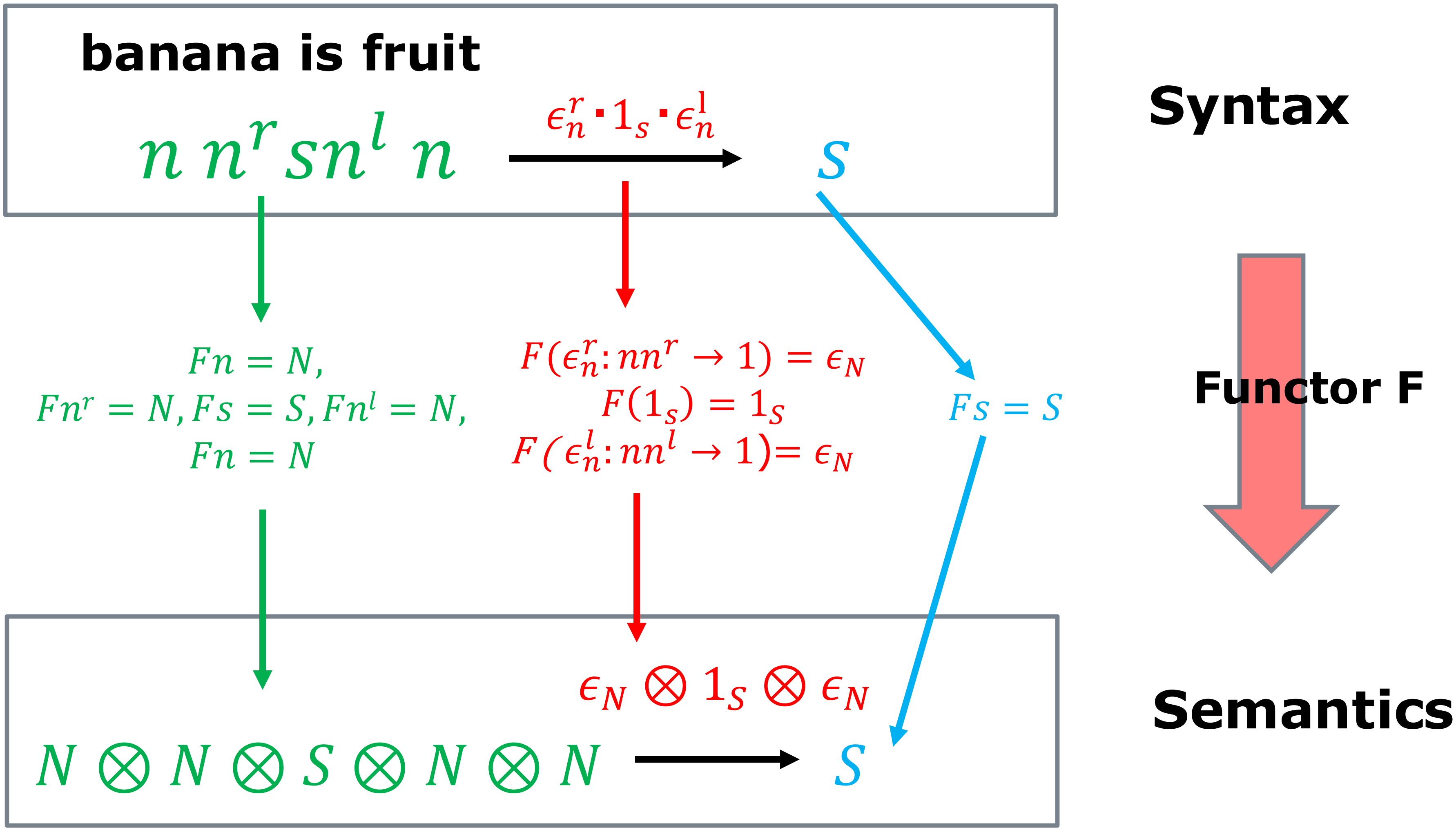
二つの基本的問題

こうして、彼女は、次の二つが基本的な問題だとします。

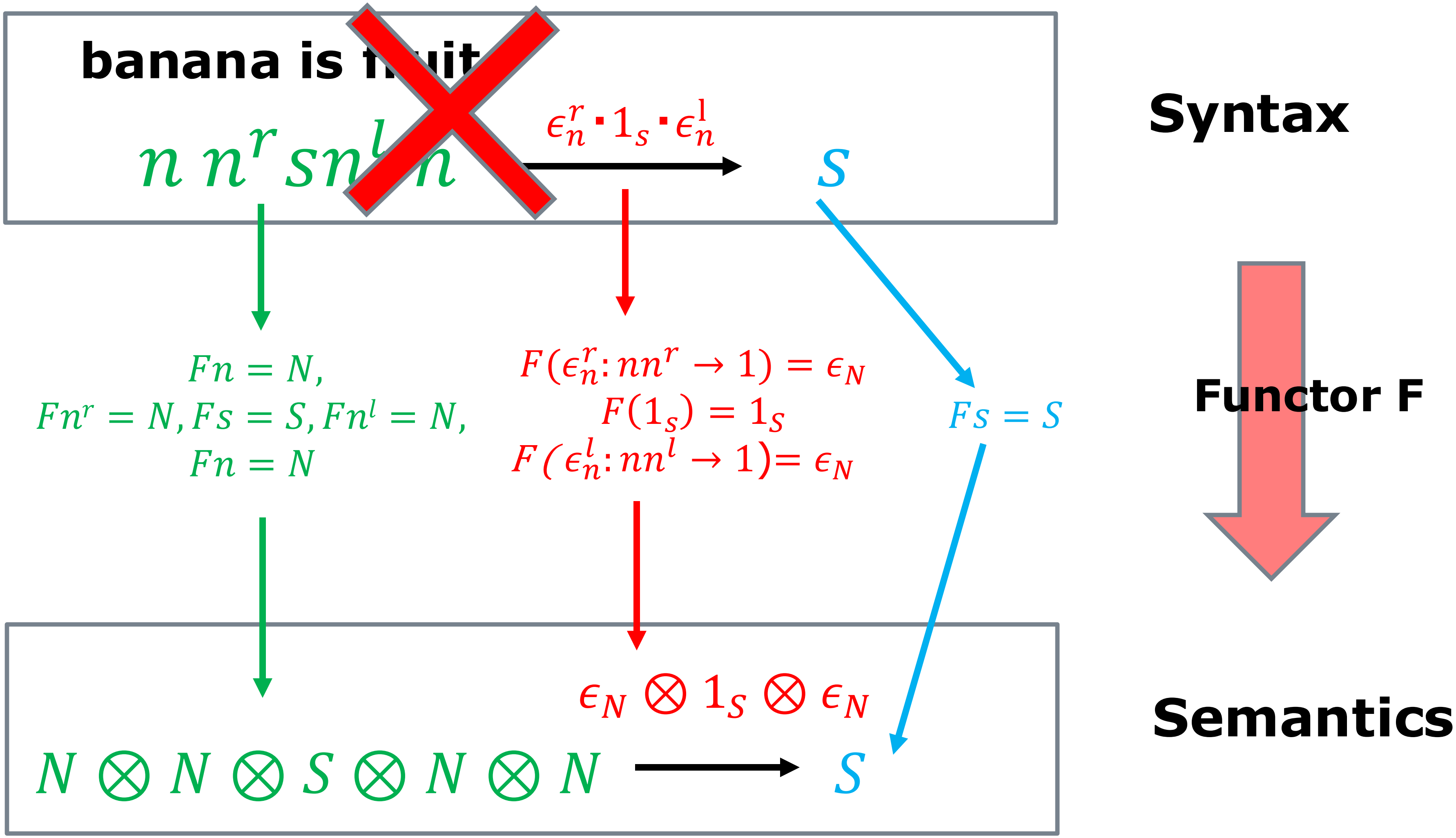
- どのような数学的構造が、構造化されていないテキストデータには存在するのか？
- どのようにして、テキストの情報は、カテゴリー構造を維持したまま保存され、モデル化することができるのか。

こうした問題意識は、この論文以降ずっと維持されることになります。

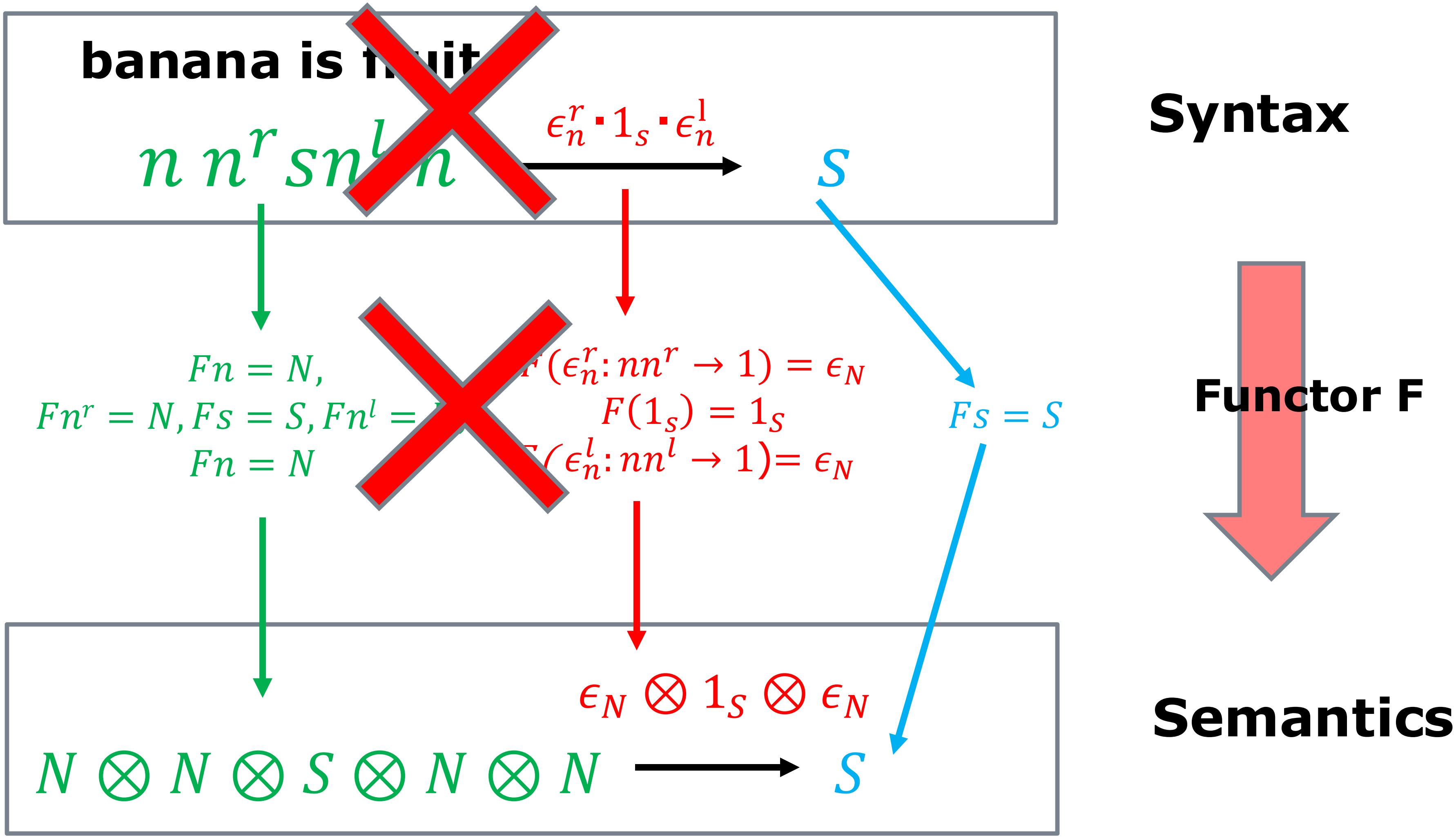
以前の彼女の図式


banana is fruit
$n\ n^r s n^l\ n$
$\epsilon_n^r \cdot 1_s \cdot \epsilon_n^l$
s
Syntax
$Fn = N,$
$Fn^r = N, Fs = S, Fn^l = N,$
$Fn = N$
$F(\epsilon_n^r : nn^r \to 1) = \epsilon_N$
$F(1_s) = 1_S$
$F(\epsilon_n^l : nn^l \to 1) = \epsilon_N$
$Fs = S$
Functor F
$\epsilon_N \otimes 1_S \otimes \epsilon_N$
$N \otimes N \otimes S \otimes N \otimes N$
S
Semantics

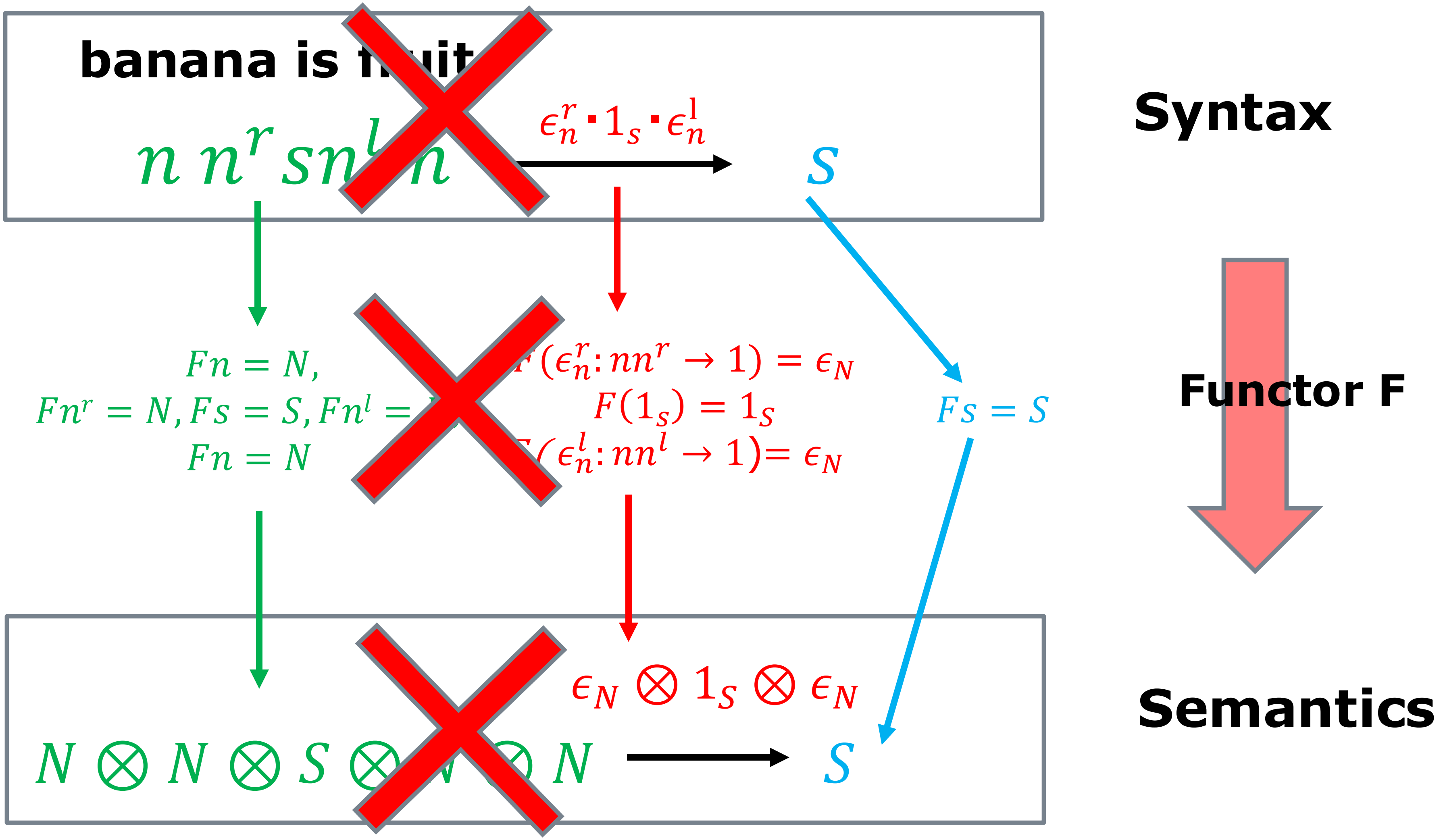
構造化されていないテキスト



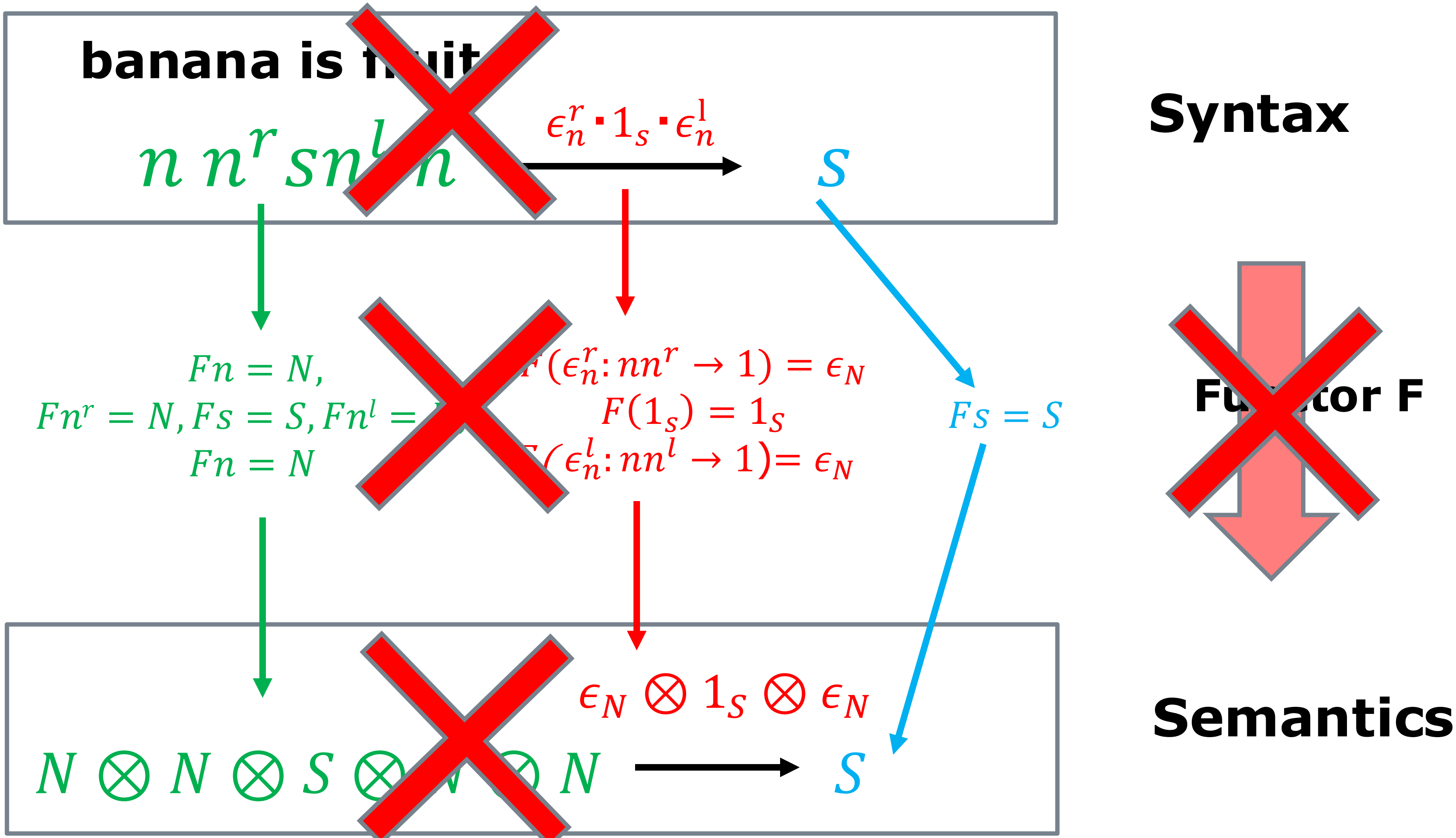
構造化されていないテキスト



構造化されていないテキスト



BradleyのDisCoCat批判



Part 4

Bradleyのcopresheaf意味論


copresheaf意味論と
enrichカテゴリー論の応用

copresheaf意味論とenrichカテゴリー論の応用

このPart 4では、前回に続いて、Bradleyの理論の発展を追っていきます。次の論文の概要を紹介します。

3. 2021年　An enriched category theory of language: from syntax to semantics
「言語のenrichedカテゴリー論：構文論から意味論へ」

4. 2025年　The Magnitude of Categories of Texts Enriched by Language Models
「言語モデルでenrich化されたテキストのカテゴリーのマグニチュード」

特に、2021年の論文は重要です。

大規模言語モデルが見せる驚くべき能力を目の当たりにして、なぜ、そんなことが可能なのかと真剣に考え始め、それに理論的に答えようとして書かれたものです。

AIは素晴らしいという情報は多いですし、AIはブラックボックスだと批判する議論も少なくないのですが、その謎に正面から挑戦した論文は、意外と少ないものです。

彼女の理論的取り組みは、貴重なものだと思います。

Bradleyの基本的な考えは、この論文の前半部に集約されている

Bradleyの2021年の論文は、”An enriched category theory of language”というタイトルがしめすように、enrich化されたカテゴリー論で言語の理論的モデルを与えることを目指したものです。

たしかに、論文の後半は、言語とenrichカテゴリーについての数学的な議論が進行します。

ただ重要なことは、LLMと意味の理論へのカテゴリー論的アプローチの導入というBradleyの基本的考え方の特徴は、ほとんどすべて論文の前半部分で展開されているということです。

2021年論文前半の構成

論文の前半は、次のように進行します。

- 言語のカテゴリー Lを、preorder category として捉える
- 表現 Sの「意味」を、カテゴリー L 内のS → T なるなる射全体からなる集合と考える
- 意味のカテゴリーは、copresheaf Set^L で表現される
- 言語のカテゴリーと意味のカテゴリーの対応は、米田の理論で与えられる
- こうして構成された表現の上で、ORやANDといった論理的な操作の意味を解釈しうることを示す

意味のカテゴリー論的解釈
copresheaf意味論

こうした進行のステップや概念構成には、言語学的にもカテゴリー論的にも、興味深い意味があります。

論文の前半部分は、最も基本的な意味のカテゴリー論的解釈を与えている部分です。セミナーでは、Bradleyが展開した、意味のカテゴリー論的解釈を「copresheaf意味論」と呼んでいます。（その言葉の意味は、後で説明します。）

今回のセッションの重点として、2021年の論文の前半で展開された、このcopresheaf意味論を、少し丁寧に紹介していきたいと思っています。

繰り返される意味の基本的構造

実は、enrich論を展開した論文の後半部分は、前半で構成された意味のカテゴリー論である copresheaf意味論を、その概念構成に利用した数学的ツール（カテゴリー、functor、copresheaf等）をまるごとenrich化して、copresheaf意味論のenrich版を再構成しようとしたものに他なりません。

さらにいえば、2025年の論文は、enrich化の手法は2021年の論文とは違うのですが、その基本的骨組みは、その新しい手法の上に copresheaf 意味論を再構築するものです。

Bradleyにとって、意味の基本構造は、2021年の論文の前半で明らかにした copresheaf 意味論なのです。

論文の後半は、この論文の主題であるLLMの理論モデルのenrichカテゴリー論上での構築を論じたものです。

- 言語のカテゴリーLを単位区間[0,1]でenrich化して確率を導入する
- functor、copresheafも、[0,1]で enrich化できることを示す
- これらのカテゴリーのenrich化のもとで、前半の構成を繰り返し、確率でenrich化された理論を構築する
- こうして構成された表現の上で、論理的な操作等の意味を解釈する

論文の大きな部分は、様々なカテゴリーのenrich化の数学的構成に当てられているのですが、今回のセッションでは、その詳細は割愛しています。

2021年と2025年の論文の enrich化の焦点の違い

2021年の論文は、あるテキストの後にあるテキストが出現する「確率」のモデルを、enrich化によって導入しました。ベクトル空間の「一点」として表現された意味は、確率によって結ばれます。これはLLMのモデルとしては、当然のことかもしれません。

2025年の論文は、あるテキストとあるテキストの間の「一般化された距離」の概念を、enrich化の手法で導入します。こうして、意味の「空間」は、さらに豊なものになります。この論文のハオライトは、「マグニチュード」とそのhomologyの導入です。

2025年の論文については、改めて独立のセミナーを開催しようと思っています。今回のセミナーでは、詳しい説明は行われません。


Bradleyのcopresheaf意味論

言語のカテゴリー Lを、
preorder category として捉える

言語を構成する意味を持つ文字列である、語・フレーズ・文・文の連続 ... を「表現」とします。任意の表現 S, Tについて、表現Sの文字列が表現Tの部分文字列であるとき、S ≦ Tと順序を定義します。

この順序は、次の性質が確かめられますので preorder（前順序）です。

1. S ≦ S （反射律）
2. S ≦ T かつ T ≦ U なら、S ≦ U （推移律）

preorderの射の合成の結合性についてです。

次のPとLの対応を見ればわかると思います。

P

$$\begin{tikzcd} S \arrow[r, phantom, "\leq"] \arrow[rr, bend left, "\leq"] & T \arrow[r, phantom, "\leq"] \arrow[rr, bend right, "\leq"'] & U \arrow[r, phantom, "\leq"] & V \end{tikzcd}$$

$$S \leq U \leq V \Rightarrow S \leq V$$

$$S \leq T \leq V \Rightarrow S \leq V$$

L

$$\begin{tikzcd} S \arrow[r, "f"'] \arrow[rr, bend left, "f \circ g"] & T \arrow[r, "g"'] \arrow[rr, bend right, "g \circ h"'] & U \arrow[r, "h"'] & V \end{tikzcd}$$

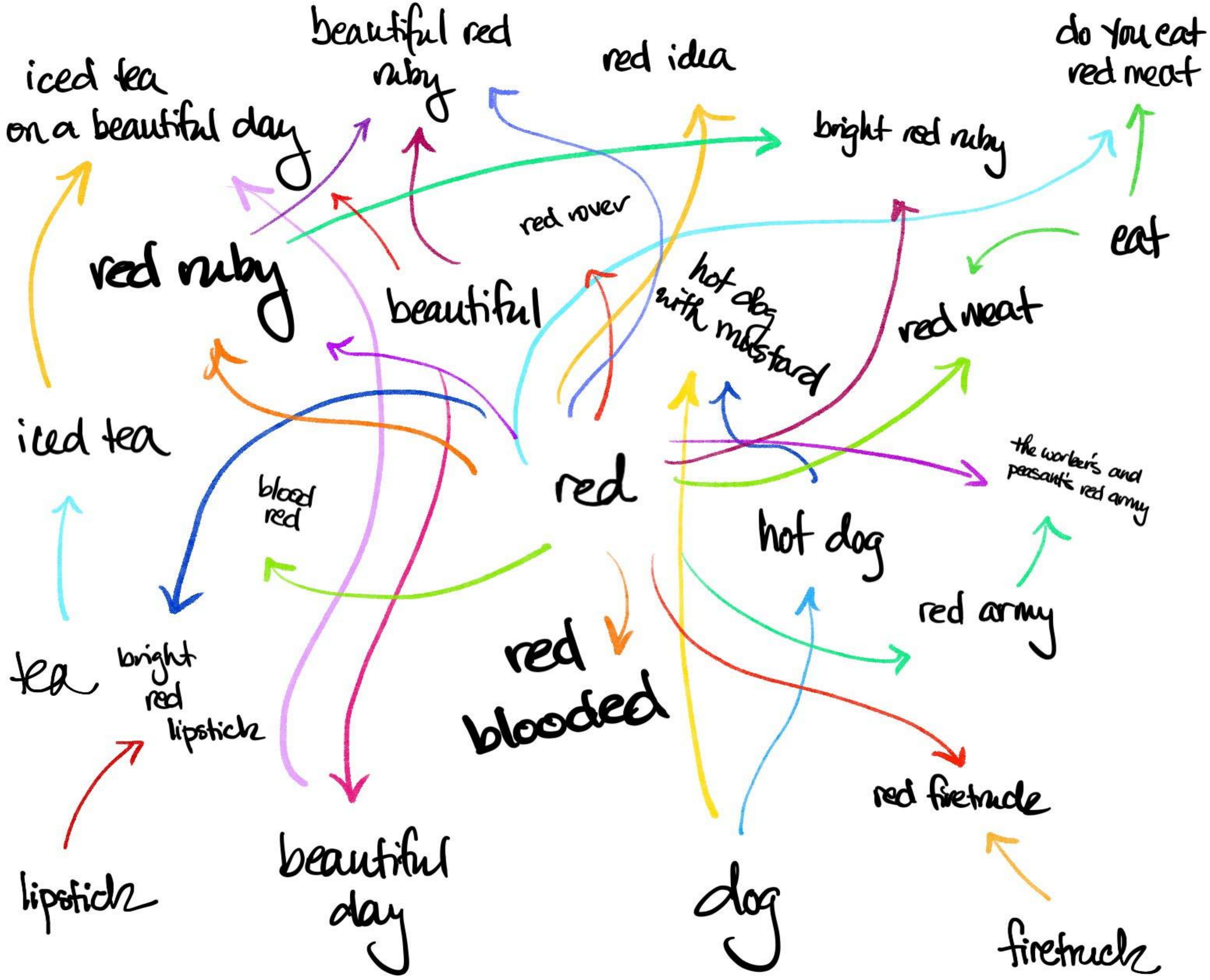
$$(f \circ g) \circ h = f \circ g \circ h$$

$$f \circ (g \circ h) = f \circ g \circ h$$

この時、言語の表現をオブジェクトとし、表現間の順序 $S \leqq T$ を $S \rightarrow T$ という射を定義すると考えると、言語は、preorder category として捉えることができることがわかります。

ここでは、次のことに注意しましょう。

- この言語のカテゴリー L では、二つの異なるオブジェクト間の射は、ただ一つ存在するか、あるいは、全く存在しないかのいずれかです。
- この言語のカテゴリー L は、ある言語が文字表現を持つ限り、日本語、英語、... といった違いを超えた言語の特徴を表現しています。


iced tea on a beautiful day
beautiful red ruby
red idea
do you eat red meat
bright red ruby
red rover
red ruby
beautiful
hot dog with mustard
eat
red meat
iced tea
blood red
red
the worker's and peasant's red army
hot dog
red army
tea
bright red lipstick
red blooded
red firetruck
lipstick
beautiful day
dog
firetruck

言語のpreorderの例

red $\leq$ red firetruck

red $\leq$ red blooded

red $\leq$ red ruby $\leq$ beautiful red ruby

red $\leq$ red ruby $\leq$ bright red ruby

red $\leq$ red meat $\leq$ eat red meat $\leq$ I eat red meat

dog $\leq$ hot dog $\leq$ hot dog with mustard

dog $\leq$ hot dog with mustard

beautiful $\leq$ beautiful day $\leq$ ice tea on a beautiful day

tea $\leq$ ice tea

…

日本語でも同じです。

赤い ≤ 赤い消防車

赤い ≤ 赤い血にまみれた

赤い ≤ 赤いルビー ≤ 美しい赤いルビー

赤い ≤ 赤いルビー ≤ 輝く赤いルビー

赤い ≤ 赤い肉 ≤ 赤い肉を食べる ≤ 私は赤い肉を食べる

ドッグ ≤ ホットドッグ ≤ カラシ付きのホットドッグ

ドッグ ≤ カラシ付きのホットドッグ

美しい ≤ 美しい日 ≤ 美しい日のアイスティー

ティー ≤ アイスティー

…

表現 Sの「意味」を、カテゴリー L 内の S → T なる射全体からなる集合と考える

Firth は、次のような言葉を残しています。

「我々は、ある語を、それが引きつれている仲間たちによって知ることになる。」

こうした意味の理解をベースにして、ある表現Sの意味を、それが「引きつれている仲間たち」の全体の集合で考えることにします。Sの「仲間たち」とは、S → T なる射が存在する表現Tのことです。

それらの全体を考えるということは、S → T なる射全体の集合を考えるということと同じことです。

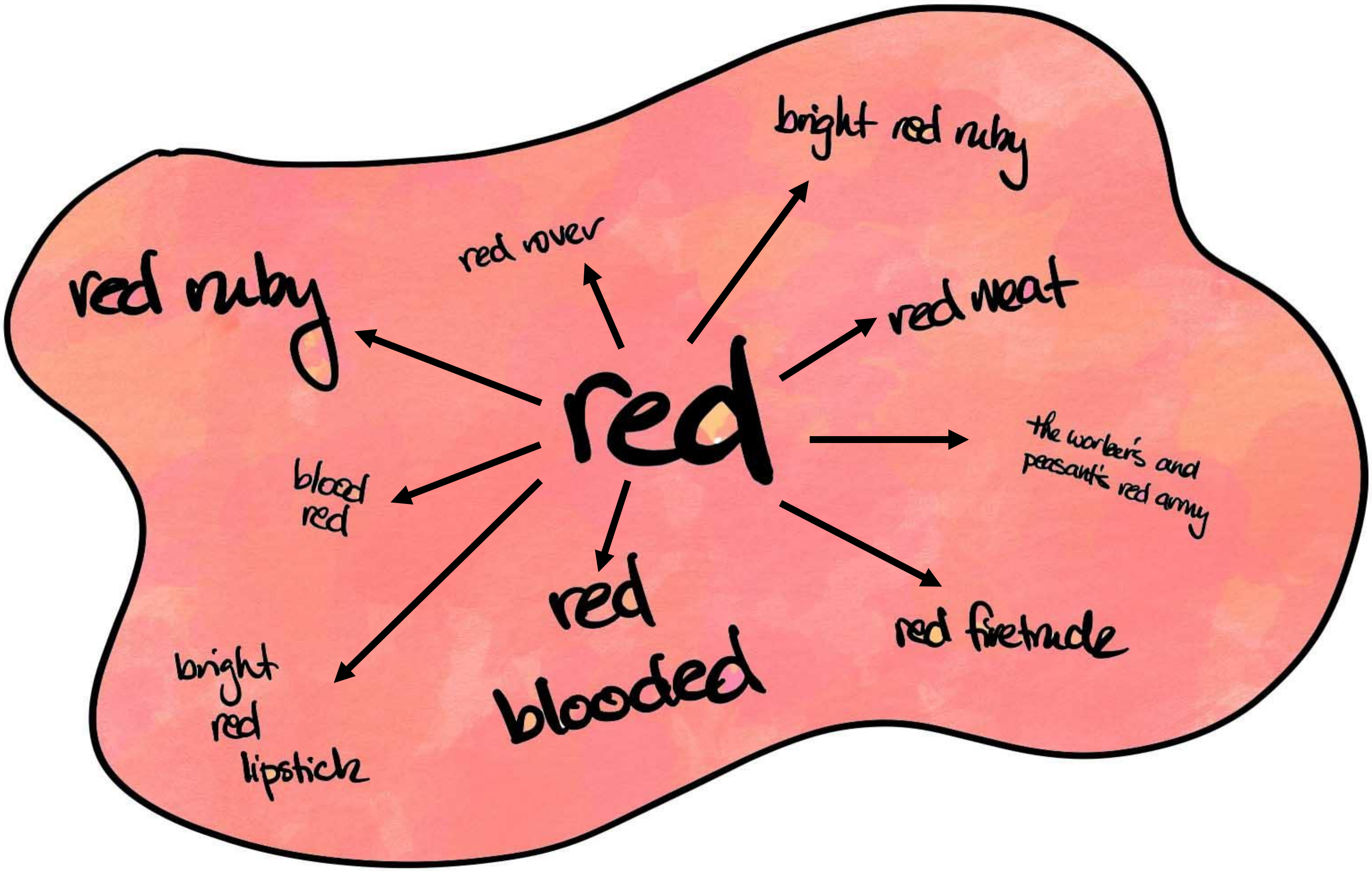
意味の文脈依存性とJ. R. Firth

ファースは「状況の文脈」という概念で意味の文脈依存的な性質に注目したことで知られ、連語的(collocational)意味に関する彼の研究は、分散意味論の分野で広く認められている。特に、彼は次の有名な引用で知られている。(wikipedia)

"You shall know a word by the company it keeps"

「我々は、ある語を、それが引きつれている仲間たちによって知ることになる。」

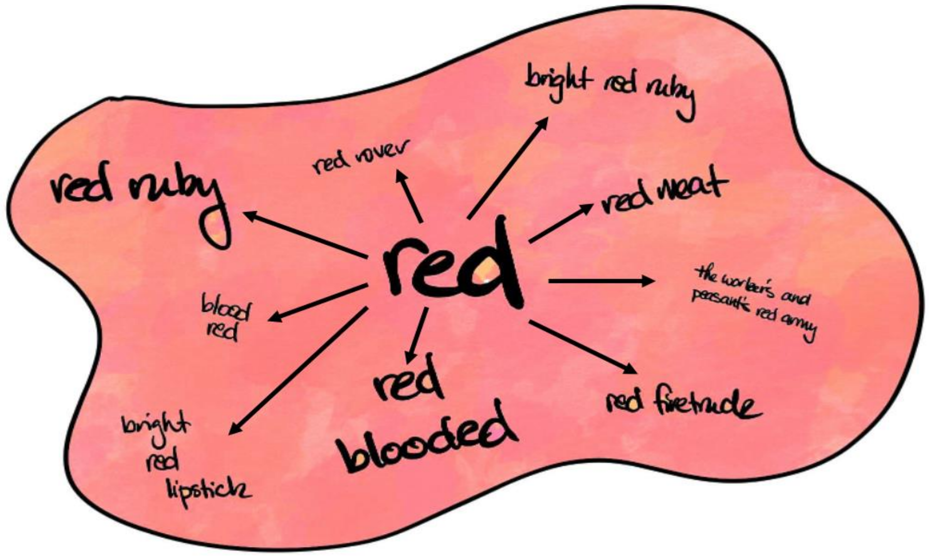
$x \leq y$を$x \rightarrow y$ で表した
“red” とその仲間たち（company）



“red” の仲間たちを表す集合

先の図から。redの仲間たちを表す集合は、次のような要素を含んでいることはわかります。{ red, red ruby, blood red, bright red lipstick, … … }　もっとも、上の図が、red の仲間たちをすべて列挙しているわけではないので、この要素の並びはまだまだ続くでしょう。

redの仲間たちを表す集合は次のように定義されます。



$$\{ y \in L \mid (\boldsymbol{red} \rightarrow y) \in L \}$$

red → y が、L の射なら、
そうしたLのオブジェクトy を
すべて集めたもの

L(S, −)という表記

カテゴリー L 内の S → T なる射を、L(S, T) と表します。

この表記を少し変形して、L(S, −) という表記を考えます。L(S, −) の後ろの ’−’ は、「ブランク」と読んで、不特定のオブジェクトにマッチします。

ただし、不特定のオブジェクトと言っても L(S, −) ですので、そのオブジェクトは、Lのオブジェクトであり、Sからそのオブジェクトへの射が存在するものの集合であることが表現されています。

L(S, −)の’ −’ は、「Sの仲間たち」の「全体」を表しています。

意味のカテゴリーは、copresheaf Set^L で表現される

これまで、意味のcategory M を構成するオブジェクトの集合を言語のcategory L のオブジェクトと同じオブジェクトから構成されると考えてきました。

それは、それでいいのですが、言語のcategory Lのオブジェクトxを一つとった時、それに対応する意味のcategory M のオブジェクトは、一つのオブジェクトからなるわけではなく、L(x, −)を満たす多数のオブジェクト’ −’ からなる集合です。

意味のcategory M: L → Set

意味のcategory Mは、言語のcategory Lのすべてのオブジェクトxに対して、L(x, −)を満たすオブジェクト’−’ を割り当てることで構成されます。

これは、category L からcategory Set へのfunctor とみなすことができます。

Lのオブジェクトxを、その意味を表すcategory Setのあるオブジェクトに割り当てる functor を L(x, −)で表すと、

L(x, −) : L → Set

と表すことができます。

functor category

重要なことは、functor は、それ自身category とみなすことができるということです。

一般に category C から category D へのfunctor から構成されるcategory を、functor category といい、D^Cで表します。（ [C, D] という表記を使うこともあります。）

- category D^C のオブジェクトは、CからDへのfunctor
- category D^C の射は、CからDへのfunctor間のnatural transformation

で、functor category を定義できます。

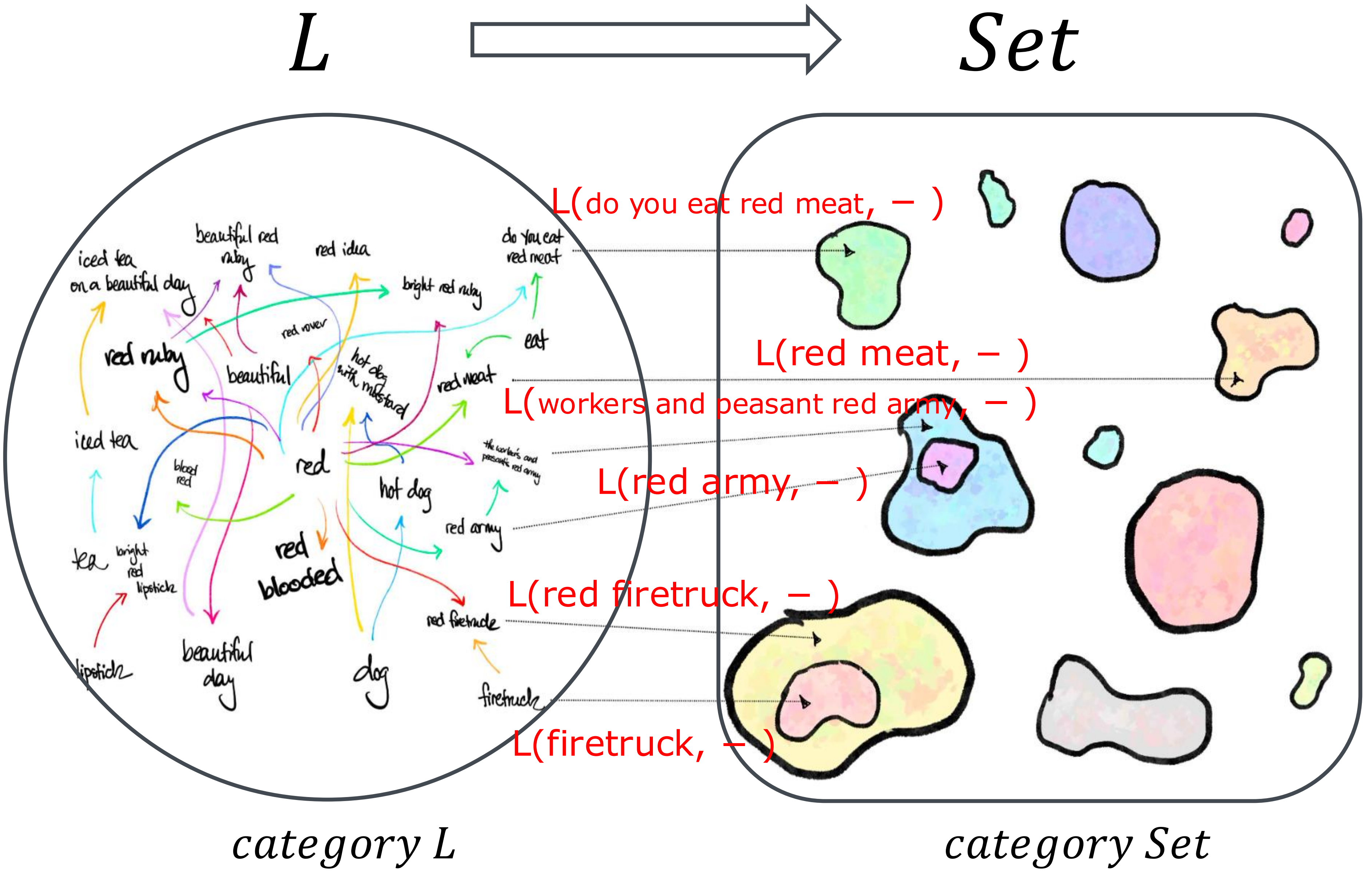
意味のcategory は、functor category Set^Lで表現できます。

copresheafとは何か

カテゴリー論では、カテゴリーCから集合のカテゴリーSetに値を持つfunctorで構成されるカテゴリーを、Set^Cで表し、C上の copresheaf と呼びます。

意味のカテゴリーは、言語のカテゴリー L から、集合のカテゴリー Setに値を持つ、functor で構成される L上のcopresheaf Set^L で表現されることになります。

Set^L のオブジェクト $L(x, -)$ の例



言語のカテゴリーと意味のカテゴリーの対応

意味のcategory M の構造は定義できたのですが、それで問題が片付いたわけではありません。

もともとの問題意識は、言語のcategory Lと意味のcategory M を関連づけようということでした。次のように。

$$\underset{\text{言語のcategory}}{\mathrm{L}} \longrightarrow \underset{\text{意味のcategory}}{\mathrm{M}}$$

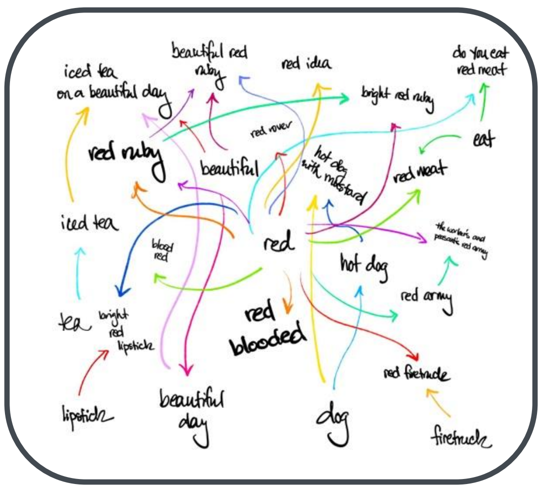
Mが定義できたので、これが今度はこうなります。

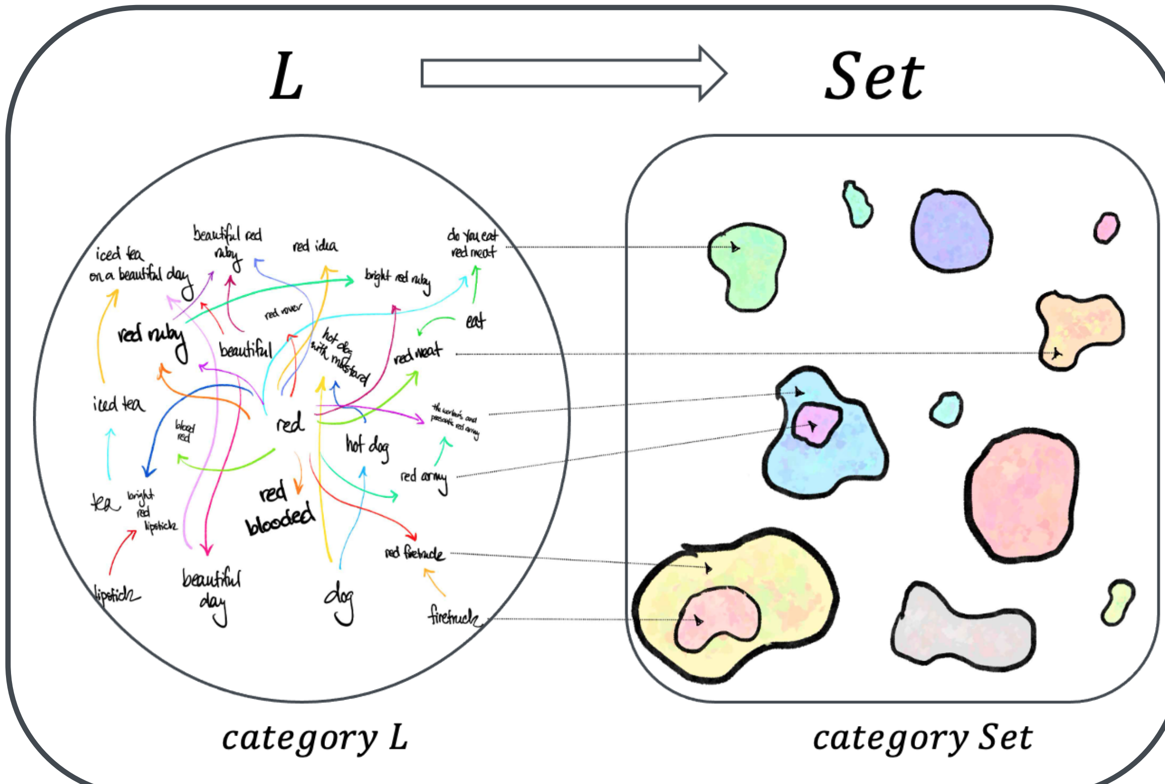
$$\underset{\text{言語のcategory}}{\mathrm{L}} \longrightarrow \underset{\text{意味のcategory}}{Set^{L}}$$

これを図で表してみましょう。

言語のcategoryと意味のcategory の関係

$L \Longrightarrow Set^L$





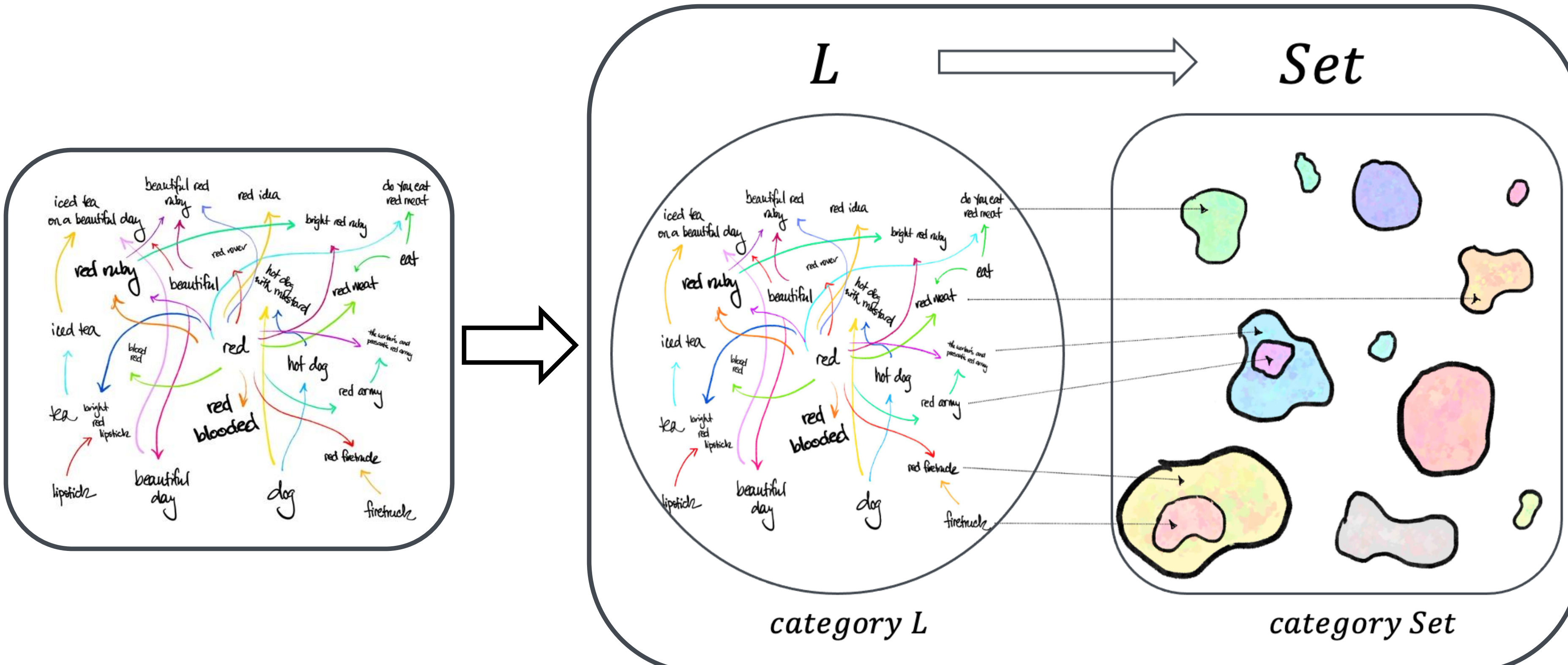
言語の
$category\ L$

$\Longrightarrow$

意味の
$category\ Set^L$

語 red の意味をサンプルに、この図式の働きを見てみましょう。

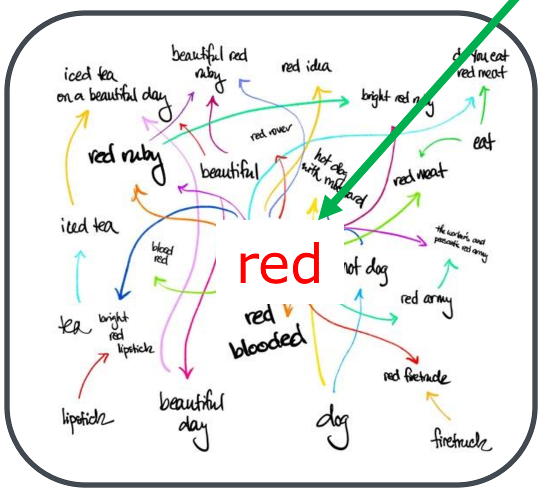
$$L \Longrightarrow Set^{L}$$

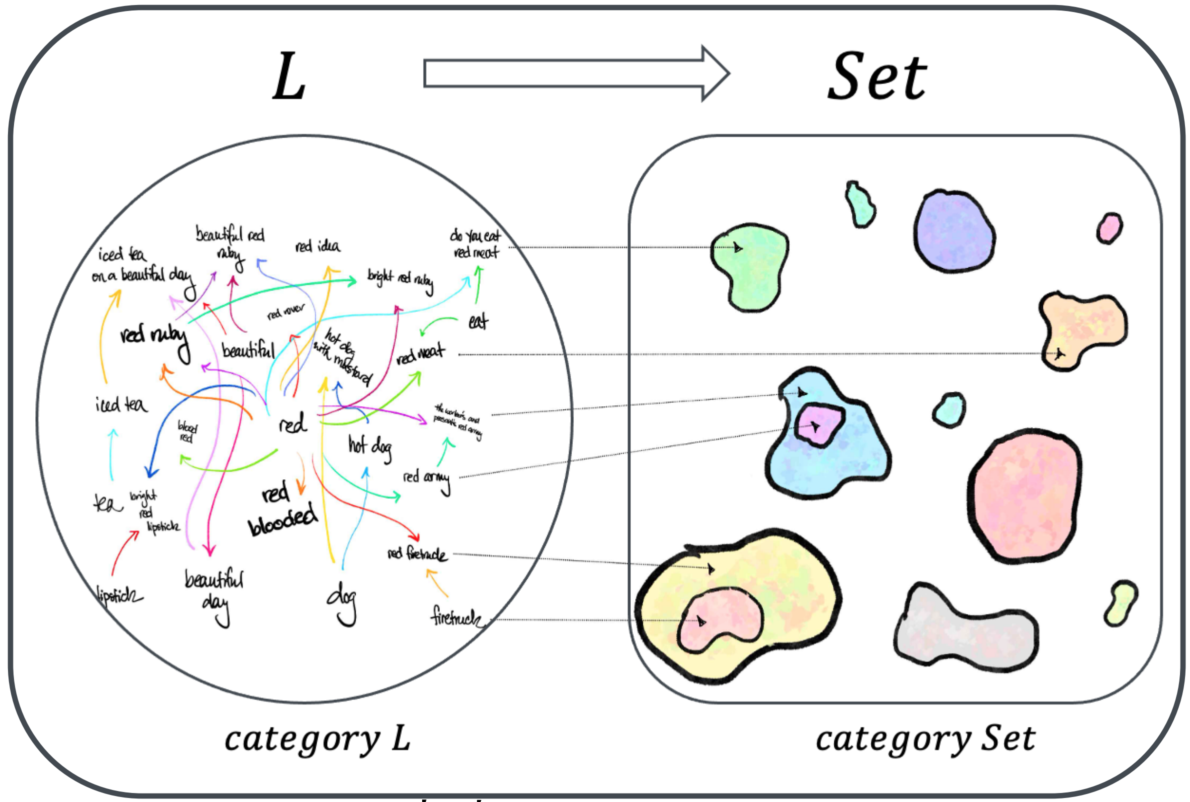


言語の $category\ L$ $\Longrightarrow$ 意味の $category\ Set^{L}$

Lのオブジェクト red を一つ選びます。

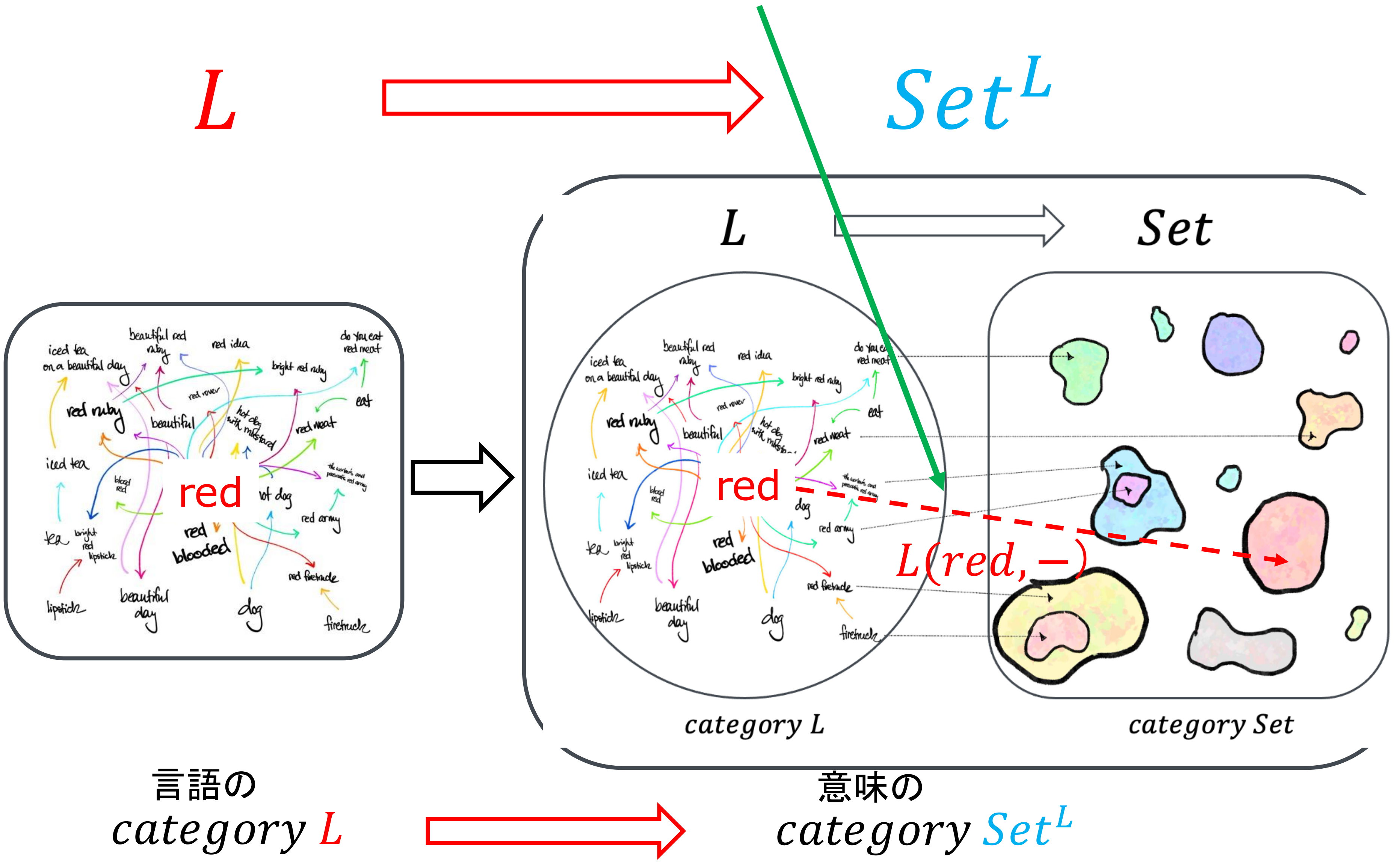
L $\Rightarrow$ Set^L



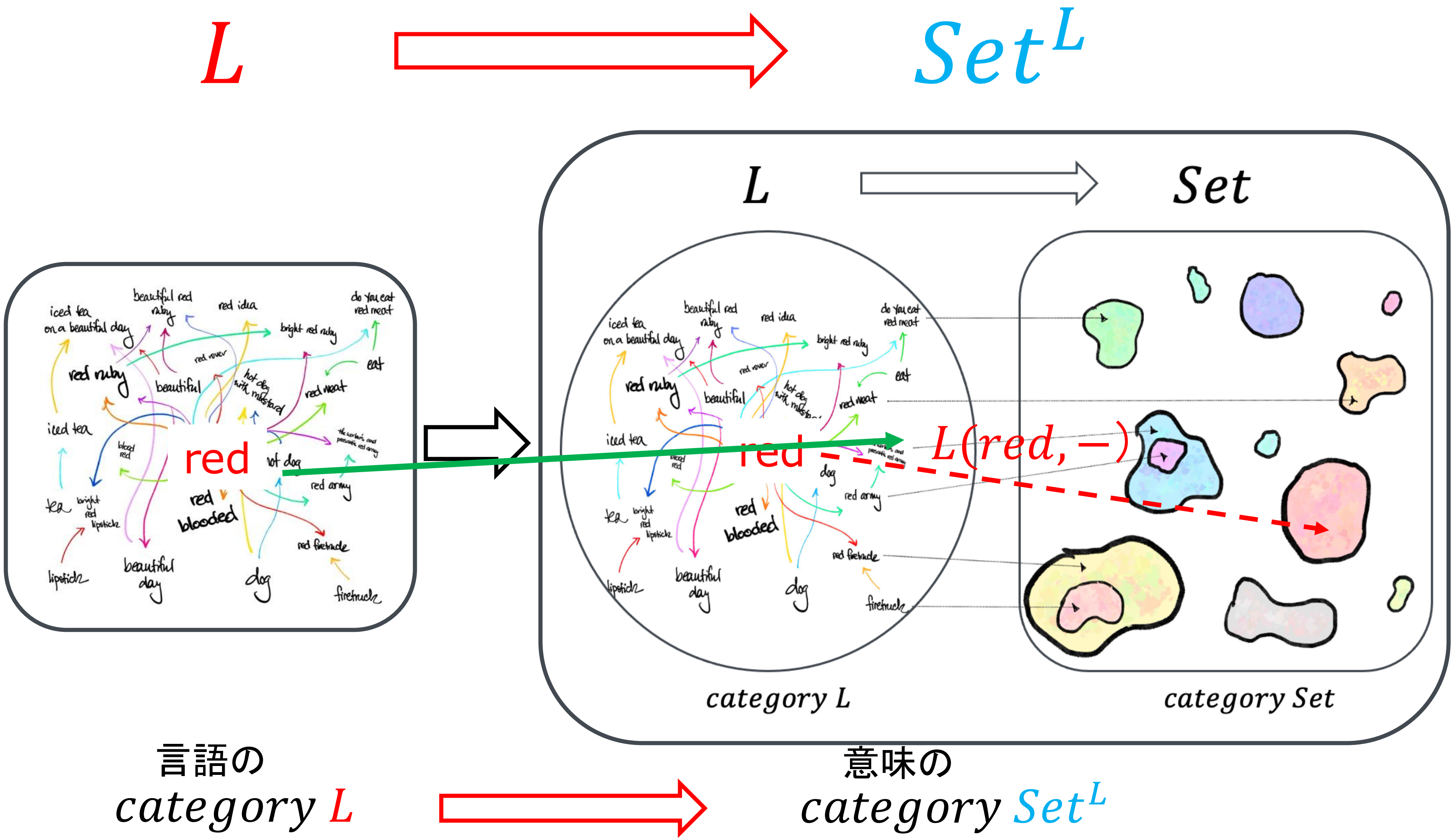


言語の category L $\Rightarrow$ 意味の category Set^L

Set^Lのオブジェクト $L(red, -)$ を考えます。

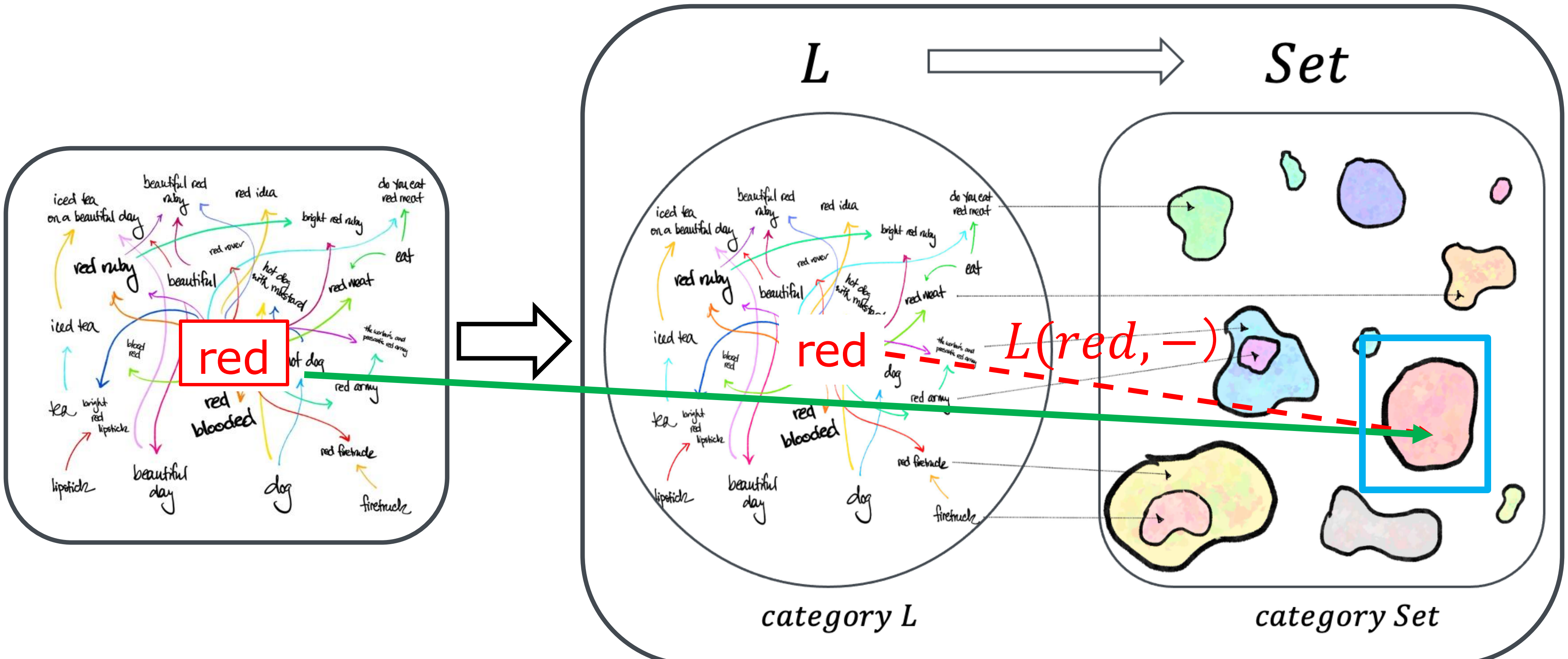


category L とcategory Set^Lの対応は、
red と$L(red,-)$ を対応づけるものです。



この対応は、語 red を
語 red の意味を表す集合に対応づけます。

$$L \Longrightarrow Set^L$$



言語の category L $\Longrightarrow$ 意味の category Set^L

言語のカテゴリーと意味のカテゴリーの対応は、米田の理論で与えられる

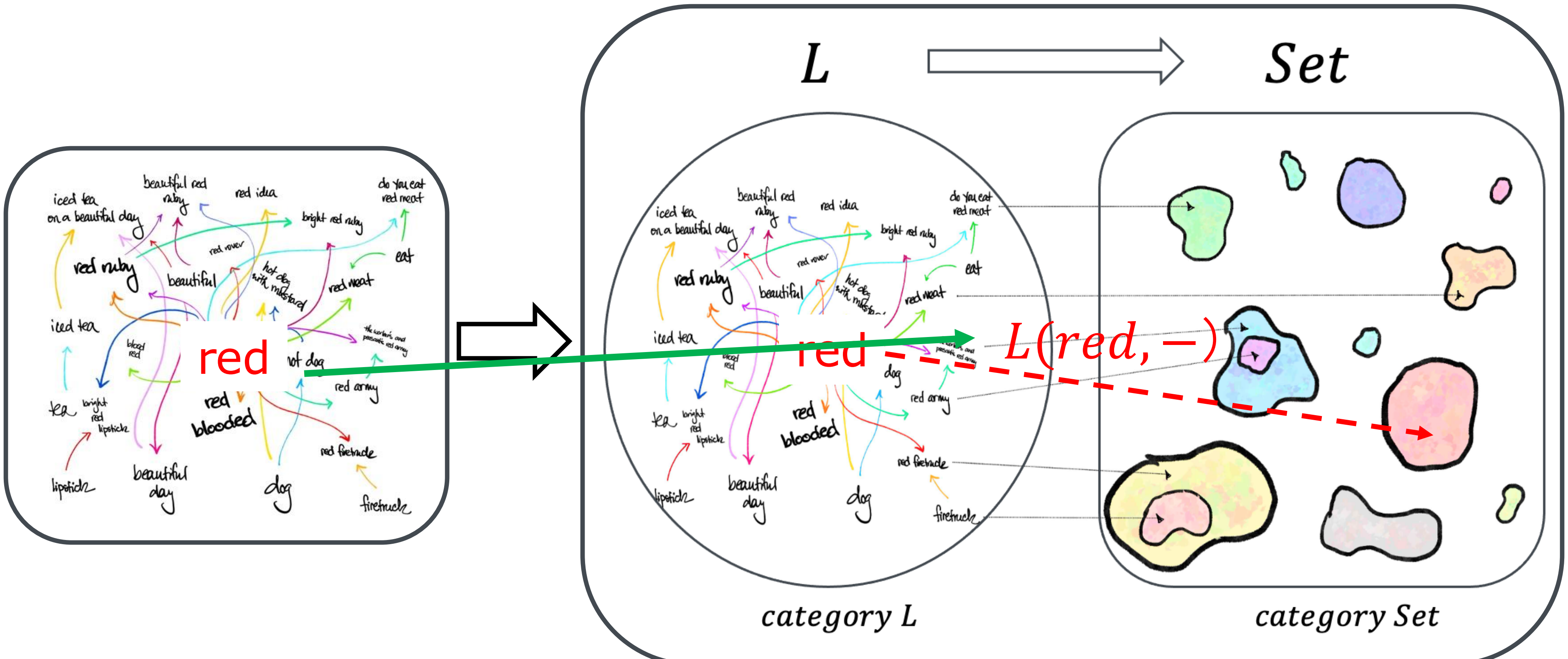
$$\begin{array}{ccc} L & \Longrightarrow & Set^{L} \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ \mathrm{red} & \longmapsto & \mathrm{L}(\mathrm{red},\ -) \end{array}$$

より一般的な、次の値の割り当てを、Yoneda embeddingと言います。

$$\begin{array}{ccc} L & \Longrightarrow & Set^{L} \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ \mathrm{x} & \longmapsto & \mathrm{L}(\mathrm{x},\ -) \end{array}$$

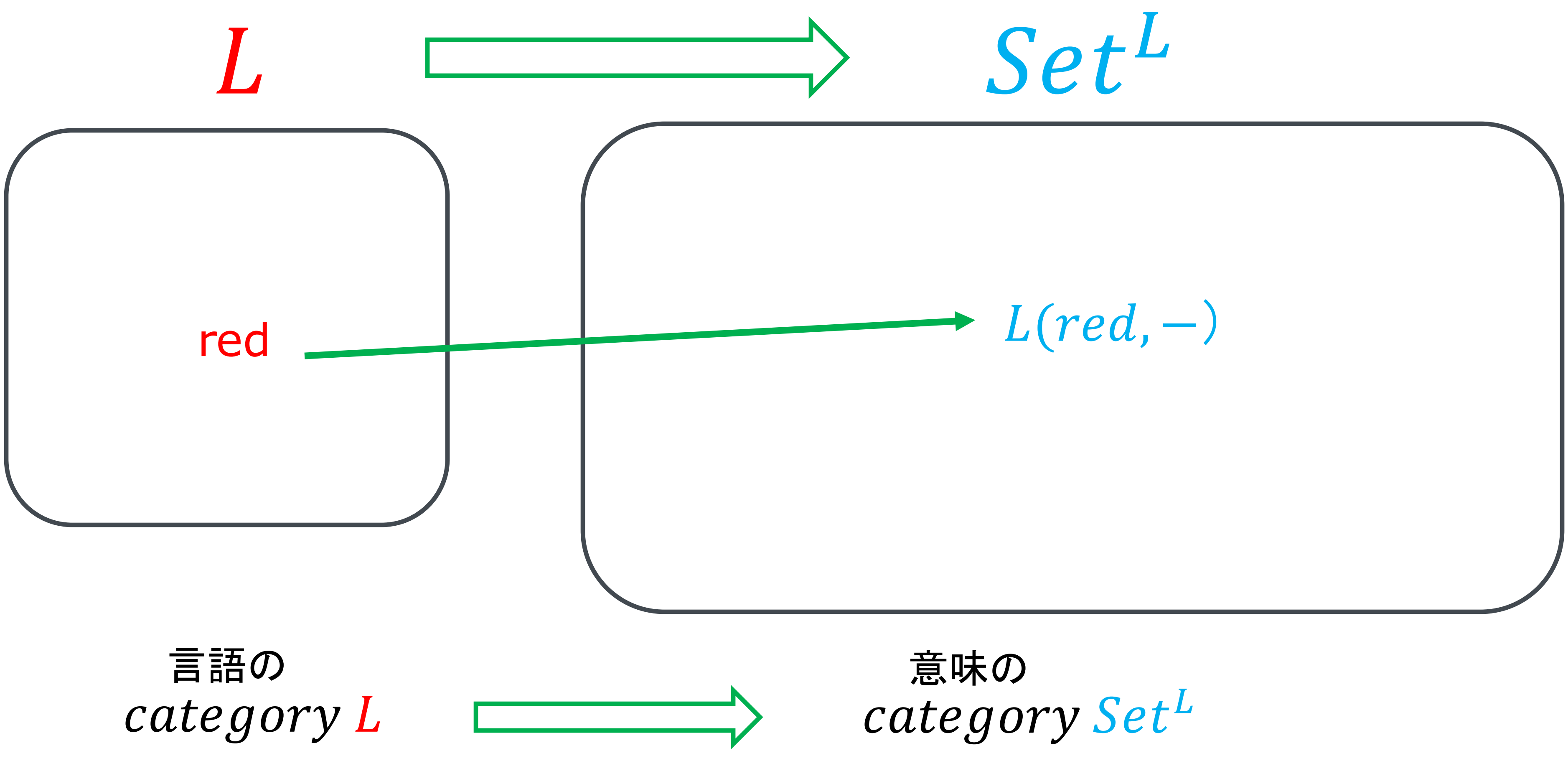
category L とcategory Set^L の対応は、
red と$L(red, -)$ を対応づけるものです。

$L \Longrightarrow Set^L$

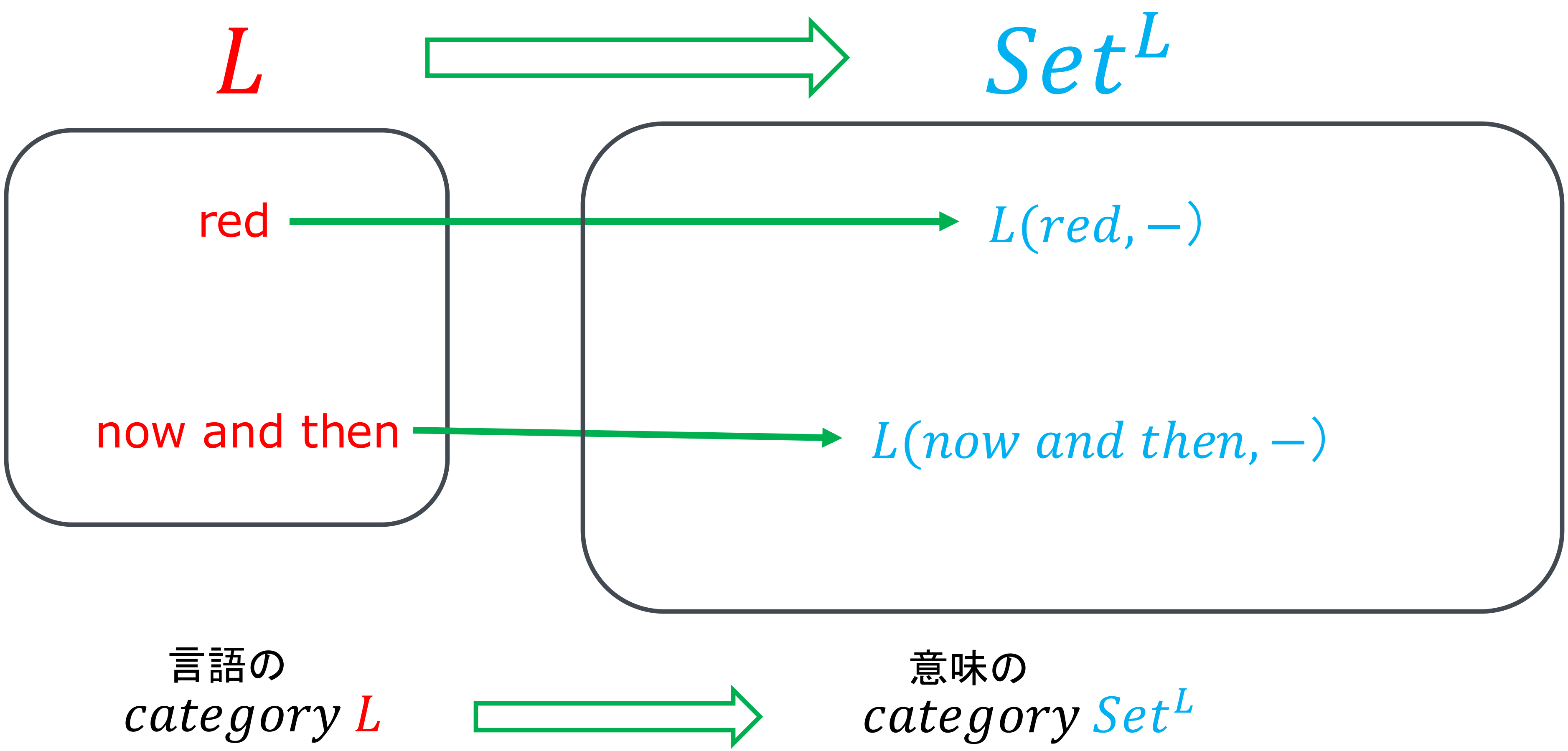


言語の
category L $\Longrightarrow$ 意味の
category Set^L

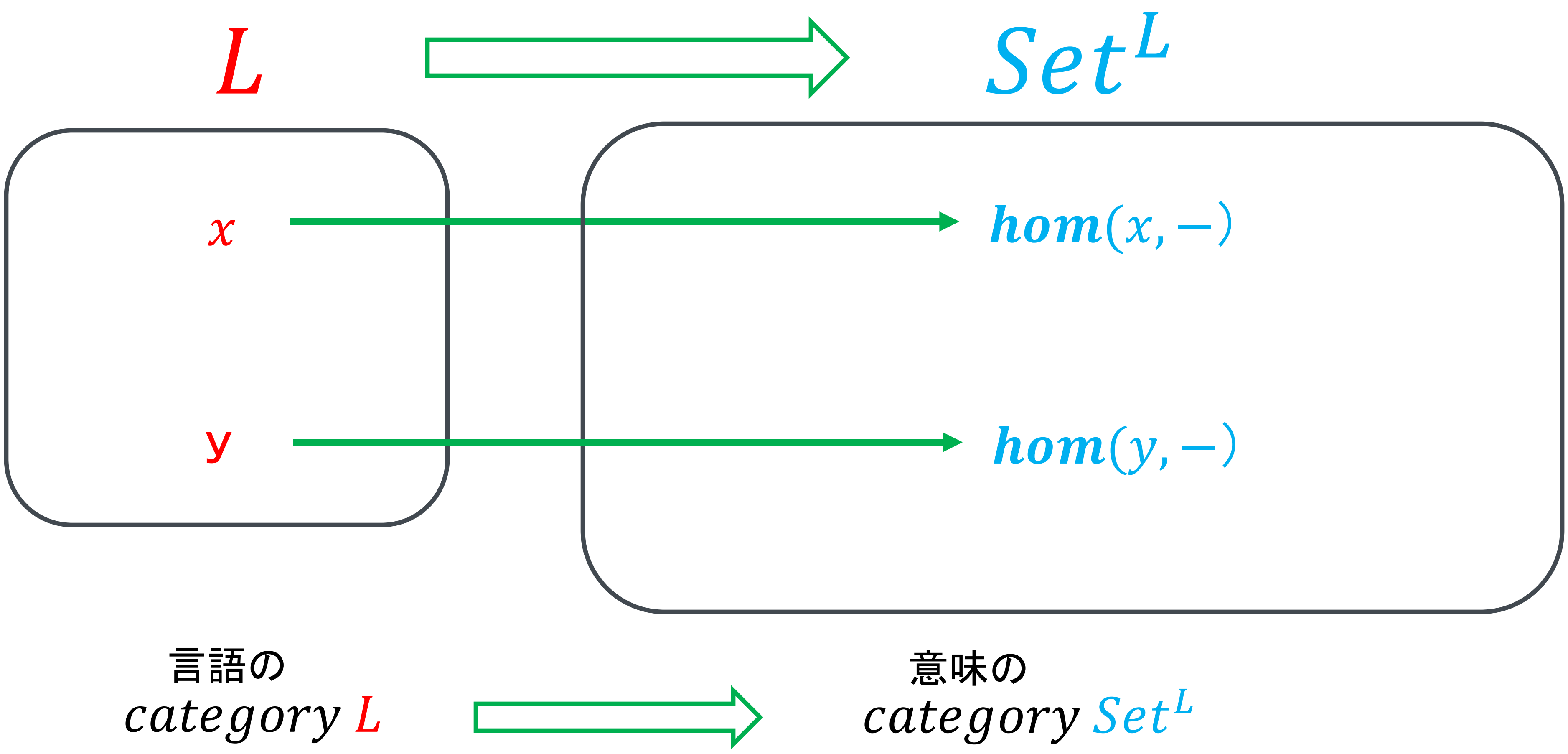
category L とcategory Set^Lの対応は、
red と$L(red,-)$ を対応づけるものです。
余分な部分を取り除くと次のような対応になります。



category L とcategory Set^Lの対応は、
red と$L(red, -)$ を対応づけるものです。
同様に $now\ and\ then$ と $L(now\ and\ then, -)$ は対応づけられます。



カテゴリー論的にいうと、category L とのオブジェクトxは、
category Set^Lのオブジェクトであるhom functor $\mathbf{hom}(\boldsymbol{x}, -)$に
対応づけられます。



集合に値を持つ表現可能なfunctorである
意味のcategory Set^Lを、
L上のcopresheafといいます。

$$Set^L$$

$$\boldsymbol{hom}(x,-)$$

$$\boldsymbol{hom}(y,-)$$

意味の
$category$ Set^L

集合に値を持つ表現可能なfunctorである意味のcategory Set^Lを、L上のcopresheafといいます。

意味の$category$ Set^L

copresheaf

Set^L

$\boldsymbol{hom}(x, -)$

$\boldsymbol{hom}(z, -)$　　$\boldsymbol{hom}(a, -)$

$\boldsymbol{hom}(y, -)$

すべてのLのオブジェクトxについて
$\boldsymbol{hom}(x, -)$なるfunctorの集まり

表現可能なfunctor

ここでは、これらの構成のアイデアのもとにあるYoneda lemmaについて、簡単に説明したいと思います。

category Cから集合のcategory Set へのfunctor

$$F : C \to \mathrm{Set}$$

が存在する時、このF を表現可能なfunctor と呼びます。

category Cの性質が最初はよく分からなくとも、性質のよくわかっているSet へのfunctor を考えると、Cの性質がSetの言葉で表現されてわかりやすくなると考えていいと思います。

functorでもSetへのfunctorは、特別なんだということです。

Hom functor

category Cの任意のオブジェクト A,B の間の射 A → B の集まりを、Hom(A, B)と表すことにします。

この時、Hom(A, −)、あるいは、Hom(−, B)で表される、次のような性質を持つCからSetへの表現可能なfunctorを考えます。

Hom(A, −) : C → Set

このfunctorをHom functor と言います。

ここでは、まず、 Hom functer Hom(A, −)について説明します。

Hom functor Hom(A, −)の性質

Hom functor Hom(A, −)は、次のような性質を持っています。

- Hom(A, −)は、Cのすべてのオブジェクトを、射の集合 Hom(A,X)にうつす。

- Hom(A, −)は、Cのすべての射 $f : X \to Y$ を、関数 $\mathrm{Hom}(A, f) : \mathrm{Hom}(A, X) \to \mathrm{Hom}(A, Y)$にうつす。
ただし、Hom(A, X)のすべてのgについて、$g \mapsto f \circ g$ とする。

$$\begin{tikzcd} X \arrow[d, "f"] \arrow[r, Rightarrow, "\mathrm{Hom}(A,\ -)"'] & \mathrm{Hom}(A,X) \arrow[d, "\mathrm{Hom}(A,\ f)"] \\ Y & \mathrm{Hom}(A,Y) \end{tikzcd}$$

Yoneda Lemma とは さわりだけ

ここまで、Hom functor Hom(A, −) について説明してきたのですが、それは、意味のcategory の構成に利用してきたのが、このHom Functor Hom(A, −) だったからです。

意味のcategory Hom(A, −) : L → Set

Yoneda lemmaのもともとの定式化は、Hom functor Hom(−, B)を使ったものでした。$\mathrm{Hom}(-,\mathrm{B}){:}\,C^{op} \to Set$

$F{:}\,C^{op} \to Set$ なるすべてのfunctor FとCのすべてのオブジェクトXについて、natural transformation $\mathrm{hom}(-,X) \to F$ は、集合$F(X)$の要素と一対一に対応する。

$$\mathrm{Nat}(\mathrm{hom}(-,X)\,,F) \cong F(X)$$

こうして構成された表現の上で、ORやANDといった論理的な操作の意味を解釈しうることを示す

- Lのオブジェクトである言語の表現x を、集合$L(x,-)$に割り当てるfunctorを考える。
- xに応じた異なる集合$L(x,-)$の割り当ては、異なるfunctorによって生み出される。
- 意味のカテゴリーは、言語のカテゴリー Lから集合のカテゴリーSetへのfunctorからなるカテゴリー $L \rightarrow Set$ である。
- このfunctorカテゴリーを Set^L （あるいは[L, Set]）と表し、L上のcopresheafと呼ぶ。

L(x, y)の性質

ただ一つの要素からなる集合を＊で表し、空集合を$\emptyset$で表すと、

$$L(x,y) = \begin{cases} * & x \to y \text{ なる射が存在する場合} \\ \emptyset & \text{それ以外の場合} \end{cases}$$

と表すことができる。

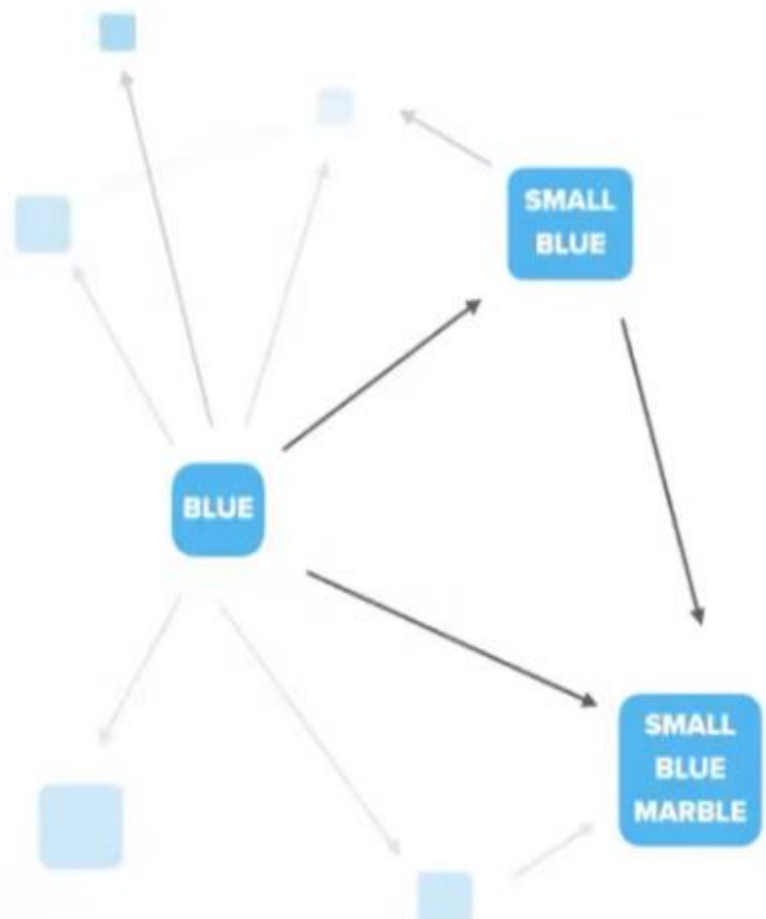
例えば、

$$L(red, red\ firetruck) \cong *$$
$$L(red, beautiful\ red\ ruby) \cong *$$
$$L(red, hot\ dog) = \emptyset$$
$$L(red, this\ is\ a\ pen) = \emptyset$$

意味のカテゴリーとしての$copresheaf$の もう一つの表現

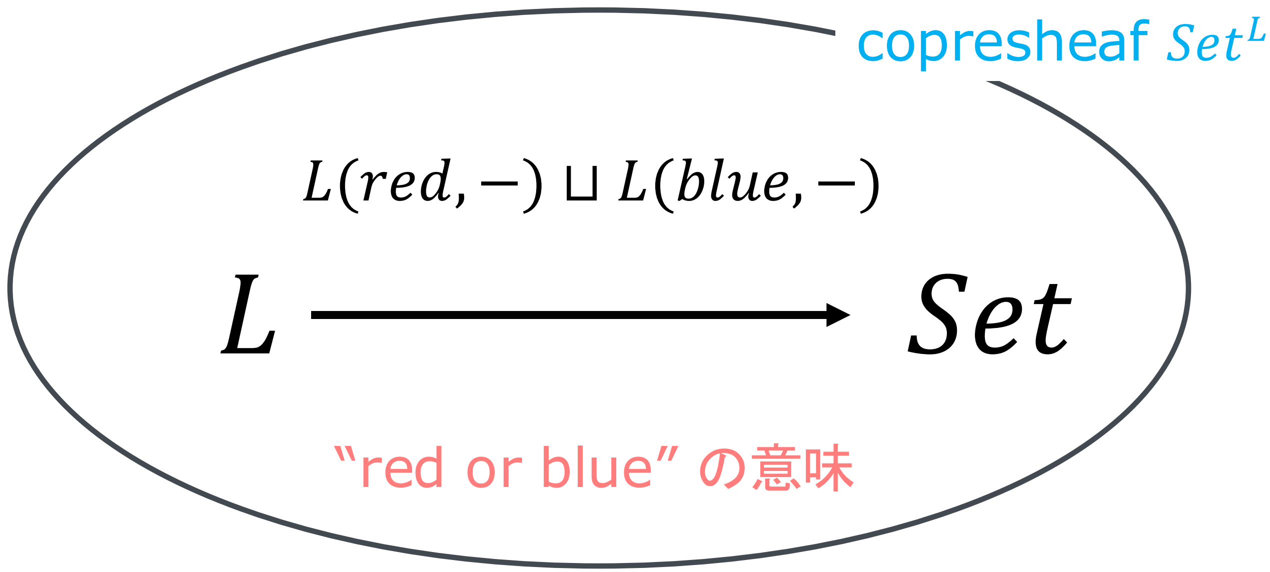
L の表現

ここで、* とマークされた
表現の全体の集合が、
blueの意味を与える。



“red or blue” の意味を copresheaf で考える

“red or blue” の意味は、意味のカテゴリー copresheaf Set^L のオブジェクトである functor $L(red, -) \sqcup L(blue, -)$ で表される。



言語のカテゴリー Lと意味のカテゴリーSet^Lを結ぶ
Yoneda embedding

$$\begin{array}{ccc} L & \longrightarrow & Set^L \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ y & \longmapsto & L(red, y) \sqcup L(blue, y) \end{array}$$

この集合は、yがred
を含む時 ＊
そうではない時は、
空集合 $\emptyset$ である。

この集合は、yがblue
を含む時 ＊
そうではない時は、
空集合 $\emptyset$ である。

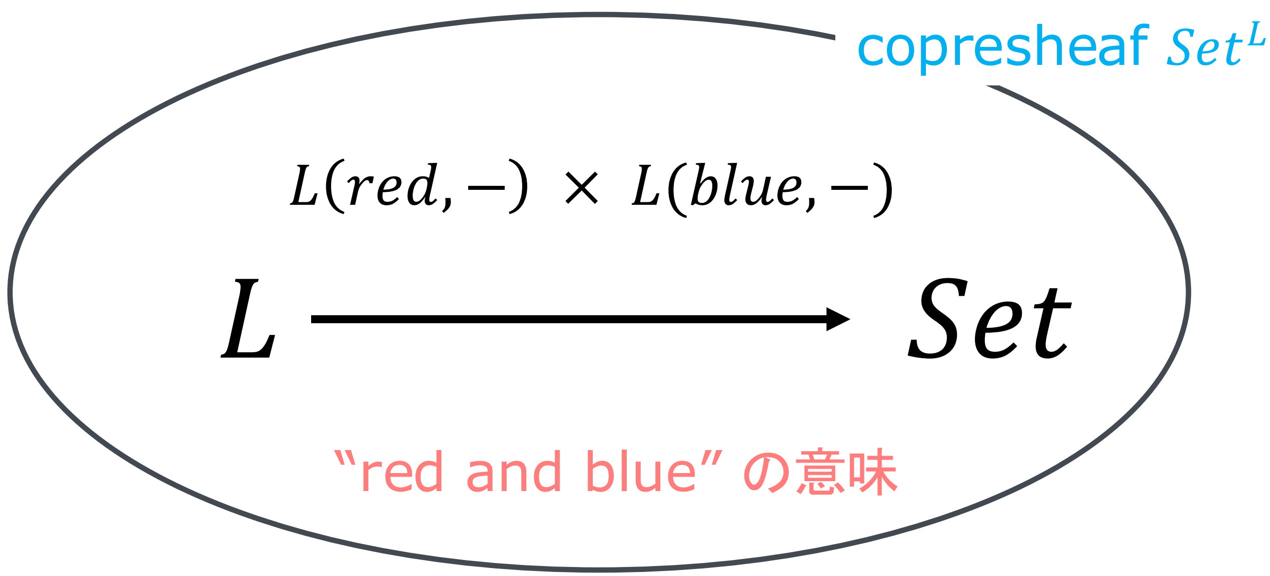
“red or blue” の意味を表す

$$L(red, -) \sqcup L(blue, -)$$

$$L(\mathrm{red}, -) \sqcup L(\mathrm{blue}, -) \overset{\text{SORT OF}}{=} \begin{bmatrix} * \\ * \\ * \\ \varnothing \\ *\,* \\ \varnothing \\ \vdots \end{bmatrix} \begin{array}{l} \text{deep red Bing cherries} \\ \text{small blue marble} \\ \text{beautiful blue ocean} \\ \text{did you put the kettle on} \\ \text{red and blue fireworks} \\ \text{Sencha green tea} \\ \vdots \end{array}$$

“red and blue” の意味を copresheaf で考える

“red and blue” の意味は、意味のカテゴリー copresheaf Set^L のオブジェクトである functor $L(red, -) \times L(blue, -)$ で表される。

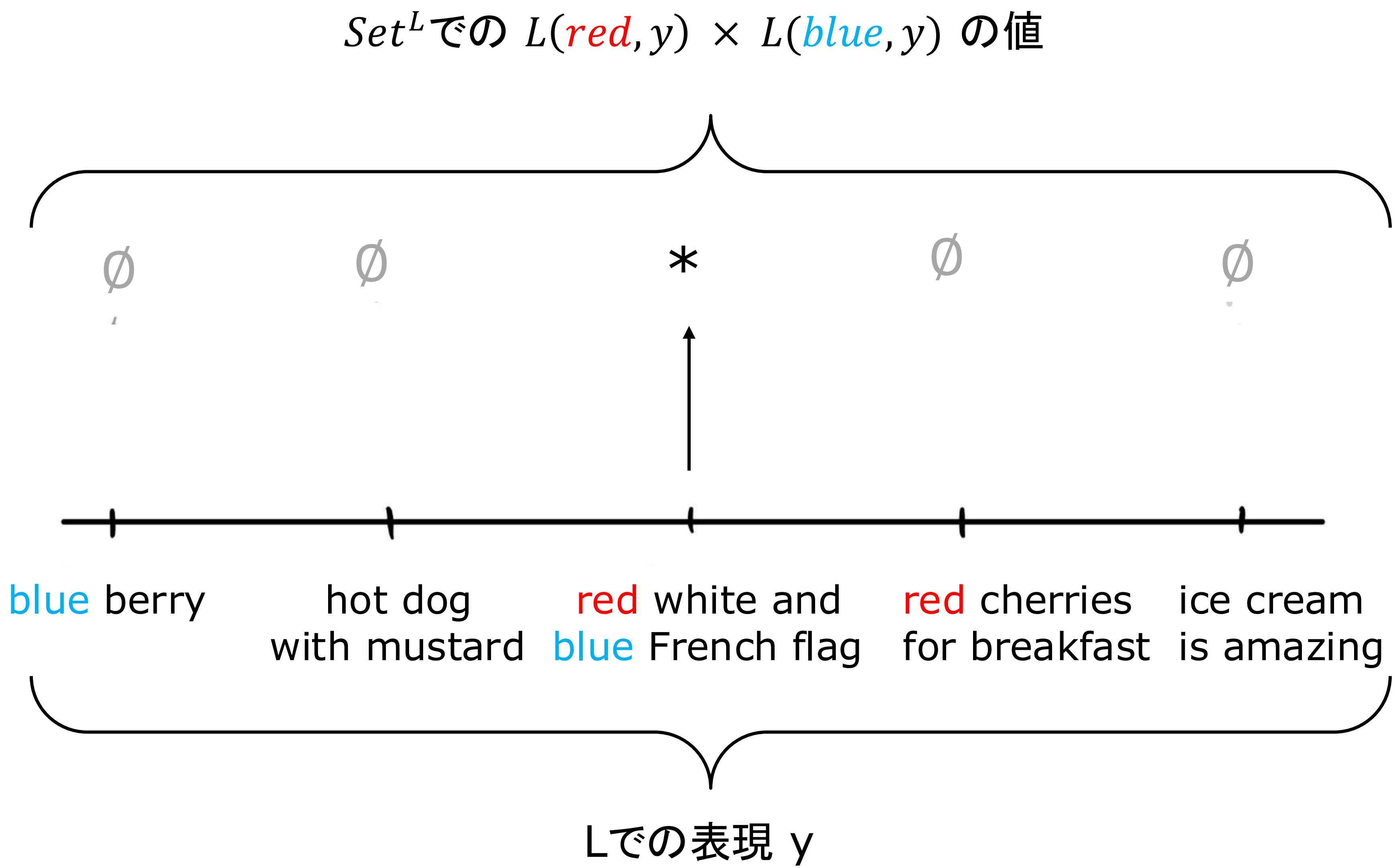


言語のカテゴリー Lと意味のカテゴリーSet^Lを結ぶ Yoneda embedding

この集合は、
yがredを含み、かつ
yがblueを含む時 ＊
そうではない時は、
空集合 ∅ である。

$(*,*) \cong *$

$$
\begin{array}{ccc}
L & \longrightarrow & Set^L \\
\Cup & & \Cup \\
y & \longmapsto & L(red, y) \times L(blue, y) = \begin{cases} * \\ \emptyset \end{cases}
\end{array}
$$


Set^Lでの $L(red, y) \times L(blue, y)$ の値
Ø
Ø
*
Ø
Ø
blue berry
hot dog with mustard
red white and blue French flag
red cherries for breakfast
ice cream is amazing
Lでの表現 y

“red and blue” の意味を表す

$$L(red, -)\ \times\ L(blue, -)$$

$$\mathrm{L}(\textbf{red}, -) \times \mathrm{L}(\textbf{blue}, -) \overset{\text{SORT OF}}{=} \begin{array}{cl} \left[\begin{matrix} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \\ * \\ \varnothing \\ \vdots \end{matrix}\right] & \begin{matrix} \text{deep red Bing cherries} \\ \text{small blue marble} \\ \text{beautiful blue ocean} \\ \text{did you put the kettle on} \\ \text{red and blue fireworks} \\ \text{Sencha green tea} \\ \vdots \end{matrix} \end{array}$$


copresheaf 意味論のenrich化

言語のカテゴリーLを単位区間[0,1]でenrich化して確率を導入する

これまでみてきた言語のcategory L では、二つの表現SとTがある時、SがTの部分文字列である時、S → T という射が存在します。

例えば、次のような射が category L には存在します。

red → red firetruck

red → red idea

S → T という射を、単なる部分文字列の関係としてではなく、表現Sが表現Tを「連想させる」という関係として考えると、普通の言語使用の局面では、red → red firetruck の方が、red → red idea よりたくさん出現するような気がします。

「連想」というのが曖昧だというなら、表現の「継続」あるいは表現の「連続」と考えて構いません。

こうした違いを、数値的に次のように表現することにします。

$$\mathrm{red} \xrightarrow{0.12} \mathrm{red\ firetruck}$$

$$\mathrm{red} \xrightarrow{0.003} \mathrm{red\ idea}$$

この例は仮のものですが、ここでのポイントは、射 red → red firetruck に割り当てられた 0.12という数字が、射 red → red ideaに割り当てられた 0.003という数字より大きいということです。

このことが、「普通の言語使用の局面では、red → red firetruck の方が、red → red idea よりたくさん出現するような気がする」ということを表現していると考えましょう。

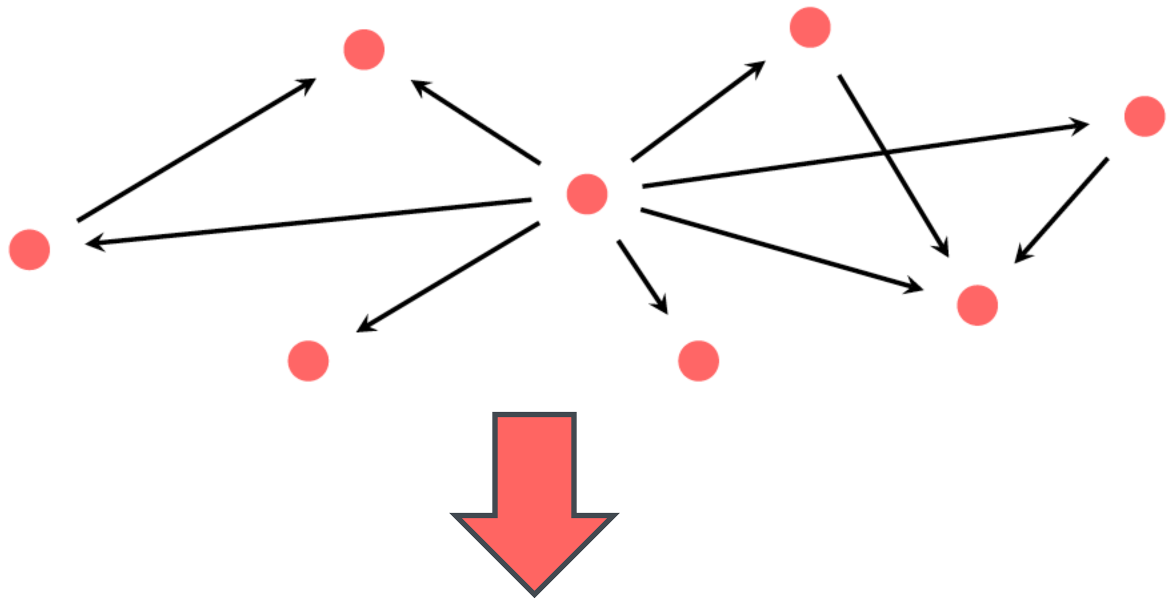
もう少しきちんと定義すれば、これらの数字は、

表現Sが現れた時、表現Sの「継続」として表現Tが現れる
条件付き確率π(T|S)

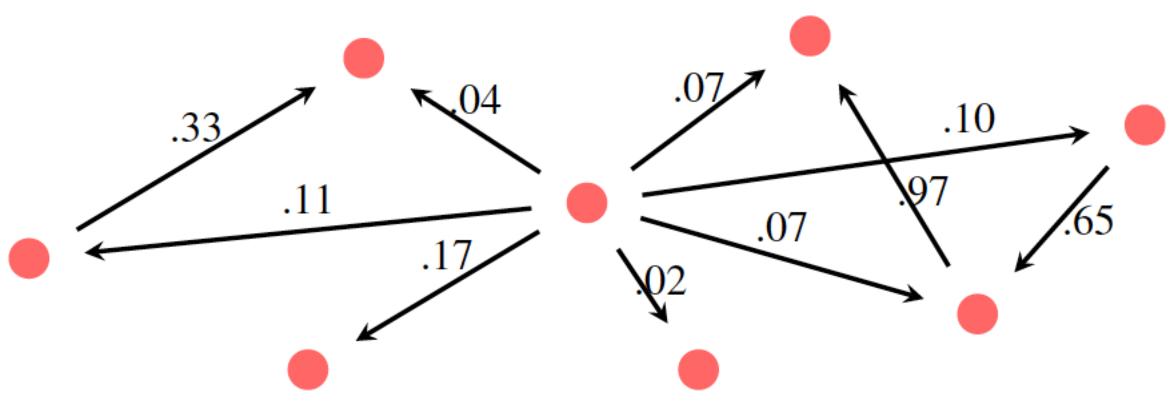
だと考えることができます。

確率を付与された言語のcategory L′のイメージ

これまでの
言語のcategory L



拡張された
言語のcategory L′



LLMと意味の理論モデルへの enriched カテゴリー論の利用

このように、あるcategoryをベースにしながら、その上に新しい特徴を与えて作られるcategoryをenriched category と言います。

今回のセミナーが取り上げようとしているTai-Danae らの論文、“An Enriched Category Theory Of Language : From Syntax to Semantics” は、まさに、このenriched categoryを用いて、大規模言語モデルと言語の数学的構造に迫ろうとしたものです。

Bradleyの構成とenrich化の課題

Bradleyの構成で興味深いのは、これまで見てきた確率なしのcopresheaf意味論の骨組みはそっくりそのまま受け継ぎながら、その構成の全体をenrich化するという道をとっていることです。

ただ、そのためには次の様な問題に答えていかなければなりません。

- Lの[0,1]上でのenrich化は、どの様に可能か？
- copresheafの[0,1]上でのenrich化は、どの様に可能か？
- Yoneda Embeddingの[0,1]上でのenrich化は、どの様に可能か？
- coproductの[0,1]上でのenrich化は、どの様に可能か？
- productの[0,1]上でのenrich化は、どの様に可能か？
- ……

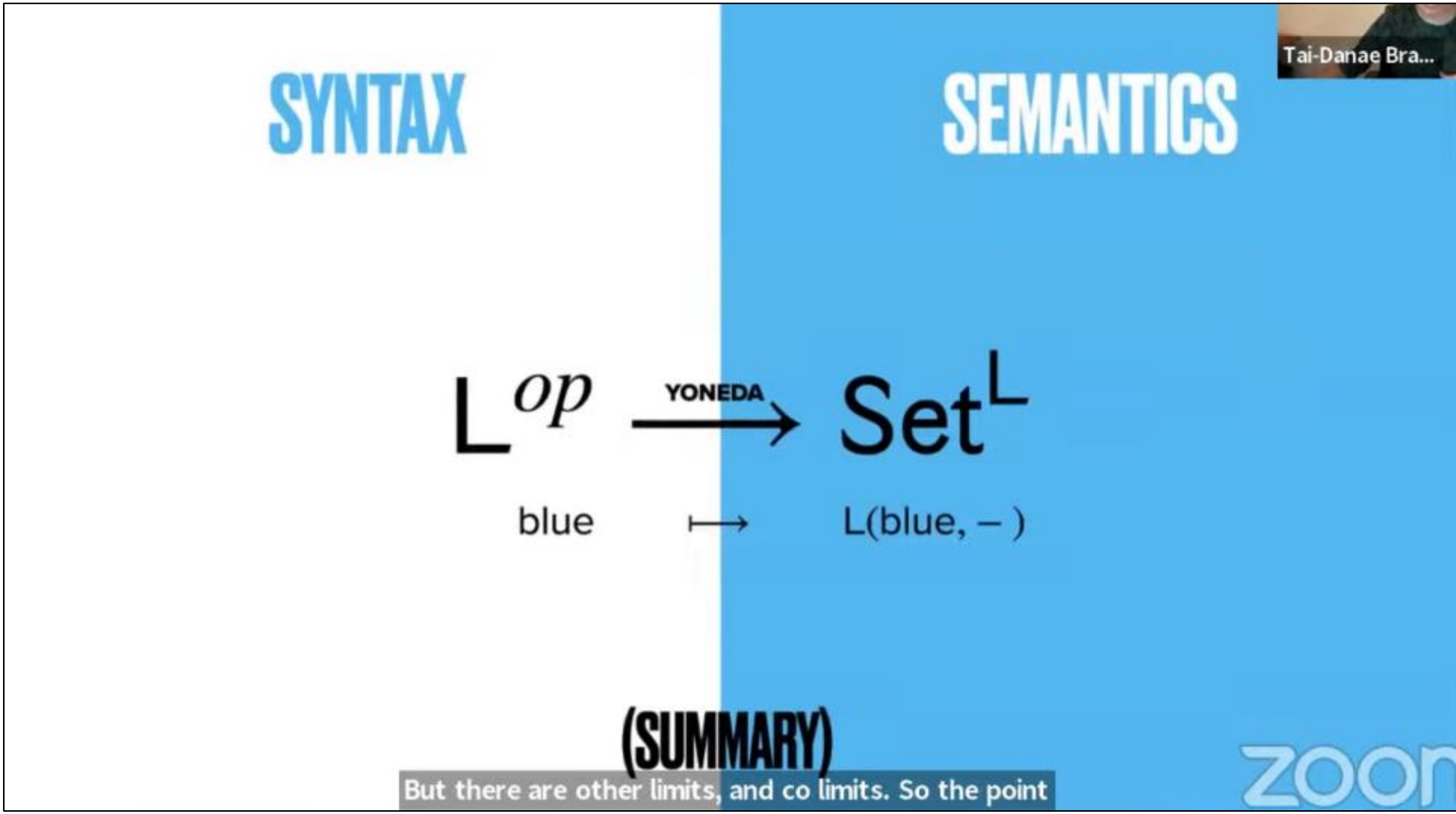
彼女は、こうした課題をやり遂げたのですが、今回のセッションでは、その数学的詳細は割愛します。

興味ある人は、「大規模言語モデルの数学的構造 II」ページの
https://www.marulabo.net/docs/llm-math2/
次のセクションを参照ください。

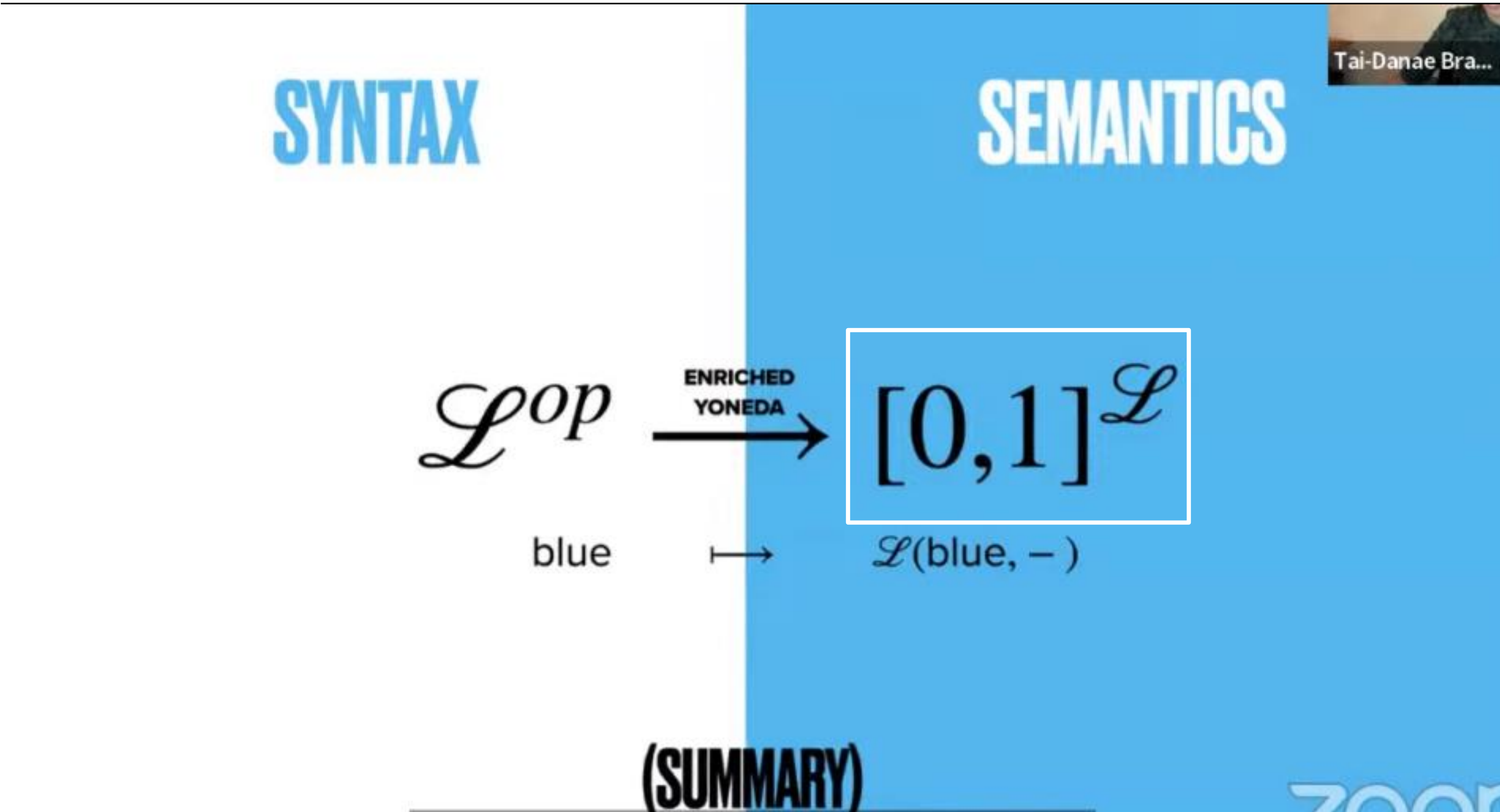
- 「enriched category論 入門」
- 「enriched category 論の言語理論への応用」

Webページから直接pdfファイルを読めるようにしました。

enrich化前のバージョン



enrich化後のバージョン



通常のカテゴリーでの射の集合

In (ordinary) **category theory**, each pair of objects has an associated **set**

$$\mathsf{C}(X, Y)$$

enrichedカテゴリーでは、射の集合が置き換わる

An Enriched Category

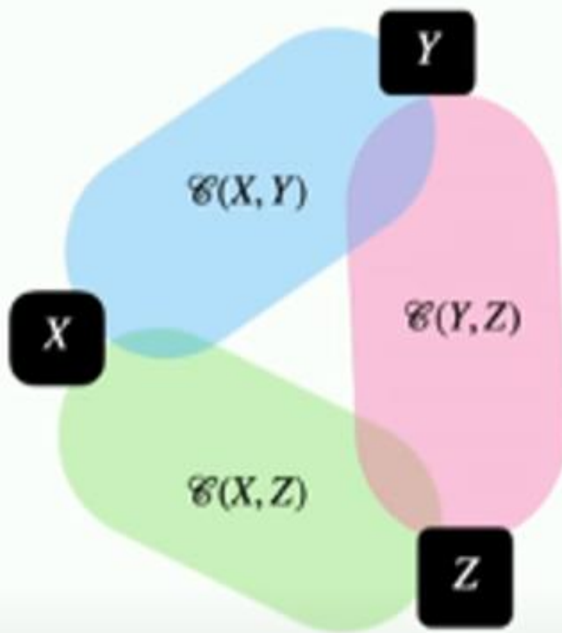
Very Loose Definition

Given a sufficiently nice category $\mathscr{V}$, a $\mathscr{V}$−**enriched category** $\mathscr{C}$ has

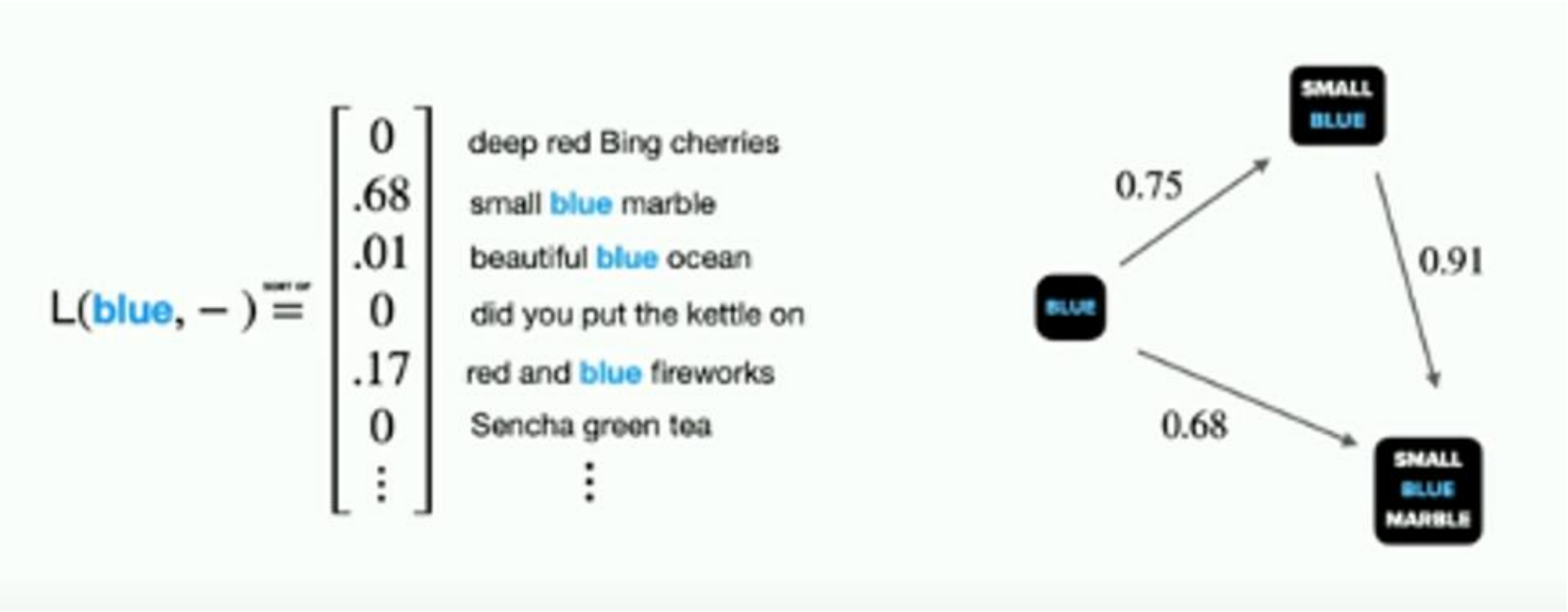
- **objects** $X, Y, \ldots$
- an **object** $\mathscr{C}(X, Y)$ in $\mathscr{V}$
- a "**composition rule**"

$$\mathscr{C}(X,Y) \otimes \mathscr{C}(Y,Z) \to \mathscr{C}(X,Z)$$

that satisfy reasonable axioms.



言語のカテゴリーLのenrich化

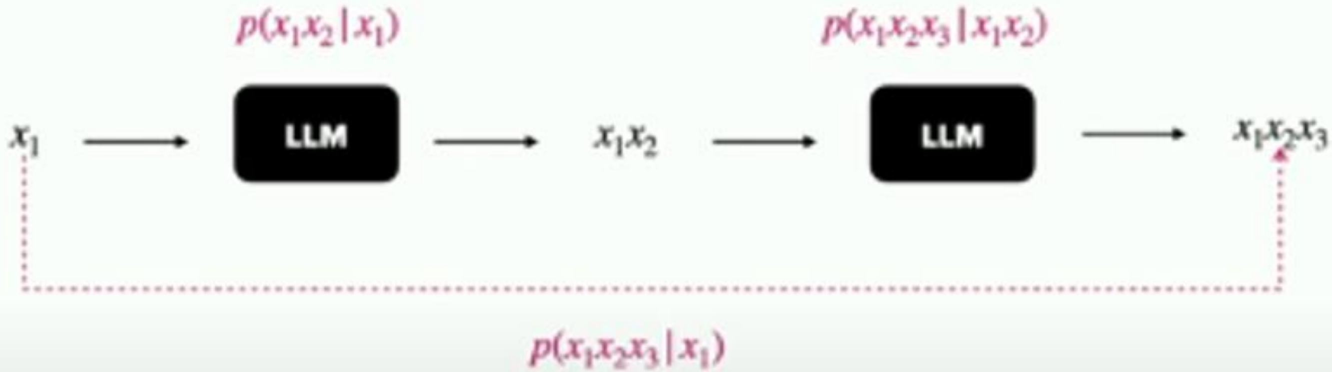


この確率は、LLMが計算しているものだ

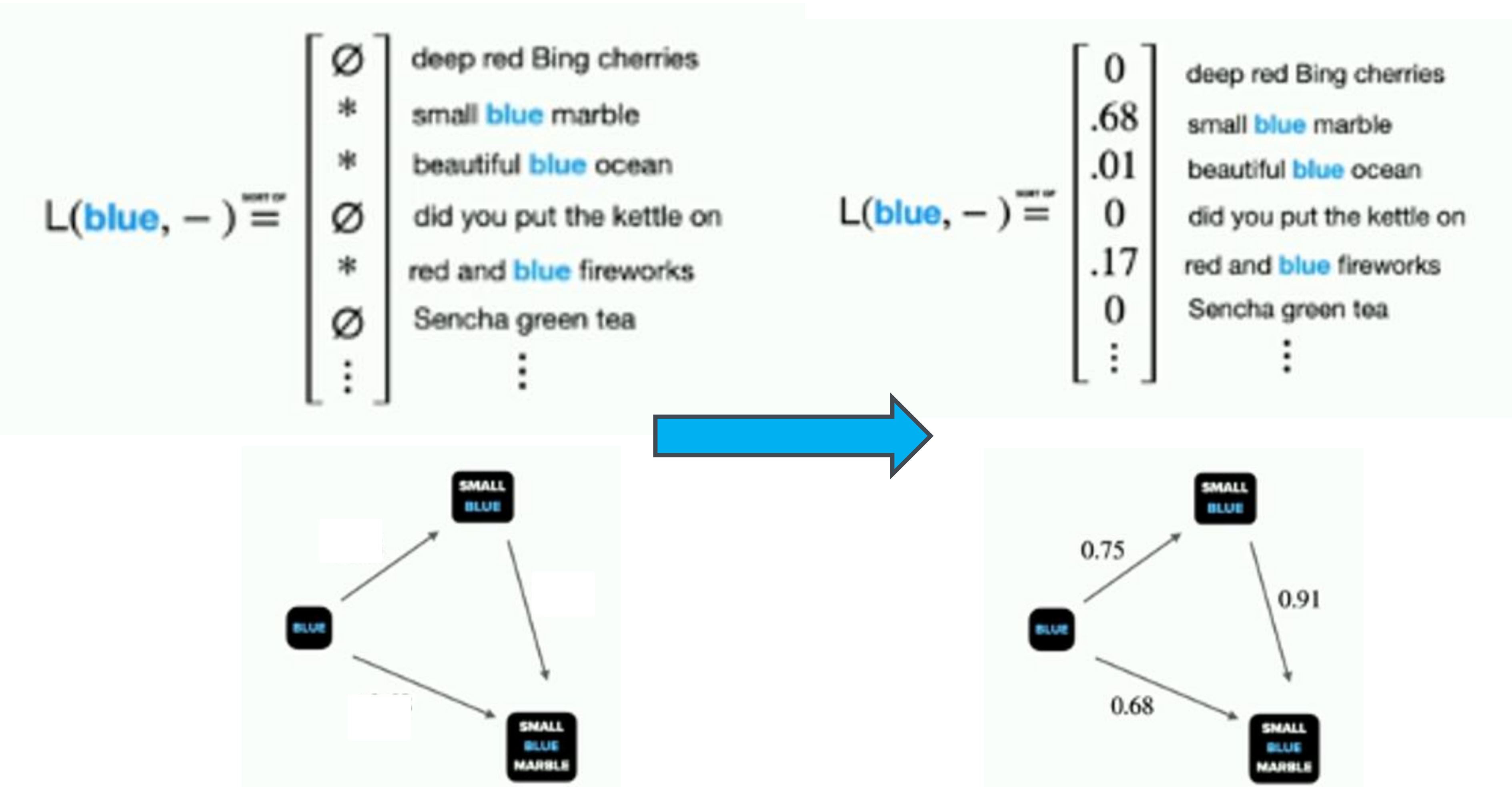
Yes, suppose we have an LLM.

For any prompt x, the LLM gives a probability distribution $p(-\,|\,x)$ on the set of tokens.

These probabilities multiply in the following sense:



enrich化前とenrich後



copresheaf-意味論の射程

5. Operate on Enriched Representations

The enriched functor category $[0,1]^{\mathscr{L}}$ has rich structure, including the *enriched* versions of limits, colimits, and Cartesian closure.

So, we can again make sense of logical operations like conjunction, disjunction, and implication.

$$\mathscr{L}(\textbf{red}, -) \sqcup \mathscr{L}(\textbf{blue}, -) \stackrel{\text{def}}{=} \begin{bmatrix} .10 \\ .22 \\ \boxed{.73} \\ 0 \\ .59 \\ 0 \\ \vdots \end{bmatrix}$$

$\max\{\mathscr{L}(\textbf{red}, y), \mathscr{L}(\textbf{blue}, y)\}$

enrich化されたcopresheafでのcoproduct

これは、enrich化されたcopresheafでは、そのcoproductは
$\mathcal{L}(red,-)\sqcup\mathcal{L}(red,-)=\max(\mathcal{L}(red,-),\mathcal{L}(red,-))$
で計算されることを表しています。

maxが出てくるのが興味深いです。Tropical代数に似ています。
Vlassopoulosらの研究に繋がります。

"Directed Metric Structures arising in Large Language Model"

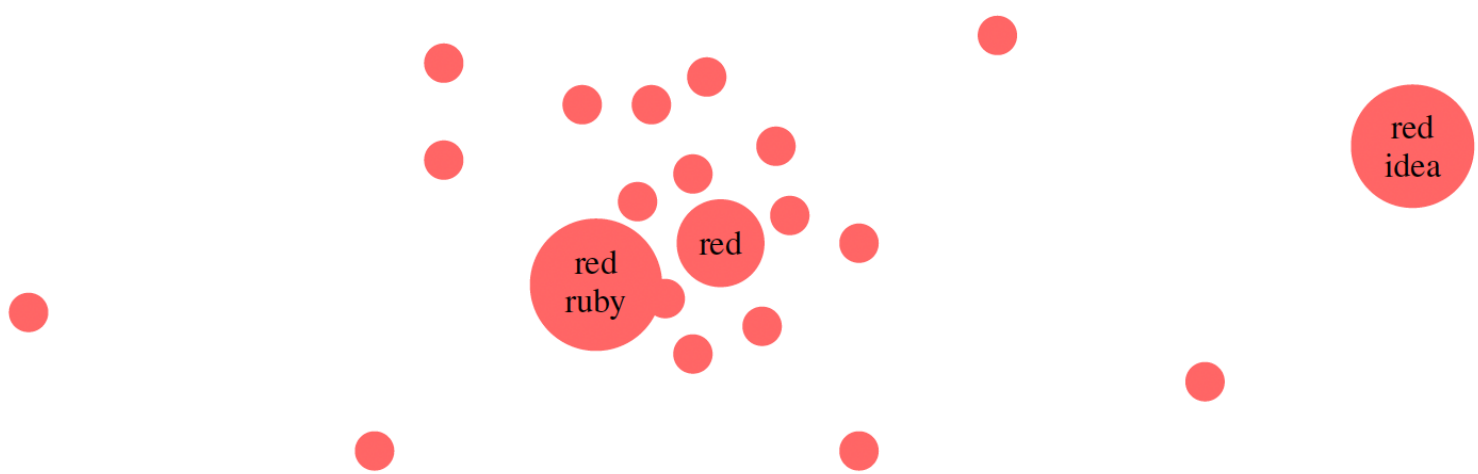
https://arxiv.org/pdf/2405.12264

copresheaf-意味論のvariant
$[0,\infty]$-$copresheaf$

この論文は、単位区間[0,1]でenrich化するcopresheaf-意味論のvariantとして、$[0,\infty]$区間でenrich化するcopresheaf-意味論も紹介されています。

それは言語を一般的な計量空間と考えるものです。表現 redは、$[0,\infty]$-$copresheaf$ $d_M(red,-)$ とみなされます。

redと継続しそうもない表現はredから離れた場所に、継続しそうな表現は、redの近くに置かれます。



この方向は、まさにBradleyが 2025年の論文で展開したことです。

“The Magnitude of Categories of Texts Enriched by Language Models”

https://arxiv.org/pdf/2501.06662