

# CCG に基づく否定スコープ認識

小島 健太郎<sup>1</sup> 加藤 芳秀<sup>2</sup> 松原 茂樹<sup>1,2</sup>

<sup>1</sup> 名古屋大学情報学部 <sup>2</sup> 名古屋大学情報連携推進本部

kojima.kentaro.a0@es.mail.nagoya-u.ac.jp

## 概要

否定スコープ認識とは否定要素の影響が及ぶ範囲である否定スコープを特定するタスクである。このタスクはヒューリスティックなルールに基づく手法やニューラルモデルに基づく手法など様々なアプローチで取り組まれている。しかし、言語理論に基づいて否定スコープを求める研究はあまり行われていない。本論文では、否定スコープを言語理論に基づいて求めるアプローチの一つとして CCG に基づく手法を提案する。本手法では、意味表現としてラムダ式を採用し、否定要素とそのスコープの関係を、ラムダ式の関数と引数の関係に関する一つの原理で説明する。否定スコープデータセットを用いて本手法の性能を評価し、手法の課題を明らかにする。

## 1 はじめに

否定とは意味を反転させる言語現象である。否定は、否定を引き起こす**否定要素** (negation cue) とその影響が及ぶ範囲である**否定スコープ** (negation scope) から構成される。例えば、文 1 においては、No が否定要素に、one knows the truth が否定スコープに該当する。以下、否定要素を太字で、否定スコープを下線で表現する。

### 1. No one knows the truth.

否定スコープ認識は、対話や質問応答などの言語理解を要するタスクにおいて重要な要素となる。

否定スコープ認識は、\*SEM2012 Shared Task において取り上げられて以来、様々なアプローチで取り組まれている。\*SEM2012 Shared Task が発表された当時、構文構造に基づくヒューリスティックな手法 [1] が高い性能を達成し [2]、その後の取り組みでも高水準の性能に達している [3]。また、ニューラルモデルや言語モデルを用いて構文構造を明示的に使わずに否定スコープを認識する手法が開発されてい

る (例えば, [4, 5])。

一方、言語理論に基づいた否定スコープ認識として、依存構造から一階述語論理式への変換により否定演算子のスコープとして捉える方法 [6] などが提案されているがまだ数は少ない。言語理論に基づく否定スコープ認識は、ニューラルモデルや言語モデルを用いる方法よりも解釈性や説明可能性に優れるという利点がある。また、ヒューリスティックなルールベースの手法と比べても、理論的な裏付けを持って否定スコープを求められるという利点がある。

そこで本論文では、否定スコープ認識を言語理論に基づいて行うアプローチの一つとして CCG (Combinatory Categorical Grammar) [7] に基づく手法を提案する。CCG においては、構文構造と意味合成に対応関係があり、意味表現はラムダ計算に基づき合成される。本手法では、意味表現としてラムダ式を採用し、否定要素とそのスコープの関係を、ラムダ式の関数と引数の関係に関する一つの原理で説明する。これにより説明可能性と一貫性の高い否定スコープ認識の実現が期待できる。

## 2 CCG

本章では、提案手法のベースとなる CCG について説明する。

CCG は単語ごとにカテゴリを付与する文法である。カテゴリには S や NP などの基本カテゴリと、カテゴリを / や \ で結合した複合カテゴリがある。カテゴリ  $X/Y$  は右側にカテゴリ  $Y$  が来るとカテゴリ  $X$  になることを表す。  $X\backslash Y$  は左側にカテゴリ  $Y$  が来るとカテゴリ  $X$  になることを表す。CCG ではカテゴリの合成と意味表現の合成が一对一に対応しており、意味表現の合成はラムダ計算として形式化できる。文 1 の CCG 構文木は図 1 のようになる。

図 1 におけるすべてのカテゴリの合成は関数適用 (function application) と呼ばれ、意味表現は、以下の規則に従って合成される。

No	one	knows	the	truth
$(S/(S\backslash NP))/N$	$N$	$(S\backslash NP)/NP$	$NP/N$	$N$
: <i>No'</i>	: <i>one'</i>	: <i>knows'</i>	: <i>the'</i>	: <i>truth'</i>
$S/(S\backslash NP): No'one'$			$NP: the'truth'$	
			$S\backslash NP: knows'(the'truth')$	
$S: No'one'(knows'(the'truth'))$				

図1 文1に対する CCG 構文木

- $X/Y:f \quad Y:a \Rightarrow X:f a$
- $Y:a \quad X\backslash Y:f \Rightarrow X:f a$

ここで、 $f$  と  $a$  は意味表現（ラムダ式）である。他にも CCG には以下の規則などがある。

- 関数合成
  - $X/Y:f \quad Y|_1 Z_1 \cdots |_d Z_d:g \Rightarrow$   
 $X|_1 Z_1 \cdots |_d Z_d:\lambda z_d \cdots \lambda z_1.f(gz_d \cdots z_1)$
  - $Y|_1 Z_1 \cdots |_d Z_d:g \quad X\backslash Y:f \Rightarrow$   
 $X|_1 Z_1 \cdots |_d Z_d:\lambda z_d \cdots \lambda z_1.f(gz_d \cdots z_1)$
- 型繰り上げ
  - $X:a \Rightarrow T/(T\backslash X):\lambda f.f a$
  - $X:a \Rightarrow T\backslash(T/X):\lambda f.f a$

ここで、 $|_i \in \{/, \backslash\}$ 、 $Z_i (1 \leq i \leq d)$  はカテゴリである。これ以降の節で用いる記法をいくつか説明する。 $Y|_1 Z_1 \cdots |_d Z_d$  について  $|_1 Z_1 \cdots |_d Z_d$  を引数スタック (argument stack) といい [8]、ギリシャ文字を使って表す。すなわち、 $Y|_1 Z_1 \cdots |_d Z_d = Y\alpha$  のような表記を用いる。また  $|\alpha| = d$  と定義する。カテゴリ  $X$  が基本カテゴリ  $Y$  を使って  $X = Y\alpha$  と表されるとき、カテゴリ  $X$  のアリティは以下のように定義される。

$$\text{arity}(X) = |\alpha| \quad (1)$$

### 3 CCG に基づく否定スコープ認識

本節では CCG を用いて否定スコープを認識する手法を提案する。また、現在の主要な CCG 構文解析器が採用する CCGbank の構文木について、本手法で用いる際の留意点とその対応策を述べる。提案手法の概略は以下の通りである。

1. 文に対する CCG 構文木を CCG 構文解析器により生成する
2. CCG 構文木の葉（単語）にラムダ式を付与する
3. CCG 構文木に従ってラムダ式を合成し、得られたラムダ式を  $\beta$  簡約する
4. 得られたラムダ式に基づき否定スコープを認識する

以下では、提案手法のキーポイントである、2. と 4.

について説明する。

#### 3.1 単語に対するラムダ式の付与

本手法では、特定の意味理論に基づく意味表現は採用せず、関数と引数の関係のみを捉えるラムダ式を採用する。具体的には、単語  $w$  に対して対応する記号  $w'$  を用意し、それを意味表現として付与する。

##### 3.1.1 付加詞

付加詞 (adjunct) は CCG において  $X\alpha|X\alpha$ （ここで  $X$  は基本カテゴリ）という形のカテゴリとして表される。このとき、左側のカテゴリ  $X\alpha$  と右側のカテゴリ  $X\alpha$  において  $\alpha$  の情報共有がなされる（例えば、 $(S\backslash NP)/(S\backslash NP)$  において、左側の  $S\backslash NP$  が受け取る  $NP$  に関する情報は、右側の  $S\backslash NP$  にも渡さなければならない）。そのため、付加詞の一部となるカテゴリ  $(X\alpha|X\alpha)\beta$  の単語  $w$  に対しては、以下のラムダ式を付与する。

$$\lambda b_1 \cdots \lambda b_{|\beta|} \lambda x \lambda a_1 \cdots \lambda a_{|\alpha|} . M \quad (2)$$

ただし、

$$M \equiv w' b_1 \cdots b_{|\beta|} (x a_1 \cdots a_{|\alpha|}) a_1 \cdots a_{|\alpha|} \quad (3)$$

このラムダ式にはまず引数スタック  $\beta$  にあるカテゴリに対するラムダ式が  $b_1, \dots, b_{|\beta|}$  に順次代入される。その後、 $X\alpha|X\alpha$  の右側の  $X\alpha$  に当たるラムダ式が  $x$  に代入され、そのラムダ式に  $a_1, \dots, a_{|\alpha|}$  が代入されるので、右側の  $\alpha$  の情報と左側の  $\alpha$  の情報が共有される。

##### 3.1.2 等位接続詞

等位接続詞は CCG では conj という特殊なカテゴリで管理され、同じカテゴリ同士をつなぐ役割を持っている。複合カテゴリを等位項とする等位構造において、等位構造全体が受け取る情報を等位項に配分する必要がある。（例えば動詞句の等位構造はカテゴリが  $S\backslash NP$  であり、その等位項は、主語  $NP$  の情報を受け取る必要がある）。付加詞と同様にこの情報の共有もラムダ式に反映する必要がある。そのため、等位構造のカテゴリが  $X$  で  $\text{arity}(X) = n$  のとき、等位接続詞のラムダ式は以下ようになる。

$$\lambda x_1 \lambda x_2 \lambda y_1 \cdots \lambda y_n . w' (x_1 y_1 \cdots y_n) (x_2 y_1 \cdots y_n) \quad (4)$$

このラムダ式にはまず、右の等位項に当たるラムダ式が  $x_1$  に代入される。 $x_1$  は変項  $y_1, \dots, y_n$  を介して等位構造全体が受け取るラムダ式を受け取る。

$x_2$  には左の等位項のラムダ式が代入され、同様に  $y_1, \dots, y_n$  を介して等位構造全体が受け取るラムダ式を受け取る。

### 3.2 ラムダ式に基づくスコープ認識

CCG 構文木の根に対して与えられるラムダ式を  $\beta$  簡約し、結果として得られるラムダ式  $l$  から次のようにして否定要素に対する否定スコープを同定する。

- $l$  に含まれる、関数適用の形をとった任意の部分ラムダ式  $fa$  について、 $func(f)$  が否定要素  $c$  に対応する意味表現の記号であり、 $w$  に対応する意味表現の記号  $w'$  が  $Sym(a)$  の要素ならば、 $w$  は  $c$  のスコープに含まれる。

ここで、 $func$  と  $Sym$  は以下のように定義する ( $M, N$  はラムダ式である)。

$$func(f) = \begin{cases} f & (f \text{ は記号}) \\ func(M) & (f = MN) \\ undefined & (f = \lambda x.M) \end{cases} \quad (5)$$

$$Sym(a) = \begin{cases} \{a\} & (a \text{ は記号}) \\ Sym(M) \cup Sym(N) & (a = MN) \\ Sym(M) & (a = \lambda x.M) \end{cases} \quad (6)$$

### 3.3 CCGbank

多くの CCG 構文解析器は CCGbank[9] を用いて学習されており、今回の手法を適用するにあたって以下の留意すべき点がある

- 量子子 no のカテゴリ
- type changing 規則
- 非局所的依存関係 (non-local dependencies)

以下ではこれらの取り扱いについて説明する。

#### 3.3.1 量子子 no のカテゴリ

CCGbank では文 1 の CCG 構文木は図 2 のように表現される。しかし、量子子に対する解析結果としては図 1 の構文木の方が望ましい。そこで no や, nothing, nobody などのカテゴリを次のように修正する。

- 構文木で no で始まる名詞句 (NP) が  $X|_1NP$  (ただし、 $X \neq NP$ ) の引数となるとき、この NP に対応するすべての NP を  $X|_2(X|_1NP)$  ( $|_2$  は  $|_1$  の向きを反転させたスラッシュ) に置き換える。

No	one	knows	the	truth
NP/N	N	(S\NP)/NP	NP/N	N
: <i>No'</i>	: <i>one'</i>	: <i>knows'</i>	: <i>the'</i>	: <i>truth'</i>
$\overline{\text{NP : } No'one'}$			$\overline{\text{NP : } the' truth'}$	
$\overline{\text{S\NP : } knows' (the' truth')}$				
$\overline{\text{S : } knows' (the' truth') (No'one')}$				

図 2 CCGbank での文 1 に対する構文木

表 1 提案手法の実験結果と他手法との比較

Method	Scope-F1	Token-F1
本手法	52.36	72.14
Fancellu ら [5]	77.77	88.72
Khandelwal ら [4]	-	92.36

#### 3.3.2 type changing 規則

CCGbank には本来の CCG がない type changing 規則が存在する。これはカテゴリ  $X$  をカテゴリ  $Y$  に変える規則である。提案手法では、字面上現れない単語がカテゴリ  $Y/X$  を取り、これをカテゴリ  $X$  に関数適用しカテゴリ  $Y$  になったと解釈して処理する。

#### 3.3.3 非局所的依存関係

CCGbank では付加詞以外でカテゴリの情報が共有されることがある。例えば、food which John likes の which に  $(NP\backslash NP)/(S\backslash NP)$  というカテゴリが付与されたとき、 $(S\backslash NP)$  の NP と  $(NP\backslash NP)$  の右側の NP は同じもの (food) を指している。このような非局所的依存関係 (non-local dependencies) を扱うために、CCGbank の User's Manual[10] を参考にして付加詞と同様に扱う。例えば、この which には以下のラムダ式を与える。

$$\lambda x \lambda y. which'(xy)y \quad (7)$$

## 4 実験

提案した手法がどの程度否定スコープを捉えることができるかを検証するために、\*SEM2012 Shared Task のデータセット [2] の検証用データを用いてその性能を評価した。CCG 構文解析器として Holographic CCG[11] を用いた。これは CCGbank[9] に従った CCG 構文木を出力する。

### 4.1 結果

実験の結果を表 1 に示す。参考値として従来手法のテストデータに対する結果も併せて示す。評価指標はスコープレベルの F1 とトークンレベルの F1 であり、\*SEM2012 Shared Task で配布された評価用のプログラムを用いた。

表2 カテゴリごとのトークンレベルの適合率, 再現率, F 値

カテゴリ	否定要素の例	出現数	Pre.	Rec.	F1.
(S\NP)\(S\NP)	not, never など	58	90.66	93.20	91.91
((S <sub>dcl</sub> \NP)\((S <sub>dcl</sub> \NP)/NP))/N	no	19	100.00	50.67	67.26
S <sub>adj</sub> \NP	improper, unknown など	12	60.71	27.87	38.20
N/N	unpleasant, unbrushed など	11	100.00	33.33	50.00
(S\NP)/(S\NP)	not, never など	9	94.81	85.88	90.12
conj	nor	6	33.33	78.38	46.77
(S <sub>dcl</sub> \NP)\((S <sub>dcl</sub> \NP)/NP)	nothing	6	100.00	19.44	32.55
((S <sub>b</sub> \NP)\((S <sub>b</sub> \NP)/NP))/N	no	5	94.44	36.96	53.13
S/S	no, never	5	17.74	100.00	30.14

先行研究と直接的な比較はできないものの, 提案手法の F 値は低く, 提案手法に問題があると考えられる. その傾向を分析するために否定要素のカテゴリごとにトークンレベルの適合率 (Pre.), 再現率 (Rec.), F 値 (F1.) を算出した. その結果を表 2 に示す (出現数が 5 未満のカテゴリは除いた).

多くのカテゴリにおいて, 適合率が高い一方で再現率が低いという傾向が見られた. 6 つのカテゴリで適合率が 90% を上回っており, そのうち 3 つのカテゴリについては出現数が少ないとはいえ適合率が 100% であった. 高い適合率と低い再現率は, 提案手法がスコープに含めるべき単語を含めない何らかの要因があることを意味している.

## 4.2 エラー分析

提案手法の問題点を明らかにするために F 値が低く, 出現数が多い ((S<sub>dcl</sub>\NP)\((S<sub>dcl</sub>\NP)/NP))/N と S<sub>adj</sub>\NP と N/N について, 実際の文, 及び CCG 構文木を確認し, 誤りの原因を調査した.

((S<sub>dcl</sub>\NP)\((S<sub>dcl</sub>\NP)/NP))/N は目的語の no に付与されるカテゴリである. このカテゴリが付与された否定要素を含む文の正解スコープと予測スコープは例えば以下である.

- 正解: he made no remark as to the contents.
- 予測: he made no remark as to the contents.

この例のように正解スコープでは後置修飾句が含まれているが, 予測スコープでは含まれていないケースが確認された. これは, CCGbank において, 後置修飾がすべて非制限用法の構造 (8) を取るためと考えられる. 制限用法の構造 (9) を与えるような CCGbank の再構築 [12] ができれば, この問題の解決が期待できる.

(NP(NP(NP/N no)(N remark))(NP\NP as to the contents)) (8)

(NP(NP/N no)(N(N remark)(N\N as to the contents))) (9)

N/N は主に unconventional や unbrushed などの否定の接頭辞を含む形容詞が名詞を修飾する場合に付与される. 例えば以下のようなケースである.

- 正解: his own unconventional appearance.
- 予測: his own unconventional appearance.

この例のように正解スコープでは所有格や冠詞などが含まれているが, 予測スコープでは含まれていないというケースが確認された. この原因の一つとして, CCGbank が名詞句内の構造 (上の例では, appearance と his own の間の述語と項の関係) を十分に表現できていないことが挙げられる.

S<sub>adj</sub>\NP は, 補語として用いられている形容詞に付与される. このカテゴリが付与された否定要素を含む文の正解スコープと予測スコープは例えば以下である.

- 正解: It is most improper
- 予測: It is most improper

この例のようにこのカテゴリの引数は主語 (NP) のみであるため, be 動詞がスコープに入らないというケースが確認された.

## 5 おわりに

本論文では否定スコープを CCG に基づいて認識する手法を提案した. 提案手法の性能は, 従来の手法と比較して低いことを確認した. エラー分析の結果, CCGbank の構文木が否定スコープ認識に適していないケースを確認した.

この手法の精度を上げていくためには, CCG の構文木やカテゴリを否定スコープ認識に適したものに修正する必要があると考えられる. 今後, 文献 [12] の CCGbank の再構築のアイデアを取り入れたアプローチについて検討したい.

## 謝辞

本研究は、一部、科学研究費補助金基盤研究（C）（No. 22K12148）により実施したものである。

## 参考文献

- [1] Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. UiO1: Constituent-based discriminative ranking for negation resolution. In **\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)**, pp. 310–318, Montréal, Canada, 7–8 June 2012. Association for Computational Linguistics.
- [2] Roser Morante and Eduardo Blanco. \*SEM 2012 shared task: Resolving the scope and focus of negation. In **\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)**, pp. 265–274, Montréal, Canada, 7–8 June 2012. Association for Computational Linguistics.
- [3] Asahi Yoshida, Yoshihide Kato, and Shigeki Matsubara. Revisiting syntax-based approach in negation scope resolution. In **Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (\*SEM 2023)**, pp. 18–23, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [4] Aditya Khandelwal and Suraj Sawant. NegBERT: A transfer learning approach for negation detection and scope resolution. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 5739–5748, Marseille, France, May 2020. European Language Resources Association.
- [5] Federico Fancellu, Adam Lopez, and Bonnie Webber. Neural networks for negation scope detection. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 495–504, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [6] Federico Fancellu, Siva Reddy, Adam Lopez, and Bonnie Webber. Universal Dependencies to logical form with negation scope. In **Proceedings of the Workshop Computational Semantics Beyond Events and Roles**, pp. 22–32, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [7] Mark Steedman. **The syntactic process**. MIT Press, Cambridge, MA, USA, 2000.
- [8] Marco Kuhlmann and Giorgio Satta. A new parsing algorithm for combinatory categorial grammar. **Transactions of the Association for Computational Linguistics**, Vol. 2, pp. 405–418, 10 2014.
- [9] Julia Hockenmaier and Mark Steedman. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. **Computational Linguistics**, Vol. 33, No. 3, pp. 355–396, 2007.
- [10] J. Hockenmaier and Mark Steedman. Ccgbank: User’s manual. 2005.
- [11] Ryosuke Yamaki, Tadahiro Taniguchi, and Daichi Mochihashi. Holographic CCG parsing. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 262–276, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [12] Matthew Honnibal, James R. Curran, and Johan Bos. Re-banking CCGbank for improved NP interpretation. In **Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics**, pp. 207–215, Uppsala, Sweden, July 2010. Association for Computational Linguistics.