

証明支援系 Coq を用いた 依存型意味論の照応解析と推論の実装の試み

小齊平ひな 高橋優太 戸次大介
お茶の水女子大学

{g1820513,takahashi.yuta,bekki}@is.ocha.ac.jp

概要

含意関係認識に対するアプローチの一つに、証明支援系 Coq を用いるアプローチがある。タクティクを用いて自動証明手続きを定義することで、さまざまなテキストに対する含意関係認識タスクがこのアプローチにより研究されてきた。その一方で、照応解析の結果を含意関係認識の中で用いるという種類のタスクを、Coq が提供するタクティクのみを用いて行う手法はこれまでに提案されていない。こうした手法への一歩として、本研究では、照応解析を Coq の refine タクティクによって行い、得られた結果を用いてテキストから仮説を導く手法を提案する。

1 はじめに

含意関係認識 (Recognizing Textual Entailment) とは、テキスト T が仮説 H を含意するかどうかを判定するタスクであり、質問応答 (question answering) など、さまざまな応用をもつ。このタスクに対しては、大規模なデータを用いてニューラルモデルに含意関係を学習させる深層学習アプローチ [1, 2], 定理証明器を用いる論理推論アプローチ [3, 4], およびこの二つを組み合わせるハイブリッドアプローチ [5] がある。

論理推論アプローチではさまざまな定理証明器が用いられており、その中には定理証明器 Coq [6] を用いるものがある [7, 8, 9, 10, 11, 12, 13]。証明支援系 (proof assistants) とも呼ばれる Coq を用いるこのアプローチには以下の利点がある。まず、依存型理論 (dependent type theory) に基づく Coq では、自然言語の文がもつ複雑な意味構造を捉えることができる。さらに、証明状態を遷移させることのできる「タクティク」と呼ばれるものを合成することで、ユーザー定義の自動証明手続きを定式化することが

できる。以上に加え、論理推論アプローチ一般がもつ利点として、含意関係認識タスクにおける判定結果の解釈・判定手続きの修正が容易である点も挙げられる。これまで、様相表現・一般化量子子などを含むさまざまなテキストに対する含意関係認識タスクが Coq アプローチにより研究されてきた。

その一方で、照応解析 (anaphora resolution) の結果を含意関係認識の中で用いるという二重タスクを、Coq が提供するタクティクのみを用いて行う手法はこれまでに提案されていない。照応解析とは、英語を例にとれば、テキストの中にある “it” や “he” などの照応表現の指示対象を指定するタスクであり、自然言語処理や形式意味論といった分野で研究されている。次の例は、照応表現を含むテキストに関する含意関係認識を行うには照応解析が必要であることを示している [14, p. 46]:

A man entered. He whistled. \implies A man whistled.

照応解析に続く含意関係認識のためのタクティク整備へ向けて、本研究ではまず、照応解析を Coq タクティクによって行う手法を定式化する。この目的のために、自然言語意味論のフレームワークである依存型意味論 (Dependent Type Semantics, DTS) を Coq で実装する。DTS では未指定項 (underspecified terms) を用いた照応解析手続き [15, 16, 17] が定式化されており、本研究の基本アイデアは、この DTS 照応解析手続きを Coq が提供する refine タクティクによりシミュレートするというものである。次に、照応解析の結果、テキストが仮説を含意する場合は、その含意関係を Coq 内の定理として証明する手法を提案する。

2 先行研究

DTS における照応解析手続きに対しては、Haskell による実装 [18, 19, 20, 21] がすでに与えられている。DTS においては、照応解析手続きは型検査アル

ゴリズムが呼び出す証明探索手続きとして捉えられ、これらのアルゴリズムおよび証明探索手続きが Haskell を用いて実装されている。それに対して本研究では、DTS における型検査アルゴリズムを Coq における型検査アルゴリズムに還元し、照応解析に対応する証明探索手続きは Coq タクティクにより与える。

含意関係認識に対するこれまでの Coq アプローチの中では、[12, 13] が、照応解析の結果を含意関係認識の中で用いるという二重タスクを扱っている。これらの先行研究では、モナドに基づく動的意味論を介して照応解析が行われている。これに対し、上述のように本研究では、Coq が提供する refine タクティクを用いることで、Coq タクティクによるのみ照応解析を行う手法を提案する。

3 提案手法

本研究が提案する手法は、照応解析に関するものと含意関係認識に関するものに分かれる。以下ではまず、Coq の refine タクティクについて説明したのち、このタクティクを用いて DTS における照応解析手続きを実装する (§ 3.1)。その際に、DTS での照応解析および含意関係認識に現れる多くの命題の証明に有効なタクティクを構成する。次に、照応解析の結果、テキストが仮説を含意する場合は、その含意関係を Coq 内の定理として証明する手法を提案する (§ 3.2)。

3.1 refine タクティクを用いた照応解析と証明自動化のためのタクティク

DTS は、自然言語の文の意味を、Martin-Löf 型理論 [22] の式にその文を翻訳することによって説明する。言い換えれば、翻訳の結果として得られる式がその文の意味表示となる。ここでの翻訳は、当の文の部分表現に割り当てられた意味表示を型理論の規則に従って合成することでなされる。

DTS において、自然言語の文に現れる “it” といった照応表現の意味表示は、もともとの Martin-Löf 型理論には含まれない未指定項 @ によって与えられる。例として、照応表現 “he” を含む次の二つの文から構成される談話を考えると、この談話の意味表示は図 1 のようになる。

1a. A man entered.

1b. He whistled.

図 1 に見られる $\left[\begin{array}{l} x : A \\ B(x) \end{array} \right]$ という表現は、 Σ 型と呼ば

$$\left[\begin{array}{l} x : \mathbf{entity} \\ u : \left[\begin{array}{l} \mathbf{man}(x) \\ \mathbf{enter}(x) \end{array} \right] \\ \mathbf{whistle}(\pi_1(@ : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right]))) \end{array} \right]$$

図 1 例文 1a, 1b の意味表示

$$\left[\begin{array}{l} x : \mathbf{entity} \\ u : \left[\begin{array}{l} \mathbf{man}(x) \\ \mathbf{enter}(x) \end{array} \right] \\ \mathbf{whistle}(\pi_1(u)) \end{array} \right]$$

図 2 照応解析結果

れるデータ型の DTS における表記であり、型 A の要素 a と型 $B(a)$ の要素 b のペア (a, b) の型である。

例えば $\left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right]$ は (以下ではこの型を Man と呼ぶ)、型 \mathbf{entity} の要素である対象 e と、型 $\mathbf{man}(e)$ の要素である「 e が男性であることの証明 t 」のペア (e, t) の型である。型と命題の同型性を示すカーリー・ハワード対応 (Curry-Howard correspondence) によれば、この型は「男性がいる」という命題と見なすことができ、ペア (e, t) はこの命題の証明となる。もし $\left[\begin{array}{l} A \\ B \end{array} \right]$ のように B が A の要素に依存しないときは、この型はたんに「 A かつ B である」という命題

に対応する。こうして、型 $\left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{man}(x) \\ \mathbf{enter}(x) \end{array} \right]$ は「あ

る男性が入ってきた」という命題に対応し、1a の意味表示となる。そして、図 1 の残りの部分が 1b の意味表示であり、「彼は口笛を吹いた」という命題に対応する。ここでの未指定項 @ は、型 Man の証明の部分がいわば空所になっていて証明を待っていることを示している。もし何らかの証明つまりペア (e, t) が与えられれば、その第一要素を取り出す関数 π_1 によって男性である対象 e を取り出すことができ、この $\pi_1(@ : Man)$ が “he” の意味表示となる。

DTS において照応解析は、証明探索によって @ を具体的な証明に置き換える操作に相当する。図 1 の例では @ の型は Man であるが、その証明として $(\pi_1(u), \pi_1(\pi_2(u)))$ が存在するため、図 1 の @ : Man をこの証明に置き換える。すると、 $\pi_1(\pi_1(u), \pi_1(\pi_2(u)))$ を簡約すると $\pi_1(u)$ となるため、図 2 のような意味表示が得られる。つまり、照応解析の結果、“he” の指示対象は $\pi_1(u)$ となる。

以上から分かるように、DTSにおける照応解析手続きを実装するには、型における空所を表現し、さらにその空所を埋める証明探索を定式化することが必要である。本研究では、この一連の手続きを、証明支援系 Coq が提供する refine タクティクを用いてシミュレートする。このタクティクは、証明が未完成の部分をひとまず空所にしたまま証明を進めることを可能にする。

Coq が提供する refine タクティクを用いた DTS 照応解析手続きは次の手順で定式化される。

- (i) DTS に従って、語彙の意味表示を定義する。
- (ii) 未指定項に対応する表現 [?asp] によって照応表現を表しつつ文の意味表示を合成し、Eval コマンドでその意味表示を計算する。
- (iii) refine タクティクを計算結果に適用して、[?asp] 以外の部分に型検査を施しつつ、空所 [?asp] を埋めることをゴールに設定する。
- (iv) 空所 [?asp] を埋める証明をタクティクによって構成し照応解析を行う。

例 1 を例に挙げて、Coq での DTS 照応解析手続きを説明する。

- (i) DTS に従って、語彙の意味表示を定義する。例えば、a_nom は名詞位置の不定冠詞 “a” の意味表示である。

```
Definition a_nom : forall n v : entity ->
  Type, Type :=
  fun n v => {x : entity & {- : n x & v x}}.

Axiom man enter whistle : entity -> Type.
```

- (ii) 未指定項に対応する表現 [?asp] によって照応表現を表しつつ文の意味表示を合成し、Eval コマンドでその意味表示を計算する。以下においては、

```
(prog_conj (a_nom man enter)
  (whistle (projT1 (?[asp1] :
    {x : entity & man x}))))
```

が 1a, 1b から構成される談話の意味表示である。

```
Theorem AManHe : Type.
Proof.
  Eval cbv in (prog_conj (a_nom man enter)
    (whistle (projT1 (?[asp] :
      {x : entity & man x})))).
```

- (iii) refine タクティクを計算結果に適用して、[?asp] 以外の部分に型検査を施しつつ、空所

[?asp] を埋めることをゴールに設定する。

```
refine {_ : {x : entity &
  {- : man x & enter x}} &
  whistle (let (a, _) :=
    (?[asp] :
      {x : entity & man x}) in a)}.
```

- (iv) 空所 [?asp] を埋める証明をタクティクによって構成し照応解析を完了させる。Coq の proof mode は、こうして得られた解析結果を、上記の定理の名前である AManHe として保存することを可能にする。

このようにして、DTS 照応解析手続きは Coq が提供するタクティクを用いて対話的に行われる。その一方で、本研究は、この手続きが含む証明探索の一部を自動化するタクティクの構成も行う。Coq のタクティク言語 Ltac を用いて構成したタクティクは以下になる。

```
Ltac tac1 := cbv in *;
  intros;
  try repeat (destruct_one_ex ||
    destruct_one_pair ||
    specialize_H ||
    rewrite_H);
  try exists_H; try apply_H.
```

図 1 およびその解析結果である図 2 が例となるように、DTS における照応解析においては、 Σ 型の要素をいったん分解し、解析の中で求められているペアに組み直すといった推論が多く現れる。この tac1 は、そうした推論を自動で行うためのタクティクとして定義している。

tac1 を構成するために作成したタクティクについてもおおまかに説明する。これらのタクティクの定義については付録 A を参照されたい。ただし、destruct_one_ex と destruct_one_pair は、Coq [6] の標準ライブラリの一つである Coq.Program.Tactics にて定義されているものであるため、定義は繰り返さない。この二つのタクティクは、DTS における照応解析の中でしばしば必要になる Σ 型の分解を行うため有用である。

specialize_H は、 Σ 型と対をなす Π 型についてのものであり、 Π 型 $\left(u : \begin{bmatrix} x : A \\ B(x) \end{bmatrix}\right) \rightarrow C$ と、型 A の要素 a および型 B(a) の要素 b が前提に含まれるとき、ペア (a, b) によって上の Π 型を例化するタクティクである。例えば、 $(u : Man) \rightarrow \mathbf{whistle}(\pi_1(u))$ は「どの男性も口笛を吹いた」という全称命題に対応するが、specialize_H は、この命題と男性 (e, t) が前提に含まれるとき、(e, t) でこの命題を例化する。照応

解析のために分解したい Σ 型が、上述の Π 型の後件である C の中に現れるとき、このタクティクによって全体の Π 型を例化することで C 中の Σ 型の分解へ進むことができる。次の `rewrite_H` は、前提に $A = B$ が現れるとき、この等式を使ってゴールの中の A を B で置き換えるタクティクである。また、タクティク `exists_H` は、Coq にあらかじめ備わっている自動証明タクティク `eauto` を用いて、ゴールが Σ 型であるときに自動でペアをつくり証明することを試みる。同様に `apply_H` は、いわゆるモーダス・ポネンス (modus ponens) を用いつつ、`eauto` による自動証明を試みるタクティクである。

以上のように、`refine` タクティクを用いた手順 (i)–(iv) により対話的に照応解析を行いつつ、`Ltac` により構成したタクティク `tac1` を用いて証明の一部を自動化することで、本研究が実装した DTS 照応解析手続きは完了する。Coq の `proof mode` は、こうして得られた解析結果を保存することを可能にし、保存された結果を用いて含意関係を判定できる。

3.2 照応解析の結果を用いる推論の実装

本研究では、照応解析の結果を含意関係認識タスクで用いるための手続きを次のように定式化する。

- (v) DTS に従って、語彙の意味表示を定義する。
- (vi) テキスト T に対する照応解析結果が前件となり、仮説 H が後件となる条件文を組み立てる。
- (vii) `Ltac` を用いて構成したタクティク `tac1` を用いて証明を行う。

次の 2. を例に挙げ、照応解析結果を用いた含意関係認識タスクへの本研究のアプローチを説明する。

2. A man entered. He whistled. \implies A man whistled.

- (v) DTS に従って、語彙の意味表示を定義する。
- (vi) テキスト T に対する照応解析結果が前件となり、仮説 H が後件となる条件文を組み立てる。いまの場合、照応解析結果として前節で保存した `AManHe` がテキストに対する照応解析結果となる。

```
Theorem ex: AManHe -> a_nom man whistle.
```

- (vii) `Ltac` を用いて構成したタクティク `tac1` を用いて証明を行う。

```
Proof.
tac1.
Qed.
```

`Proof.` で証明を始め、`tac1` を用いて自動でゴー

ルを見つけることかできたので、`Qed.` で証明を終える。

4 成果

§§ 3.1–3.2 にて説明した提案手法に基づき、前節で扱った例 2. に加え、次の例 3. に対しても証明を構成することができた。

- 2. A man entered. He whistled. \implies A man whistled.
- 3. John bought a car. He checked the motor. \implies John checked a motor.

例 3. においては、二文目における “he”, “the motor” が照応表現となり、それぞれ次のような意味表示が割り当てられる：

```
projT1 (?[asp1] : {x : entity & man x})
```

```
projT1 (?[asp2] : {y : entity & motor y})
```

上述のように、これらを含む談話全体の意味表示に `refine` タクティクを適用することで照応解析手続きは進む。いまの場合、`?[asp1]` と `?[asp2]` を証明で埋める（言い換えれば、“he” と “the motor” について照応解析をする）という二つのサブゴールが生じる。

“he” の指示対象として `John` を指定するには、公理として `man john` を加えれば、タクティクにより容易に照応解析をすることができる。一方で、“the motor” の指示対象として `John` が買った車のモーターを指定するには、全ての車にモーターがあるという一般的な背景知識を公理として措定し、`John` が買った車にもモーターがあることを導出できるようにする必要がある。こうした導出も、本研究が構成したタクティク `tac1` により行うことができる。あとは、以上の照応解析結果を `CarMotor` として保存し、例 3. の意味表示となる条件文

```
CarMotor -> a_acc motor check john
```

をタクティクにより証明すればよい。

5 おわりに

本研究では、DTS における未指定項を用いた照応解析手続きを Coq により実装し、さらに照応解析の結果を推論の中で用いるための手続きを定式化した。今後の課題として、同様に Coq を用いるアプローチである [12, 13] の提案手法との比較がある。加えて、照応解析の結果を用いる推論の中でも本研究が扱わなかった、結論が照応表現を含む推論の Coq による実装を目指す。

謝辞

本研究は、JST CREST JPMJCR20D2 の支援を受けたものである。

参考文献

- [1] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. In Y. Bengio and Y. LeCun, editors, **4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings**, 2016.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)**, pp. 4171–4186. Association for Computational Linguistics, 2019.
- [3] Lasha Abzianidze. LangPro: Natural Language Theorem Prover. In L. Specia, M. Post, and M. Paul, editors, **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017 - System Demonstrations**, pp. 115–120. Association for Computational Linguistics, 2017.
- [4] Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. MonaLog: a lightweight system for natural language inference based on monotonicity. In **Proceedings of the Society for Computation in Linguistics 2020**, pp. 334–344, New York, New York, January 2020. Association for Computational Linguistics.
- [5] Aikaterini-Lida Kalouli, Richard S. Crouch, and Valeria de Paiva. Hy-nli: a hybrid system for natural language inference. In D. Scott, N. Bel, and C. Zong, editors, **Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020**, pp. 5235–5249. International Committee on Computational Linguistics, 2020.
- [6] The Coq Development Team. The Coq proof assistant reference manual, Version 8.16.1, 2022. <https://coq.inria.fr/>.
- [7] Stergios Chatzikyriakidis and Zhaohui Luo. Natural language inference in coq. **J. Log. Lang. Inf.**, Vol. 23, No. 4, pp. 441–480, 2014.
- [8] Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. Higher-order logical inference with compositional semantics. In L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, editors, **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015**, pp. 2055–2061. The Association for Computational Linguistics, 2015.
- [9] Stergios Chatzikyriakidis and Zhaohui Luo. Proof assistants for natural language semantics. In M. Amblard, P. de Groote, S. Pogodalla, and C. Retoré, editors, **Logical Aspects of Computational Linguistics. Celebrating 20 Years of LACL (1996-2016) - 9th International Conference, LACL 2016, Nancy, France, December 5-7, 2016, Proceedings**, Vol. 10054 of **Lecture Notes in Computer Science**, pp. 85–98, 2016.
- [10] Koji Mineshima, Ribeka Tanaka, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. Building compositional semantics and higher-order inference system for a wide-coverage japanese CCG parser. In J. Su, X. Carreras, and K. Duh, editors, **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016**, pp. 2236–2242. The Association for Computational Linguistics, 2016.
- [11] Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. On-demand injection of lexical knowledge for recognising textual entailment. In M. Lapata, P. Blunsom, and A. Koller, editors, **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers**, pp. 710–720. Association for Computational Linguistics, 2017.
- [12] 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.
- [13] Jean-Philippe Bernardy and Stergios Chatzikyriakidis. Applied temporal analysis: A complete run of the FraCaS test suite. In **Proceedings of the 14th International Conference on Computational Semantics (IWCS)**, pp. 11–20, Groningen, The Netherlands (online), June 2021. Association for Computational Linguistics.
- [14] Ribeka Tanaka. **Natural Language Quantification and Dependent Types**. PhD thesis, Ochanomizu University, 2021.
- [15] Daisuke Bekki. Representing anaphora with dependent types. In N. Asher and S. Soloviev, editors, **Logical Aspects of Computational Linguistics - 8th International Conference, LACL 2014, Toulouse, France, June 18-20, 2014. Proceedings**, Vol. 8535 of **Lecture Notes in Computer Science**, pp. 14–29. Springer, 2014.
- [16] Daisuke Bekki and Koji Mineshima. Context-passing and underspecification in dependent type semantics. In S. Chatzikyriakidis and Z. Luo, editors, **Modern Perspectives in Type-Theoretical Semantics**, Vol. 98 of **Studies in Linguistics and Philosophy**, pp. 11–41. Springer, Cham, 2017.
- [17] Daisuke Bekki. Proof-theoretic analysis of weak crossover. In **Proceedings of the Eighteenth International Workshop of Logic and Engineering of Natural Language Semantics 18 (LENLS18)**, pp. 75–88, 2021.
- [18] Daisuke Bekki and Miho Satoh. Calculating projections via type checking. In **Proceedings of TYTTLES**, 2015.
- [19] 佐藤未歩. 依存型意味論の証明探索とその実装, 2016.
- [20] Hinari Daido and Daisuke Bekki. Development of an automated theorem prover for the fragment of DTS. In **Proceedings of the 17th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS17)**, 2020.
- [21] 大洞日音. DTS の部分体系を用いた定理自動証明器への等号型の導入, 2022.
- [22] 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.

A 付録

本研究では、DTS での照応解析および含意関係認識に現れる多くの命題の証明にとって有効なタクティク `tac1` を構成した。このタクティクを構成する各々のタクティクの定義は以下になる。

```
Ltac specialize_H :=
  match goal with
  |[x : ?A |- _] => match goal with
    |[H : ?B x |- _] => match goal with
      |[H1 : forall u : {x : A & B x }, ?C |- _] =>
        specialize (H1 (existT B x H))
      end
    end
  end

end.

Ltac rewrite_H :=
  match goal with
  |[H : ?A = ?B |- _] => rewrite H
  end.

Ltac exists_H :=
  match goal with
  |[H : ?A |- {_ : ?A & _}] => exists H; solve[info_eauto]
  end.

Ltac apply_H :=
  match goal with
  |[H : _ |- _] => apply H; solve[info_eauto]
  end.
```