

Coq を用いた依存型意味論の照応解析の実装

小齊平ひな (指導教員: 戸次 大介)

1 はじめに

近年の意味解析における照応解析では、語の間の意味的類似度を用いて照応表現 (apaphoric expression) の先行詞を同定するといった深層学習・機械学習によるアプローチが主流となっている。一方、依存型意味論 (Dependent Type Semantics, DTS) は、依存型理論による自然言語の意味論の一つであり、特に照応 (anaphora) と前提 (presupposition) は DTS が説明できる意味現象の中核となっている。照応現象にはこのような文の統語構造や言語現象を捉えていない解析では扱うことのできない例、たとえば E タイプ照応 [Evans 1980] や橋渡し照応 [Clark 1975], 前提現象 [Heim 1983], 弱交差現象の例が存在することが知られている。

依存型意味論 (Dependent Type Semantics; DTS) [Bekki 2014; Bekki and Mineshima 2017][1][2] は、上記のような例を含む様々な照応現象を統一的に扱うことのできる自然言語の意味論である。DTS では近年、未指定型 (underspecified types) と呼ばれる新たな理論的装置が導入され、照応解析は、未指定型を介した証明探索として再定式化された。DTS のための証明探索アルゴリズムを定式化し、実装することができれば、照応解析の計算を自動で行うことのできるシステムの構築が可能となる。

本論文では、未指定型 (underspecified types) を介した DTS のための証明探索アルゴリズムを証明支援系 Coq を用いて実装を行なった。また、どの表現が照応表現の先行詞になりうるか (到達可能性) を予測し、照応表現の到達可能なすべての表現について、その表現が先行詞である場合の意味を与えることだけでなく、その前後の文脈から到達可能な表現のうちどの表現が整合性が取れているかを判定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 [3, 7, 6, 8] 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。これらの研究では、DTS の以前のバージョンである未指定項 (underspecified term) を実装していた。それに対して本研究では、DTS の新しいバージョンである未指定型 (underspecified type) を介した 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、[5, 4] 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、依存型理論に基づく依存型意味論を介して

照応解析を行い、Coq が提供する `refine` タクティクを用いることで、Coq タクティクによってのみ照応解析を行う手法を提案する。

3 実装

未指定型 $\left[\begin{array}{c} x@A \\ B \end{array} \right]$ を介した照応探索手続きを、Coq を用いて次の手順で定式化する。

- (i) 到達可能な先行詞を探索する。
- (ii) 照応表現を含むテキストに到達可能な先行詞を当てはめたテキストを生成する。

まず、未指定型を Coq の帰納的型 `aspT` として定義する。

```
Inductive aspT (A : Type) (a : A) (B : Type)
  : Type :=
  resolve : B -> aspT A a B.
```

ここで型 `aspT` は、型 `A`、型 `A` をもつ項 `a`、型 `B` の三つから構成される複合的な型 `aspT A a B` である。ただし、型 `B` は項 `a` に依存していてもよい。例として以下の談話を考える。

(1) The lawyer asked the witness a question, but he was reluctant to answer it.¹

(1-0) There is a lawyer, a witness, and a question.

(1a) The lawyer asked the witness a question.

(1b) He was reluctant to answer it.

(1) の談話では、*he* と *it* が照応表現であり、`aspT` を用いて (1b) の意味表示を次のように表す。

```
aspT {x:entity & human x -> False} ?[asp1]
(aspT {y:entity & man x} ?[asp2]
  (reluctant (answer (projT1 ?asp1))
    (projT1 ?asp2)))
```

?[asp] は空所であり `aspT` の型 `A` を持つ項である。(1a) のテキストから (1-0) のように *a lawyer* と *a witness* と *a question* が前提され、それぞれは次のような型をもち、常識的な背景知識を公理として追加する。*a lawyer* の意味表示:

$$\left[\begin{array}{c} x : \text{entity} \\ \text{lawyer}(x) \end{array} \right]$$

a witness の意味表示:

$$\left[\begin{array}{c} x : \text{entity} \\ \text{witness}(x) \end{array} \right]$$

a question の意味表示:

$$\left[\begin{array}{c} x : \text{entity} \\ \text{question}(x) \end{array} \right]$$

¹<https://cs.nyu.edu/~davise/papers/WinogradSchemas/WSCollection.xml>, # 10.

公理として追加する常識的な背景知識：

```
Hypothesis LawyerHuman : forall x:entity,
  lawyer x -> human x.
Hypothesis WitnessHuman : forall x:entity,
  witness x -> human x.
Hypothesis QuestionNotHuman : forall x:
  entity,
  question x -> (human x -> False).
```

Coq が提供する `refine` タクティクを (1b) の意味表示に適用して、`?[asp]` 以外の部分に型検査を施しつつ、空所`?[asp1]` と `?[asp2]` を埋めることをゴールに設定する。照応解析は、空所を用いて表された照応表現をゴールに設定し、前提から照応の指示対象を探索するという証明探索と言い換えることができる。ここで、自動照応探索タクティクを用いて自動的に証明探索を行う。自動照応探索タクティクは、`destruct_one_ex` や `destruct_one_pair` で DTS の Σ 型およびペア型を分解し²、`specialize_H` で Π 型の例化を行い、`eexists` でゴールに対して任意の存在変数を与えて、`apply_H` でゴールの型に合うものを前提や仮定から適用して証明するものである。

```
Ltac tac1' n :=
  match n with
  | 0 => cbv in *; intros; try repeat (
    destruct_one_ex || destruct_one_pair ||
    specialize_H || rewrite_H); eexists; try
    apply_H
  | S ?m => move_H' n; cbv in *; intros; try
    repeat (destruct_one_ex ||
    destruct_one_pair || specialize_H ||
    rewrite_H); eexists; try apply_H
  end.
```

このタクティクを用いることで、テキスト内の全ての可能な先行詞を探索し証明探索結果として登録できる。証明探索結果は `Type` という型で登録し、(1b) から次のような先行詞を当てはめたテキストが生成される。

(1b-1) The lawyer was reluctant to answer the question.

(1b-2) The witness was reluctant to answer the question.

4 整合性判定手法

§ 3 で生成された照応表現の到達可能なすべての表現を当てはめたテキストから、到達可能な表現のうちの表現が整合性が取れているかを判定する。本研究では、上位分類として接続詞ごとに、下位分類として用法ごとに、照応表現の前後の文脈から、到達可能な対象のうち、ある対象が照応表現の指す対象であるか、または照応表現の指す対象でないかを判定する手法を提案する。

5 おわりに

本研究では、到達可能性をもつ表現が照応表現の先行詞であるか否かを前後の文脈との整合性があるか否かで判定するという手法で実装を行なった。照応探索結果の整合性判定手法では文脈の中でも特に接続詞に

²`destruct_one_ex` と `destruct_one_pair` は、次の Coq 標準ライブラリの中で定義されているものである：<https://coq.inria.fr/doc/V8.18.0/stdlib/Coq.Program.Tactics.html>

着目して、上位分類として接続詞ごとに、下位分類では用法ごとに分類し、`because` の原因・理由の用法や `but`, `although` の逆接の用法および `when`, `after` の時制の用法を定式化することで、より多くの照応解析の問題を扱うことができるようになった。

今後の課題としては、照応解析において全ての先行詞の可能性を枚挙するという手続きを Coq タクティクによって可能な限り自動化する点が挙げられる。

謝辞： 本研究は JST CREST JPMJCR20D2 の助成を受けたものである。

参考文献

- [1] Bekki, D.: Representing Anaphora with Dependent Types, in Asher, N. and Soloviev, S. eds., *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).
- [2] Bekki, D. and Mineshima, K.: Context-Passing and Underspecification in Dependent Type Semantics, in Chatzikyriakidis, S. and Luo, Z. eds., *Modern Perspectives in Type-Theoretical Semantics*, Vol. 98 of *Studies in Linguistics and Philosophy*, pp. 11–41, Springer, Cham (2017).
- [3] Bekki, D. and Satoh, M.: Calculating projections via type checking, in *Proceedings of TYTLES* (2015).
- [4] Bernardy, J.-P. and Chatzikyriakidis, S.: Applied Temporal Analysis: A Complete Run of the Fra-CaS Test Suite, in *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pp. 11–20, Groningen, The Netherlands (online) (2021), Association for Computational Linguistics.
- [5] Bernardy, J. and Chatzikyriakidis, S.: A Wide-Coverage Symbolic Natural Language Inference System, in Hartmann, M. and Plank, B. eds., *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).
- [6] Daido, H. and Bekki, D.: 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).
- [7] 佐藤未歩 依存型意味論の証明探索とその実装 (2016).
- [8] 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).