大規模言語モデルの数学的構造 I

言語へのカテゴリー論的アプローチ入門

はじめに

大規模言語モデルの働きがよくわからないという人は少なくないと思います。僕自身もそうです。大規模言語モデルの振る舞いとその背後にある理由をもっとよく知りたいと考えている人は、きっと沢山いらっしゃると思います。

二回連続のシリーズを予定しているセミナー「大規模言語モデルの数学的構造」は、こうした疑問に、大規模言語モデルにカテゴリー論に基づくモデルを与えることで応えようとしたTai-Danae Bradley らの興味深い論文を紹介することを主要な目的にしています。

今回のセミナー

大規模言語モデルの数学的構造 I
-- 言語へのカテゴリー論的アプローチ入門

は、この連続セミナーの第一回目です。

連続セミナーの二回目は、

大規模言語モデルの数学的構造 II
-- enriched categoryによる言語モデル

というタイトルで、来年の一月末開催予定です。

ご期待ください。

日頃あまり見慣れない用語や概念が出てくるので、最初はわかりにくいところが多いかもしれません。

ただ、それは当然かもしれません。

なぜなら、ここでの議論のベースになっているのは、50年以上前に数学の巨人であるグロタンディックやローベールたちが純粋数学の世界で作り上げた、functorial semantics, presheaf, topos という道具たちだからです。

しかし、大規模言語モデルの不思議さ以上に不思議なことが現在起きています。

それは、かつての巨人たちが作り上げた抽象的な数学理論を、私たちが具体的な例を通じて容易に理解する道が開かれつつあるということです。

その舞台が、カテゴリー論と比較してはるかに多くの人が関心を持っている大規模言語モデルだと、僕は考えています。

Yoneda lemma の名前ぐらいは知っていたのですが、その「実際的な応用」をIT技術者に語ることがあるなんて、まったく考えたことはありませんでした。

今回のセミナーの準備を通じて、僕は、技術者には身近な大規模言語モデルを通じて、技術者が抽象的に見えるカテゴリー論の基礎を学習することが可能なのだと思い始めています。

現実の大きな変化に対応して、きっと何か大きな理論的なパラダイムシフトが進行しているのだと思います。きっと、皆が、当たり前のように、巨人の肩の上に乗ればいいのです。

大規模言語モデルや人工知能を語るにはカテゴリー論が必要で、逆に、カテゴリー論を語るには言語理論や人工知能論といった背景が必要なのだという認識が、きっと一般的なものになるだろうと考えています。

Agenda

- はじめに

- Part 1 構成的分散意味論の展開
  - 語の意味をどう捉えるか？
  - 言語の構成性をどう捉えるか？
  - DisCoCatの登場
  - Quantum NLP

- Part 2 大規模言語モデルの特徴
  - Tai-Danaeの問題提起
  - あらためて、セミナーの目的について
  - Tai-Danaeの道具箱

Agenda

- Part 3 言語をカテゴリーとして捉える
  - preorderとしての言語
  - categoryとしての言語
  - functor:意味を表現する方法

- Part 4 意味をカテゴリーとして捉える
  - functor category と意味
  - Yoneda embeddingと意味の表現
  - 言語のcategoryに確率を導入する -- enriched category

Part 1

構成的分散意味論の展開

大規模言語モデルの数学的構造 I
言語へのカテゴリー論的アプローチ入門

Part 1　構成的分散意味論の展開

Agenda

- 語の意味をどう捉えるか？
- 言語の構成性をどう捉えるか？
- DisCoCatの登場
- Quantum NLP

語の意味をどう捉えるか
分散意味論の系譜

大規模言語モデルの数学的構造 I

意味の使用説:Wittgenstein

意味の使用説とは、意味はその使用から説明されるべきだ、という考え方である。Wittgensteinに始まる。

“meaning of a word is its use in a language”

「ある語の意味は、ある言語におけるその使用である」

TuringのWittgenstein批判

チューリングは、言葉の意味を知る事は、その用法を知る事だといったヴィトゲンシュタインらの見解に痛烈な皮肉をあびせている。

すなわち、「機械」や「考える」という言葉の使い方をいくら調べた所で「機械は考える事ができるか」という問の意味も答えも明らかになるわけではない。それとも、「ギャラップの世論調査の様な統計的研究」が必要という事になるのだろうかと.

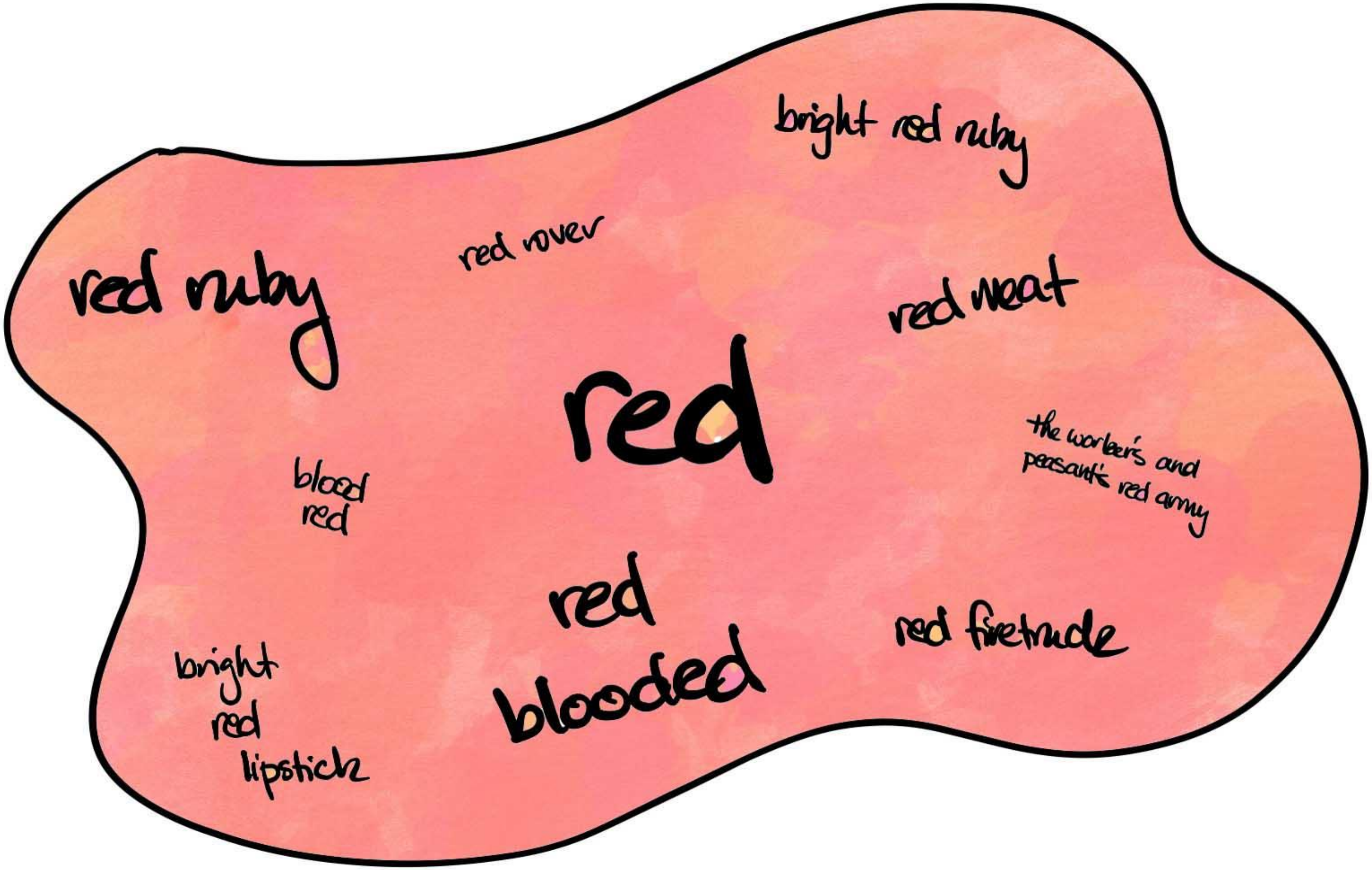
意味の文脈依存性とJ. R. Firth

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。」

例: “red” とその仲間たち（company）



分散意味論のベクトル空間でのモデル化と ニューラルネットワークの利用

分散意味論は、フレームワークとして線形代数を使うようになる。基本的なアプローチは、分散情報を高次元ベクトルで表現し、意味の類似性をベクトルの類似性 cosine similarityで定義する。

こうした動きは、統計的言語モデルからニューラルネットワーク上での言語モデル構築へと進んでいく。

- Bengio:"A Neural Probabilistic Language Model"
- Mikalov:Word2Vec
- …

参考資料

二つの分散意味論

- Deep Learningの分散意味論:
  BERT,GPT3を準備したもの

- DisCoCat の分散意味論

Deep Learningの分散意味論: BERT,GPT3を準備したもの

- **Linguistic Regularities in Continuous Space Word Representations**
  Tomas Mikolov et al.
  2013 (Word2Vec 論文)
  https://aclanthology.org/N13-1090.pdf

- **Sequence to Sequence Learning with Neural Networks**
  Ilya Sutskever et al,
  2014/12/14 (Sentence2Sentence論文)
  https://arxiv.org/pdf/1409.3215.pdf

- **Neural machine translation by jointly learning to align and translate**
  Bahdanau, D., Cho, K., and Bengio, Y,
  2016/05/19 (Attention 論文)
  https://arxiv.org/pdf/1409.3215.pdf

DisCoCat の分散意味論

- **Mathematical Foundations for a Compositional Distributional Model of Meaning**
  Bob Coecke, Sadrzadeh, Stephen Clark, 2010/05/23
  https://arxiv.org/abs/1003.4394

- **A quantum teleportation inspired algorithm produces sentence meaning from word meaning and grammatical structure**
  Stephen Clark, Bob Coecke, Edward Grefenstette, Stephen Pulman, Mehrnoosh Sadrzadeh,
  2013/10/11 https://arxiv.org/abs/1305.0556

- **Meaning updating of density matrices**
  Authors:Bob Coecke, Konstantinos Meichanetzidis, 2020/01/03
  https://arxiv.org/pdf/2001.00862.pdf

- **Foundations for Near-Term Quantum Natural Language Processing**
  Authors:Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, Alexis Toumi, 2020/12/07
  https://arxiv.org/abs/2005.04147

言語の構成性をどう捉えるか

大規模言語モデルの数学的構造 I

「ことばと意味の構成性」
$Compositionality$

このセッションでは、ことばと意味の関係を「構成性」という観点から取り上げたいと思います。

「構成性」、あまり聞きなれない言葉かもしれません。英語だと、"$compositionality$" と言います。あるものが、もっと基本的なものから「構成」されていることを言います。

ことばの「構成性」

「構成性」は、いろんなところで見ることができます。

ことばも、そうした「構成性」を持っています。

例えば、「黒い猫が走る」ということばは、「黒い」という形容詞、「猫」という名詞、「が」という助詞、「走る」という動詞から構成されています。

文法という「構成ルール」

ここまでで挙げた「構成」の例は、それぞれ違う「構成ルール」に基づくものですが、少し乱暴ですが、それらの「構成ルール」を、「数式の文法」、「プログラムの文法」、「ことばの文法」というように、「文法」と呼ぶことができます。

(別の言い方もあります。こうしたレベルの記号や語からの「構成ルール」を「統語規則」とか「syntax」ということがあります。)

意味の「構成性」

それでは、「意味」にも、「構成性」があるのでしょうか？

ことばを例に考えれば、ことばは意味を運ぶものです。その意味では、ことばと意味との関係は、直接的で、ことばと意味を分けるのは、少し難しいかもしれません。

ただ、「黒い猫」の意味は、「黒い」という語の意味と「猫」という語の意味から「構成」されると考えることは自然なことのように思えます。

「（黒い猫）（が）（走る）」という文の意味も、（黒い猫）と（走る）の意味から「構成」されると考えるのは可能です。

意味は、どう表現されるか？

意味の「構成性」を考える時難しいのは、構成された「意味」は、もはやもとの記号や語の並びそのものとは違うものだと言うことはわかっているのですが、それを、うまく表現することが難しいことかもしれません。

先にも述べたように、語のレベルで言うと特にそうなのですが、語そのものと語の意味は、強く結びついていて、分離するのは少し面倒です。でも、語から構成された文になると、文そのものとその意味は、分離しやすくなります。

ことばと意味の「構成性（compositionality）」基本的な観察

- 文は文法に従って語から構成される
- 文の意味は、語の意味に依存する
- 文の意味は、文の文法的構成に依存する

多様な「自然な意味」

記号や語の構成ルールに従った構成、文法的なレベルでの構成物は、そのうえ、コンテキストに応じて様々な「自然な意味」を持ちます。

プログラムでの、x = x + 1 という代入文は、「x という変数に x + 1 という値を代入する」という意味を持つと考えるのが自然です。

また、1 + 2 = 3 と言う数式の意味は、「「1 + 2 = 3」は正しい計算である」と考えることができます。

「意味」を一つの枠組みで捉える

ただ、これらの例でも、「意味」の「解釈」は、最初の直接的対象であった、「文法的構成」のそれぞれの要素あるいはその全体を、「意味」の構成に取り込んでいるように見えます。それぞれの例でも、意味の「構成性」は、透けて見えています。

これらの、いくつかの「意味」の「解釈」を、一つの枠組みで捉えることができるでしょうか？

DisCoCatの登場
構成的な分散意味論

大規模言語モデルの数学的構造 I

構成型論理モデルと分布型確率モデル
Coeckeの整理

単語は文の構成要素であるが、文の意味はそこに含まれる単語の意味をはるかに超える。実際、私たちは単語の辞書は持っているが、文を構成する単語の意味から文の意味を推論するのに辞書は必要ないように思われる。

自然言語における意味を与えるプロセスを発見することは、言語学やコンピュータサイエンスにおける最も基本的な課題のひとつであり、その発見は、文書検索やテキストの自動生成など、多くの言語関連タスクを自動化するアプリケーションを作る上で役立つだろう。

現在までに、構成型論理モデルと分布型確率モデルが、自然言語における意味付与の問題に対する2つの相補的な部分解を提供してきた。

論理的アプローチは、文の意味は文中の単語の関係から導かれるというフレーゲの原理を中心とする数理論理学の古典的な考えに基づいている。

分布モデルはより新しいもので、ウィトゲンシュタインの「使用としての意味」の哲学に関連しており、単語の意味はその文脈から決定することができるというものである。

理論面では論理モデルが王者であったが、実践面では確率論的なライバルが最良の予測を提供してきた。

「論理的形式」と「文脈的使用」という意味の定義特性の二者択一は、「意味の基礎構造とは何か」という問いを以前にも増して未解決のままにしている。

"Technical Proposal: Algorithmic and Logical Aspects when Composing Meanings” Bob Coecke and Mehrnoosh Sadrzadeh"

構成的分散意味論の基本的問題意識と方法

語から構成される文の構成性＝文法から、語の意味から構成される文の意味＝Semanticsを導くこと。

- SyntaxとSemanticsの対応付けに、カテゴリー論のFunctorial Semanticsのフレームを利用する。

- Syntax＝文法 を数学的カテゴリーとして捉える
  - Pregroup Grammar を利用する ： **PregX**
- Semantics＝意味を数学的カテゴリーとして捉える
  - 語と文の意味の表現に、ベクトル空間を利用する ： **FVect**

$$\boldsymbol{F} : \boldsymbol{PregX} \rightarrow \boldsymbol{FVect}$$

$$Functor : Syntax \rightarrow Semantics$$

Mathematical Foundations for a Compositional Distributional Model of Meaning

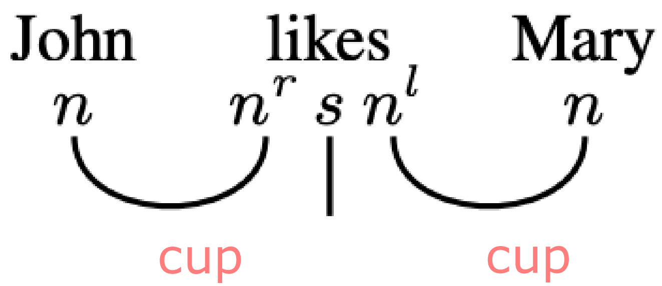
Bob Coecke, Mehrnoosh Sadrzadeh,
Stephen Clark

https://arxiv.org/abs/1003.4394

2010年

DisCoCat
Pregroup文法での”John likes Mary”の分析

- 語の型　John : n , likes : $n^r s n^l$, Mary : n
- 還元ルール　$p \cdot p^r \to 1$　　$p^l \cdot p \to 1$
- ダイアグラム



- 文の型sの導出

$$n \cdot (n^r \cdot s \cdot n^l) \cdot n \to$$
$$(n \cdot n^r) \cdot s \cdot (n^l \cdot n) \to 1 \cdot s \cdot 1 \to s$$

DisCoCat
Functor: pregroupからベクトル空間へ

- Functor $\mathcal{F}$ は、pregroupの型をベクトル空間に移す。

$$n \to N, s \to S, n^r s n^l \to N \otimes S \otimes N$$

  - 他動詞の型 $n^r s n^l$は、次数3のテンソル $N \otimes S \otimes N$ に移る
    これは、$N \otimes N \to S$ の双線形写像と見ることが出来る
  - 形容詞の型nn^lは、線形写像$N \to N$を 表わす行列と見ることが出来る

- Functor $\mathcal{F}$ は、pregroupでの還元(reduction)をベクトル空間の縮約(contraction)に移す。

- 文$s = w_1 w_2 \cdots w_n$の意味$\boldsymbol{s}$は、pregroupの還元ルールαの元で、次の式で与えられる。

$$\boldsymbol{s} = \mathcal{F}(\alpha)[\boldsymbol{w_1} \otimes \boldsymbol{w_2} \otimes \cdots \otimes \boldsymbol{w_n}]$$

語と文の「意味」と「文法」を同時に表現するカテゴリー $\mathbf{FVect} \times P$

$$
\begin{tikzcd}
& language \arrow[dl, "meaning"'] \arrow[d] \arrow[dr, "grammar"] & \\
\mathbf{FVect} & \boxed{\mathbf{FVect} \times P} \arrow[l, "\pi_m"] \arrow[r, "\pi_g"'] & P
\end{tikzcd}
$$

$$\mathrel{\rotatebox[origin=c]{90}{$\in$}}$$

語w_iの意味を表す拡張された空間:(W_i, p_i)

$W_i \in FVect$: 語w_iの意味ベクトル

$p_i \in P$: 語w_iの文法型

What is applied category theory

3.2 Natural Language Processing

Tai-Danae Bradley
2018年
https://arxiv.org/pdf/1809.05923.pdf

文法のCategory

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

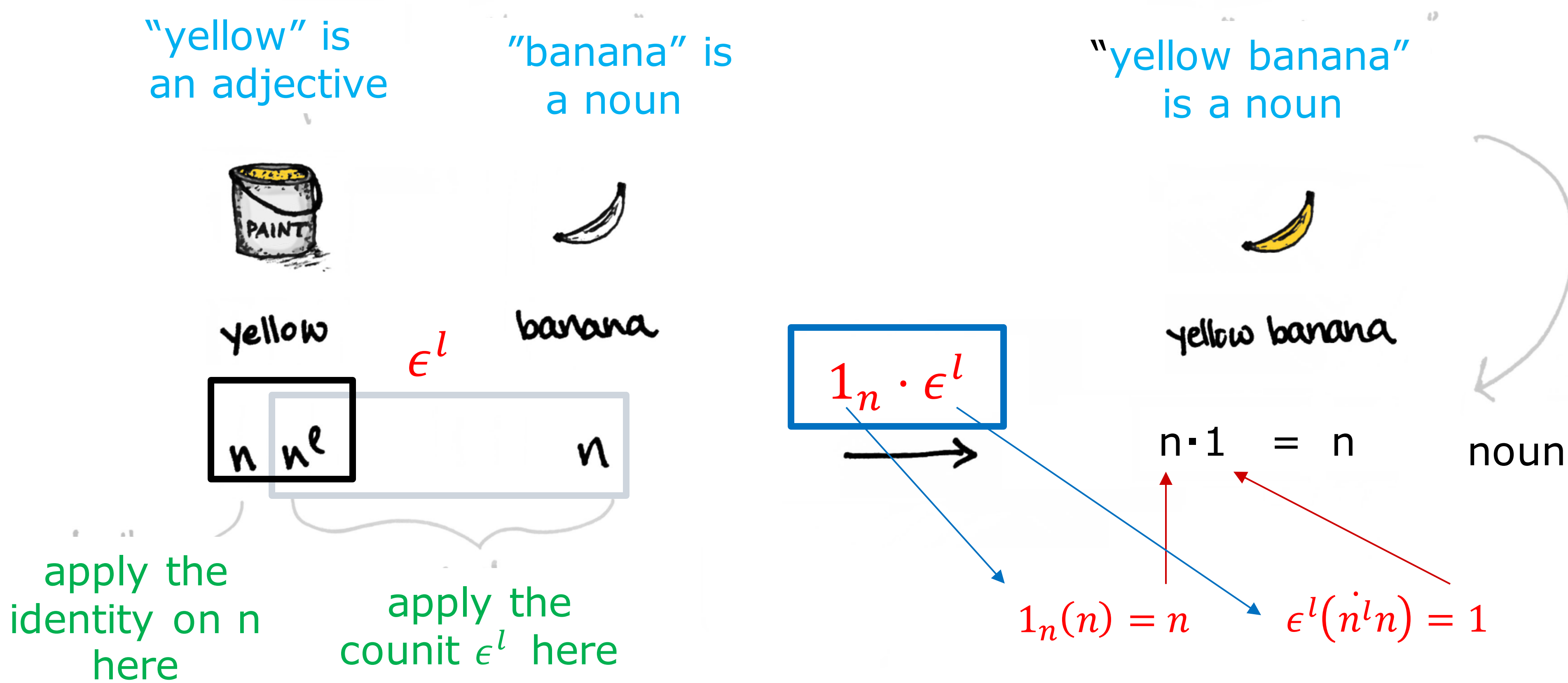
$n^r s n^l$ ＝ 他動詞

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。

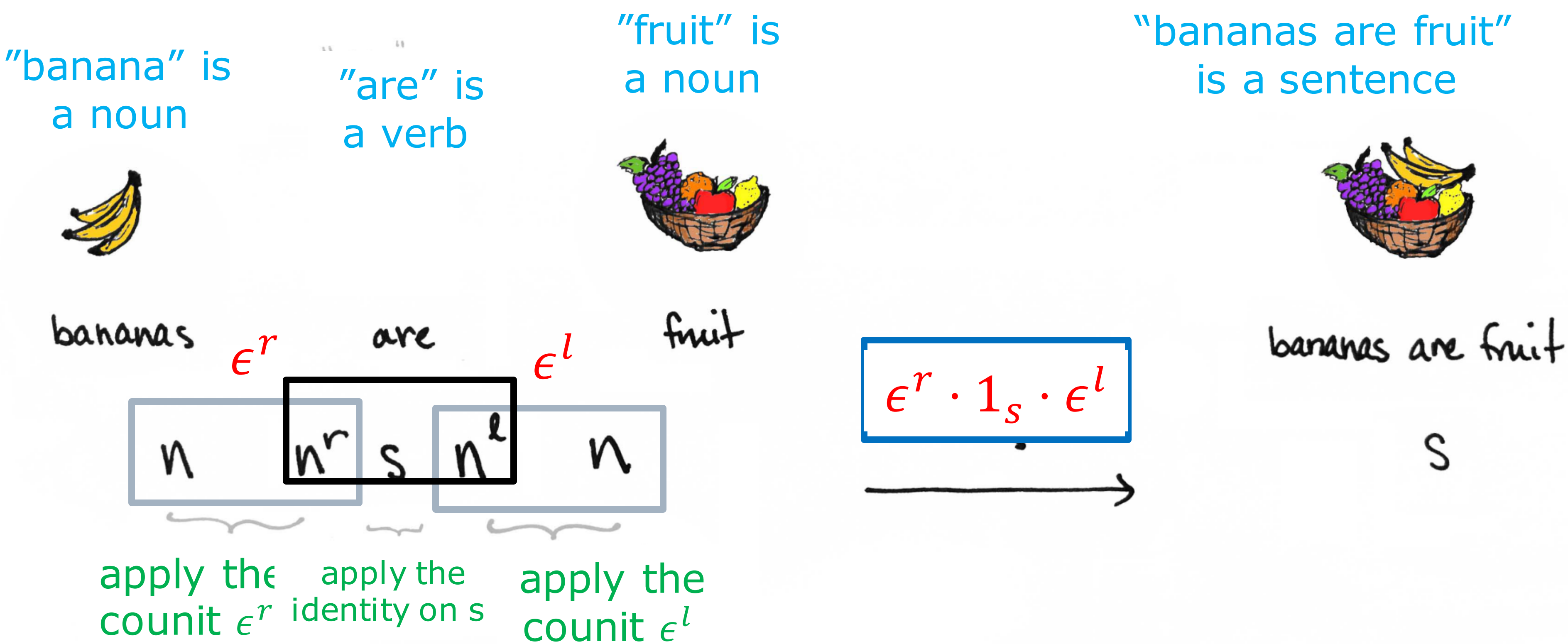
$w^l w = 1$ （ε^l を適用すると消える）

$w w^r = 1$ （ε^rを適用すると消える）

“yellow banana” は名詞として扱われる



“bananas are fruit” は文である



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

The Functor F: Syntax → Semantics

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**

On morphisms

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

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

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。

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

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

$\mathrm{F}(n^r)=F\left(n^l\right)=N$　(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。

$\mathrm{e.g.}\ F\left(n^r s n^l\right)\cong F(n^r)\otimes F(s)\otimes F\left(n^l\right)=N\otimes S\otimes N$

DisCoCatでの構成的意味の理解
Syntax → Semantics

Syntax

banana is fruit

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

Functor F

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

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

$Fs = S$

Semantics

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

banana⊗is⊗fruit

Quntum-NLP

大規模言語モデルの数学的構造 Ｉ

CoeckeのQuantum-NLP

このセッションでは、DisCoCatの創始者Bob Coeckeが、現在どのような研究を行っているのかを見てみようと思う。

彼は、DisCoCat の枠組みを、ことばの意味を量子状態として捉える方向で発展させ、QNLP 量子論的自然言語処理 Quantum Natural Language Processing を展開した。

実際の量子コンピュータ上でQNLPの実証実験が始まっている。

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

Abstract

近い将来の量子自然言語処理（QNLP）の概念的、数学的基礎を提供し、量子コンピュータ科学者にやさしい言葉でそれを行う。

説明的な表現方法を採用し、実証的な証拠と数学的な一般性についての形式的な記述のための参考文献を提供する。

我々は、我々が採用している自然言語の量子モデルが、言語的な意味と豊かな言語構造（特に文法）を正しく標準的にに結合していることを思い出している。

特に、意味と構造を結合するために量子的なモデルが必要であるという事実は、QNLPが量子系のシミュレーションと同じように、量子ネイティブであることを立証している。

さらに、量子ハードウェア上で古典データをエンコードするための、現在主流のノイズの下での中規模量子（NISQ）パラダイムのさまざまな量子回路は、NISQを特別にQNLPに適したものにしている。

言語構造は、明らかに指数関数的に高い計算コストを必要とするな古典な文法のえンディングとは対照的に、ほとんどただでエンコードすることができる。

図式的な推論はQNLPの中核をなすものである。

まず、量子モデルは、カテゴリー論的量子力学の図式的な形式化によって、言語を量子プロセスとして解釈する。次に、これらの図式はZX-calculusによって量子回路に変換される。そして、意味のパラメータ化が、学習すべき回路変数となる。

我々の言語構造の量子回路へのエンコーディングは、言語構造の中核にウィトゲンシュタインの「意味とは文脈である」と言う議論を据えることによって、現在主流のAI技術を超える、新しい言葉の意味を確立するアプローチを具体化するものである。

自然言語の量子モデル

複数の語の意味は、複数のqubitの状態で表現されるとしよう。

特に、hat（帽子）のような名詞 n の意味は、1-qubitの状態 $|\psi_n > \in \mathbb{C}^2$で表現されるとする。

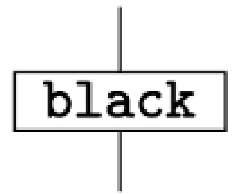
$$\text{名詞 } n \text{ の意味} \Leftrightarrow 1 - qubit\text{の状態 } |\psi_n > \in \mathbb{C}^2$$

$$\text{名詞 hat の意味} \Leftrightarrow 1 - qubit\text{の状態 } |\psi_{hat} > \in \mathbb{C}^2$$

名詞に形容詞を適用する (1)

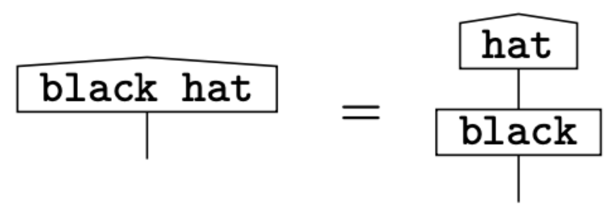
$hat \rightarrow black\ hat$ 写像としての形容詞

black hatの形容詞 blackは、名詞 hatの状態を black に変える。形容詞aを1-qubitの写像$\eta_a\colon \mathbb{C}^2 \rightarrow \mathbb{C}^2$ と表わすことが出来る。

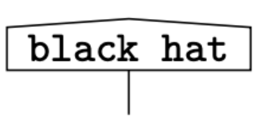
$\eta_{\mathtt{a}} \rightsquigarrow$ 

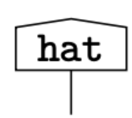
名詞nへの形容詞aの適用の意味$|\psi_{a \cdot n} >$ は次のように表現される。

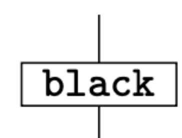
$$|\psi_{a \cdot n} > = \eta_a(|\psi_n >)$$

図式で表すと、次のようになる。 

ここで、

$|\psi_{\mathtt{a \cdot n}}\rangle \rightsquigarrow$ 

$|\psi_{\mathtt{n}}\rangle \rightsquigarrow$ 

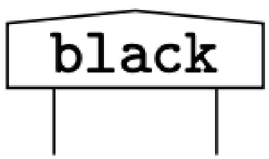
$\eta_{\mathtt{a}} \rightsquigarrow$ 

名詞に形容詞を適用する (2)

$hat \to black\ hat$ 状態としての形容詞

Blackとhatの意味から black hat の意味を得るもう一つの方法がある。それは、形容詞を 2-qubitの状態$|\psi_a >\in \mathbb{C}^2 \otimes \mathbb{C}^2$と考えることである。

$$|\psi_{\mathtt{a}}\rangle \rightsquigarrow$$



2×2の行列の要素は、次の対応で2-qubitの状態と対応する。

$$\eta_{00}|0\rangle\langle 0| + \eta_{01}|0\rangle\langle 1| + \eta_{10}|1\rangle\langle 0| + \eta_{11}|1\rangle\langle 1| \quad \mapsto \quad \eta_{00}|00\rangle + \eta_{01}|01\rangle + \eta_{10}|10\rangle + \eta_{11}|11\rangle$$

Choi-Jamiolkowski correspondence

名詞に形容詞を適用する (2)
$hat \rightarrow black\ hat$ を図式で表す 状態としての形容詞

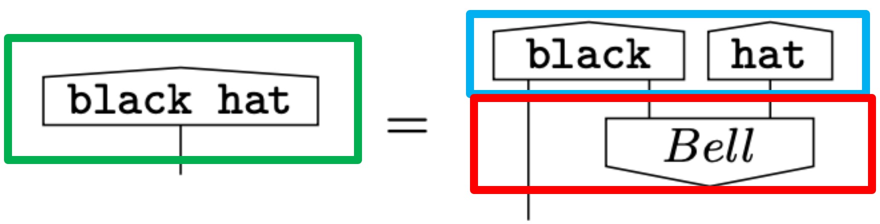
この時、名詞に形容詞を適用した時の意味は、次の式で表される。

$$|\psi_{a\cdot n}> = (\mathbb{I} \otimes <Bell|\,) \circ (|\psi_a> \otimes |\psi_n>\,)$$

ここで $<Bell| = <00| + <11|$ で、$\mathbb{I}$ は同一性である。

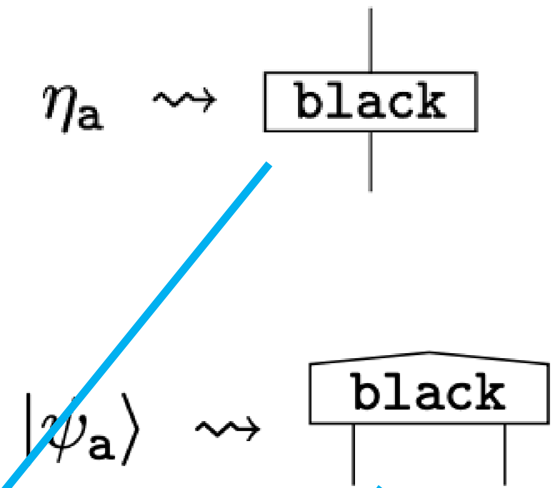
$$|\psi_{a\cdot n}> = (\mathbb{I} \otimes <Bell|\,) \circ (|\psi_a> \otimes |\psi_n>\,)$$

は、図式では、次のように表現される。

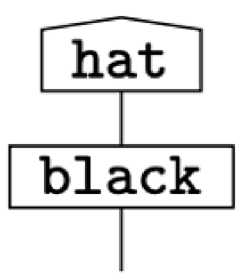


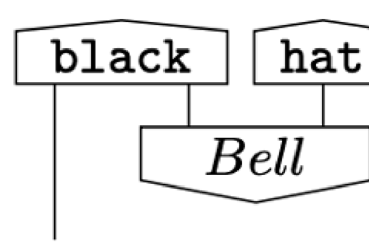
形容詞の二つの表現

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



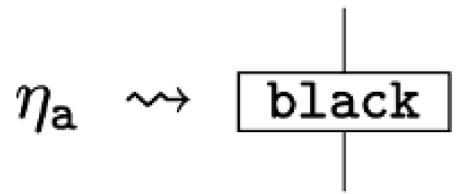
形容詞を 2-qubitの状態
$|\psi_a> \in \mathbb{C}^2 \otimes \mathbb{C}^2$と考える



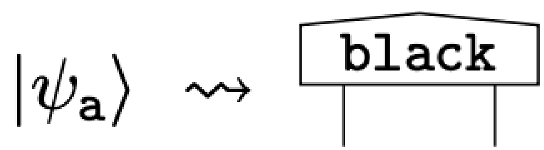


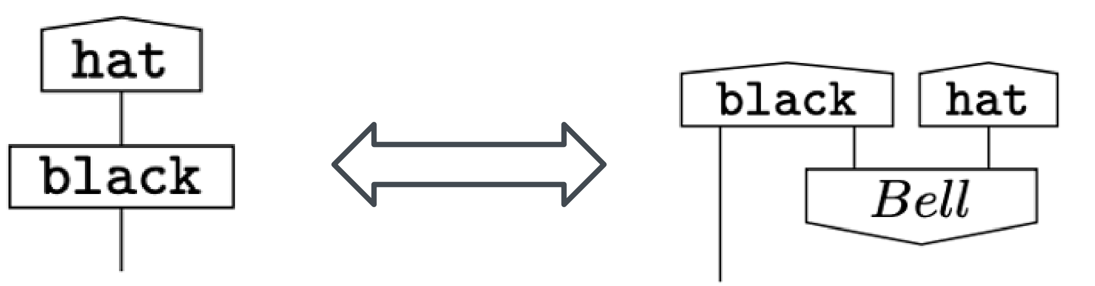
形容詞の二つの表現

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

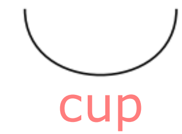




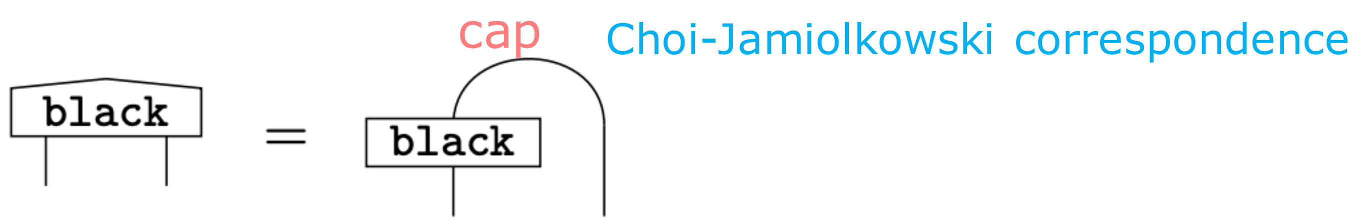
$|Bell>$ と $<Bell|$
cap と cup

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

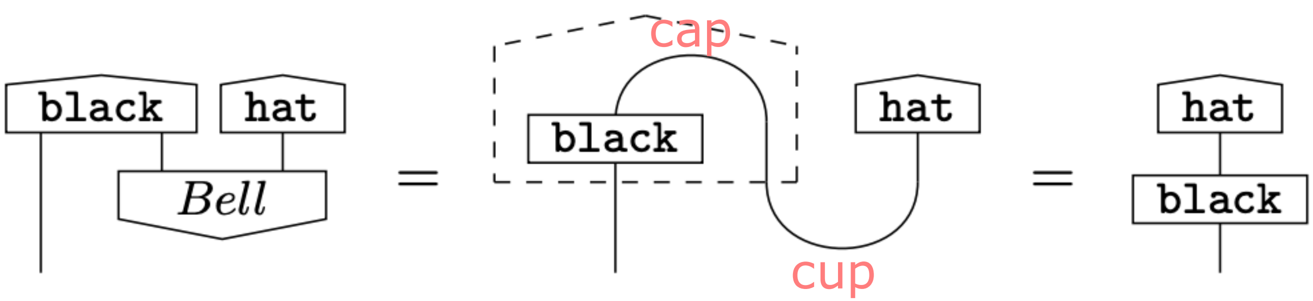
$|Bell\rangle =$ 

$\langle Bell| =$ 

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。

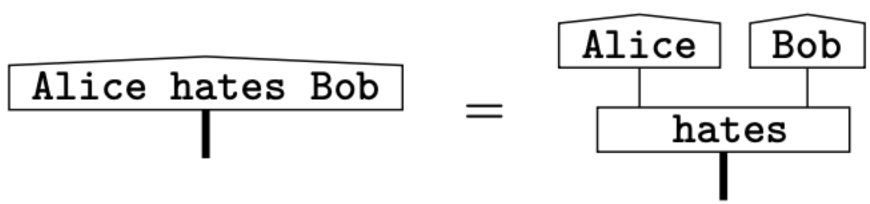
動詞tvに主語n_sと目的語n_oを与える

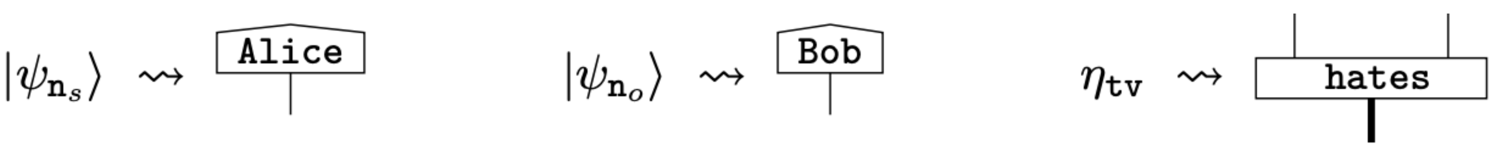
写像としての動詞

$|\psi_{n_s}>\in\mathbb{C}^2, |\psi_{n_o}>\in\mathbb{C}^2, |\psi_{n_s\cdot tv\cdot n_o}>\in\mathbb{C}^{2k}$ とする。

kは、文の意味空間のサイズ。

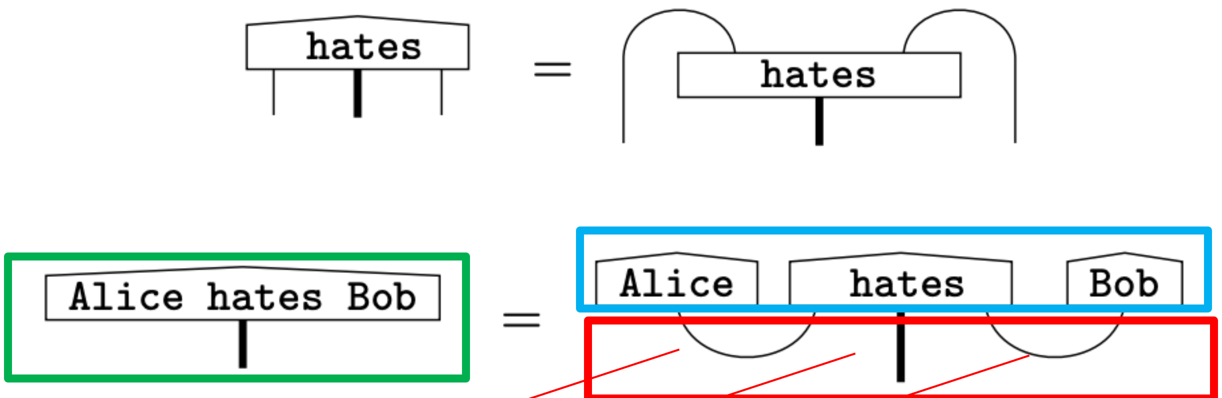
$$|\psi_{n_s\cdot tv\cdot n_o}>=\eta_{tv}(\,|\psi_{n_s}>\otimes\ |\psi_{n_o}>)$$





動詞tvに主語n_sと目的語n_oを与えるもう一つのアプローチ　状態としての動詞

$$|\psi_{tv}> \in \mathbb{C}^2 \otimes (\mathbb{C}^{2k}) \otimes \mathbb{C}^2$$



$$|\psi_{n_s \cdot tv \cdot n_o}> =$$

$$(< Bell \mid \otimes \, \mathbb{I} \otimes < Bell \mid) \circ (\,|\psi_{n_s}> \otimes |\psi_{tv}> \otimes |\psi_{n_o}>)$$

QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, Bob Coecke

https://arxiv.org/abs/2102.12846v1

2021年

Abstract

量子自然言語処理(QNLP)は、量子ハードウェア上で実行することを目的とした自然言語処理モデルの設計と実装を扱うものである。本論文では、100文以上のデータセットに対して、Noisy Intermediate-Scale Quantum(NISQ)コンピュータ上で行った最初のNLP実験の結果を紹介する。

Coeckeら(2010)による意味の構成モデルと量子論の形式的な類似性を利用し、量子回路に自然にマッピングされる文の表現を作成する。これらの表現を用いて、量子ハードウェア上で簡単な文の分類タスクを解く2つの自然言語処理モデルの実装と訓練に成功した。

本論文では、これらの実験の主要な原理、プロセス、課題を、自然言語処理研究者が理解しやすい方法で詳細に説明し、実用的な量子自然言語処理への道を開く。

量子コンピュータは、現在の標準的技術より計算速度を指数関数的に向上させるという前提の元で、量子コンピュータ技術はコンピュータサイエンスの最先端分野として急速に発展している。

ただ、最近まで、量子コンピューティングの研究のほとんどは、純粋に理論的であったり、古典的なハードウェア上でのシミュレーションに関心を持っていたのだが、ノイズの下での中間スケール量子（ Noisy Intermediate Scale Quantum ：NISQ）デバイスと呼ばれる研究者が利用できる最初の量子コンピュータの登場により、暗号（Pirandolaら、2020）、化学（Caoら、2019）、生物医学（Caoら、2018）といった幅広いテーマにまたがる有望な実用結果や応用がすでに得られている。

この新しい計算のパラダイムが、自然言語処理にも利用できるのか、という疑問は当然に生まれている。このような応用は、言語関連の問題で計算の高速化を目的とするためだけではない。

計算の高速化を超えて、量子システムがどのように数学的に記述され、また、情報の「量子的」エンコーディングがどのように行われているのかを研究することが、言語の意味の表現と処理の概念的・実用的な進歩につながる可能性がある。

このような観点にインスパイアされた、量子自然言語処理（QNLP）は、まだ始まったばかりの研究分野であり、量子ハードウェア上で実行できるように設計された自然言語処理モデルの開発を目指している。

この分野には素晴らしい理論的研究が存在するが、提案されている実験は古典的なシミュレーションである。例外は最近の研究（Meichanetzidis et al.、2020）で、非常に小規模な概念実証実験が初めて量子ハードウェア上で実行された。

この論文では、量子ハードウェア上で実行される言語処理タスクからなる2つの完全な中規模実験を紹介する。

この実験の目的は、NLPタスクにおいて古典的な実装よりも量子的な優位性を示すことではない。現在利用可能な量子コンピュータの機能が限られているため、このようなことはまだ不可能であると考えている。

本研究では、QNLPが実際にどのようなものであるかについて、NLPコミュニティに詳細な説明を提供することを主な目的としている。従来のモデリングとコーディングのパラダイムがどのように量子に適した形に移行できるかを示し、現在のNISQコンピュータが課す課題と制限を探る。


Bob Coecke

CoeckeのOxfordでの担当コースシラバス
Classical and Quantum Compositional Distributional Meaning (2022-23)

自然言語（コンピュータとは異なる）の意味をモデル化することは、人工知能において最も困難な問題の一つである。この問題を解決することで、テキストの要約、検索、機械翻訳、言語生成、質問応答など、さまざまなテキストや言語処理アプリケーションの品質やインパクトを劇的に向上させる可能性がある。

この問題に対しては、長年にわたって多くの異なるアプローチが考案されてきた。その一つが「形式的意味論」で、自然言語を高次の論理に「コンパイル」するプログラミング言語として扱う。また、「分散意味論」は、言葉の意味を、出現文脈によって決まる高次元の意味空間の点としてモデル化するものである。

https://courses.maths.ox.ac.uk/course/view.php?id=833

最近の研究では、この2つのアプローチの長所を調和させ、意味の構成的な分散型（ベクトルベース）モデルを作り出すという課題に取り組んでいる。

このコースは、この新しく急速に成長している分野の理論的な端緒となるもので、フレーズやセンテンスの意味を形成するために単語の意味をどのように構成するかに焦点を当てている。

特に、量子自然言語処理（QNLP）という新しい分野は、現在DisCoCatとして知られているこの言語モデルに基本的な方法で依存している。QNLPは、量子コンピュータの最も有望な手段の1つと考えられている。

https://courses.maths.ox.ac.uk/course/view.php?id=833

Part 2

大規模言語モデルの特徴

Part 2 大規模言語モデルの特徴

Agenda

- Tai-Danaeの問題提起
- あらためて、セミナーの目的について
- Tai-Danaeの道具箱

大規模言語モデルの特徴と
Tai-Danaeの問題意識

大規模言語モデルの数学的構造 I

なぜ、Tai-Danaeの議論に注目するのか

なぜ、DisCoCatの流れの中で、Tai-Danae の議論に注目するのでしょうか？ ある論文の冒頭で、彼女はこう言っています。

「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、まず、現在の「大規模言語モデル」の「印象的」な性能に向けられています。その上で、「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」ことに注目しています。

Language Modeling with Reduced Densities
https://arxiv.org/abs/2007.03834v4

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？

ここでは、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みておきましょう。

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。

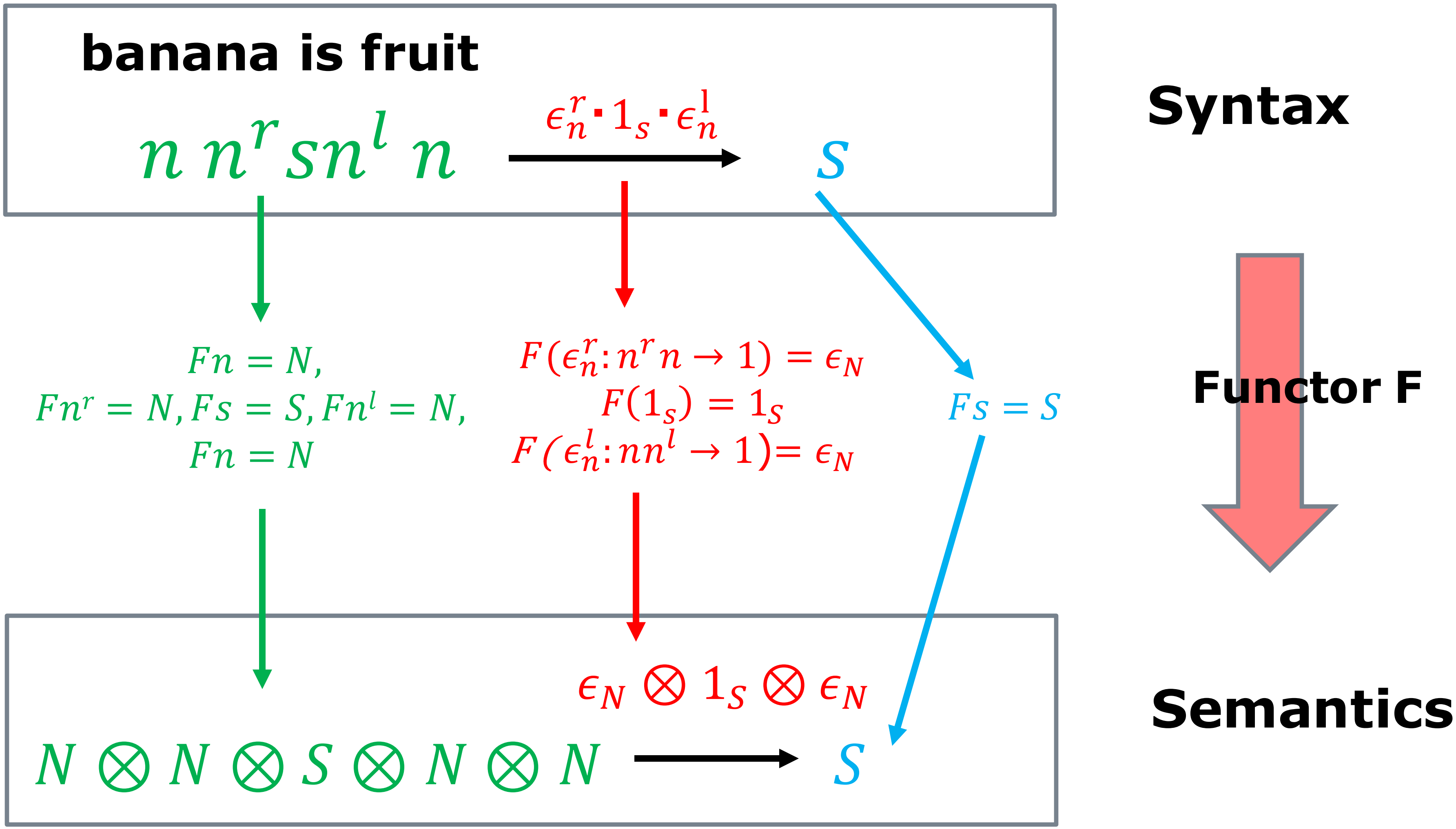
より驚くべき大規模言語モデルの能力

問題は、それだけではありません。

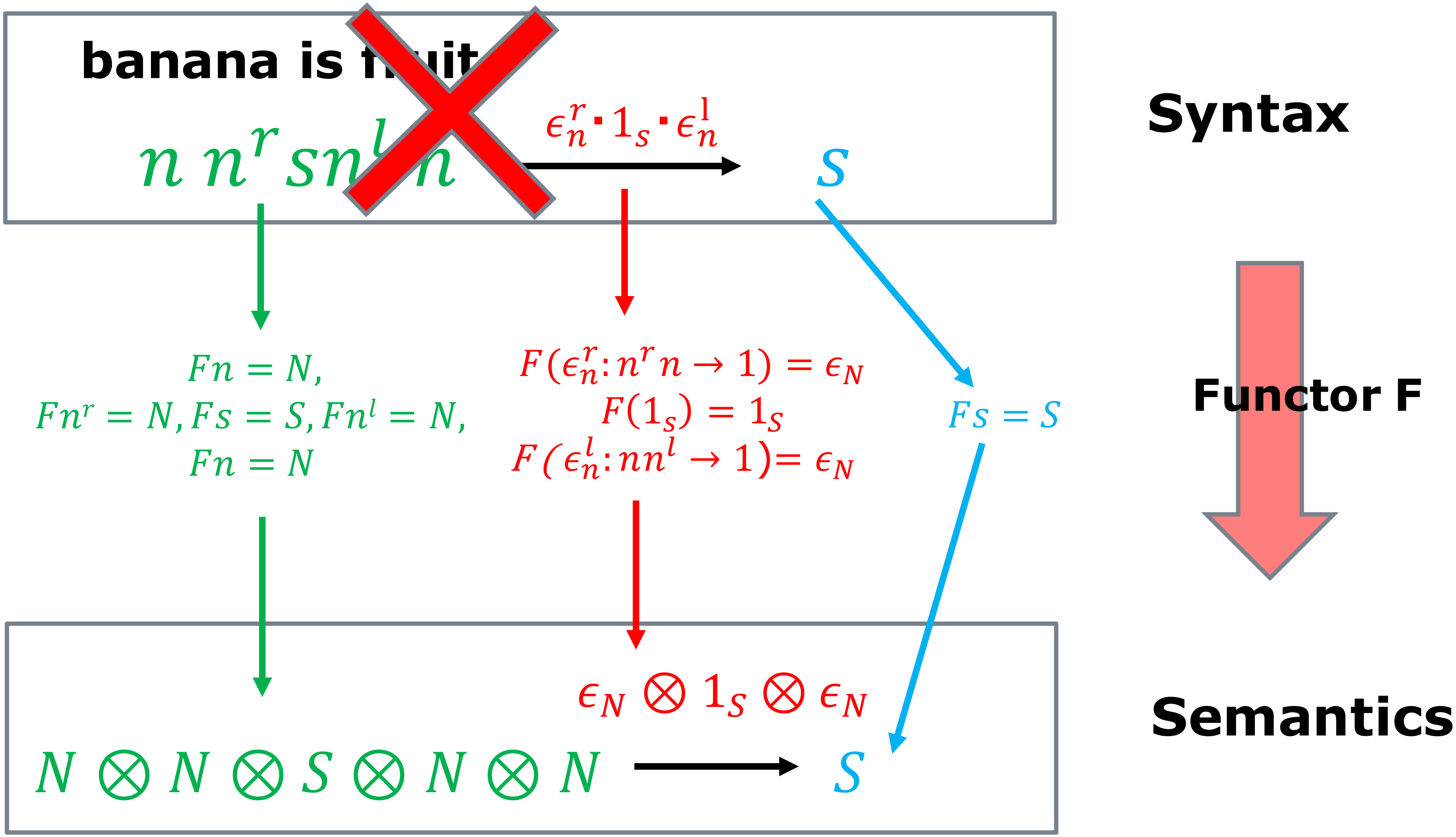
「驚くべきは、これらのLLMが、ラベルのないテキストサンプルを使って、次の単語を事前に指示するように学習できることである。
文法的、意味的な入力は提供されないが、それにもかかわらず、複雑な構文構造、意味情報、世界知識が学習され、実証される。」

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、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。

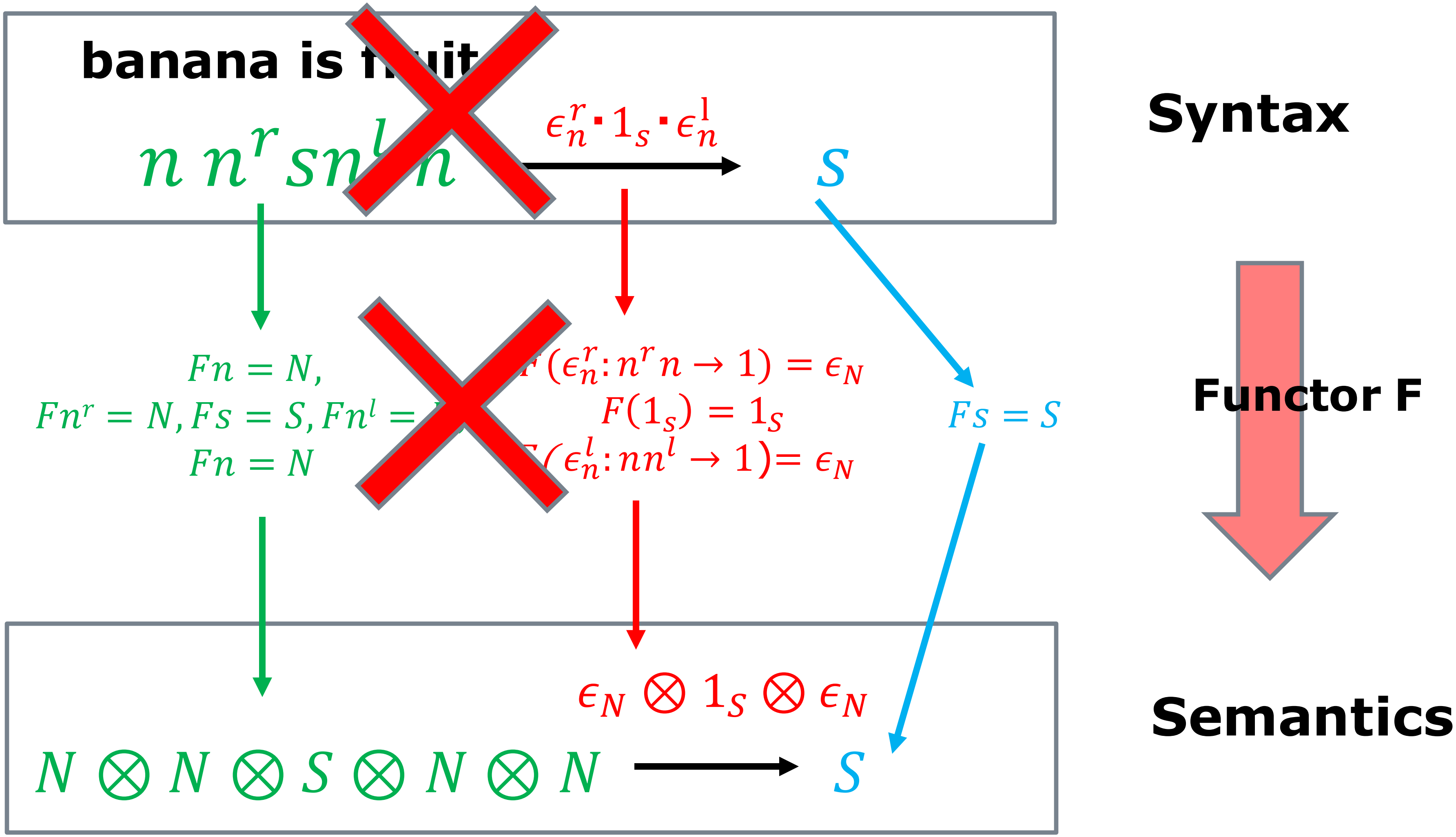
以前の彼女のDiscoCatの図式



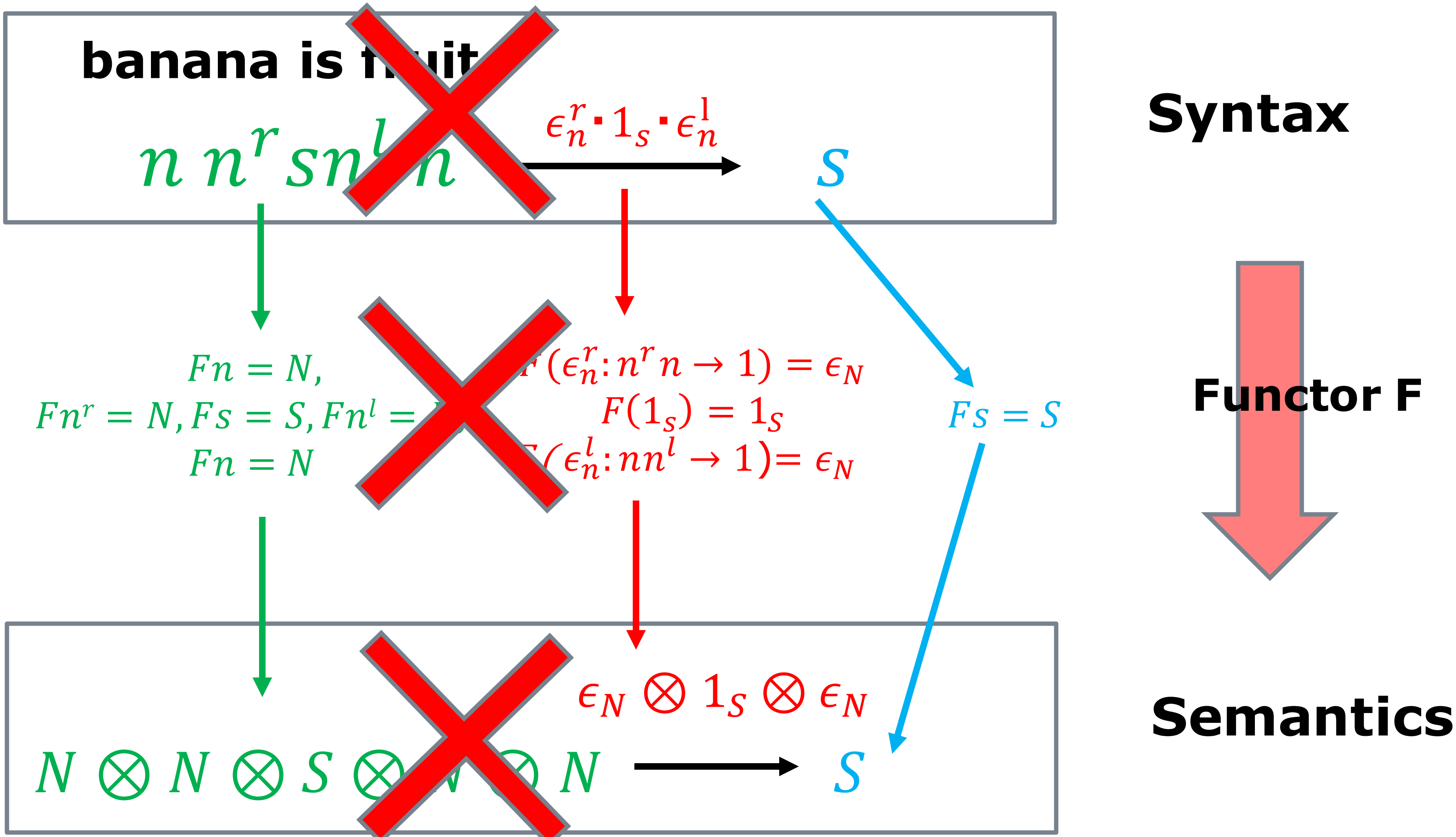
入力されるテキストは構造化されていない



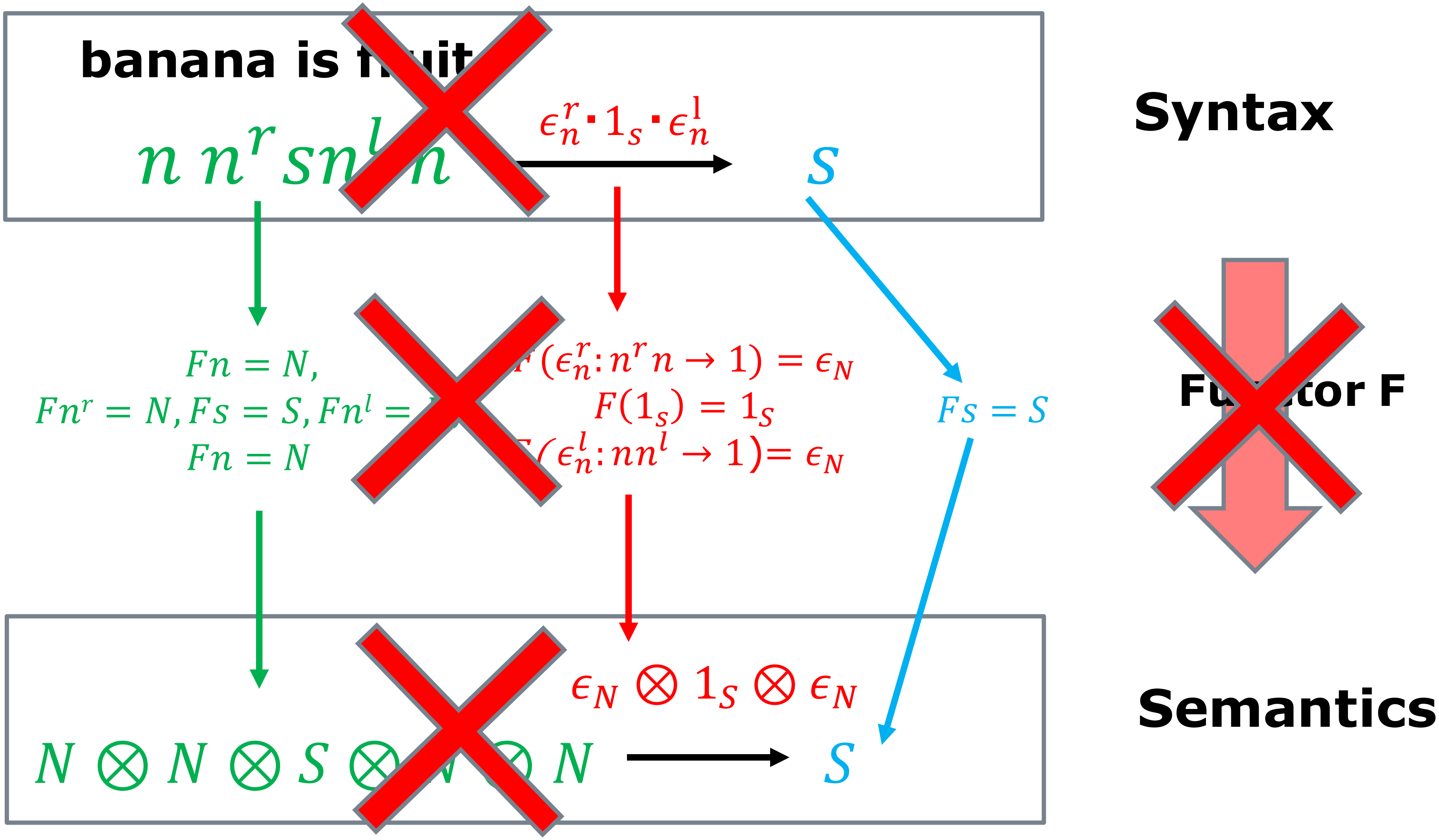
構造化されていないテキスト



構造化されていないテキスト



大規模言語モデルは、DiscoCatモデルでは説明できない



彼女の目指すもの
構成的分散意味論の再構築

数学的カテゴリーとしての文法理論も、文法カテゴリーから意味のカテゴリーへのFunctor意味論も、大規模言語モデルの振る舞いを基礎づけるには使えそうもありません。

ただ、それにもかかわらず、彼女は、SyntaxからSemanticsへの構成的アプローチを捨てたわけではありません。今回のセミナーで主要に紹介するのは、彼女たちの次の論文です。

「本研究は、テキスト継続の確率分布から意味情報への移行が可能であるという現実世界の証拠への応答である。この移行のための数学的枠組みを提案する。」

あらためて、セミナーの目的について

セミナーの目的を再確認する

改めて今回のセミナー「大規模言語モデルの数学的構造」の目的を確認したいと思います。

まずは、個人的な背景から。

正直に言うと、僕には大規模言語モデルの働きについてよくわからないところがいろいろあります。ただ、その振る舞いには何かの理由があるはずです。また、その理由は数学的に表現できるとも考えています。

そうした疑問の中、DisCoCatの時代から注目していた Tai-Danae Bradleyの議論の「変化」に注目するようになります。

DisCoCatの理論は、言語の文法的構成性を言語の意味的構成性に結びつける、強力なカテゴリー論的枠組みを提供する素晴らしい理論です。

ただ、現実の大規模言語モデルの振る舞いは、DisCoCatの枠組みでは説明できないのです。

なぜなら、DisCoCatのモデルではモデルに与えられる入力は、あらかじめ pregroupとして形式的に記述される文法構造を持ち、そうしたものとして解析されていることを前提としているからです。大規模言語モデルは、そうではありません。

ある論文の冒頭で、Tai-Danae Bradleyは、こう言っています。

「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、まず、現在の「大規模言語モデル」の「印象的」な性能に向けられています。その上で、「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」ことに注目しています。

まったく、同感です。

Language Modeling with Reduced Densities
https://arxiv.org/abs/2007.03834v4

Tai-Danaeは、別の論文で、次のように語ります。

「最先端の言語モデルは、どのような入力テキストからも自然言語テキストの続きを返すことができる。首尾一貫したテキストの拡張を生成するこの能力は、このモデルが文法や意味論の知識を含む、かなりの高度化を達成していること意味する。
本論文では、今日の大規模な言語モデルによって学習されるような、与えられたテキストの拡張に関する確率分布から、意味情報を含む豊かなカテゴリに移行すための数学的枠組みを提案する。」

これは、まさに、僕が知りたいことです。

An enriched category theory of language: from syntax to semantics
https://arxiv.org/abs/2106.07890

今回のセミナーの目的

今回のセミナー「大規模言語モデルの数学的構造」の主要な目的は、は、Tai-Danae らの論文

"An enriched category theory of language: from syntax to semantics"
https://arxiv.org/abs/2106.07890

の内容を紹介することにあります。

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

大規模言語モデルの特徴と
Tai-Danaeの道具箱

大規模言語モデルの数学的構造 I

Tai-Danaeはどんな概念装置を利用したか？

このセッションでは、大規模言語モデルの特徴を捉えるために、どのような概念装置を利用したかを見ていきたいと思います。

今回、紹介するのはその概略です。

DisCoCatモデルとの比較

彼女の大規模言語モデルの数学モデルの概要を理解するには、を、それをDisCoCatのモデルと比較するのがわかりやすいと思います。

DisCoCatのモデルというのは、SyntaxとSemanticsの対応を次のように捉えるものです。

$$Functor\ F:\ PreGroup \rightarrow FVect$$

ここで $PreGroup$は、LambekのPregroup文法のカテゴリーで、$FVect$は、意味を表現する有限ベクトル空間のカテゴリー、$Functor\ F$ は、カテゴリーからカテゴリーへの一般的なFunctorです。

大規模言語モデルの訓練用データはPreGroupではない

大規模言語モデルに与えられる訓練用データは、ただの平文で、文法構造が Pregroup文法で解析済みのものではありません。

ただ、これまでそうした平文の集まりは、「何の構造もない」と言ってきたのですが、文をフレーズに、フレーズを語に分解して、それらの全体を考えてみます。文・フレーズ・文を、文字の並びとして考えてみると、これらの文字列の間には、単純ですが、ある関係があることがわかります。

それは、文字列xが文字列yに含まれるという関係です。これを $x \leq y$ と表すことにします。

例えば、

”blue” ≤ “small blue” ≤ “small blue marbles”

という関係が成り立っています。

言語LをPreorderのカテゴリーとして考える

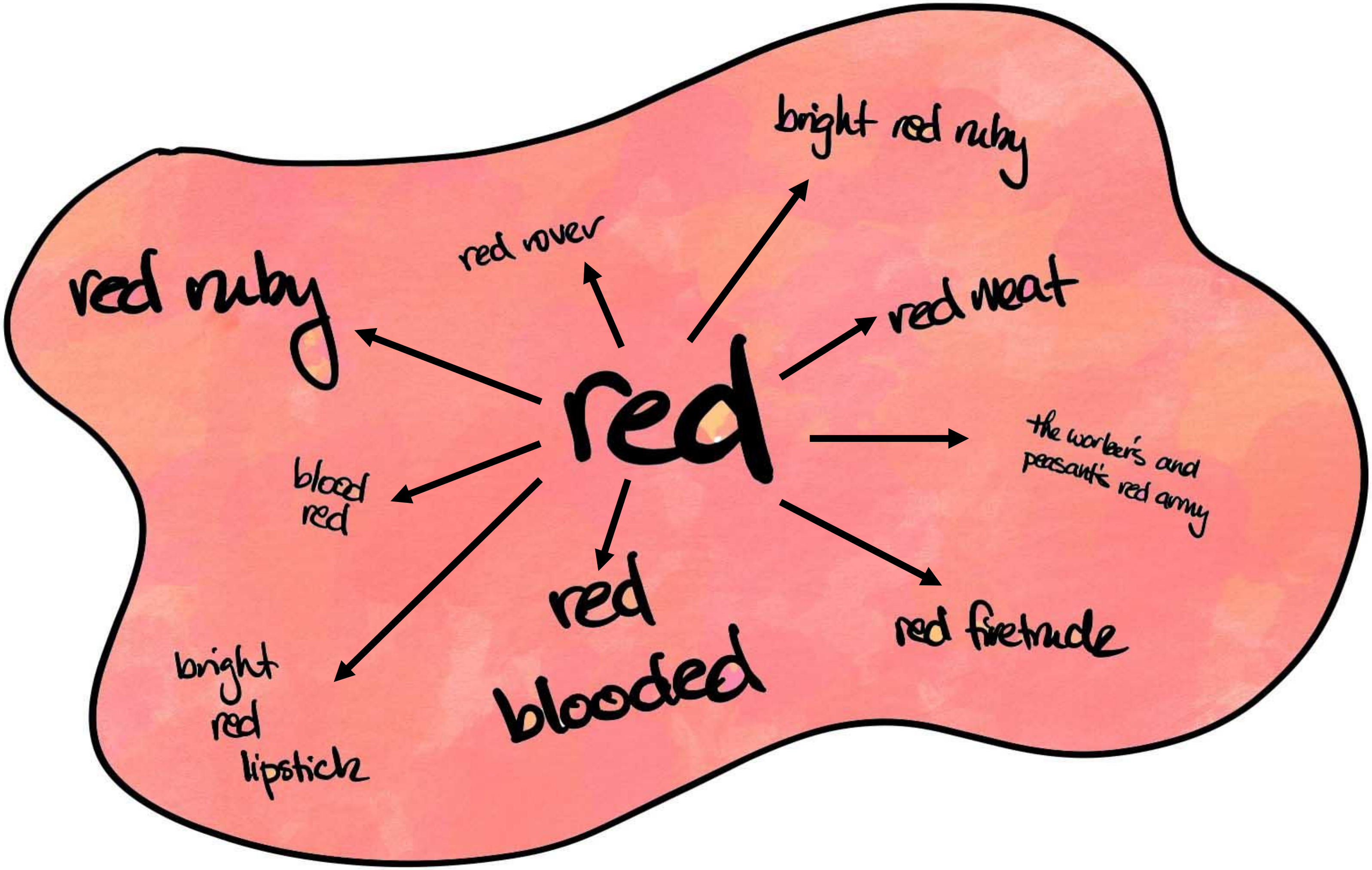
xがyの部分文字列であるという関係$x \leq y$をPreorderと言います。

$x \leq y$を$x \to y$ で表すと、文字列の全体は、$x \to y$を射とするカテゴリーと考えることができます。

これを言語Lだと考えます。

言語は、Preorderを射とするカテゴリーということになります。

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



preorderのもう一つの意味

先のような説明だと、preorderの関係が定義されているのは、語とその語を含む文字列の間のように見えます。確かにそれは、Firth流に語の意味を考えるにはいいモデルになっています。

ただ、preorder の関係にとって、語が一番基本的かというとそうではありません。語はもちろんのこと、意味のあるフレーズ、文、さらには、任意の長さの文の並びも、要するに自然言語の文字列表現は全て、preorder によって順序づけられます。

このことは、preorderが、大規模言語モデルが得意とする、表現の継続あるいは連続の、自然なモデルでもあることを意味しています。

DisCoCatの意味を表現するFVectは？

意味を表現するFVectは、どう変わったのでしょう？

彼女のモデルでは、”語の意味は、それが引きつれている仲間で決まる”というFirth流の意味論が採用されています。語”red”の意味は、先の図でredと矢印で結ばれている文字列全てで表現されることになります。

このカテゴリーを copresheaf と言います。

定義で言うと、カテゴリーC上のcopresheaf とは、カテゴリーCから集合のカテゴリーSetへのFunctor です。

$$copresheaf\ cps: C \rightarrow Set$$

これを、Set^Cで表します。

copresheafが、Firth流の意味を表現するカテゴリーになりうることについては、後の章で詳しく説明したいと思います。

ここで重要な役割を果たすのは、「米田のレンマ Yoneda Lemma 」という定理です。

また、このカテゴリーは、大規模言語モデルの意味の世界に論理的な構造を導入する役割も果たします。

多分多くの人には見知らぬカテゴリーcopresheaf Set^C ですが、構成はベクトルの構成と似ているところがあります。

ベクトルVというのは(実ベクトルなら)、ある集合Xから実数Rへの関数です。

$$V : X \to \mathbb{R}$$

これは、$\mathbb{R}^X$と表すことができます。

FVect $\mathbb{R}^X$の代わりに、Set^C が登場しても、似ているところがあると思っていいのです。

DisCoCatのSyntaxとSemanticsを対応づけるFunctor は？

新しい枠組みでは、DisCoCatのFunctorの代わりに、Yoneda embeddingというのを使います。

これもあとでcopresheafと一緒にきちんと説明します。

DisCoCatモデルと新しいモデル

DisCoCatのモデルというのは、SyntaxとSemanticsの対応を次のように捉えるものです。

$$PreGroup \quad \overset{Functor}{\longrightarrow} \quad FVect$$

新しいモデルでは、SyntaxとSemanticsの関係はこうなります。

$$Preorder \quad \overset{Yoneda}{\longrightarrow} \quad copresheaf$$

ただ、それだけではないのです

今までのがモデル構成の最初の段階です。次の段階があるのです。(実はもっとあるのですが)

それは、最初の言語Lを、preorder のカテゴリーと特徴づけたこととと関連します。

それは元々は、“red” → “red ocean” のような単純な射からなるカテゴリーだったのですが、この射に、確率を紐づけます。

$$\text{“}red\text{”} \xrightarrow{0.24} \text{“}red\ ocean\text{”}$$

$$\text{“}red\text{”} \xrightarrow{0.031} \text{“}red\ idea\text{”}$$

一般に、あるカテゴリーの射がある性質を持つように拡張したカテゴリーを enriched category と呼びます。

彼女の新しいモデルは、このenriched category を用いて、大規模言語モデルの確率的性質を記述しようとするモデルなのです。

この確率論的拡張は、語とその近傍の表現の間だけでなく、任意の表現の間にも、適用されます。

前者の確率論的拡張が、言語の文法性の構築に寄与し、後者の確率論的拡張が、大規模言語モデルの、継続する文、表現の連続を生み出す能力を基礎付けることになるのです。

彼女が利用する概念装置

彼女が利用する概念装置の概略（名前だけですが）をまとめてみました。

- preorder
- copresheaf
- Yoneda Lemma
- Yoneda embedding
- enriched category

つぎから、これらのカテゴリーの働きを、大規模言語モデルとの関係を含めて説明したいと思います。

Part 3

言語をカテゴリーとして捉える

大規模言語モデルの数学的構造 I
言語へのカテゴリー論的アプローチ入門

Part 3 言語をカテゴリーとして捉える

Agenda

- preorderとしての言語
- categoryとしての言語
- functor：言語のcategoryと意味のcategoryを結ぶ

preorderとしての言語

言語のプリミティブな構造を考える

大規模言語モデルの数学的構造 Ｉ

DisCoCatとpregroup文法

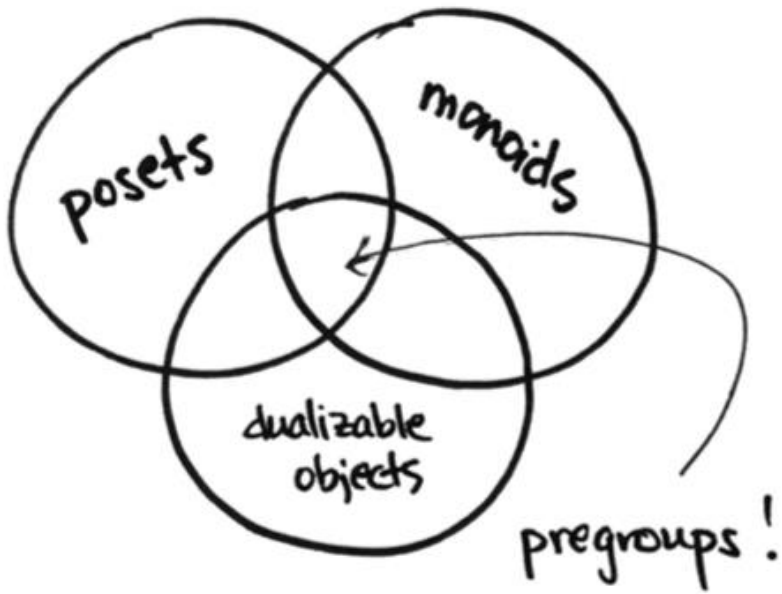
DisCoCatの言語理論の前提の一つは、自然言語が構成的な文法構造を持つことです。それは、通常は、Lambekのpregroup文法で記述されています。

DisCoCatとLambekのpregroup文法については、2022年12月のマルレク「ことばと意味の「構成性」について」 https://www.marulabo.net/docs/discocat と、その中の次の資料を参照ください。
https://www.marulabo.net/docs/discocat/#Pregroup_Grammar

pregroupの構造

pregroupは、基本的には、いくつかの代数的構造を持つ「半順序集合（partial order set: poset）」です。カテゴリー論的には、monoidal categoryの一種で、compact closed categoryになります。それは「半順序」より、複雑な構造を持っています。

poset + monoid + "duals" = pregroup



pregroupからpreorderへ

大規模言語モデルが、「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」とみなそうとする Tai-Danae にとって、pregroupという構造を前提にすることはできません。

彼女が選んだのは、pregroupがもつ余分な代数的な構造(monoidとself dual)を捨て、さらに残った半順序(partial order)からも、ある性質を捨てて得られる 非常にプリミティブな前順序(preorder)という構造から大規模言語モデルの振る舞いを数学的に再構築するという道でした。

その意味では、彼女は、テキストデータが、「全く構造化されていない」と考えたわけではありません。最低限の前提ですが、言語は、preorder という構造を持つのです。

preorderとは何か

順序≤が定義された集合Pで、Pに属する任意の要素 a, b, c が次の性質を持つとき、順序≤ はpreorder（前順序）と言われます。

1. $\forall a \in P.\ a \leq a$ （反射律）
2. $\forall a, b, c \in P. a \leq b$ で $b \leq c$ なら $a \leq c$ （推移律）

このpreorderの条件は、この二つの条件に加えて次の条件を満たすpartial order (半順序)より弱いものです。

3. $\forall a, b \in P. a \leq b$ かつ $b \leq a$ なら $a = b$

ちなみに、以上の条件に加えて $\forall a, b \in P. a \leq b$ または $b \leq a$ なる条件を満たす順序を total order（全順序）と言います。

語の並びとしての言語と preorder

先に、言語は preorder という構造を持つとしたのですが、この仮定が不自然なものではないことを見ることにしましょう。

ここでは、言語は語の並びだと考えることにします。実は、これも、正確に言うと言語の構造についての仮定なのですが、大規模言語モデルの入力に与えられるテキストデータは、どの言語のテキストであれ、語の並びと考えることはできそうです。

このとき、語の並びからなる二つの表現SとTがあるとき、SとTとの間の順序≤を次のように定義します。

- SがTの部分文字列である時、$S \le T$
- そうでない時、SとTの間には、順序関係は存在しない。

具体的な例で見てみましょう。

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

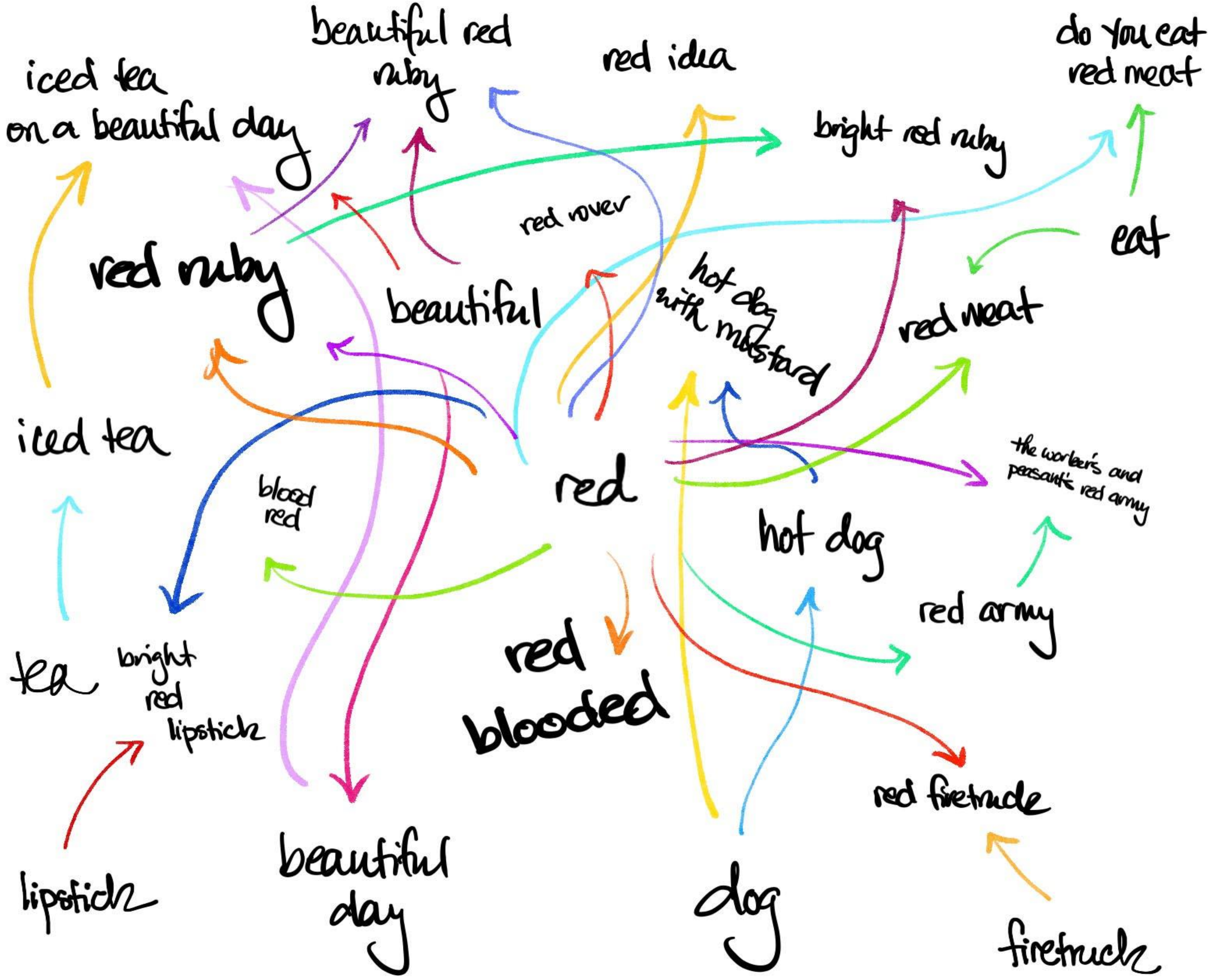
ドッグ　≤ カラシ付きのホットドッグ

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とTが、S ≤ T という順序関係がある時、S→ Tと、SからTに向かう矢印で表すことにします。次の図のように。



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
the worker's and peasant's red army
blood red
red
hot dog
tea
bright red lipstick
red army
red blooded
red firetruck
lipstick
beautiful day
dog
firetruck

「暗い空の下高い丘の上に一人の少女がいる。」という文章から連想できる文章を作ってみてください

A=“暗い空の下高い丘の上に一人の少女がいる。”
B="少女は星空を見上げて、心に願い事をした。"

C="彼女は高い丘の上から町の灯りを眺めていた。"

D="少女は深夜の風に髪をなびかせながら立っていた。"

E="彼女は静寂の中で自分の思考に耳を傾けていた。"

文Aと文Bを連続させて新しい表現を作ることをA+Bで表すと、次のようなpreorder があることがわかる。

$$A \leq A+B \leq A+B+C \leq A+B+C+D \leq A+B+C+D+E$$

preorderは、表現の継続・連続を表現する。

categoryとしての言語

大規模言語モデルの数学的構造 I

preorderとしての言語
前回のふりかえり

大規模言語モデルの入力に与えられるテキストデータを、「語の並び」からなる「表現」の集まりと考えることにしましょう。

二つの表現SとTがあるとき、文字列の包含関係に基づいて二つの表現SとTとの間の順序 ≤ を、次のように定義できます。

- SがTの部分文字列である時、S ≤ T
- そうでない時、SとTの間には、順序関係は存在しない。

この順序 ≤ は、反射律と推移律を満たしますので、preorder（前順序）です。

「表現」の集まりとしての言語は、preorderの構造を持ちます。

preorderとしての言語から categoryとしての言語へ

このセッションでは、前回見たpreorderの構造を持つ対象は、category としても考えることができると言う話をします。

まず、category とは何かを見ておきましょう。

categoryとは何か？
categoryを構成するもの

category Cは、次のものからできています。

- オブジェクト (object): Cを構成する要素
- 射 (morphism): Cの二つのオブジェクトx, yを結ぶもの。

この射fを $f: x \to y$ と表します。この時、xをfのdomain、yをfのcodomainと呼びます。

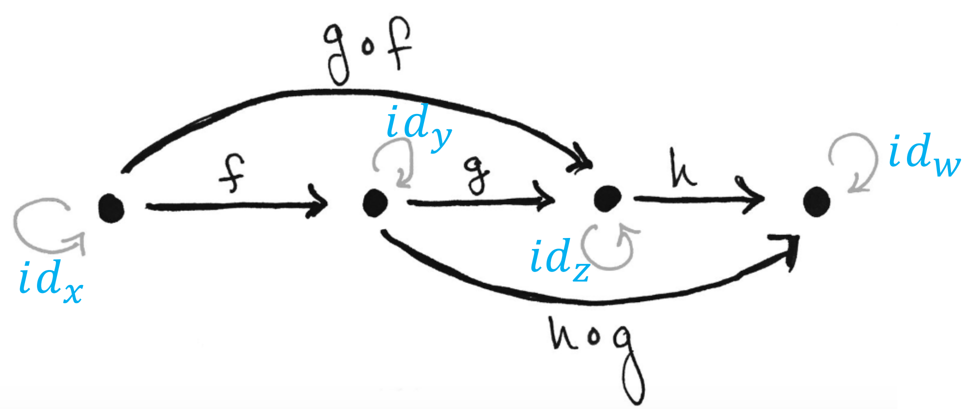
$f: x \to y,\ g: y \to z$ なる 射f, gに対して、$g \circ f: x \to z$ なる射が存在します。これを射 f, gの合成 (composition) といいます。

射 fのcodomainと射 gのdomainが一致する時、射の合成$g \circ f$ が定義されるということです。

ccategoryとは何か？
categoryが満たすべき性質

category C は、次の性質を満たさなければなりません。

- **同一射の存在**:
  Cのすべてのオブジェクト x について、xを同じxと結ぶ射
  $id_x : x \to x$ が存在する。
- **射の合成の結合性**:
  $f : x \to y$, $g : y \to z$, $h : z \to w$の時、$(h \circ g) \circ f = h \circ (g \circ f)$が成り立つ



preorderとしての言語はcategoryである

先に見た、preorderとしての言語 P がcategoryとしても解釈できることを見ていきましょう。

このcategoryとしての言語ををLとします。

Lのオブジェクトを、Pの要素と同じものとします。

Pの要素をS,Tとすると、S,Tは、Lのオブジェクトでもあります。それは言語の語の並びからなる表現です。

Lの射 $\mathrm{f}: S \rightarrow T$ を、Pで S ≤ T である時かつその時に限り、定義されるものとします。

Lのオブジェクトと射が定義できたので、このLがcategoryの要件を満たすことを見ていきましょう。

まずは、射の合成についてです。

Pはpreorderで推移律を満たしますので、Pの要素 S,T,Uについて、 S ≤ T かつ T ≤ U なら S ≤ U が成り立ちます。
S,T,Uは、Lのオブジェクトですので、このことは、Lにおいて f:S→T かつ g:T→U なら h:S→U が成り立つことを意味します。
このhを射 f,g の合成$g \circ f$ と解釈することができます。

次は、同一射の存在についてです。

Pは反射律を満たしますので、すべてのPの要素Sについて、S ≤ S が成り立ちます。このことは、すべてのLのオブジェクトSについて、同一射$id_S: S \rightarrow S$ が存在することを意味します。

最後は、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

$$\begin{tikzcd} S \arrow[r, phantom, "\leq"] \arrow[rr, bend left, no head, "\leq"] & T \arrow[r, phantom, "\leq"] \arrow[rr, bend right, no head, "\leq"'] & U \arrow[r, phantom, "\leq"] & V \end{tikzcd}$$

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

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

L

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

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

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

なぜ、言語をcategory として捉えるのか？

なぜ、「構造のないテキスト」の集まりに見える大規模言語モデルのテキスト・データを、あらためてcategory として捉える必要があったのでしょう？

プリミティブな順序構造であるpreorderdだけだと、大規模言語モデルが行なっているように見える言語の意味理解にはつながりません。それは、文のsyntaxを分析するだけでは、語や文のsemanticsの理解が進むわけではないのと同じです。

確かに、大規模言語モデルの理解には、pregroupの文法カテゴリーは強すぎるものでした。その前提を弱める必要がありました。

ただ、カテゴリー論的フレームを放棄する必要はないのです。

なぜ、preorderのcategory なのか？

順序が定義された集合は、preorder, partial order, total order そして pregroup といろいろあるのですが、それらはいずれも、categoryの言葉に翻訳できます。

今回、preorderがcategoryであることを説明する中で、改めて気づいたのですが、preorder は、categoryを構成する順序集合の中で、もっとも仮定が少ないものになっています。

なぜ、category論が重要なのか？

pregroupという文法的categoryとLawvereのFunctorial Semanticsのフレームを利用して、syntaxの構成性とsemanticsの構成性を結びつけることに成功したDisCoCatの理論は、画期的な飛躍だったと思います。

それは、同時に、言語とその意味の理解には、カテゴリー論が極めて有用であることを示しました。

今回のセミナーで紹介するTai-Danaeの理論は、言語とその意味の理解にとどまらず、現代の人工知能技術の中核である大規模言語モデルの理解にとっても、カテゴリー論が強力なツールを提供できることを意味しています。

Tai-Danaeの素晴らしいアイデア
意味を導くのに「米田レンマ」を使う

しかも、その方法は、意想外（僕にとっては）のものでした。

彼女は、プリミティブなpreorderで特徴づけられる言語の構成性を表現するcategory L から、意味の世界へのジャンプを、一般的なFunctorial Semanticではなく、「米田レンマ」を使うことで成し遂げるのです。

この彼女のアプローチを特徴づける素晴らしいアイデアを、次回のセッションでは紹介しようと思います。

functor:
言語のcategoryと意味のcategoryを結ぶ

大規模言語モデルの数学的構造 I

意味の世界はどこにある？

大規模言語モデルは、言語の意味を理解しているように見えます。それでは、大規模言語モデルが理解しているように見える「意味の世界」はどこにあるのでしょう？

先のセッションでは、言語を語の並びである表現の集まりだと考えれば、文字列の包含関係で preorder の順序が定義でき、それは category としても考えられるという話をしました。

ただ、この言語のcategory L をいくら眺めても、そこにあるのは、具体的な文字列や語や表現だけで、どこにも意味は見当たりません。

意味を表現する方法を考える

これまでの流れで考えてみると、言語のcategory Lとは別に、言語の意味を表現するcategory M (meaningのMです）が存在すると考えるのが自然なアプローチだと思います。

ただ、この二つのcategory L, Mは、別々バラバラなものではなく結びついていて、言語のcategory Lが与えられるとその言語の意味のcategory Mが生み出されるという関連があるはずです。

イメージだけですが、こんな感じです。

言語のcategory　　　　意味のcategory

$$L \longrightarrow M$$

言語のcategoryと意味のcategoryを結んで言語の意味を表現する

もし、意味のcategory Mがうまく定義できるなら、言語のcategory Lと意味のcategory M を結びつけることで、言語の意味の表現ができそうです。

もっとも、この段階では「LとMを結びつけられたらいいかも」と言っているだけで、意味のcategory Mがどんなものかは何もわかっていません。意味のcategory Mをどう構成すればいいかは、次回のセッションでもう少し具体的に触れていきたいと思います。

今回のセッションでは、「category とcategoryを結ぶ」ということを、考えてみたいと思います。

functor
categoryとcategoryを結ぶ

カテゴリー論では、あるcategoryともう一つのcategoryを結びつける方法がキチンと定義されています。それをfunctor と言います。

言語の意味を表現するのに、カテゴリー論的枠組みを使うのなら、functorを利用することが必要になりそうです。

ここでは、functorと二つのfunctorを結びつけるnatural transformationの定義を見ておくことにします。

functorとはなにか？

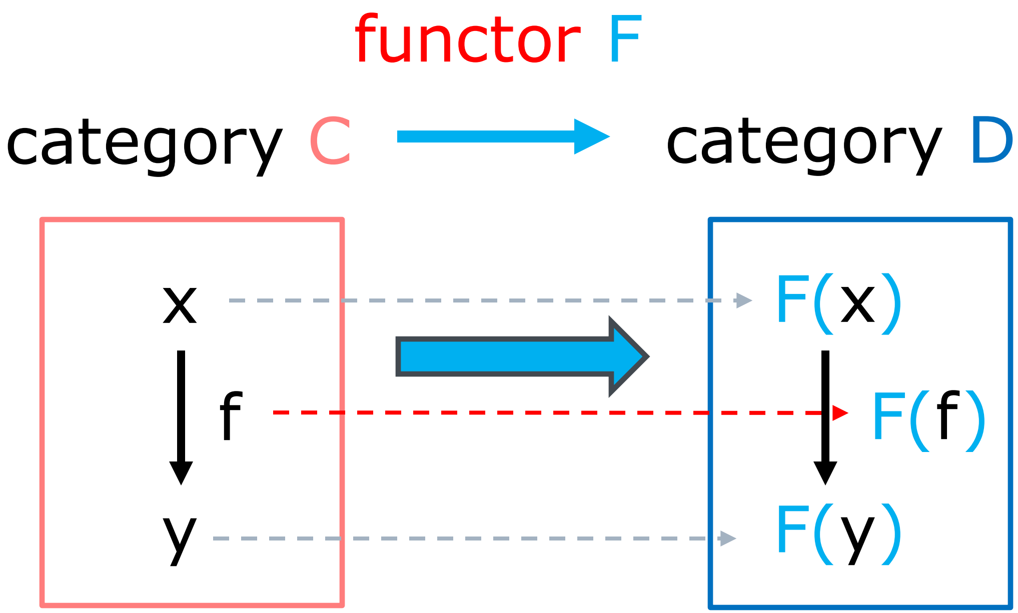
category C から category D へのfunctor F: C → D は、次のように定義されます。

まず、Cのオブジェクトと射について、functor Fがどう作用するかです。

- Cのすべてのオブジェクトx について、F(x)はDのオブジェクトである。
- Cのすべての射 f : x → y について、F(f)はDの射 F(f) : F(x) → F(y) である。

要するに、functor Fによって、CのオブジェクトはDのオブジェクト F(x)にうつされ、Cの射 fはDの射 F(f)にうつるということです。

図で表すと、次のようになります。



functorは、次の条件も満たす必要があります。

- 射 g, f がCで合成可能な時、すなわち、射 $g \circ f$ が C に存在する時、$F(g \circ f) = F(g) \circ F(f)$。
- FはCのすべてのオブジェクトxについて、Cの同一射をDの同一射にうつす。$F(id_x) = id_{F(x)}$ 。

natural transformation

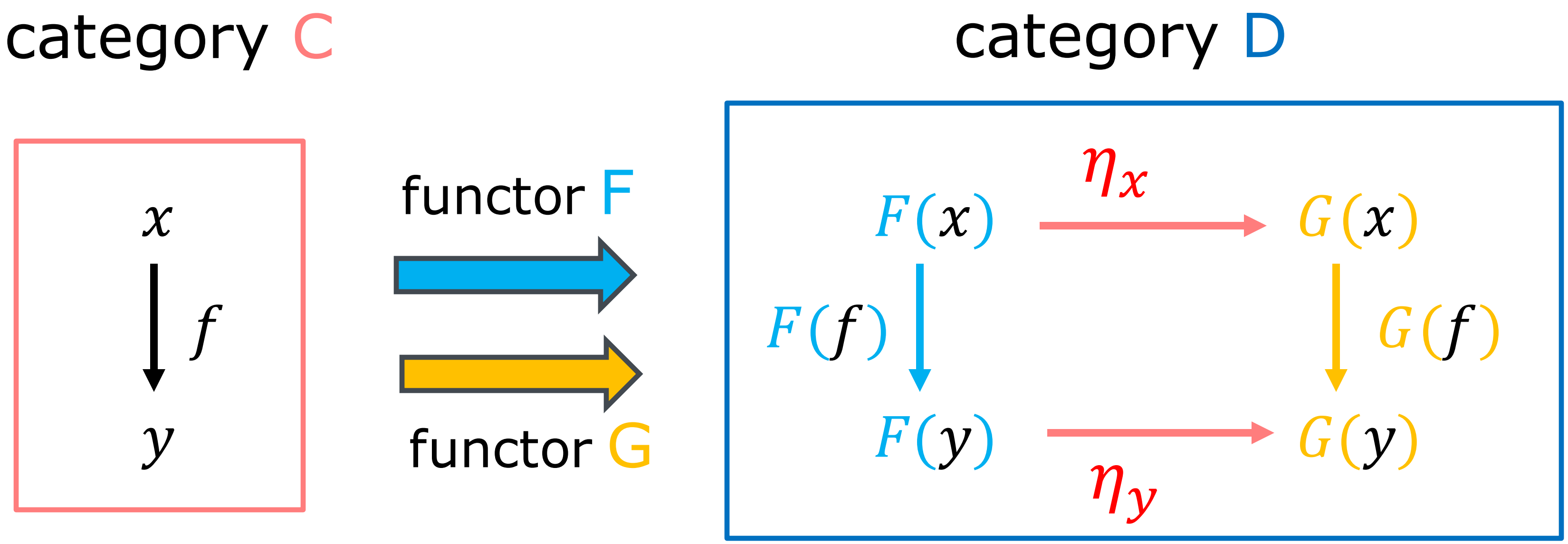
二つのfunctor F,Gがあって、F,Gともに、category Cから category DへのfunctorだとしますI。

この時、次の条件を満たすη(エータ)を、FからGへのnatural transformation と言います。

- Cのすべてのオブジェクトx について、η_xはF(x)からG(x)の射である。$\eta_x : F(x) \to G(x)$
- $f : x \to y$ がCの射である時、次の式が成り立つ。$G(f) \circ \eta_x = \eta_y \circ F(f)$

natural transformationの働きを
可換図式で表すと、次のようになります。

functor F, G はともに、
category Cからcategory D
へのfunctor



$$G(f) \circ \eta_x = \eta_y \circ F(f)$$

Part 4

意味をカテゴリーとして捉える

大規模言語モデルの数学的構造 I
言語へのカテゴリー論的アプローチ入門

Part 4 意味をカテゴリーとして捉える

Agenda

- functor category と意味
- Yoneda embeddingと意味の表現
- 言語のcategoryに確率を導入する -- enriched category

functor category と意味

大規模言語モデルの数学的構造 I

これまで見てきたいくつかの意味論

意味の世界にアプローチする手がかりは、どこかにないでしょうか？　特に、その意味の世界をカテゴリーとして捉えるヒントはどこかにないでしょうか？

このセミナーの中では、意味の世界をカテゴリーとして捉えようとする二つの理論 DisCoCatとQNLPを紹介してきました。そこでは意味の世界は、有限ベクトル空間あるいはヒルベルト空間のカテゴリーとしてモデル化されていました。

また、それらの理論では言語のカテゴリー Lは、pregroup文法に従うものとして、高度に構造化されたものだとされていました。

ただ、それらは、「構造を持たないテキスト」をその入力とするように見える大規模言語モデルの数学的モデルとしては、使いにくいものでした。

改めて、DisCoCatやQNLP以前の、まだカテゴリー論化されていない意味の理論を振り返ってみましょう。

ここでは、次の二つの意味の理論を、振り返ってみましょう。

- 意味の使用説:Wittgenstein
- 意味の文脈依存説:Firth

意味の使用説：Wittgenstein

意味の使用説とは、意味はその使用から説明されるべきだ、という考え方である。Wittgensteinに始まる。

“meaning of a word is its use in a language”

「ある語の意味は、ある言語におけるその使用である」

TuringのWittgenstein批判

チューリングは、言葉の意味を知る事は、その用法を知る事だといったヴィトゲンシュタインらの見解に痛烈な皮肉をあびせている。

すなわち、「機械」や「考える」という言葉の使い方をいくら調べた所で「機械は考える事ができるか」という問の意味も答えも明らかになるわけではない。それとも、「ギャラップの世論調査の様な統計的研究」が必要という事になるのだろうかと.

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。」

Firthの解釈をベースに意味を考えてみよう

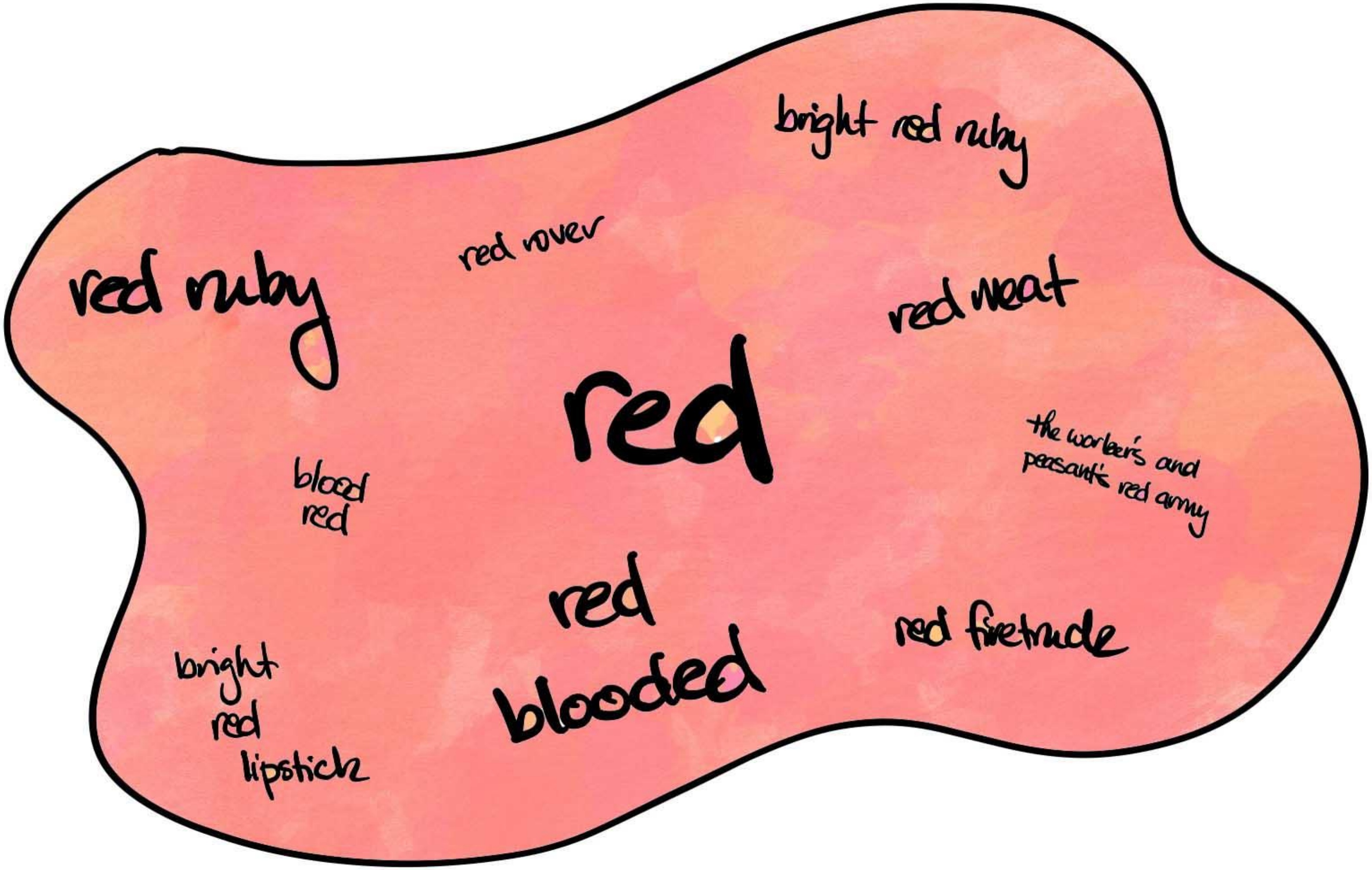
このセッションでは、Firthの解釈をベースに意味を考えてみます。

例えば、語 red が与えられた時、red の「仲間たち」は、red という語を含んだ表現の全体だとしましょう。

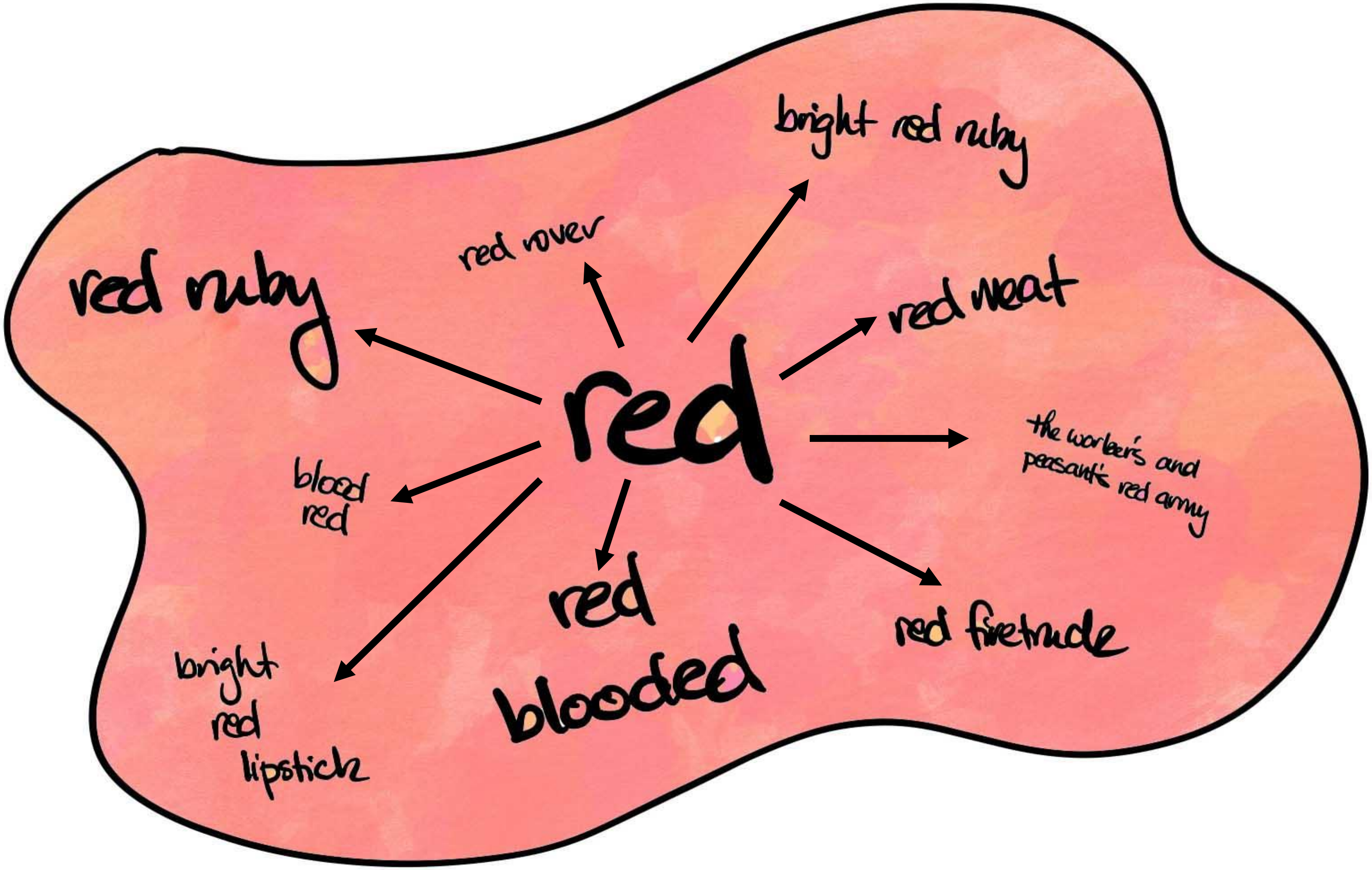
言語には、部分文字列の包含関係で preorder ≦が定義できます。category化された言語 L は、このpreorder ≦ を、category Lの射 → とみなしたものです。

category L では、red の「仲間たち」とは、red を始点として、L の射 →　でつながるすべての表現のことです。

例: “red” とその仲間たち（company）



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



表現の意味を、
その表現の「仲間たち」の「全体」と考える

ここで、ある言語における「表現」（語の並びのことです）の「意味」を、その表現の「仲間たち」のなす「全体」の集まりと考えることにします。

言語のcategory L でこの考えを整理してみましょう。（カッコの中で、カテゴリー化される以前の解釈をしています。）

LのオブジェクトS,T(SもTも語の並びとしての表現です)で、TがSの「仲間」になるのは、S → T なる射が category L に存在する時、かつその時に限ります。（SがTの部分文字列だということです。）

ある条件を満たす表現の全体を考える

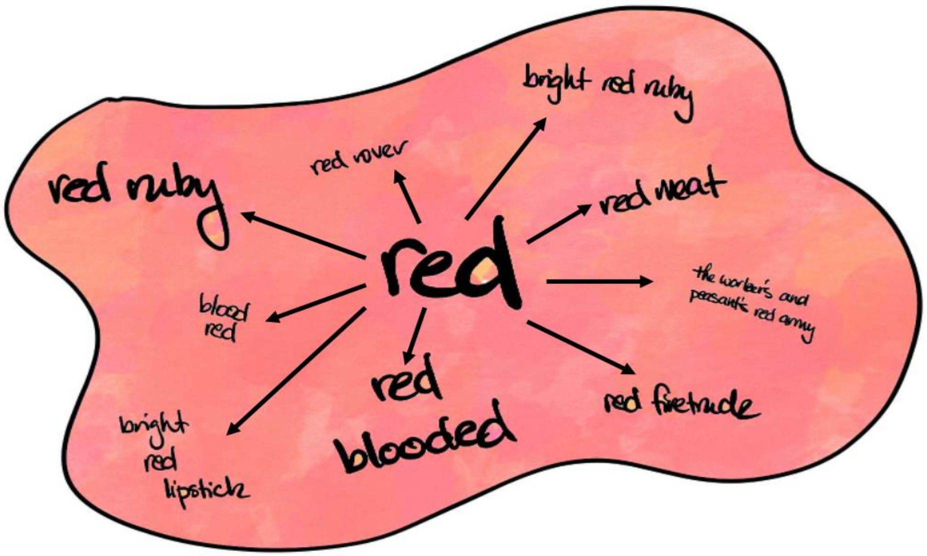
この時、Sと「仲間である」という条件を満たすLの表現全体の集まりを、Sの「意味」と考えるということです。

もう少し例を挙げて説明してみましょう。

category Lのオブジェクトである red という表現をとり、Lの中の任意のオブジェクトy を考えます。Lの射で、red → y という射が存在すれば、そうしたオブジェクトy全体の集合を考え、その集合が、red の意味だと考えるのです。

“red” の意味を表す集合

先に見たredの意味を表す集合は次のように定義されます。



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

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

上の図から。redの意味を表す集合は、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。

意味のcategory Mのオブジェクト

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xだけからなるわけではなく、xを含むcategory L の多数のオブジェクトからなる「集合」です。

集合は、その要素をオブジェクトとし、集合の要素間に定義される関数を射と考えると、容易に、それがcategoryの要件を満たすことを示すことができます。集合のcategory をSetで表すことにします。

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、集合のcategory Set のある要素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、

$$L(x, -) : L \to Set$$

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。

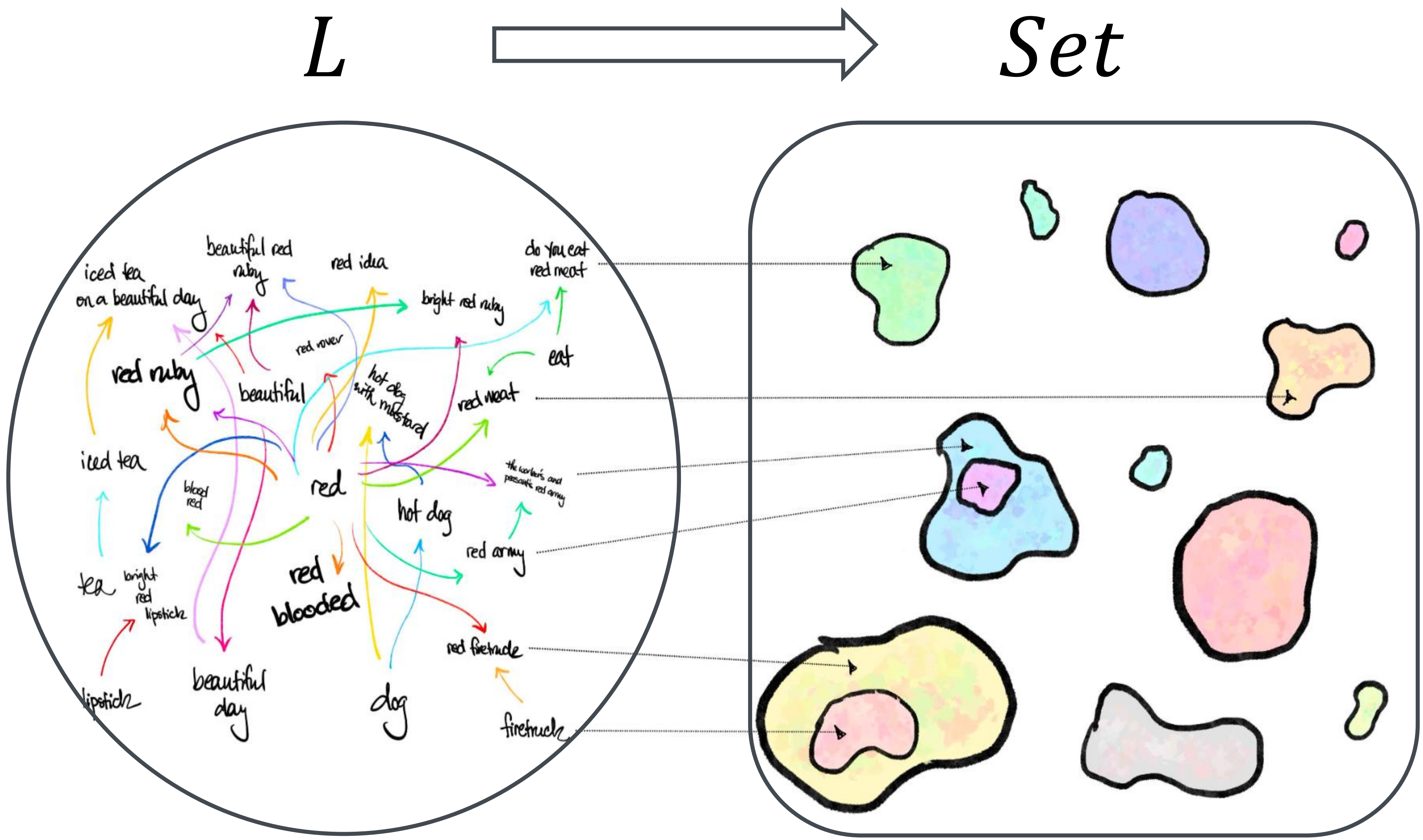
一般に category C から category D へのfunctor から構成されるcategory を、functor category といい、D^Cで表します。（ [C, D] という表記を使うこともあります。）

- 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。

functor category Set^L のイメージ



category L *category Set*

Yoneda embeddingと意味の表現

大規模言語モデルの数学的構造 I

ふりかえり
意味のcategory M の形

前回のセッションで、意味のcategory M の形を見てきました。

それは、言語のcategory L のオブジェクト x から、Lの射 x → y によってうつされるすべてのオブジェクト yの集合によって表現されます。

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

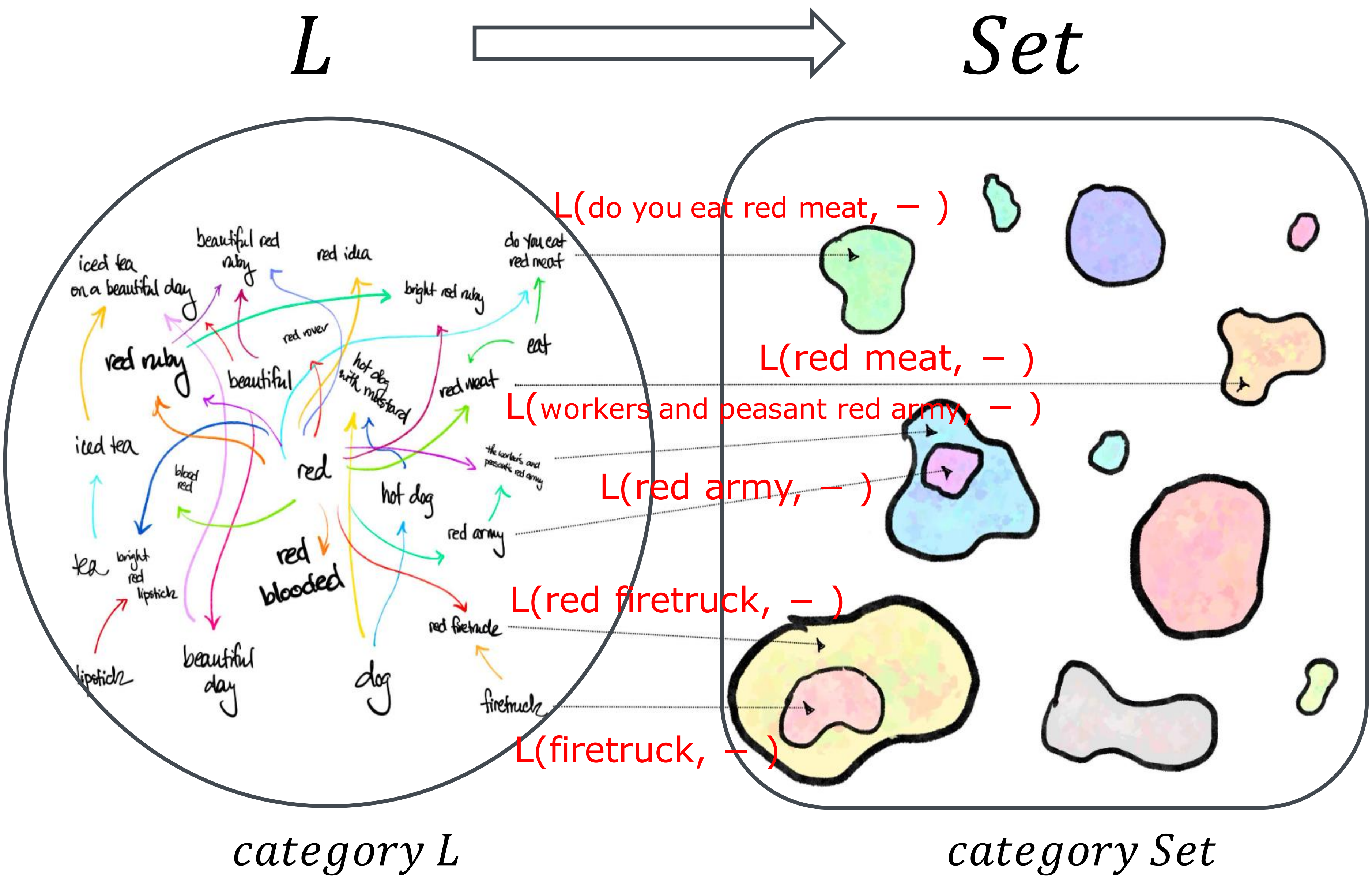
と表すことができます。

意味のcategory M のオブジェクトは、この

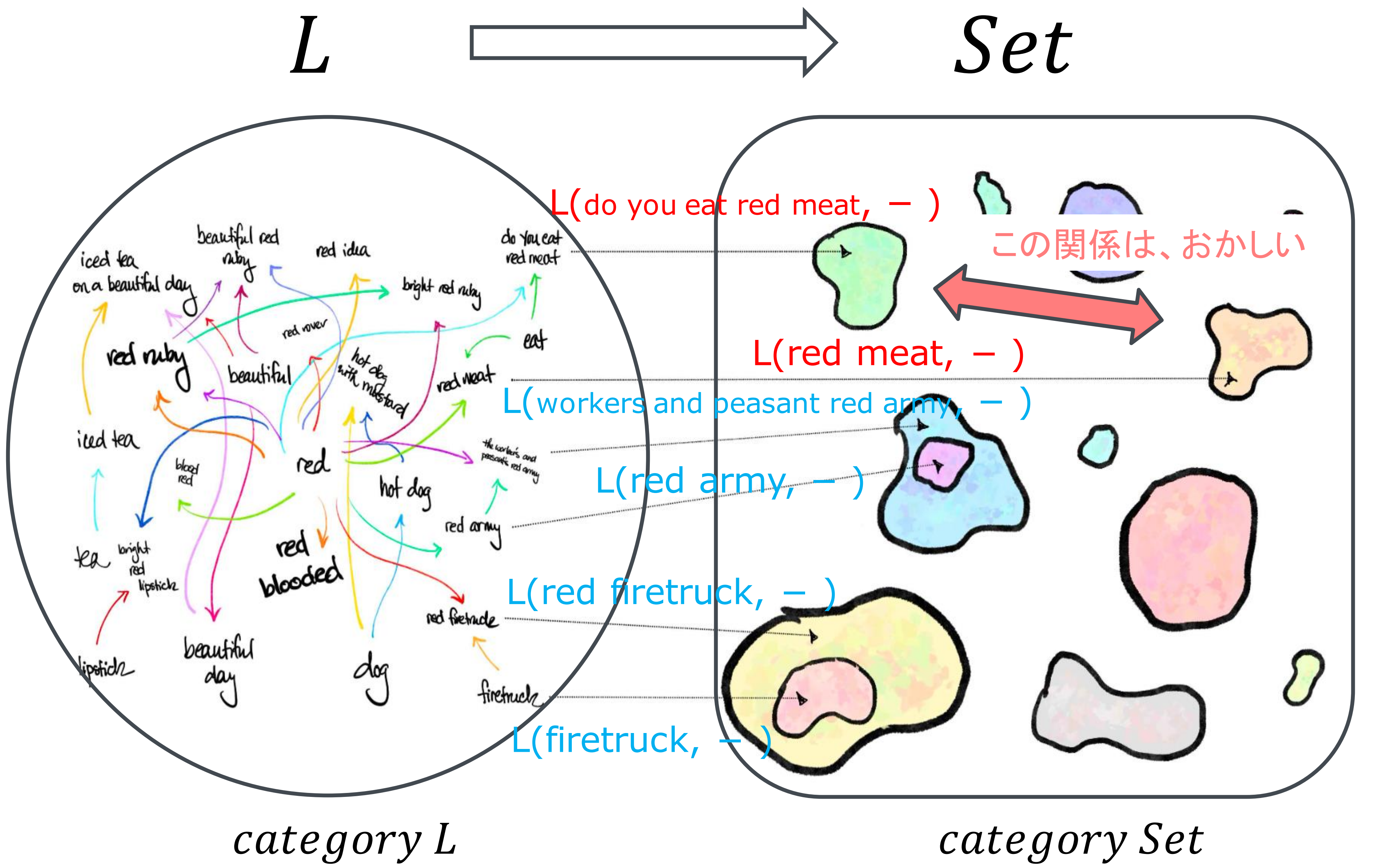
L(x, −) : Set^L

に他なりません。

Set^L のオブジェクト $L(x, -)$ の例



Set^L のオブジェクト $L(x, -)$ の例



もとの問題意識に戻る

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。

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

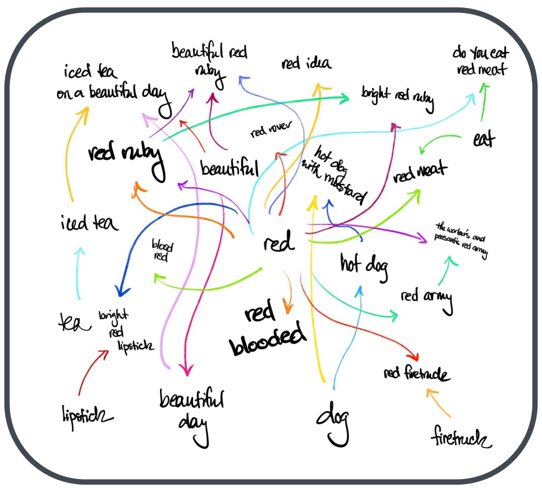
Mが定義できたので、これが今度はこうなります。

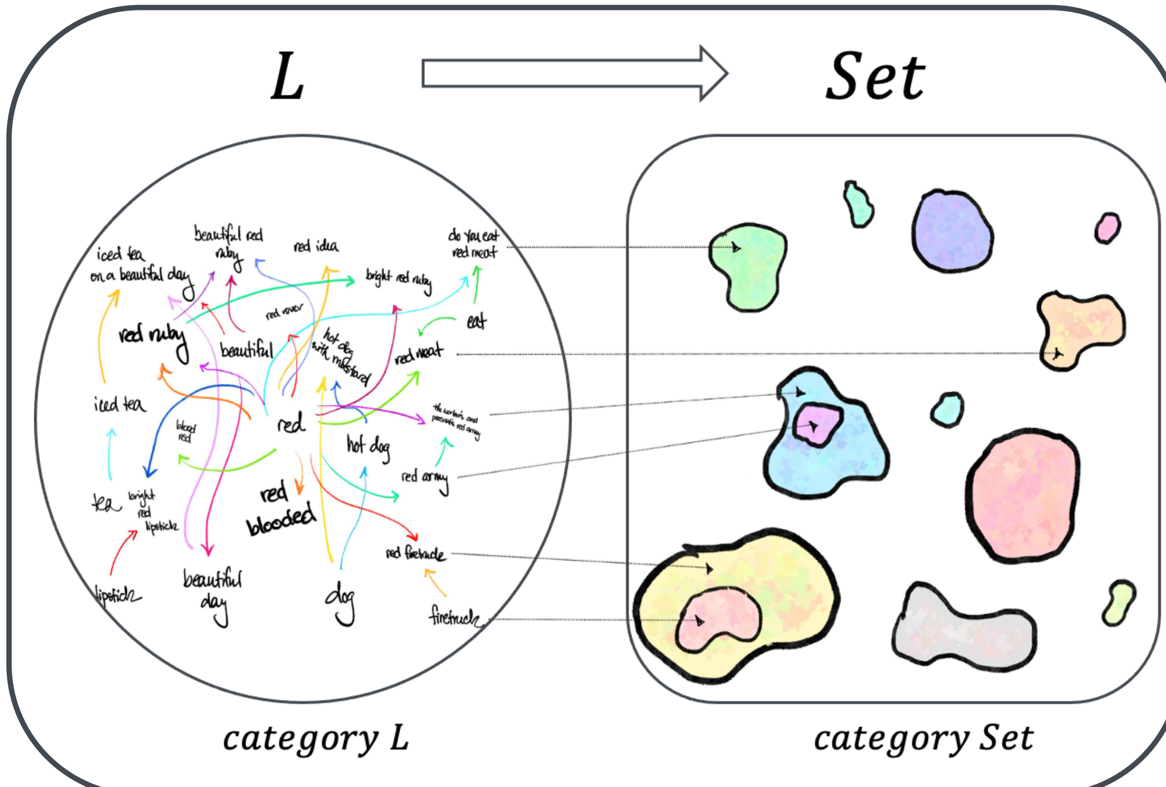
$$\underset{\text{言語のcategory}}{L} \longrightarrow \underset{\text{意味のcategory}}{Set^L}$$

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

$L \Longrightarrow Set^L$





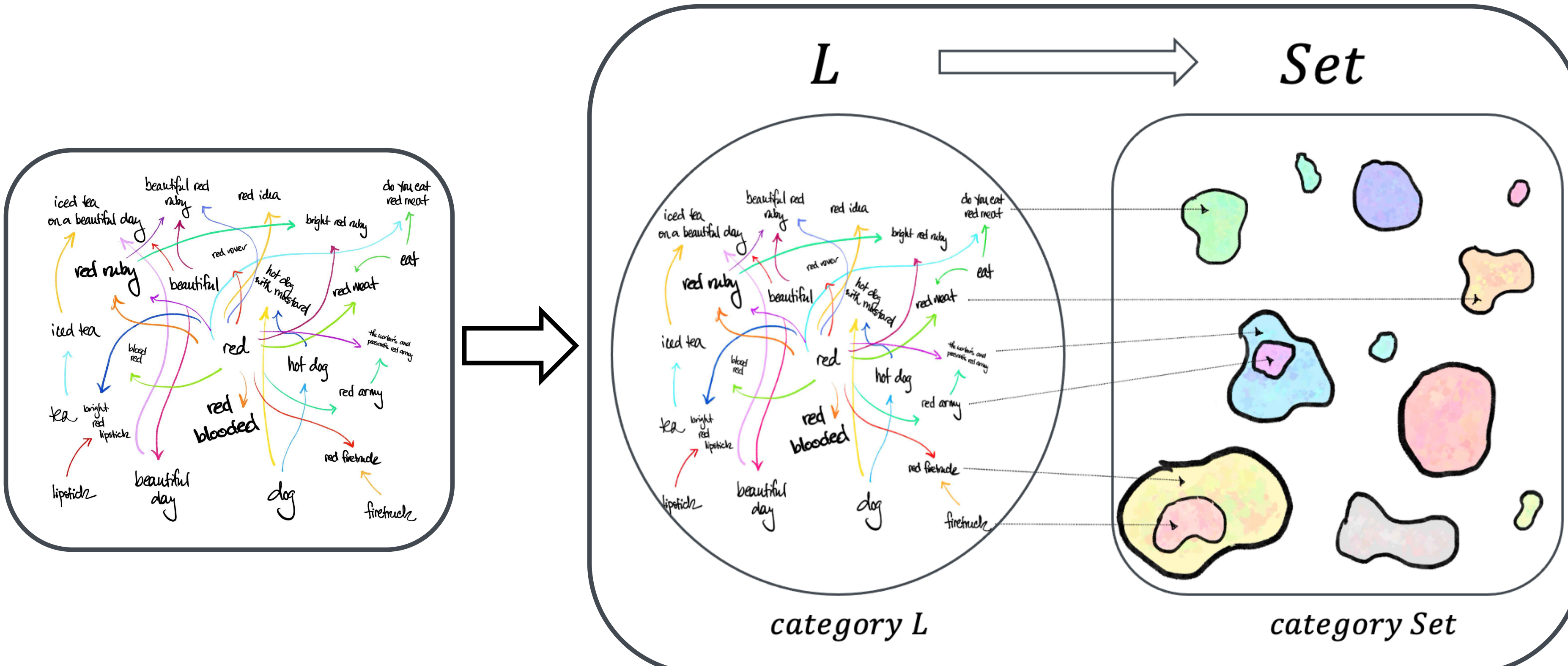
言語の
$category\ L$

$\Longrightarrow$

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。

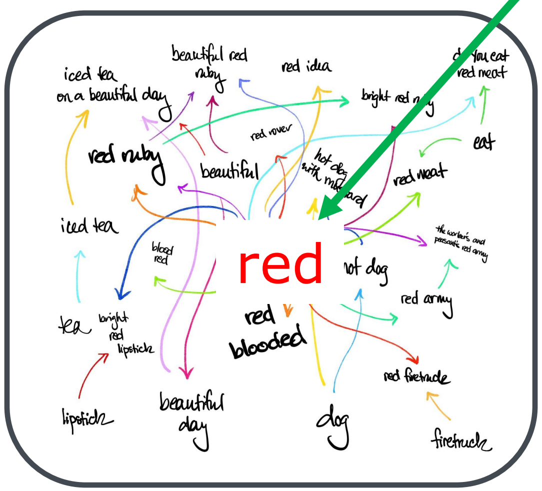
$$L \Longrightarrow Set^{L}$$

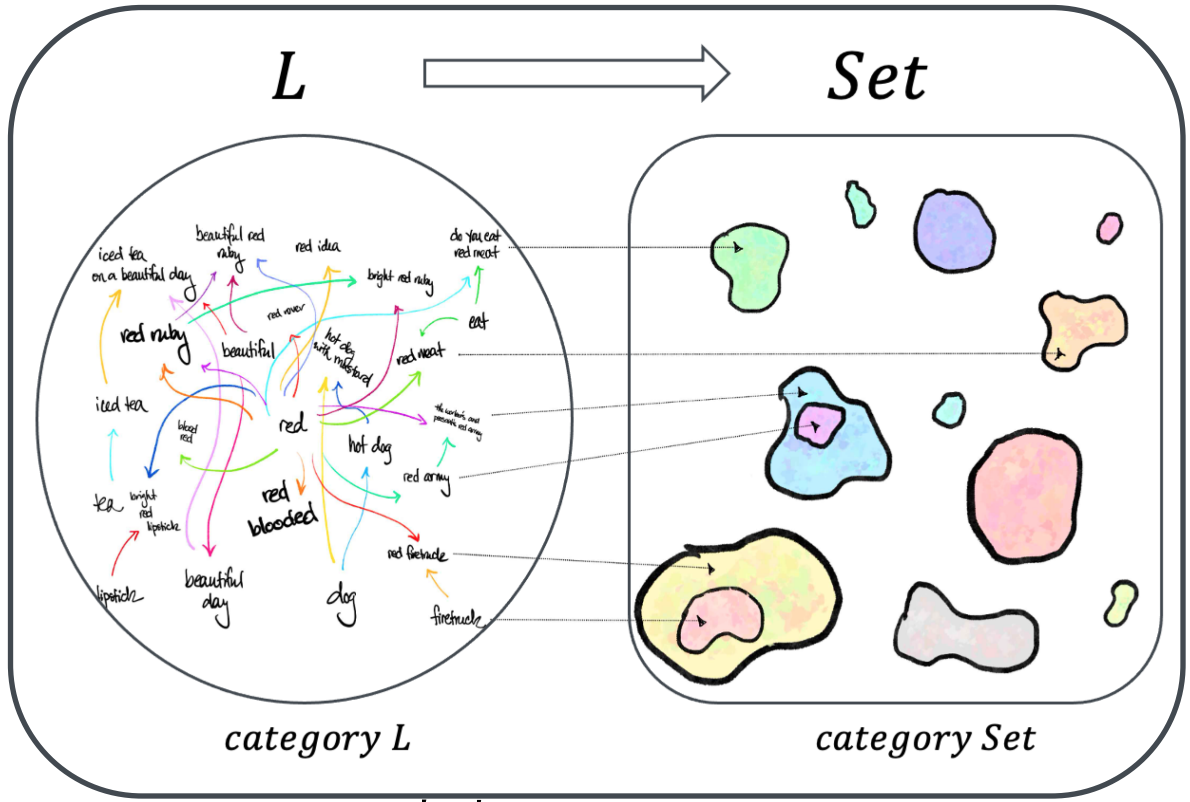


言語の
category L $\Longrightarrow$ 意味の
category Set^{L}

Lのオブジェクト red を一つ選びます。

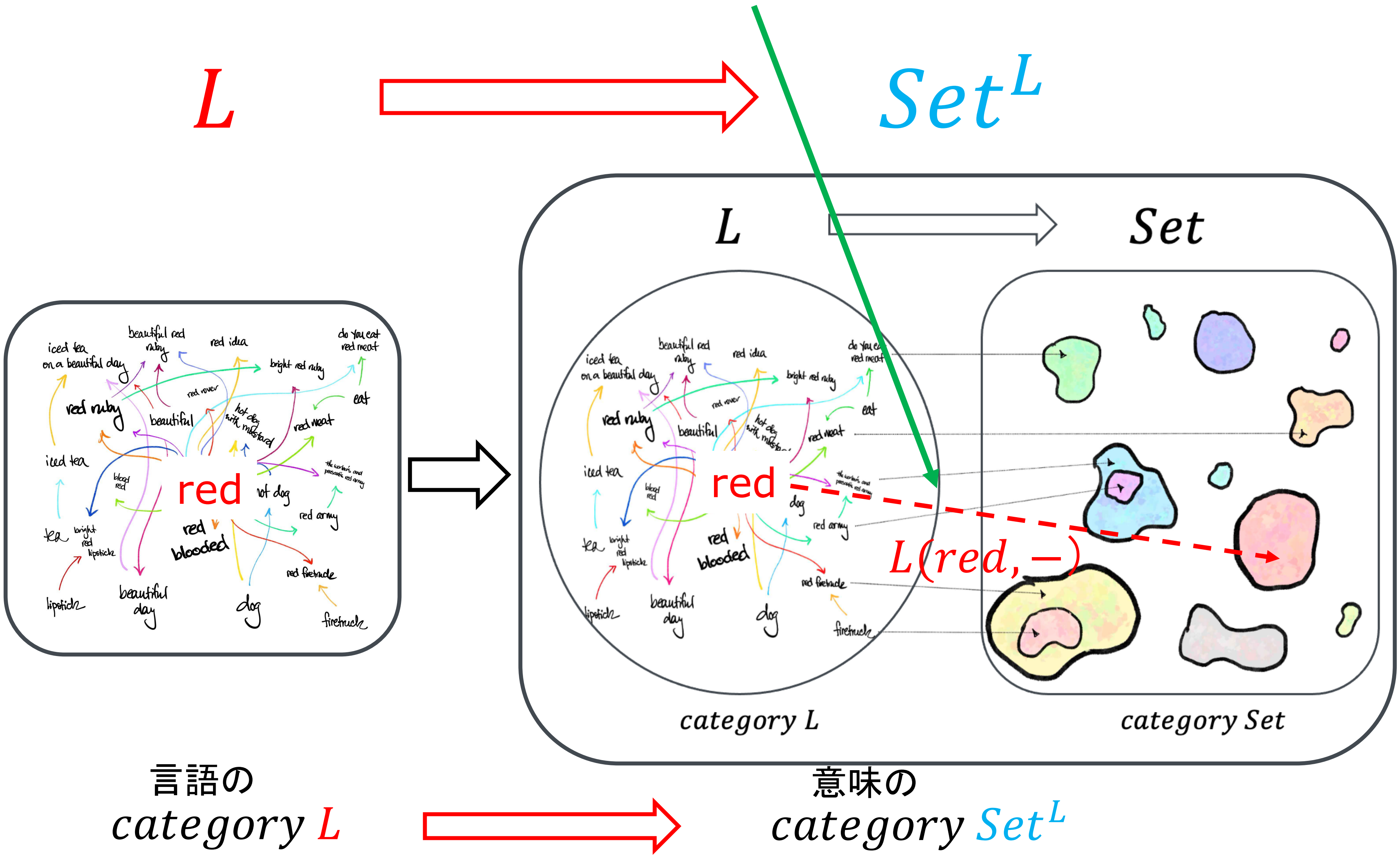
$L \Longrightarrow Set^L$



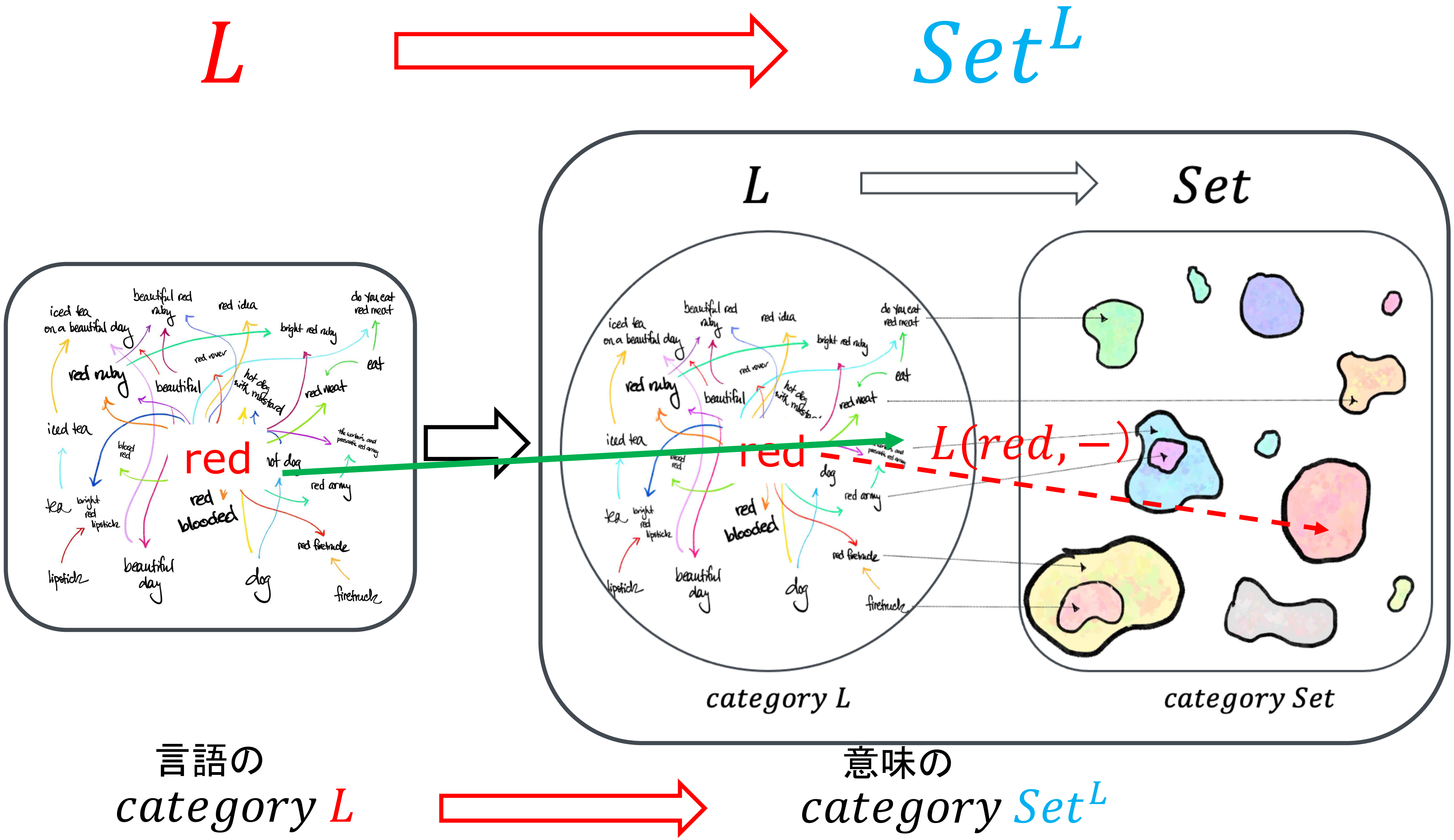


言語の
$category\ L$ $\Longrightarrow$ 意味の
$category\ Set^L$

Set^Lのオブジェクト $L(red,-)$ を考えます。

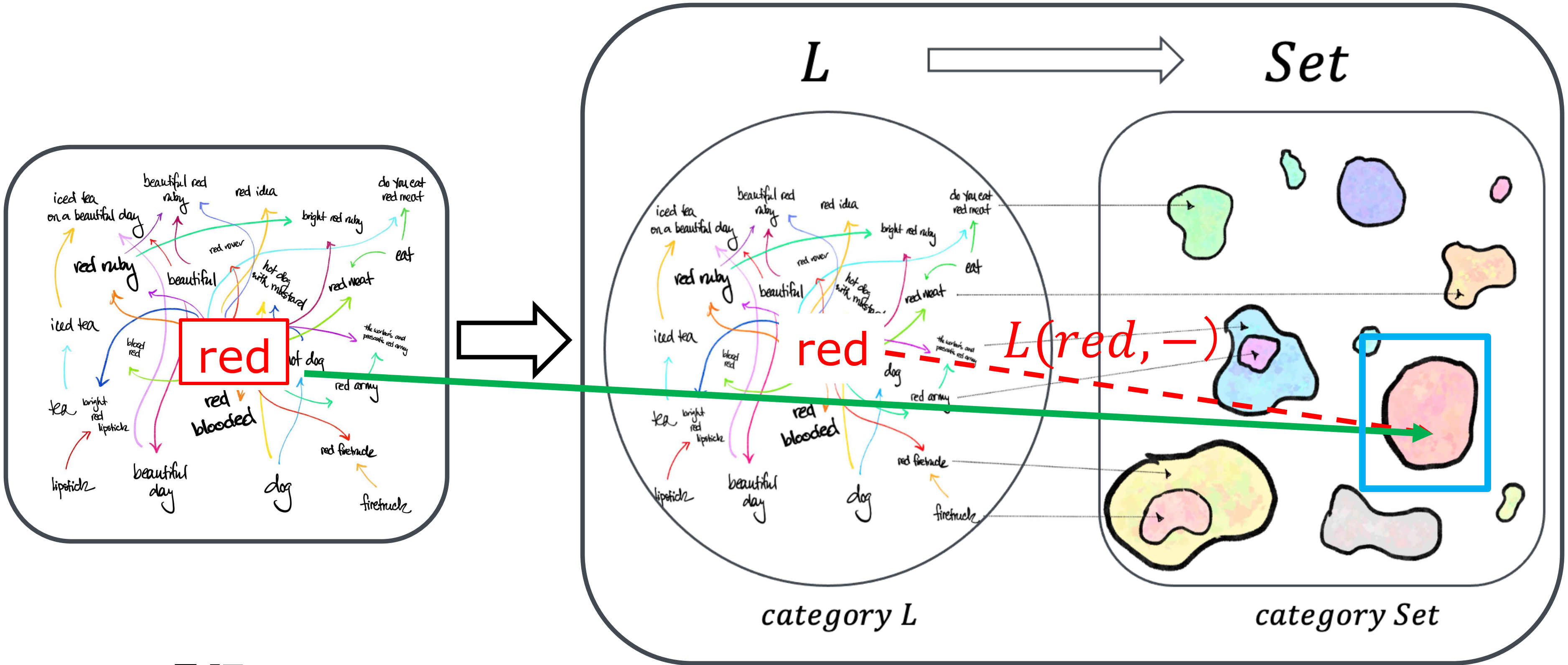


category L とcategory Set^Lの対応は、
red と$L(red,-)$ を対応づけるものです。



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

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

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

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

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

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

$$F : C \to Set$$

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 $A \to B$ 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。

$$\mathrm{Hom}(A, -) : C \to \mathrm{Set}$$

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。

$$\begin{array}{ccc} X & & \mathrm{Hom}(A,X) \\ \downarrow f & \underset{\mathrm{Hom}(A,\,-)}{\Longrightarrow} & \downarrow \ \mathrm{Hom}(A, f) \\ Y & & \mathrm{Hom}(A,Y) \end{array}$$

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。$\mathrm{Hom}(-,\mathrm{B}) : C^{op} \to Set$

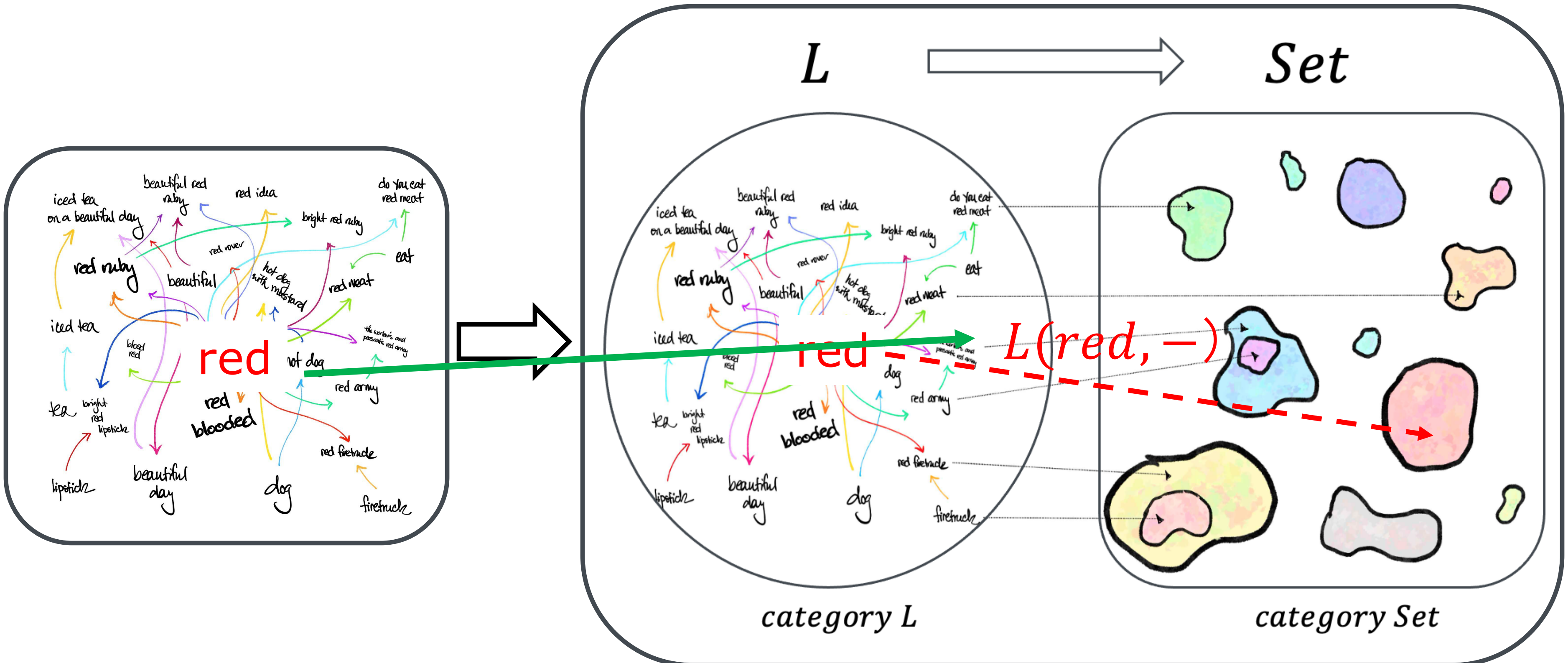
$F: C^{op} \to Set$ なるすべてのfunctor FとCのすべてのオブジェクトXについて、natural transformation $\mathrm{hom}(-, X) \to F$ は、集合$F(X)$の要素と一対一に対応する。

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

セッションのまとめ

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

$$L \Longrightarrow Set^L$$



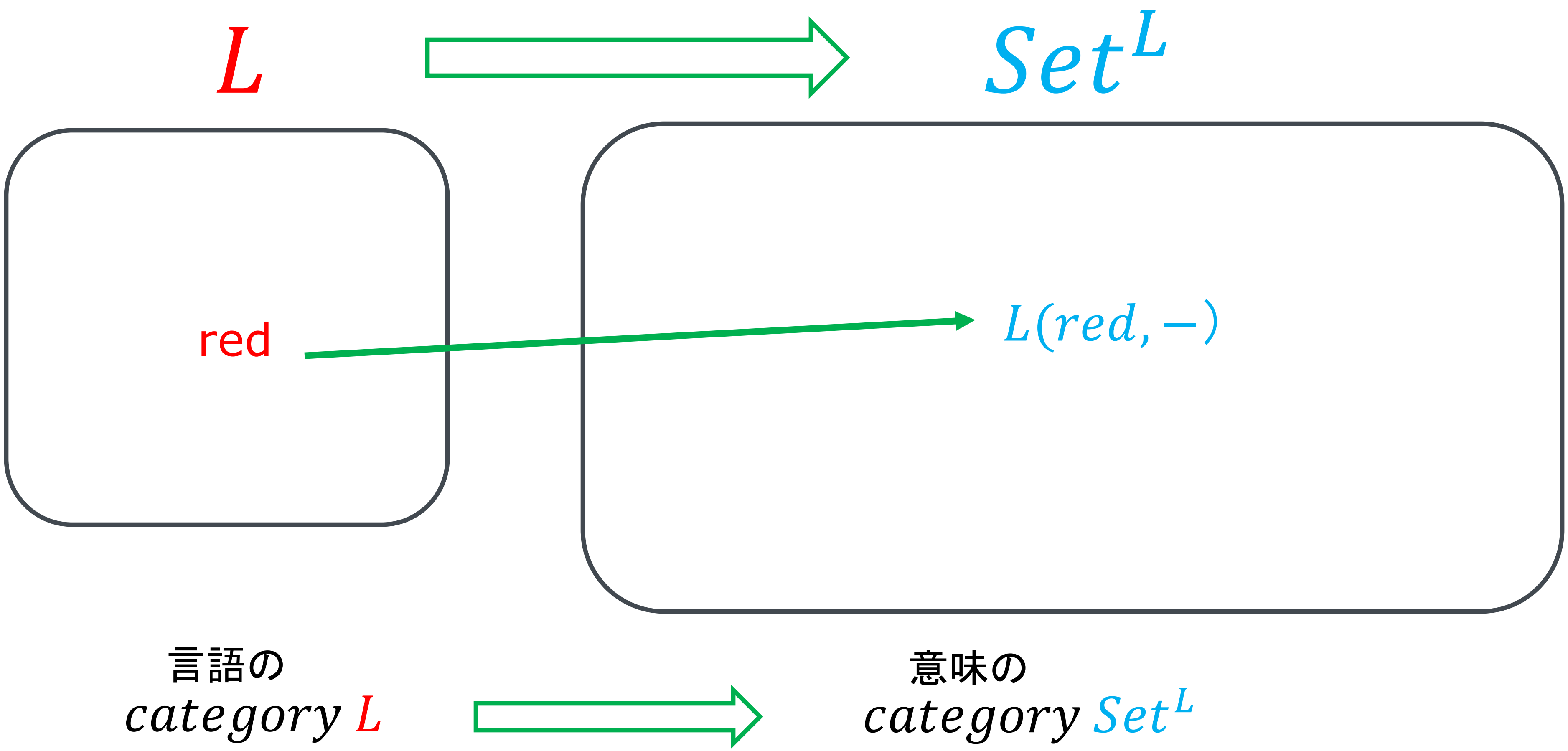
言語の
category L

$\Longrightarrow$

意味の
category Set^L

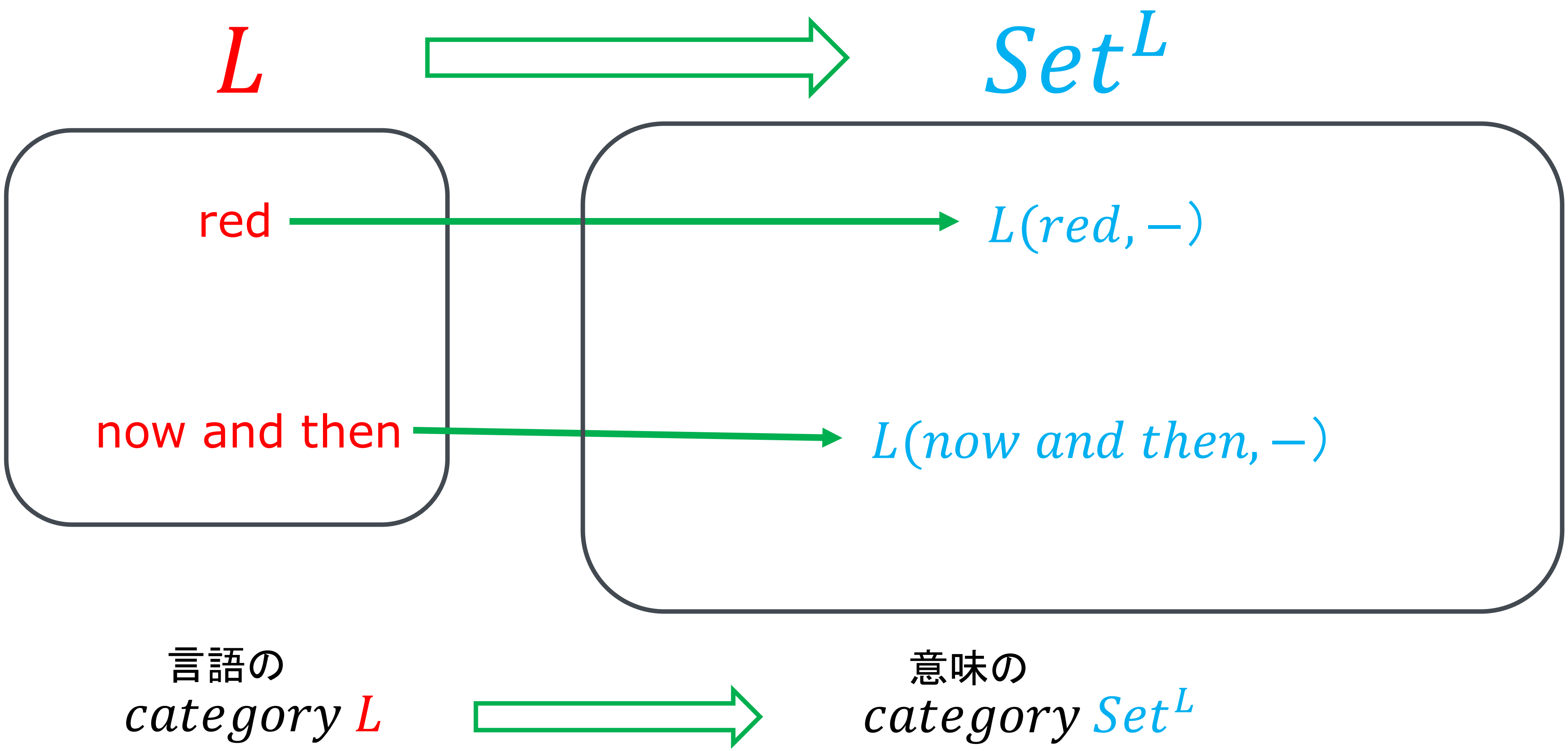
セッションのまとめ

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。



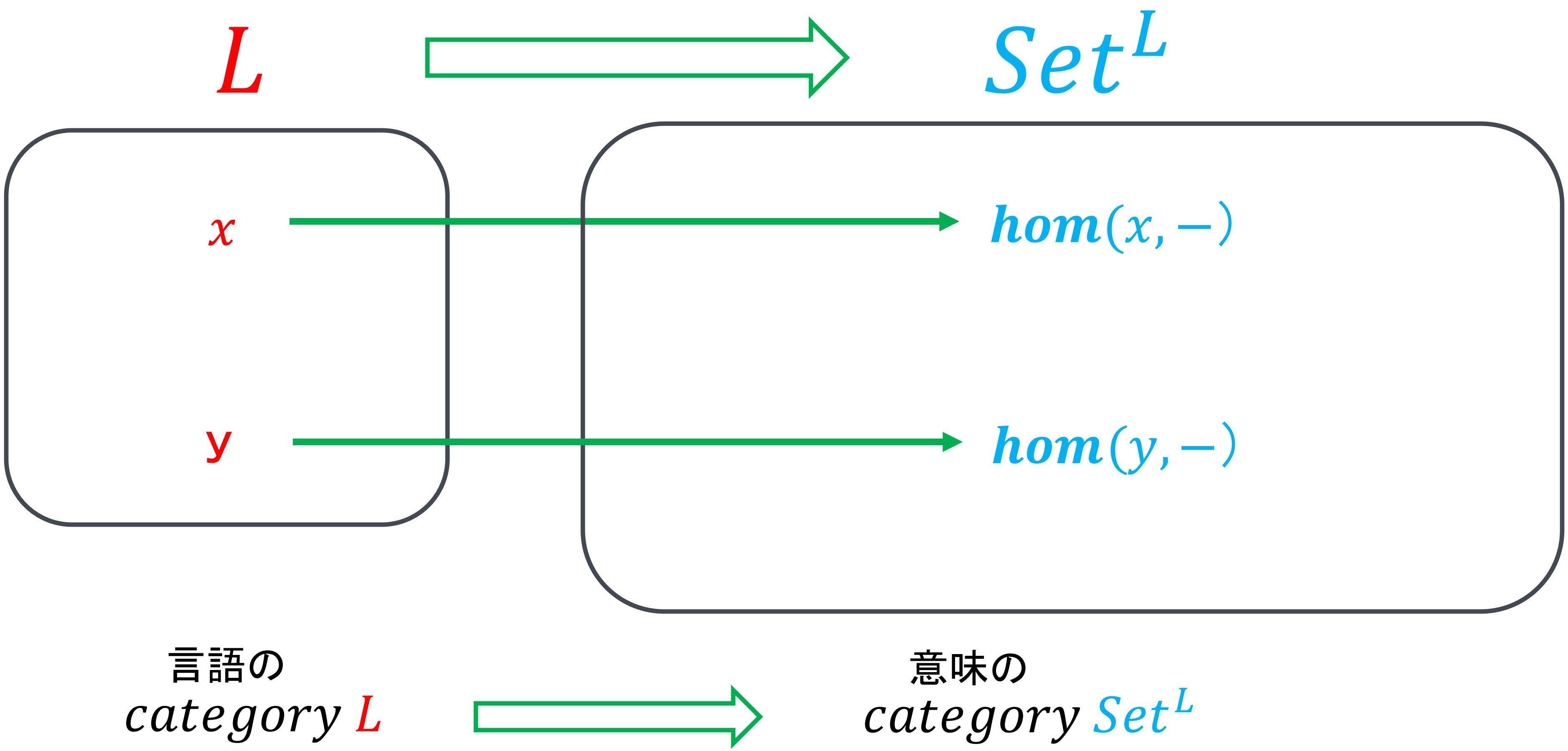
セッションのまとめ

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。



セッションのまとめ

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

THE YONEDA LEMMA

Mac Lane

Categories for the Working Mathematician

Noboru Yoneda is rightly honored for his well known lemma: if F is a functor from a category C to Sets, then the natural transformations from hom(c, -) to F corresponds by a bijection to the set F(c).

$$\mathrm{Nat}(\mathrm{Hom}(c, -), F) \cong F(c)$$

THE YONEDA LEMMA

Mac Lane

Categories for the Working Mathematician

Yoneda enjoyed relating the story of the origins of this lemma, as follows. He had guided Samuel Eilenberg during Eilenberg's visit to Japan, and in this process learned homological algebra. Soon Yoneda spent a year in France (apparently in 1954 or 1955).

There he met Saunders Mac Lane. Mac Lane, then visiting Paris, was anxious to learn from Yoneda, and commenced an interview with Yoneda in a cafe at the Gare du Nord. The interview was continued on Yoneda's train until its departure. In its course, Mac Lane learned about the lemma and subsequently baptized it.

THE YONEDA LEMMA

SAUNDERS MAC LANE

Received December 1, 1996

Noboru Yoneda is rightly honored for his well known lemma: if F is a functor from a category C to Sets, then the natural transformations from $\hom(c,-)$ to F corresponds by a bijection to the set $F(c)$.

$$\mathrm{Nat}(\mathrm{Hom}(c,-),F) \cong F(c)$$

Yoneda enjoyed relating the story of the origins of this lemma, as follows. He had guided Samuel Eilenberg during Eilenberg's visit to Japan, and in this process learned homological algebra. Soon Yoneda spent a year in France (apparently in 1954 or 1955). There he met Saunders Mac Lane. Mac Lane, then visiting Paris, was anxious to learn from Yoneda, and commenced an interview with Yoneda in a café at the Gare du Nord. The interview was continued on Yoneda's train until its departure. In its course, Mac Lane learned about the lemma and subsequently baptized it.

Yoneda made other important contributions to homological algebra. The functor Ext.(G,A) had been defined in terms of short exact sequence $A \to X - G$; In 1954, he showed that the related function $\mathrm{Ext}^n(G,A)$ could be defined by long exact sequences (in the J. Fac. Sci. Tokyo, Sec. 1,7 pp.193-227; subsequently he showed that the products here could be given by composition of such sequences. He was the first to formulate the notion of an "end" of a bifunctor, in a 1960 paper in the same journal (vol. 8, 507-526). This notion has been widely used, as by Day and Kelly and by Mac Lane. In Short, Yoneda has made decisive contributions to albegra.

We mourn his recent death.

言語のcategoryに確率を導入する
enriched category

大規模言語モデルの数学的構造 I

言語のcategoryに確率を導入する

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。

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

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

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。

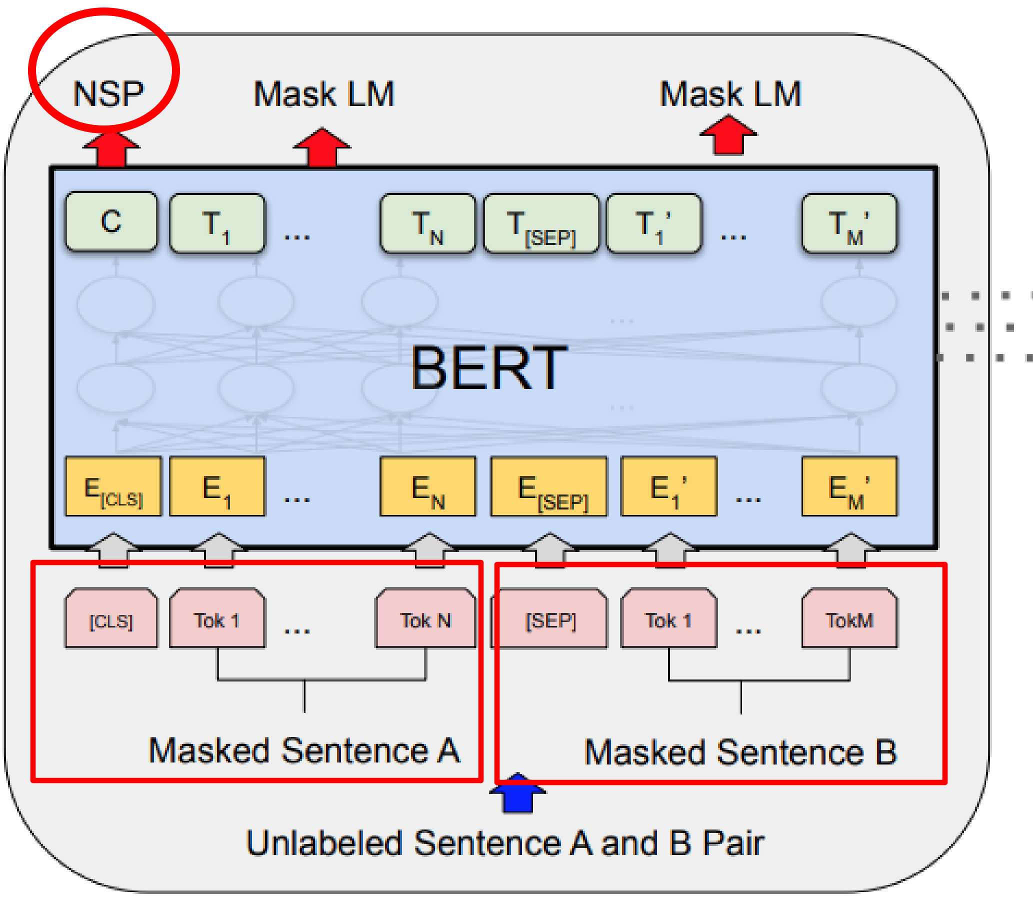
もう少しきちんと定義すれば、これらの数字は、表現Sが現れた時、表現Sの「継続」として表現Tが現れる条件付き確率π(T|S)だと考えることができます。

大規模言語モデルでの
Next Sentence Prediction 導入の意味

僕は、翻訳モデルから大規模言語モデルへの飛躍をもたらした最大のものは、文と文の「継続」あるいは「連続」を学習することを可能とする能力の獲得にあったと考えています。

もっともプリミティブなものは、おそらくBERTのpretrainingで導入されたNext Sentence Prediction のフラグだったと思います。

Pre-training BERTのタスク
Next Sentence Prediction (NSP)



Pre-training

- NSP:
  Next
  Sentence
  Prediction

  IsNext/NotNext

NSPの入力:Sentence Pair

Sentence BがSentence Aと
つながることを判定する

Next Sentence Predictionの例

Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

BERTの例でいえば、category L で次のような射が、ある確率を割り当てられて存在することを意味しています。

the man went to convenience store → the man went to convenience store he bought a gallon of milk

ただ、次のような射は存在しません。あるいは、割り当てられる確率はゼロです。

the man went to convenience store → the man went to convenience store penguin are flight less birds

このように、category L に確率を導入することで、大規模言語モデルの表現の継続の振る舞いを、より具体的にシミレートできます。

確率導入のもう一つの意味
文法性の付与

それだけではありません。言語のcategoryへの確率の導入は、言語に文法性を与えることに繋がります。

英語だと、射 cat → black cat は、射 cat → cat black より出現確率は高いのですが、フランス語だと、逆に、射 chat → chat noir の方が射 chat → noir chatより出現確率が高くなります。

この確率の違いは、英語では形容詞の後ろに名詞が続くのに対して、フランス語では名詞の後ろに形容詞が続くという、二つの言語の文法の違いを反映しています。

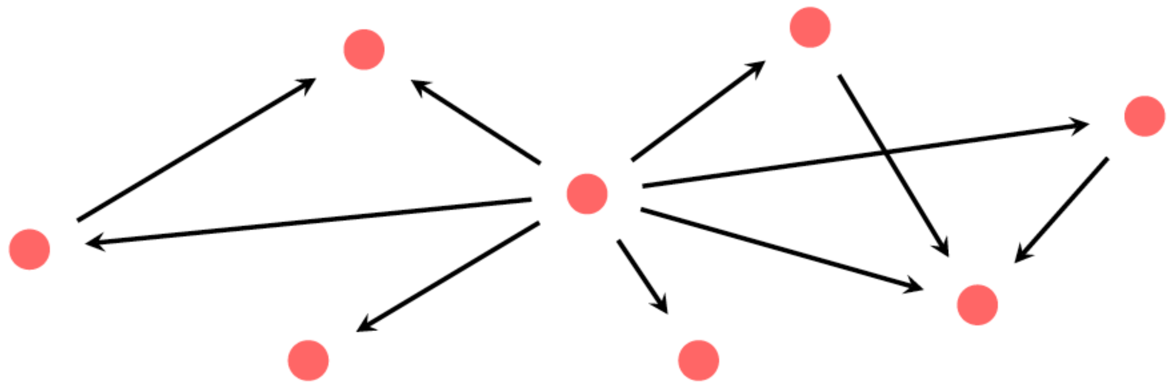
こうして、「形容詞は名詞の前にくる」(あるいは、形容詞は名詞の後ろにくる」)といった、文法ルールの形は取らないのですが、射に割り当てる確率を、その代理として、文法にあたるものを表現できます。

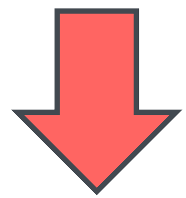
これまでは、言語からできるだけ構造を排除して、残ったpreorder の性質からだけ、言語のcategory L を作ってきたのですが、pregroupのような強い性質を前提にしなくとも、確率の導入だけで、言語のcategory Lに文法性を回復できるのです。

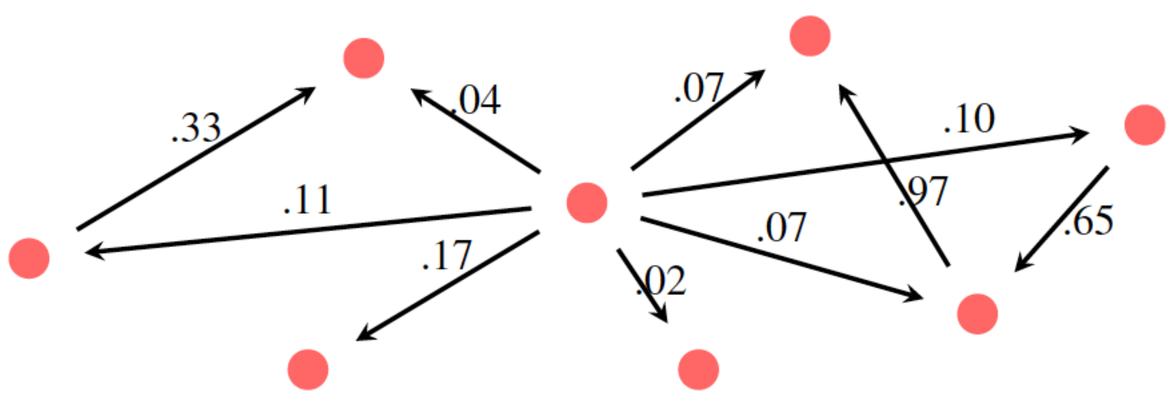
こうして拡張された言語のcategory を L′ としましょう。

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′

enriched category

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。

セミナーを前半と後半の二つに分けます

セミナーの目的を、前述のTai-Danae らの論文を理解することにおいているのですが、まだ、この論文の基本的なツールであるenriched categoryの紹介ができていません。

ここまでのショートムービーだけでも既に3時間を超えています。一回のセミナーに全てを詰め込むのは、かえってわかりにくいものになると感じています。

そこで、セミナーを、enriched category 導入以前（実はまだ話すべきことが残っていて、終わっていません）の前半部分と、enriched category 導入以後の論文本体の紹介の後半部分の二回に分割しようと考えています。

ただ、いわばenriched category導入以前にフォーカスした現在進行中のセミナー前半の内容は、とても大事なものです。

特に、言語のcategory Lを、集合に値を持つL上のcopresheafを意味のcategoryとして、Yoneda embedding で対応づける手法は、セミナー前半のコンテンツの中核です。

セミナー後半は、確率を導入した拡大された言語のcategory L′に、この構成をそっくり適用したものになっています。