係り受け木を用いたツリーバンク自動生成による CCG 解析分野適応

吉川 将司 [1]　能地 宏 [2]　峯島 宏次 [3]　戸次 大介 [3]
奈良先端科学技術大学院大学 [1]　産業技術総合研究所 [2]　お茶の水女子大学 [3]

yoshikawa.masashi.yh8@is.naist.jp, hiroshi.noji@aist.go.jp,
mineshima.koji@ocha.ac.jp, bekki@is.ocha.ac.jp

1　はじめに

自然言語の文に対して、その意味を、論理式やグラフ構造で付与したデータセットを構築することは、高コストであり、それを用いて教師あり学習により意味解析器を構築するようなアプローチでは、応用範囲に限界がある．一方で近年、構文解析技術の向上と言語学理論の発展により、組み合わせ範疇文法 [7, 24](Combinatory Categorial Grammar; 以降 CCG) と高階論理の組み合わせによる意味解析技術が発展している．このアプローチでは、CCG 解析器の学習は必要であるが、CCG 木から高階論理による意味表示への変換は、教師なしの方法で行うことができる．また出力の論理式は、量化や否定の論理構造を捉え、それを用いた推論システム [19] は、一部の含意関係認識タスクでは最新の深層モデルに劣らない精度を示すなど有用である．

本研究は、後者のアプローチを更に推進し、医療分野や音声会話など、より広範なテキストに対する高度な推論システムの構築を目的とする．その実現のためには、ニューステキスト (Wall Street Journal; WSJ [3, 18]) のみから学習した既存の解析器が、それらの分野のテキストに対して頑健であることが重要であるが、学習データに現れない未知の語彙（病名、専門用語）や文構造（言い淀み、分野特有の表現）が現れた場合に、解析性能が大きく下がってしまう．この問題を解決するには解析器の分野適応が不可欠である．

研究 [6] では、CCG カテゴリが持つ豊富な情報量を活かし、新たなテキストに対し終端のカテゴリ列のみ付与したデータから学習することで、新たな分野への CCG 解析器の分野適応が可能であることが示されている．しかしながら、例えば医療分野テキストに対し CCG カテゴリを付与するには、医療テキストを理解し、かつ文法理論を熟知している必要がある．また、終端の情報だけでは、近年のより強力な解析手法 [12, 14] を学習させることができないという問題もある．

そこで本研究では、これらの問題を同時に解決するため、アノテーションコストが安価で、既に大量のコーパス資源が存在する係り受け木を利用することで、新たな分野に対する高品質な CCG コーパス（CCGBank）を自動構築する手法を提案する．具体的な手法としては、まず WSJ 上のアノテーションを利用して、係り受け木から CCG 木への深層変換モデルを学習する（図 1a）．新たな分野の係り受けコーパスをその変換モデルで CCG 木へ変換し（1b）、そのデータで既存の CCG 解析器を再学習させることでその分野のテキストに頑健な解析器を構築する（1c）．

近年、深層学習を用いてグラフや木などの構造を扱う技術が発展しており、本研究はこれらを活用することができる．また、文から CCG 木への変換である一般的な構文解析のボトルネックは、大量の未知語への対処であるが、係り受け木を入力とする木構造間の変換モデルでは、文の構造については分野間での差異がほとんどなく、よって「未知の構造」というものが存在しないと考えられるため、提案法によって高品質な CCGBank を生成できると期待できる．

提案法の汎用性を示すため、既存の CCG 解析の分野適応のベンチマークである医療、疑問文 [6] に加え、Switchboard コーパス [8] を変換して構築した CCGbank を用い、話し言葉解析の実験を行った．また、数学の問題文 [17] の解析実験データを構築し、数学入試問題の意味解析 [21] において完全にデータ駆動の手法が貢献できるかを評価した．

2　変換モデル

A* CCG 解析の手法 [14] を応用することによって、文 $\boldsymbol{x} = (x_1, ..., x_N)$ の各単語に対するベクトル表現があれば、それを用いて CCG 木をデコードすることができる．木構造間の変換モデルを構成するには、単語に加え、係り受け木の情報もベクトルにエンコードすればよく、そのために本研究では双方向 Tree LSTM [5] を利用する．[14] によれば、CCG 木 $\boldsymbol{y}$ の構造は終端の CCG カテゴリの列 $\boldsymbol{c} = (c_1, ..., c_N)$ と CCG 木の句構造を表した係り受け構造 $\boldsymbol{d} = (d_1, ..., d_N)$ が決まればほぼ一意に決まるため、[1]文 $\boldsymbol{x}$ の係り受け木を $\boldsymbol{z}$ とすると、以下のように確率モデルを定義できる．[2]

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{N} p_{tag}(c_i|\boldsymbol{x}, \boldsymbol{z}) \prod_{i=1}^{N} p_{dep}(d_i|\boldsymbol{x}, \boldsymbol{z}). \quad (1)$$

[1]木が一項規則 (unary rule) を含む場合その限りではない．
[2]ここで各 c_i と d_i の独立性を仮定している．

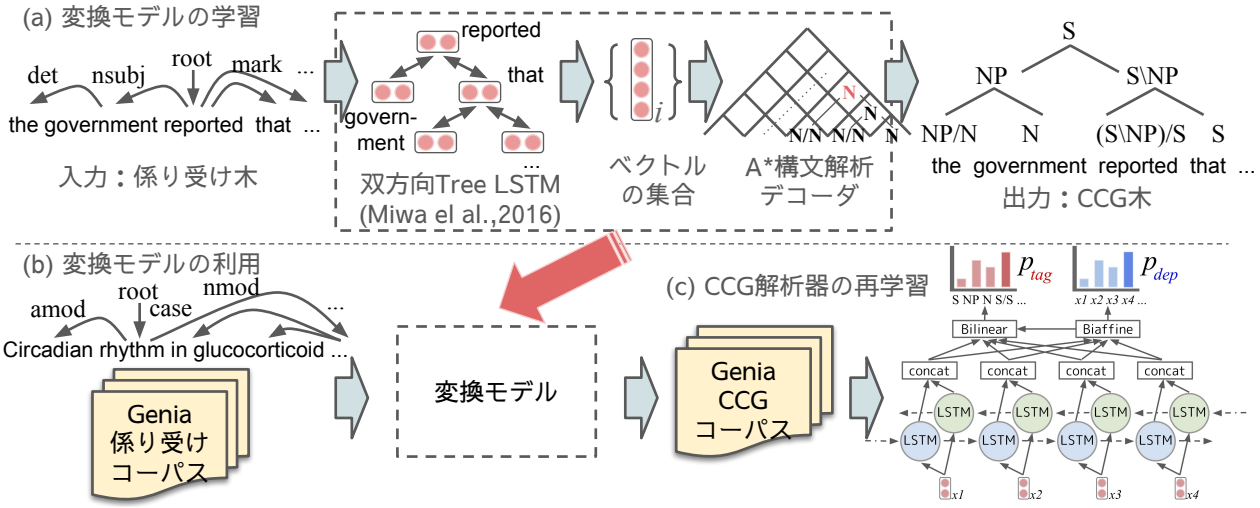

図 1: 提案法の概要図.

以下、入力の係り受け木を三つ組 $\boldsymbol{z} = (\boldsymbol{p}, \boldsymbol{d}', \boldsymbol{\ell})$ とし、変換モデルの詳細を述べる．各 x_i に対し、p_i はその品詞、$d'_i \in \{0, ..., N\}$ は係り受け木における親のインデックス（0 はルートとする）、また ℓ_i は対応する係り受けエッジのラベルである.[3]

係り受け木をエンコードする双方向 Tree LSTM は 2 つの Tree LSTM から成る．ボトムアップ Tree LSTM は、各単語 x_i について、係り受け木における子の隠れベクトル $\{\boldsymbol{h}^{\uparrow}_j | d_j = i\}$ と、x_i の表現ベクトル $\boldsymbol{e}_i$ から隠れベクトル $\boldsymbol{h}^{\uparrow}_i$ を再帰的に計算する．トップダウン Tree LSTM は、親の隠れベクトル $\boldsymbol{h}^{\downarrow}_{d_i}$ と $\boldsymbol{e}_i$ から、ベクトル $\boldsymbol{h}^{\downarrow}_i$ を計算し、双方向 Tree LSTM 全体では各 x_i に対し隠れベクトルの連結 $\boldsymbol{h}_i = \boldsymbol{h}^{\uparrow}_i \oplus \boldsymbol{h}^{\downarrow}_i$ を返す.

提案法では以下のように係り受け構造をエンコードする．ここで、$\boldsymbol{e}_v$ は v に対応する表現ベクトルで、Ω は双方向 LSTM、$\Xi_{\boldsymbol{d}'}$ は双方向 Tree LSTM の一連の操作の略記とする.

$$\boldsymbol{e}_1, ..., \boldsymbol{e}_N = \Omega(\boldsymbol{e}_{p_1} \oplus \boldsymbol{e}_{x_1}, ..., \boldsymbol{e}_{p_N} \oplus \boldsymbol{e}_{x_N}),$$
$$\boldsymbol{h}_1, ..., \boldsymbol{h}_N = \Xi_{\boldsymbol{d}'}(\boldsymbol{e}_1 \oplus \boldsymbol{e}_{\ell_1}, ..., \boldsymbol{e}_N \oplus \boldsymbol{e}_{\ell_N}).$$

デコーダは、[14] と同様に $\{\boldsymbol{h}_i\}_{i \in [1,N]}$ を基にそれぞれ Biaffine 層 [2] と双線形写像を用いて $p_{dep|tag}$ を計算し、A*構文解析により $P(\boldsymbol{y}|\cdot)$ を最大化する CCG 木 $\boldsymbol{y}$ を出力する．A*構文解析は、グラフ探索問題に対する A*アルゴリズムの一般化であり、優先度付きキューを使って次に探索する部分木を制御する．このようなデコーダを採用する強みには、出力結果の最適性の保証に加え、次に述べる制約付きデコーダがある.

2.1 制約付きデコーダ

提案法は、機械学習によるツリーバンクの自動生成法であるが、既存の言語資源を利用して生成される木の構造を制御したい場合が多く考えられる．例えば、医療分野の CCGBank を生成するときに、固有表現に誤って NP 以外のカテゴリが付与されることを、病名辞書等を用いることによって防ぎたい場合などである．提案法の A*構文解析に基づくデコーダでは、探索空間に対し制約をかけることで、そのような制御を行いながら木構造の生成を行うことが可能である.

制約はカテゴリ c、始点 i、終点 j の三つ組 (c, i, j) であり、単語列 $x_i, ..., x_j$ から成るカテゴリ c の CCG 句を表す．制約付きデコーダは、A*構文解析において優先度付きキューに探索点（部分木；同様にルートのカテゴリと区間で (c', k, l) と表す）を追加するときに次の条件を満たすものをブロックすることで行われる.

- 区間が重複する（$i < k \leq j < l \vee k < i \leq l < j$）.
- $i = k$ かつ $j = l$ のとき：カテゴリが異なる（$c \neq c'$）、もしくはあるカテゴリ c'' について $c'' \to c'$ なる一項規則が文法に存在しない.

最後の一項規則についての条件は、名詞句を NP にする制約で (NP (N dog)) などの構造を誤って除外しないために必要である．同時に複数の制約を課す場合には、優先度付きキュー追加時にそれらすべてに対し上の条件が満たされるかをチェックすれば良い．また、木の終端の（単語に付与される）カテゴリを c にしたい場合は、p_{tag} の値を操作して行う：$p_{tag}(c|\cdot) = 1$ かつ他のすべてのカテゴリ $c' \neq c$ について $p_{tag}(c'|\cdot) = 0$.

3 実験

3.1 実験設定

CCG 解析器 depccg [14] に対し、提案法で生成したデータによる再学習で性能が改善するかを評価する．depccg の手法は $P(\boldsymbol{y}|\cdot)$ (式 1) における係り受け木 $\boldsymbol{z}$ への依存を除き、デコーダへの入力を $\boldsymbol{h}_1, ..., \boldsymbol{h}_N =$

[3] 入力の係り受け木と CCG 木の係り受け構造は一般に異なる：$\boldsymbol{d} \neq \boldsymbol{d}'$．本研究では $\boldsymbol{d}'$ は Universal Dependencies [9] に従い、$\boldsymbol{d}$ は Head First 木 [14] である（§ 3）.

表 1: 提案法の WSJ23 上での変換性能の上界評価．UF と LF はそれぞれラベルなし/あり F1 である．

手法	UF1	LF1
depccg	94.0	88.8
+ ELMo [15]	94.98	90.51
提案法	**96.48**	**92.68**

$\Omega(\boldsymbol{e}_{x_1}, ..., \boldsymbol{e}_{x_N})$ として、提案法同様に A*構文解析を行うことに等しい．[14] では、単語の表現ベクトル $\boldsymbol{e}_{x_i}$ に、GloVe[4]と接頭（尾）辞ベクトルの連結を用いているが、本研究では、接辞ベクトルの代わりに文脈を考慮した単語ベクトル (ELMo) [15] を用いた場合（$\boldsymbol{e}_{x_i} = \boldsymbol{x}_{x_i}^{GloVe} \oplus \boldsymbol{x}_{x_i}^{ELMo}$）もベースラインに含める．提案法の変換モデルでも同様の単語ベクトルを用いる．

入力の係り受け木はすべて Universal Dependencies (UD) v1 [9] に従い、句構造木のコーパスから UD 木を得る場合には Stanford Converter[5]で変換する．また出力側で用いる係り受け構造 $\boldsymbol{d}$ は係り受けの親が常に左の語である Head First 木（[14] 参照）である．変換モデルは Penn Treebank [18] を UD 木にしたものと CCGBank [3] の WSJ2-21 部を用いて学習する．

3.1.1 医療テキスト

研究 [6] は、医療分野のテキストに対する CCG 解析の分野適応の研究を行っており、Genia コーパス [23] の 1,000 文に終端のカテゴリ列を付与したデータ[6]（以降、GENIA1000）を学習用に公開している．評価は、CCG 木から変換して得られる意味表現である Stanford grammatical relations (GR) [13] に基づき、500 文に GR を付与した BioInfer コーパス [20] を用いて行う．医療テキストと次の疑問文の実験では、[6] と同様に GR への変換が成功した文のみからスコアを計算する．

本研究では、医療論文に句構造木を付与した Genia コーパス [23] を CCG 木に変換し、4,432 文からなる CCGBank を構築した．変換時には、医療分野の複雑な固有名詞を必ず NP にマップするために、制約付きデコーダを用いた．具体的に、句構造木上で NP 句である語列は CCG 木でも NP となるように制約する．

3.1.2 疑問文

ニューステキストには疑問文がほとんど含まれないため、疑問文の解析には分野適応を行う必要がある．医療テキストと同様に [6] によって疑問文解析の評価データ（500 文）、学習データ（1,328 文；木の終端のカテゴリ列のみ．[6] 以降 Qus と呼ぶ.）が用意されている．評価も同様に CCG 木から GR に変換して行う．疑問文解析の実験では、QuestionBank [10] から [6] の評価データに含まれないもののみを抽出して CCG 木に変換し（3,622 文）、解析器の再学習に用いる．

3.1.3 音声会話

音声会話の書き起こしに対し句構造木を付与した Switchboard コーパス [8] を CCG 木に変換して、話し言葉に対する CCG 解析の実験を行う．[7]Switchboard コーパスには、言い淀み箇所のアノテーションが付与されているため、CCG 木への変換ではこれを利用し、制約つきデコーダで言い淀みである単語にはカテゴリ X を付与した．[8]結果として学習/評価/開発それぞれ 59,029/3,799/7,681 文からなる CCG コーパスを構築した．[9]今回の CCG 解析の実験では、木から言い淀み箇所をすべて取り除き、言い淀み除去済みの話し言葉文に対する解析性能を評価する．[10] 話し言葉の CCG 解析の評価データが存在しないため、評価は解析結果の CCG 木を [11] の手法により句構造木に変換し、Switchboard コーパスの正解の句構造木との間で EVALB スコアを計算することで行う．[11] の変換法は完全でないため、スコアには CCG 解析の誤りに加え、変換に伴う誤りが影響することに注意されたい．

3.1.4 数学問題

また、本研究では数学の問題文 [17] に対する CCG 解析の分野適応の実験を行った．[17] により公開されている学習データの 63 文に人手により UD 木と品詞の情報を付与し、評価データ 62 文に対しても正解の CCG 木を付与することで実験データを作成した．[11] 評価は標準の CCG 解析の方法に従い、CCG 木から変換して得られる述語項構造の予測性能を報告する．図 2 はこの分野のテキストに対する解析結果の例である．

3.2 結果と分析

学習した変換モデルで WSJ23 部の係り受け木を CCG 木に変換し、CCG 解析の標準の評価尺度を用いることで、変換モデルの性能を評価した（表 1）．学習データと同じ分野のテキストであるから、これは変換能力の上界を評価していることになる．表 1 の ELMo を用いた場合の depccg の F1 値は、現在の CCG 解析器の最高性能であるが、変換モデルはそれを大きく上回るスコアを示しており、係り受け木から CCG 木への変換が高精度にできていることがわかる．

[4] https://nlp.stanford.edu/projects/glove/

[5] https://nlp.stanford.edu/software/stanford-dependencies.shtml. version 3.9.1 を用いた．

[6] depccg の学習に用いる際は、RBG パーザ [22] に Head First 木を学習させて係り受け構造を付与した．

[7] UD への変換結果は、品詞タグにノイズが多く含まれるため、spacy (https://spacy.io/) を用いて品詞タグを付与し直した．

[8] カテゴリ X は任意のカテゴリ C と結合可能とする：$C \rightarrow C\ X$ かつ $C \rightarrow X\ C$.

[9] データの分割は [4] と同じものに従う．

[10] 言い淀み除去と構文解析を同時に行う手法 [4] の開発と評価は今後の課題である．

[11] ここで付与した CCG 木は基本的に CCGBank [3] に即したものである．数値に対し整数、実数の区別を CCG の素性値で区別するなど細かい粒度の情報の付与 [21] は今後の課題である．

表 2: 医療分野での実験結果．P, R はそれぞれ適合率、再現率．

手法	P	R	F1
C&C [1]	77.8	71.4	74.5
EasySRL [16]	81.8	82.6	82.2
depccg	83.11	82.63	82.87
+ ELMo	85.87	85.34	85.61
+ GENIA1000	85.45	84.49	84.50
+ 提案法	**86.90**	**86.14**	**86.52**

表 3: 疑問文での実験結果．

手法	P	R	F1
C&C [1]	-	-	86.8
EasySRL [16]	88.2	87.9	88.0
depccg	65.48	65.29	65.38
+ ELMo + Qus	**90.55**	**89.86**	**90.21**
+ 提案法	90.07	89.77	89.92

表 4: Switchboard での実験結果．

手法	P	R	F1
depccg	74.73	73.91	74.32
+ ELMo	75.76	76.62	76.19
+ 提案法	**78.03**	**77.06**	**77.54**

表 5: 数学問題での実験結果．

手法	UF1	LF1
depccg	88.49	66.15
+ ELMo [15]	89.32	70.74
+ 提案法	**95.83**	**80.53**

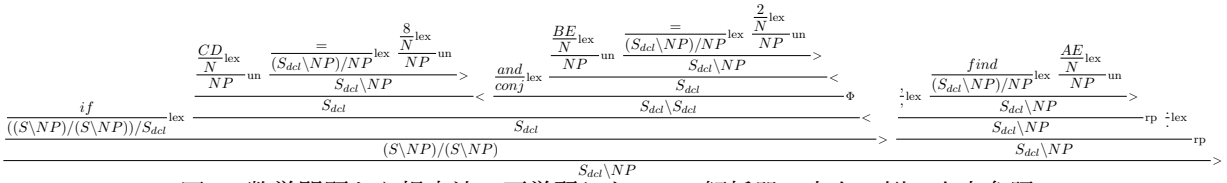

図 2: 数学問題から提案法で再学習した CCG 解析器の出力の例．本文参照．

表 3,3,4,5 に医療テキスト、疑問文、話し言葉、数学問題における結果を示す．疑問文以外において、ELMo と提案法によるデータの追加が相加的に有効であった．疑問文の評価データには、似た構造の文が多く、Qus データで学習したモデルに提案法によるデータを追加しても効果が少なかったと考えられる．注目すべき点として、数学問題における改善が特に大きい．数学問題では、数学特有の表現や記号が多く、これらに対して分野適応が有効であった．図 2 の文では、分野適応前は等号を述語と認識できず、文全体を誤って名詞句の塊と解析していたが、分野適応後は適切に “If S_1 and S_2, S_3.” という構造を認識できるようになった．

4 終わりに

本論文では、ツリーバンクの自動生成による CCG 解析の分野適応手法を提案した．評価を行った分野では、いずれにおいても精度の改善を見た．今後の課題としては、エラー分析を精密に行い、提案法による分野適応の限界を評価する．また、医療、話し言葉、数学問題の CCG 解析を応用した推論の研究を行う予定である．

謝辞 本研究は、JSPS 科研費 18J12945、JST AIP-PRISM JPMJCR18Y1、JST CREST 「知識に基づく構造的言語処理の確立と知識インフラの構築」プロジェクトの支援を受けたものである．

参考文献

[1] Stephen Clark and James R. Curran. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *CL*, 2007.

[2] Timothy Dozat and Christopher D. Manning. Deep Biaffine Attention for Neural Dependency Parsing. 2017.

[3] Julia Hockenmaier and Mark Steedman. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *CL*, 2007.

[4] Matthew Honnibal and Mark Johnson. Joint Incremental Disfluency Detection and Dependency Parsing. *TACL*, 2014.

[5] Makoto Miwa and Mohit Bansal. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proc. of ACL*, 2016.

[6] Laura Rimell and Stephen Clark. Adapting a Lexicalized-Grammar Parser to Contrasting Domains. In *Proc. of EMNLP*, 2008.

[7] Mark Steedman. *The Syntactic Process*. The MIT Press, 2000.

[8] J. J. Godfrey *et al.* SWITCHBOARD: telephone speech corpus for research and development. In *Proc. of ICASSP*, 1992.

[9] Joakim Nivre *et al.* Universal Dependencies v1: A Multilingual Treebank Collection. In *LREC*, 2016.

[10] John Judge *et al.* QuestionBank: Creating a Corpus of Parse-Annotated Questions. In *Proc. of ACL*, 2006.

[11] Jonathan K. Kummerfeld *et al.* Robust Conversion of CCG Derivations to Phrase Structure Trees. In *Proc. of ACL*, 2012.

[12] Kenton Lee *et al.* Global Neural CCG Parsing with Optimality Guarantees. In *Proc. of EMNLP*, 2016.

[13] M. Marneffe *et al.* Generating Typed Dependency Parses from Phrase Structure Parses. In *Proc. of LREC*, 2006.

[14] Masashi Yoshikawa *et al.* A* CCG Parsing with a Supertag and Dependency Factored Model. In *Proc. of ACL*, 2017.

[15] Matthew E. Peters *et al.* Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[16] Mike Lewis *et al.* LSTM CCG Parsing. In *Proc. of NAACL*, 2016.

[17] Minjoon Seo *et al.* Solving Geometry Problems: Combining Text and Diagram Interpretation. In *Proc. of EMNLP*, 2015.

[18] Mitchell P. Marcus *et al.* Building a Large Annotated Corpus of English: The Penn Treebank. *CL*, 1993.

[19] Pascual Martínez-Gómez *et al.* On-demand Injection of Lexical Knowledge for Recognising Textual Entailment. In *Proc. of EACL*, 2017.

[20] Sampo Pyysalo *et al.* BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 2007.

[21] Takuya Matsuzaki *et al.* Semantic Parsing of Pre-university Math Problems. In *Proc. of ACL*, 2017.

[22] Tao Lei *et al.* Low-Rank Tensors for Scoring Dependency Structures. In *Proc. of ACL*, 2014.

[23] Yuka Tateisi *et al.* Syntax Annotation for the GENIA Corpus. In *Proc. of IJCNLP*, 2005.

[24] 戸次大介. 「日本語文法の形式理論 －活用体系・統語構造・意味合成－」. くろしお出版 日本語研究叢書 2 4 , 2010.