

自然言語推論システム Neural DTS の学習アルゴリズムの実装

飯沼瑞稀 戸次大介

お茶の水女子大学

{iinuma.mizuki, bekki}@is.ocha.ac.jp

概要

本研究では、依存型意味論 (Dependent Type Semantics, DTS) を基盤とした自然言語推論システム Neural DTS の学習アルゴリズムを実装し、その挙動を検証した。DTS は文の意味を型理論に基づいて厳密に表現する一方、曖昧な述語を表現したり、言語表現同士の類似度を計算したりすることはできない。この課題に対処するため、Bekki らにより提案された Neural DTS の枠組みを基に、その学習アルゴリズムを実装した。また、自然言語文データセットで分類器の学習を行い、DTS の述語と名前を埋め込むことによって構築したニューラル判定器の性能を検証した。

1 はじめに

記号推論はニューラルネットの推論と対比されるが、現代的な記号推論システムの一例として型理論 (type theory) が挙げられる。型理論は数学の証明を形式化する目的で提案され [1], カリー・ハワード対応によりプログラミング言語としての性質も注目されてきた。特に依存型理論 (dependent type theory) [2] は自然言語の意味論への応用が進んでおり [3], その理論的枠組みと計算論的特性を活かして自然言語の意味解析が行われている。

依存型理論に基づく代表的な理論には、依存型意味論 (Dependent Type Semantics, DTS) [4], Modern Type Theories [5], および Type Theory with Records (TTR) [6] がある。これらは自然言語の含意認識タスクに応用され [7], ニューラルネットを統合した手法も提案されている [8]。これらの手法は、従来のシンボリックなアプローチにおける高い説明可能性と、ニューラルネットの持つ柔軟性を組み合わせることで、記号推論の推論力を拡張しようとしている点で注目されている。

本研究では、DTS にニューラルネットを統合した自然言語推論システムである、Neural DTS [9, 10]

の学習アルゴリズムを実装し、日本語文を対象とした性能検証を行う。また、本研究ではデータ構造や学習方法に関する新たな工夫も提案し、記号推論とニューラル推論を統合する新たなアプローチの可能性を探ることを目指す。

2 先行研究

本節では、依存型理論を用いた自然言語意味論と、ニューラルネットを統合した手法に関する先行研究を紹介する。

2.1 TTR とニューラルネットの統合

TTR は、依存型理論を基盤にした意味の理論で、情報や意味をレコード (属性と値のペア) として表現することを特徴とする。自然言語の意味論や認知科学において、複雑な情報構造の表現や推論を行うのに用いられる。Larsson et al. [8] では、TTR と、ニューラルネットワークを利用して記号的データをベクトル形式で表現するアプローチである Semantic Pointer Architecture (SPA) [11] を統合することで、形式意味論とニューラルネットワークを結びつける新しい手法を提案している。ここでは、TTR の形式的意味オブジェクトを SPA のベクトル表現にマッピングすることで、記号的推論とニューラル推論を統合する方法が示されている。この統合により、記号的な型理論に基づく推論がニューラルネットワーク上で直接実現される一方、ニューラル分類器との統合や学習アルゴリズムの具体的な実装については今後の課題として残されている。

2.2 DTS とニューラルネットの統合

Neural DTS は数学的背景 [9] と学習アルゴリズム [10] がすでに提案されている。

Neural DTS では、ニューラルネットを活用することで、通常の DTS では証明できない命題を推論することが可能になる。ここでは、具体例を用いて、通常の DTS での推論と Neural DTS での推論の違い

について説明する。

次のような知識が前提として与えられているとする：

- 前提 1. すべての麺類は安い
- 前提 2. ラーメンは麺類である
- 前提 3. ラーメンは安い

しかし、「蕎麦は麺類である」に関する情報は与えられていない。この時、仮定として「蕎麦が安いかどうか」を推論する。

DTS では、次のような流れで蕎麦が安いことを表す命題である、**isCheap**(soba) の妥当性を検証する：

1. **意味表示の生成**: 前提となる文を DTS の型を用いて表現する。

前提 1: $c : (x : e) \rightarrow \text{isNoodle}(x) \rightarrow \text{isCheap}(x)$

前提 2: $\text{isNoodle}(\text{ramen})$

前提 3: $\text{isCheap}(\text{ramen})$

仮定: $\text{isCheap}(\text{soba})$

2. **型環境の構築**: 文脈を表す型環境を以下のように設定する。型システムでは、型付け規則を適用する際に、型環境を参照して項の型を決定する：

$$\Gamma = \{c : (x : e) \rightarrow \text{isNoodle}(x) \rightarrow \text{isCheap}(x), \text{isNoodle}(\text{ramen}), \text{isCheap}(\text{ramen})\}$$

3. **証明の試行**: 定理証明器は、以下のように **isCheap**(soba) を導くための証明を探索する。ただし、**isNoodle**(soba) に関する直接的な証拠が存在しないため、証明は失敗する。

$$\frac{\frac{c : (x : e) \rightarrow \text{isNoodle}(x) \rightarrow \text{isCheap}(x) \quad \text{soba} : e \quad (\text{CON})}{c(\text{soba}) : \text{isNoodle}(\text{soba}) \rightarrow \text{isCheap}(\text{soba}) \quad (\text{PIE})} \quad ? : \text{isNoodle}(\text{soba})}{c(\text{soba})(s) : \text{isCheap}(\text{soba}) \quad (\text{PIE})}$$

しかし、Neural DTS では、この命題に対してニューラル分類器を呼び出し、**isNoodle**(soba) のスコアを計算する：

$$\text{pred}(\text{soba}, \text{isNoodle}) := \text{sigmoid}(W_{\text{out}}(\text{sigmoid}(W_{\text{hidden}}(\text{embed}(\text{soba}) \oplus \text{embed}(\text{isNoodle}))))).$$

スコアが閾値 τ を超える場合、**isNoodle**(soba) は成立すると判定される。

$$\text{pred}(\text{soba}, \text{isNoodle}) \geq \tau \implies \text{成立}.$$

この例からわかるように、通常の DTS の推論では明示的な前提に基づいて証明を行うため、未知の

情報を補完する機能は持たないが、Neural DTS では前提に含まれない知識でも学習済み分類器により判定可能になる。

Neural DTS の学習アルゴリズム [10] は、以下の手順で構成される。

1. **テキストからの意味表示生成**: 実世界のテキストデータに対して構文解析と意味合成を行い、DTS の意味表示を生成する。
2. **正例と負例の生成**: 推論規則に基づいて正例の述語を抽出し、ランダムに選ばれた負例と組み合わせる。
3. **損失関数の最適化**: 損失関数を定義し、正例が 1、負例が 0 になるようにニューラルネットワークのパラメータを学習する。

また、Neural DTS の部分的な実装として、Neural DTS に対する型検査アルゴリズムを用いる手法が提案されている [12]。この部分的な実装では、(photograph, copyrightHolder, Person) のような (entity1, relationentity2) という二項述語の命題において、否定、連言、選言のみを扱った。さらに、分類器の判定結果を型システム内で扱えるかどうかという点について、ニューラル分類器と型システムの融合の可能性が検討された。しかし、これは量化的でない述語論理における学習であり、自然言語への本格的な応用には至っていない。

3 提案手法

本研究では、NeuralDTS の学習アルゴリズムの実装と実験を行った。以下ではまず、DTS の述語と名前を埋め込むニューラル分類器を `hasktorch`¹⁾ により実装する (§ 3.1)。そして、日本語 CCG パーザ `lightblue` [13] に接続し、自然言語の文からデータを作成し、学習と分類を行えるようにする (§ 3.2)

3.1 `hasktorch` による分類器の実装

比較実験として、MLP (Multi Layer Perceptron) と NTN (Neural Tensor Network) の 2 つのバージョン (Socher らによるオリジナル版 [14] と Ding らによるバージョン [15]) を実装した。

3.1.1 MLP の実装

DTS の推論システムと組み合わせるための、埋め込み層・隠れ層・出力層の 3 層から構成される単純

1) <https://github.com/hasktorch>

な MLP によるニューラル分類器を実装する。ここでの実装には、hasktorch の Github リポジトリにある MLP による排他的論理和の実装を参考にしている。ここでの埋め込み層は、 n 個の entity と n 項述語をそれぞれ埋め込んだのちに連結するものとなっている。例えば 2 項のとき、MLP は以下のように定式化される。

$$f(e_1, e_2, p) = \sigma(W_3(\sigma(W_2(\sigma(W_1(e_1 \oplus e_2 \oplus p) + b_1)) + b_2)) + b_3)$$

ここで、 e_1 および e_2 は entity の埋め込み、 p は述語の埋め込み、 $e_1 \oplus e_2 \oplus p$ はそれらの埋め込みの結合を表す。また通常通り、 W_i および b_i は各 $i \in \{1, 2, 3\}$ における重みとバイアスであり、 σ は活性化関数としてシグモイド関数を表す。

3.1.2 NTN の実装

NTN は、各入力ペアに対してテンソル演算を適用し、非線形な関係をモデル化することができるニューラルネットワークモデルである。今回は、Socher et al. (2013) [14] と、それを改良した Ding et al. (2015) の NTN モデル [15] を実装した。

Socher NTN は、双線形テンソル積 $e_1^T W_r^{[1:k]} e_2$ を用いて以下のように定式化される：

$$g(e_1, r, e_2) = U_p^T \tanh\left(e_1^T W_p^{[1:k]} e_2 + V_p \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_p\right)$$

ここで、 $V_p \in R^{k \times 2d}$, $U_p \in R^k$, $b_p \in R^k$ である。

Ding NTN は以下のように定式化される：

$$g(e_1, r, e_2) = \sigma\left(e_1^T W e_2 + V_p \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_p\right)$$

Ding NTN は双線形テンソル積を用いず簡略化し、計算効率を向上させている。

3.2 学習データの準備

本研究では、以下の手順で学習データを準備する。

1. 構文解析と意味表示の生成:

CCG (Combinatory Categorical Grammar) [16] は、基本的なカテゴリと少数の結合規則を用いる形式文法である。その特徴として、統語構造と意味表示の結合を緊密にモデル化することで、統語構造から直接意味表示を構築する能力を持つ点が挙げられる。

lightblue [13] は、CCG を基盤として、日本語の

統語解析と意味解析を統合的に実現するために開発されたパーザである。

この lightblue を用いることで、入力された自然言語文を解析し、意味表示を生成することができる。例えば、「太郎が歌う」という文は次のような意味表示に変換される：

$$S : (e_0 : \text{event}) \times (u_0 : \text{歌う}(e_0, \text{太郎}))$$

2. 述語とエンティティの辞書作成:

意味表示から entity と述語を抽出し、それぞれ辞書として格納する。例えば、entity は「太郎」、述語は「歌う」に対応する。

entity の辞書:

$[(\pi_1(S), 0), (\pi_1(\pi_2(S)), 1), (\pi_1(\pi_2(\pi_2(S))), 2), (\text{太郎}, 3)]$

述語の辞書:

$[(\text{歌}, 0)]$

3. 正例の生成:

依存型理論の (ΣE) 規則を用いて、述語リストから成り立つ述語 (正例) を抽出する。(ΣE) 規則は、存在量子子を表現する Σ 型 $(x : A) \times B$ に対する除去規則である。

ΣE 規則

$$\frac{M : (x : A) \times B}{\pi_1(M) : A} (\Sigma E) \quad \frac{M : (x : A) \times B}{\pi_2(M) : B[\pi_1(M)/x]} (\Sigma E)$$

その後、(ΣE) 規則を適用して得た項の中で、**述語 (項)** のような形になっている項のみ取り出す。例えば、 Σ 型で記述される「ある出来事 e_0 が太郎によって歌われる行為である」という命題を (ΣE) 規則を適用したあとフィルタリングすると以下ようになる：

入力: $(e_0 : \text{evt}) \times \text{歌}(e_0, \text{太郎})$

出力: $\text{歌}(e_0, \text{太郎})$

4. 負例の作成:

負例は、正例に出現した entity と述語をランダムに組み合わせることで生成し、正例と同数になるよう調整する。ただし、生成する負例の組み合わせは、正例に含まれないものとする。

3.2.1 データセットの補強

上記のように生成されたデータセットの entity は、 $\pi_1(S)$ のように各文固有のものがほとんどであるため、出現頻度が少なくなってしまう。そのため、entity の出現回数を増やす工夫を行った。日本語 WordNet を用いて、述語名の同義語を検索し、そ

の述語名を同義語で置き換えたデータを追加した。同義語は上から5つまで使用した。

4 実験

日本語文を対象として、Neural DTS の学習アルゴリズムの有効性を評価するために実験を行った。

4.1 実験手法

データセットとして、日本語 WordNet の例文 48274 件のうち 1000 件を使用した²⁾。日本語 WordNet は、日本語における多様な語彙を含んでおり、世界知識として Neural DTS に学習させるデータとして適している。うち、parse エラーなどを除き、entity や述語を取り出すために使用した文は 791 件である。ここから、entity は 4491 件取り出された。データを補強した結果、辞書にある述語の数は 1848 件から 7572 件、成り立つ述語の件数は 2310 件から 8606 件に増加した。この補強データは学習データにのみ追加している。

また、entity の偏りを防ぐため、テストデータは各文につき最大1つずつランダムに成り立つ述語を抽出して利用した。結果、テストデータのサイズは正例、負例共に 595 件となった。その際、汎化性能の確認のため、テストデータのランダム抽出は 4 回行い、その平均精度を出力した。

4.2 実験結果

2 項の場合の学習結果を以下に示す。

表 1 モデルスペック

パラメータ	MLP	S.NTN	D.NTN
Entity 埋め込み次元	256	128	256
Pred 埋め込み次元	256	128	256
テンソル次元	-	-	256
隠れ層次元 1	216	-	-
隠れ層次元 2	32	-	-
出力次元	1	1	-
ドロップアウト率	-	-	0.1

MLP と NTN のいずれも、実用上十分な精度には達しなかった。部分的な実装 [12] を用いた実験では、MLP や NTN を利用した推論で精度が 90% を超えていたことから、今回の実験で精度が低かった原因は、学習データに述語を追加したにもかかわらず、entity 間の分布が疎であったことに起因すると

表 2 ハイパーパラメータ

パラメータ	MLP	S.NTN	D.NTN
学習率	5e-2	5e-2	1e-2
Optimizer	GD	Adam	Adam
バッチサイズ		256	
エポック数		1000	
損失関数	binary cross-entropy		
活性化関数	Sigmoid		

表 3 推論スコア

モデル	Accuracy	Precision	Recall	F1-Score
MLP	0.78	0.75	0.76	0.77
S.NTN	0.51	0.51	0.52	0.51
D.NTN	0.50	0.51	0.50	0.50

考えられる。特に、今回の手法で得られる entity は、 $\pi_1(S1)$ のように各文固有のものがほとんどであり、汎用的な entity の利用が難しかったことが影響していると考えられる。

5 おわりに

Bekki et al. [10] によって提案された学習アルゴリズムを実装したが、十分な精度は得られなかった。

今後の課題として、データセットや分類器の選択、パラメータ調整など現在の手法を洗練させる工夫に加え、負例の生成手法やデータセット補強といった、Bekki et al. [10] の学習アルゴリズム自体を改良するための具体的なアプローチを検討する必要がある。特に、entity の分布を補完するための追加データの収集や生成、あるいはデータの再サンプリング手法の導入が有効であると考えられる。

精度が十分に得られた後は、DTS での定理証明器に分類器を組み込み、推論をシームレスに行えるシステムの実現を目指したい。これにより、記号的推論とニューラル推論を統合した新たな手法を確立し、自然言語推論のさらなる発展に寄与することを目指す。将来的には、提案手法を幅広いタスクに適用し、その汎用性や有効性を実証したいと考えている。

2) <https://bond-lab.github.io/wnja/>

謝辞

本研究の一部は、JST CREST JPMJCR20D2 の支援を受けたものである。

参考文献

- [1] Per Martin-Löf. An intuitionistic theory of types. In G. Sambin and Jan M. Smith, editors, **Twenty-five years of constructive type theory**, Vol. 36 of **Oxford Logic Guides**, pp. 127–172. Clarendon Press, 1998.
- [2] Per Martin-Löf. **Intuitionistic Type Theory**. Bibliopolis, Naples, 1984.
- [3] G. Sundholm. Constructive generalized quantifiers. **Synthese**, Vol. 79, pp. 1–12, 1989.
- [4] Daisuke Bekki and Koji Mineshima. Context-passing and underspecification in dependent type semantics. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, **Modern Perspectives in Type-Theoretical Semantics**, pp. 11–41. Springer International Publishing, Cham, 2017.
- [5] Stergios Chatzikyriakidis and Zhaohui Luo. On the interpretation of common nouns: Types versus predicates. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, **Modern Perspectives in Type-Theoretical Semantics**, pp. 43–70. Springer International Publishing, Cham, 2017.
- [6] Robin Cooper. Adapting type theory with records for natural language semantics. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, **Modern Perspectives in Type-Theoretical Semantics**, pp. 71–94. Springer International Publishing, Cham, 2017.
- [7] Stergios Chatzikyriakidis and Jean-Philippe Bernardy. A wide-coverage symbolic natural language inference system. In Mareike Hartmann and Barbara Plank, editors, **Proceedings of the 22nd Nordic Conference on Computational Linguistics, NoDaLiDa 2019, Turku, Finland, September 30 - October 2, 2019**, pp. 298–303. Linköping University Electronic Press, 2019.
- [8] Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and Andy Luecking. TTR at the SPA: Relating type-theoretical semantics to neural semantic pointers. In **Proceedings of the Natural Logic meets Machine Learning (NALOMA) 2023 Workshop**, pp. 41–50, 2023.
- [9] Daisuke Bekki, Ribeka Tanaka, and Yuta Takahashi. Integrating deep neural network with dependent type semantics. **Proceedings of the Symposium Logic and Algorithms in Computational Linguistics 2021 (LACompLing2021)**, Axcel Ljungstrom, Rousanka Loukanova, Peter LeFanu Lumsdaine, Reinhard Muskens (eds.) **Stockholm University, 2021, DiVA Portal for Digital Publications**, pp. 37–37, 12 2021.
- [10] Daisuke Bekki, Ribeka Tanaka, and Yuta Takahashi. Learning knowledge with neural DTS. In **Proceedings of the 3rd Natural Logic Meets Machine Learning Workshop (NALOMA III)**, pp. 17–25, Galway, Ireland, August 2022. Association for Computational Linguistics.
- [11] Chris Eliasmith. **How to Build a Brain: A Neural Architecture for Biological Cognition**. Oxford Series on Cognitive Models and Architectures. Oxford University Press, 2013.
- [12] Mizuki Inuma, Yuta Takahashi, Sora Tagami, and Daisuke Bekki. Neural DTS: A Hybrid NLI System Combining Two Procedural approaches. In **Proceedings of Procedural and Computational Models of Semantic and Pragmatic Processes, ESSLLI 2023 Workshop**, 2023.
- [13] Daisuke Bekki and Ai Kawazoe. Implementing variable vectors in a CCG parser. In **Proceedings of the Logical Aspects of Computational Linguistics (LACL 2016)**, Vol. 10054 of **Lecture Notes in Computer Science**, pp. 52–67. Springer, Cham, 2016.
- [14] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, **Advances in Neural Information Processing Systems**, Vol. 26, pp. 926–934. Curran Associates, Inc., 2013.
- [15] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In **Proceedings of the 24th International Conference on Artificial Intelligence**, pp. 2327–2333, 2015.
- [16] Mark Steedman. **The Syntactic Process**. The MIT Press, Cambridge, MA, 2000.