

CCG パーザ lightblue と証明支援系 Coq の接続

松原 舞 (指導教員：戸次 大介)

1 はじめに

含意関係認識 (Recognizing Textual Entailment, RTE) [3] は、前提が仮説を含意するかを判定するというタスクである。例を以下に示す。

- (1) あるイタリア人が世界最高のテノール歌手になった。
- (2) 世界最高のテノール歌手になったイタリア人がいた。

前提である (1) が仮説である (2) を含意するかを判定するこのタスクに対して、大規模言語モデルを用いる手法が近年注目されている。その一方で、含意関係を判定するだけでなく、その判定の厳密な検証を与えるためには意味論的な推論を用いることが有効である。

先行研究ですでに高階論理のもとで推論するシステム [5] が提案されており、本研究ではそれとは主に統語解析の点で異なるシステムを提案することを目標とする。

提案手法となるパイプラインは、日本語 CCG パーザ lightblue¹ [1, 8] を用いて前提文および仮説文を統語解析し、依存型意味論 (Dependent Type Semantics, DTS) [2] における意味表示を与え、その意味表示を証明支援系 (proof assistant) Coq の式に変換し Coq タクティクにより証明を試みるというものである。

2 先行研究：cgc2lambda

cgc2lambda [5] は日本語・英語の文に対応した高階論理のもとで推論を行うシステムであり、RTE タスクにも用いられている。

日本語の意味的な現象に基づく含意関係のデータセットである JSeM² [4] の一般量量子、複数性、形容詞、動詞、命題的態度の 5 つのセクションに焦点を当てて推論を行い、評価した。スコアは以下の通りである。

Section	#Problem	Accuracy
一般量量子	337	78.0
複数性	41	56.1
形容詞	65	63.1
動詞	36	75.0
命題的態度	44	86.4
Total	523	75.0

エラーは主に構文解析の誤りに由来しており、ここに課題が残されている。

3 研究背景

3.1 lightblue

lightblue は日本語の文を組合せ範疇文法 (Combinatory Categorical Grammar, CCG) に基づいてパーズする統語解析器である。統語構造および DTS の意味合成過程のリストを木構造で出力する。次の図 1 は (1) を lightblue でパーズした際の出力の一部である。

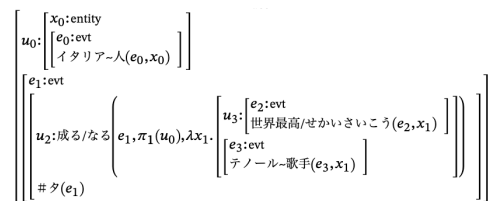


図 1: lightblue によるパーズの出力例

3.2 Coq

Coq [7] は依存型理論に基づく証明支援系であり、タクティクと呼ばれる証明の状態を遷移させるためのコマンドを使って、証明を構成する。Coq は自動証明を行うためのタクティクも備えており、自動定理証明器としても用いることができる。これらのタクティクは Ltac というタクティク言語により扱うことができ、この言語にもともと備わっているタクティクおよび構文を用いるだけで、多様な RTE タスクに対応したタクティクを比較的簡潔に作成することができる。

4 提案手法

先行研究では主に構文解析の精度に課題があると考えられているため、本研究では cgc2lambda で用いられた解析器とは異なる解析器である lightblue を用いた推論のシステムを提案する。

4.1 推論のパイプライン

本研究で採用した推論のパイプラインは以下の図 2 の通りである。



図 2: 推論のパイプライン

まず、日本語の文を lightblue に入力し、パーズする。得られた DTS における意味表示をもとに Coq で推論を行うためには、Coq のための意味表示に変換する必要がある。

DTS における意味表示を Coq のための意味表示に変換するコード、意味表示をもとに命題を作り Coq を呼び出すコードを実装することで lightblue から Coq を呼び出して自動で推論することを実現した。

また、Coq は証明の中で構成されたプログラムを返す。それを lightblue で出力するためには再度 DTS における意味表示に変換するコードを実装する必要がある。これは今後実装する。

lightblue は関数型言語 Haskell で実装されており、これらのコードも同様に Haskell で実装する。

以下、冒頭の (1), (2) を例に実際の推論を紹介する。

¹<https://github.com/DaisukeBekki/lightblue>

²<https://github.com/DaisukeBekki/JSeM>

4.2 DTS を Coq に変換

(1) を lightblue でパーズして得た DTS における意味表示は図 3. のようになる.

```
U.Sigma (U.Sigma (U.Con "entity")
  (U.Sigma (U.Con "evt")
    (U.App (U.App (U.Con "イタリア~人") (U.Var 0)) (U.Var 1))))
(U.Sigma (U.Con "evt")
  (U.Sigma (U.App (U.App (U.App (U.Con "成る/なる") (U.Var 0)) (U.Proj U.Fst (U.Var 1)))
    (U.Lam (U.Sigma (U.Sigma (U.Con "evt")
      (U.App (U.App (U.Con "世界最高/せかいさいこう") (U.Var 0)) (U.Var 1)))
        (U.Sigma (U.Con "evt")
          (U.App (U.App (U.Con "テノール~歌手") (U.Var 0)) (U.Var 2))))))
    (U.App (U.Con "#タ") (U.Var 1))))
```

図 3: DTS における意味表示

この DTS における意味表示を, 実装したコードで Coq のための意味表示に変換すると図 4. のようになる.

```
{u0: {x0:Entity &
  {e0:Evt &
    _c_イタリア人 x0 e0}} &
{e1:Evt &
  {u1:_c_成る_なる (fun x1 => {u2:{e2:Evt &
    _c_世界最高_せかいさいこう x1 e2} &
    {e3:Evt &
      _c_テノール歌手 x1 e3}}) (projT1 u0 e1 &
    _c_MCNTa e1}}}
```

図 4: Coq のための意味表示

4.3 推論

(2) も同様に, Coq のための意味表示に変換する. これらの意味表示を元に, 実装したコードで (前提文の Coq のための意味表示)→(仮説文の Coq のための意味表示) という命題を作り, Coq を呼び出して推論を行う (付録). また, 語義に関する常識的な公理を追加している.

5 今後の計画

5.1 実装

今後の実装の計画としてはまず次のものが挙げられる. 現時点では Coq の推論の結果は直接標準出力に表示される. そうした推論結果を lightblue が受け取ることで変換し lightblue から出力できるよう実装を進める. このように実装することで, たとえばある含意関係の正しさを検証する証明が Coq から得られた場合, それを木構造として出力できるようになる.

次に, Coq タクティクの実装が挙げられる. 現時点では, Coq の中で (1) から (2) を自動で証明することはできておらず, 部分的に手動で証明を完成させている. 自動証明のためには, 語義に関する常識的公理を自動で適用する必要がある, そうした適用の際には公理の引数も自動で構成する必要がある. 以上の手続きのために, Coq タクティクがもつ機能を精査し, 公理適用のためのタクティクを定義する. そして, すでに提案されている自動証明のためのタクティク [6] と接続し, Coq での自動証明を実現する.

5.2 評価

本研究が提案する RTE のためのパイプラインを評価するために, 先行研究である ccg2lambda の評価データとなっている JSeM のセクションを用いる. そして, 本研究の提案手法と ccg2lambda のスコアを比較しつ

つ, エラー分析を行う.

6 終わりに

本研究では, CCG による統語解析と DTS における意味合成に基づく推論システムの実現に向けて, 日本語 CCG パーザ lightblue と証明支援系 Coq を接続して得られる推論パイプラインを提案した. このパイプラインは, さまざまな言語現象を説明できる自然言語の意味論である DTS を用いて RTE タスクにアプローチすることを可能にする. 上述の計画に沿って引き続き実装を進め, 本システムの実現を目指す.

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] Bekki, D. and Kawazoe, A.: Implementing Variable Vectors in a CCG Parser, in Retoré, C. and Pogodalla, S. eds., *Logical Aspects of Computational Linguistics (9th international conference, LACL2016, Nancy, France, December 2016 Proceedings)*, pp. 52–67, Springer, Heiderburg (2016).
- [2] Bekki, D. and Mineshima, K.: Context-Passing and Underspecification in Dependent Type Semantics, in *Modern Perspectives in Type-Theoretical Semantics*, Springer (2017).
- [3] Ido Dagan, M. S., Dan Roth and Zanzotto, simo F. M.: *Recognizing Textual Entailment: Models and Applications*, Morgan & Claypool Publishers (2013).
- [4] Kawazoe, A., Tanaka, R., Mineshima, K. and Bekki, D.: An Inference Problem Set for Evaluating Semantic Theories and Semantic Processing Systems for Japanese, in *LENLS12*, pp. 67–73 (2015).
- [5] Koji Mineshima, P. M.-G. Y. M., Ribeka Tanaka and Bekki, D.: Building compositional semantics and higher-order inference system for a wide-coverage Japanese CCG parser, in *EMNLP* (2016).
- [6] Kosaihira, H., Takahashi, Y. and Bekki, D.: Implementation of Anaphora Resolution Using the Refine Tactic of Coq, Abstract Booklet of the 2023 Workshop on Proof Theory and its Applications, University of Barcelona (2023), https://www.ub.edu/prooftheory/event/tps2023workshop/Booklet_of_abstracts_TPSW23.pdf.
- [7] The Coq Development Team, : The Coq proof assistant reference manual: Version 8.18 (2023), <http://coq.inria.fr>.
- [8] 戸次大介: 日本語文法の形式理論, くろしお出版 (2010).

```

lightblue -- -zsh -- 82x48
[matsubaramai@MacBook-Air-7 lightblue % stack run dts2coq
Welcome to Coq 8.17.1
Entity is declared
Evt is declared
_c_イタリア人 is declared
_c_世界最高_せかいさいこう is declared
_c_テノxmdashxル歌手 is declared
_c_成る_なる is declared
_c_MCNタ is declared
naru is declared
naru2 is declared
1 goal

=====
(exists
 (u0 : exists (x0 : Entity) (e0 : Evt), _c_イタリア人 x0 e0)
 (e1 : Evt) (_ : _c_成る_なる
  (fun x1 : Entity =>
   exists
    (_ : exists e2 : Evt, _c_世界最高_せかいさいこう x1 e2)
    (e3 : Evt), _c_テノxmdashxル歌手 x1 e3)
   (ex_proj1 u0) e1), _c_MCNタ e1) ->
exists
 (x0 : Entity) (_ : exists e0 : Evt, _c_世界最高_せかいさいこう x0 e0)
 (_ : exists
  (e1 : Evt) (_ : _c_成る_なる
   (fun x1 : Entity =>
    exists e2 : Evt, _c_テノxmdashxル歌手 x1 e2) x0 e1),
   _c_MCNタ e1) (e3 : Evt), _c_イタリア人 x0 e3)
1 goal

H : exists
 (u0 : exists (x0 : Entity) (e0 : Evt), _c_イタリア人 x0 e0)
 (e1 : Evt) (_ : _c_成る_なる
  (fun x1 : Entity =>
   exists
    (_ : exists e2 : Evt, _c_世界最高_せかいさいこう x1 e2)

    (e3 : Evt), _c_テノxmdashxル歌手 x1 e3)
    (ex_proj1 u0) e1), _c_MCNタ e1)
=====
exists
 (x0 : Entity) (_ : exists e0 : Evt, _c_世界最高_せかいさいこう x0 e0)
 (_ : exists
  (e1 : Evt) (_ : _c_成る_なる
   (fun x1 : Entity =>
    exists e2 : Evt, _c_テノxmdashxル歌手 x1 e2) x0 e1),
   _c_MCNタ e1) (e3 : Evt), _c_イタリア人 x0 e3)

```

図 5: lightblue から Coq を呼び出して行った推論の出力 (抜粋)

```

Theorem trm : {u0: {x0:Entity &
  {e0:Evt &
    _c_イタリア人 x0 e0}} &
  {e1:Evt &
    {u1:_c_成る_なる (fun x1 => {u2: {e2:Evt &
      _c_世界最高_せかいさいこう x1 e2} &
      {e3:Evt &
        _c_テノxmdashxル歌手 x1 e3}}) (projT1 u0) e1 &
      _c_MCNタ e1}}}}
->
{x0:Entity &
 {u0:{e0:Evt &
  _c_世界最高_せかいさいこう x0 e0} &
 {u1:{e1:Evt &
  {u2:_c_成る_なる (fun x1 => {e2:Evt &
    _c_テノxmdashxル歌手 x1 e2}) x0 e1 &
    _c_MCNタ e1}} &
 {e3 : Evt &
  _c_イタリア人 x0 e3}}}}}.

Proof.
  tacl.
  exists X.
  split.
  naruTac.
  tacl.
  split; tacl.
  exists X0.
  split; tacl.
  naru2Tac.
  assumption.
Defined.

```

図 6: Coq で行われる推論