日本語 CCGBank は言語学的に妥当か

戸次大介 [1] 谷中瞳 [2]
[1] お茶の水女子大学 [2] 東京大学
bekki@is.ocha.ac.jp hyanaka@is.s.u-tokyo.ac.jp

概要

日本語 CCGBank は、日本語 CCG パーザの開発において学習・評価データとして利用されている CCG ツリーバンクである。しかし、日本語 CCGBank は係り受けツリーバンクからの自動変換によって生成されたものであり、その言語的妥当性については検証が必要である。本論文では、日本語 CCGBank における受身・使役の分析に焦点を当て、それが意味解析システム ccg2lambda の意味合成と相まって、特に使役受身文において経験的に誤った予測をもたらすことを示す。メタレベルでは、本論文はツリーバンクを言語学的分析と見做して反証する試みであり、その方法論の例示である。

1 日本語 CCG ツリーバンク

ツリーバンクからパーザを構築する、という工程は 1990 年代の確率的 CFG パーザの時代に確立されたものである。しかし、当時はこのようなアプローチは言語学としての形式統語論には適用できない、と考えられていた。その理由は「形式統語論は柔軟性に欠け、実際のテキストの構造を網羅的に記述することはできない」と信じられていたからである。この誤解は、組合せ範疇文法 (Combinatory Categorial Grammar, CCG: Ades and Steedman (1982); Steedman (1996, 2000)) の理論的発展、および英語 CCGBank (Hockenmaier and Steedman, 2005) 等の CCG ツリーバンクの登場によって払拭されることとなった。その後の C&C parser (Clark and Curran, 2007) や EasyCCG (Lewis and Steedman, 2014) 等の CCG パーザの開発は、形式統語論の理論である CCG のためのパーザが、確率的 CFG パーザに準ずる工程によってツリーバンクから生成可能である、ということを明らかにした点において印象的な出来事であった。

この流れは、日本語 CCG パーザの研究にも影響を与えた。Bekki (2010) によって、日本語の統語構造も CCG による網羅的な記述が可能であることが一定以上の水準において示されたことをうけて、Uematsu et al. (2013) による日本語 CCGBank が構築され、続いて Jigg (Noji and Miyao, 2016), depccg (Yoshikawa et al., 2017) といった日本語 CCG パーザが開発されるに至った。現在の研究環境では、高速で頑健な複数の CCG パーザが利用可能であり、その精度はニューラル言語モデルの発達によってさらなる改善をみている。

日本語 CCGBank の開発における難点として、当時、日本語の CFG ツリーバンクが存在しなかったことが挙げられる[1)]。CCGbank (Hockenmaier and Steedman, 2005) が CFG ツリーバンクである Penn Treebank から生成されたのに対し、日本語 CCGBank 開発当時に利用可能であった日本語のための大規模ツリーバンクは、係り受けコーパスである京都大学テキストコーパス[2)]しかなかったため、Uematsu et al. (2013) では係り受け構造から CCG 統語構造への自動変換を試みたのである。

ところが CCG の統語構造は一般に、項構造や統語素性など、CFG 木と比較しても精緻な情報を持っている。したがって、CFG よりもさらに情報量の少ない係り受け木は、多くの言語学的情報で埋め合わせなければならない。この補完は自明ではなく、そのために ad-hoc な規則を随所で措定する必要があった。たとえば、受身／使役の動詞性接尾語「れ（る）」「せ（る）」を、統語範疇 $S\backslash S$ を持つ語として

1） 近年では、日本語の大規模な CFG ツリーバンクが NINJAL parsed corpus of modern Japanese (https://npcmj.ninjal.ac.jp/) の一部として利用可能であり、それを用いて範疇文法の一種である AB 文法のツリーバンクを生成する試みも行われている (Kubota et al., 2020)。しかし、日本語 CCGBank の何が経験的に問題であるのか、なぜ問題なのか、という問いに対して踏み込んだ議論はなされていない。Universal Dependency のような係り受けツリーバンクから新たな CCGBank を生成する試み Tran and Miyao (2022) も進行中であり、上述の問いに答えることの重要性は益々高まっている。

2） https://github.com/ku-nlp/KyotoCorpus

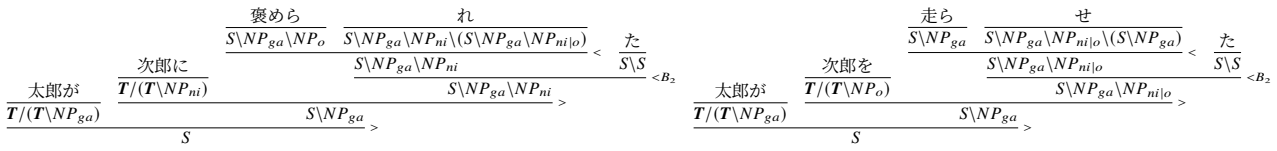

図 1　Bekki (2010) における (1a)(2a) の統語構造

解析したのはその一例である。これがなぜ誤りといえるのかについては次節以降で論じる。

CCGBank は CCG パーザの学習・評価データとして機能するため、CCG パーザは CCGBank の分析の誤りを継承する。その意味において、CCGBank における統語的分析の妥当性は、CCG パーザの（精度に表れない）性能の上限を定める。しかし、ツリーバンクに含まれる統語構造の妥当性に関しては、自然言語処理は語る術を持たない。一方、理論言語学の観点からの研究はこれまでほとんど行われていない。

そのような背景から、本論文では、日本語 CCGBank が示す統語構造を理論言語学の観点から批判的に検討することを試みる。具体的には、日本語 CCG (Bekki, 2010) と CCGBank が異なる分析を与えている「受動文」と「使役文」に着目し、日本語 CCGBank の分析にどのような経験的誤りがあるのか、その一例を明らかにする。また、このような研究の持つ意義についても第 4 節で論じる。

2　日本語における受動文と使役文

まずは日本語の受動文・使役文にまつわる経験的な事実と、それらが日本語 CCG (Bekki, 2010) においてどのように説明されているかを簡潔に解説する[3)]。受動文の主節におけるガ格名詞句は埋め込み節におけるニ格またはヲ格名詞句に対応する。このことを推論の形式で表せば (1c) のようになる。

(1)　a.　太郎が次郎に褒められた。
　　b.　次郎が太郎を褒めた。
　　c.　(1a) ⟹ (1b)

次に、使役文の主節におけるニ格もしくはヲ格名詞句は埋め込み節におけるガ格名詞句に対応する。こちらも推論の形式で表せば (2c) のようになる。

(2)　a.　太郎が次郎を走らせた。
　　b.　次郎が走った。
　　c.　(2a) ⟹ (2b)

Bekki (2010) に依れば、文 (1a)(2a) の統語構造は図 1 である。ここで意味の理論として依存型意味論 (Dependent type semantics, DTS: Bekki (2014); Bekki and Mineshima (2017); Bekki (2021)) を採用し[4)]、語／句の意味表示がそれぞれ以下のようであると仮定する。

(3)　太郎が：$\lambda P.P(\mathbf{t})$
　　次郎{に | を}：$\lambda P.P(\mathbf{j})$
　　褒めら：$\lambda yxk.(e:\mathbf{e})\times\mathbf{praise}(e,x,y)\times k(e)$
　　走ら：$\lambda xk.(e:\mathbf{e})\times\mathbf{run}(e,x)\times k(e)$
　　た：id

すると、受動態の動詞性接尾語「れ（る）」および使役態の動詞性接尾語「せ（る）」の意味表示が、受動文・使役文の分析の中核となる。

(4)　れ：$\lambda Pyx.Pxy$

(5)　せ：$\lambda Pyxk.Py(\lambda e.\mathbf{cause}(e,x)\times ke)$

これらの動詞性接尾語は第一項である埋め込み節の項と自らの項の対応関係を「把握」している。すなわち受動文においては、主節のニ格名詞句が埋め込み節のガ格名詞句に、主節のガ格名詞句が埋め込み節のヲ格またはニ格名詞句に意味的に対応する。継続 k は意味合成の最終段階で $\lambda e.\top$ によって埋められる[5)]と仮定すると、意味合成の結果、文 (1a)(2a) の意味表示はそれぞれ以下のようになる。

(6)　$(e:\mathbf{e})\times\mathbf{praise}(e,\mathbf{j},\mathbf{t})\times\top$

(7)　$(e:\mathbf{e})\times\mathbf{run}(e,\mathbf{j})\times\mathbf{cause}(e,\mathbf{t})\times\top$

3)　受動文については、間接受動文（いわゆる迷惑受身）と直接受動文の二種類が存在する (Bekki, 2010, pp.222-226) が、ここでは直接受動文のみについて論じる。この限定は議論に影響しない。

4)　ここでは Bekki (2010) で採用されている型付き動的論理に代えて DTS を採用しているが、さらにイベント意味論と意味合成を両立する手法として標準的な技法である継続渡し形式 (continuation-passing style) を採用している。一方、簡便のためテンスの分析は省略しており「た」の意味は恒等関数である。このような理論設定は、本論での論点を意味合成から推論まで明らかにするという目的のもとではもっとも簡略化された設定の一つである。

5)　$\top$ は証明項を一つだけ持つ枚挙型 (enumeration type) で「真」の役割を果たす。

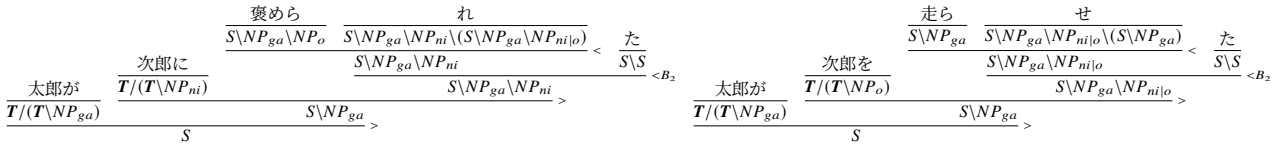

図 2　CCGBank における (1a)(2a) の統語構造

一方、文 (1b)(2b) の意味表示は以下のようになる。

(8)　$(e : \mathbf{e}) \times \mathbf{praise}(e, \mathbf{j}, \mathbf{t}) \times \top$

(9)　$(e : \mathbf{e}) \times \mathbf{run}(e, \mathbf{j}) \times \top$

(6)(7) はそれぞれ (8)(9) を含意するので、推論 (1c)(2c) はいずれも正しく予測される。

この分析の予測の妥当性は受動・使役の接尾語「れ（る）」「せ（る）」を含む様々な構文についての推論データによって検証することができるが、端的な例は使役受身である。日本語においては (10a) のように、使役と受身は接続することができる。

(10)　a. 次郎が太郎に走らせられた。
　　b. 太郎が次郎を走らせた。
　　c. (10a) $\Longrightarrow$ (10b)

文 (10a) は文 (2a)（＝(10b)）の受動化であり、(10b) を含意する。「走らせられ」の意味表示は、「走ら」の意味表示に「せ」(5) を関数適用したのちに「れ」(4) を関数適用したもので、以下のようになる。

(11)　$\lambda yxk.\,(e : \mathbf{e}) \times \mathbf{run}(e, x) \times \mathbf{cause}(e, y) \times ke$

したがって文 (10a) 意味表示は

(12)　$(e : \mathbf{e}) \times \mathbf{run}(e, \mathbf{j}) \times \mathbf{cause}(e, \mathbf{t}) \times \top$

となるが、これは (7) そのものであるから、(2a) を、したがって (2b) を含意することが正しく予測される。このように、日本語 CCG の受動文・使役文の分析は、統語構造から意味表示、そして推論に至る過程を「れ（る）」「せ（る）」が現れる広範囲の構文において正しく予測・説明する。

3　CCGBank と $S\backslash S$ 分析

日本語 CCGBank では、太郎が次郎に褒められた、の統語構造は図 2 のようになっており、受身の動詞性接尾語「れ（る）」も使役の動詞性接尾語「せ（る）」もともに統語範疇が $S\backslash S$ の語として分析されている。

日本語 CCGBank 自体は意味論の分析を持たないが、日本語 CCG のための意味論的分析として、含意関係システム ccg2lambda (Mineshima et al., 2015) において採用されている高階論理による意味表示とその意味合成過程が知られている。ccg2lambda の分析は、日本語 CCG パーザである Jigg または depccg の出力する統語構造に依存しており、それらのパーザの出力は日本語 CCGBank の記述に依存している。

まず、ガヲ動詞である「褒め（る）」の意味表示は、以下のようになっている。

(13)　$\lambda Q_2 Q_1 C_1 C_2 K.Q_1(\lambda x_1.Q_2(\lambda x_2.$
$\exists e(K(\mathbf{praise}, e)\,\&\,C_1(x_1, e, \mathbf{Ag})\,\&\,C_2(x_2, e, \mathbf{Th}))))$

この意味表示は、標準的な二項動詞の意味表示、もしくは前節の意味表示と比べても、かなり複雑である。その理由は、ccg2lambda では、意味役割である **Ag** と e, x_1 の間の関係、および **Th** と e, x_2 の間の関係を相対化して C_1, C_2 という関数の変数によって表しているからである。Q_2 にヲ格名詞句、Q_1 にガ格名詞句を取って統語範疇 S となったのち、トップレベルで機械的に $\lambda ES.S(\lambda xeT.(T(e) = x), \lambda xeT.(T(e) = x), id)$ を適用する。これによって、C_1 と C_2 に $\lambda xeT.(T(e) = x)$ が代入されて $\mathbf{Ag}(e) = x$, $\mathbf{Th}(e) = y$ が指定され、また K に id が代入されるのである。「次郎が」の意味表示を $\lambda P.P(\mathbf{j})$,「太郎を」の意味表示を $\lambda P.P(\mathbf{t})$ とすれば、能動文 (1b) の意味表示は、

(14)　$\exists e(\mathbf{praise}(e)\ \&\ \mathbf{Ag}(e) = \mathbf{j}\ \&\ \mathbf{Th}(e) = \mathbf{t})$

となる。次に、受動文 (1a) の意味表示を考える。ccg2lambda においては「れ（る）」の意味表示は意味テンプレート[6)]によって以下のように与えられる。

(15)　$\lambda Q_2 Q_1 C_1 C_2 K.V(Q_2, Q_1,$
$\lambda x_1 eT.C_1(x_1, e, \mathbf{Th}), \lambda x_2 eT.C_2(x_2, e, \mathbf{Ag}), K)$

V には隣接する他動詞「褒めら」の意味表示が与えられる。よって「褒められ」の意味表示は以下のようになる。

6）ccg2lambda においては「れ（る）」の場合に見られるように、意味テンプレートの適用によって意味表示の型が統語範疇と準同型ではなくなる場合がある。このこと自体への反論はありうるが、ここでは本題ではないので深入りしない。

(16) $\lambda Q_2 Q_1 C_1 C_2 K.Q_1(\lambda x_1.Q_2(\lambda x_2.$
$\exists e(K(\mathbf{praise}, e) \,\&\, C_1(x_1, e, \mathbf{Th}) \,\&\, C_2(x_2, e, \mathbf{Ag}))))$

すなわち、(15) において C_1, C_2 が受け取っている意味役割を捨て、C_1 には **Th** を、C_2 には **Ag** を与え直すのである。これに $\lambda P.P(\mathbf{t})$（太郎を）、$\lambda P.P(\mathbf{j})$（次郎が）、そして $\lambda S.S(\lambda xeT.(T(e) = x), \lambda xeT.(T(e) = x), id)$ を順次適用することで、(1b) は以下のようになる。

(17) $\exists e(\mathbf{praise}(e) \,\&\, \mathbf{Ag}(e) = \mathbf{j} \,\&\, \mathbf{Th}(e) = \mathbf{t})$

これはまさに (14) と同一であるから、(1c) の推論が正しく予測されることになる。同様に、使役の「せ（る）」についても、

(18) $\lambda Q_2 Q_1 C_1 C_2 K.V(Q_2, Q_1,$
$\lambda x_1 eT.C_1(x_1, e, \mathbf{Cause}), \lambda x_2 eT.C_2(x_2, e, \mathbf{Ag}), K)$

のような意味テンプレートを与えれば、まず「走らせ」の意味表示は以下のようになる。

(19) $\lambda Q_2 Q_1 C_1 C_2 K.Q_1(\lambda x_1.Q_2(\lambda x_2.$
$\exists e(K(\mathbf{run}, e) \,\&\, C_1(x_1, e, \mathbf{Cause}) \,\&\, C_2(x_2, e, \mathbf{Ag}))))$

したがって、(2b) は以下のようになる。[7)]

(20) $\exists e(\mathbf{run}(e) \,\&\, \mathbf{Cause}(e) = \mathbf{t} \,\&\, \mathbf{Ag}(e) = \mathbf{j})$

これは (2a) すなわち $\exists e(\mathbf{run}(e) \,\&\, \mathbf{Ag}(e) = \mathbf{j})$ を含意するので、(2c) についても正しく予測される。

ところが、この個別のケースについては正しく見える分析が、使役受身について誤った予測を生み出す。「走らせられ」の意味表示は、ccg2lambda では (19) に (15) を適用することによって、

(21) $\lambda Q_2 Q_1 C_1 C_2 K.Q_1(\lambda x_1.Q_2(\lambda x_2.$
$\exists e(K(\mathbf{run}, e) \,\&\, C_1(x_1, e, \mathbf{Th}) \,\&\, C_2(x_2, e, \mathbf{Ag}))))$

となるが、これは「走ら」に直接受身の「れ（る）」を適用したもの、すなわち「走られ」の意味表示と同一になってしまっている。ここからは、(10b) = (2a) も、(2b) も、いずれも含意しない。

この誤りの原因は、「れ（る）」が第一項に対して **Th** を、第二項に対して **Ag** を与える、と大域的に決め打ちしていることにある。使役受身の例が示すことは、「れ（る）」は第一項を、埋め込み節の用言の第二項と対応付け、第二項を、埋め込み節の用言の第一項と対応付ける、という以外の捉え方が許されないということである。この標準的分析が要求することは、埋め込み節の用言の第一、第二項が「れ（る）」の意味表示において操作可能であることである。このことは統語論において「れ（る）」の取るべき第一項の統語範疇が $S\backslash NP\backslash NP$ であることを要求するのである。

7) 実際の ccg2lambda は、使役文については受動文とは異なる分析を採用している。その分析の是非については紙幅の都合上、機会を改めて論じたい。

4 おわりに

日本語 CCGBank の受動文と使役文の分析、すなわち「れ（る）」と「せ（る）」が統語範疇 $S\backslash S$ を持つ、という統語的分析は、使役受身文において誤った予測を生み出すことを示した。これは、日本語 CCGBank の分析が (1c)(2c)(10c) のような受身・使役が関わる推論を説明しうる意味論的分析を、現時点で持たないということである。第 2 節で述べた日本語 CCG における分析がこれらの推論をすべて正しく予測する以上、修正の責任は CCGBank 側にある。

本論文の議論には、二つの目的がある。一つ目は、ツリーバンクを必ずしも言語学的分析とみなしていない言語学コミュニティへの呼びかけである。ツリーバンクの記述に対して、言語学的分析に対するのと同じように反例を提示する、という過程が、ツリーバンクと言語学双方の健全性にとって重要な研究活動であると考える。

二つ目は、ツリーバンクの記述を言語学コミュニティにおいて合意済みとみなしてしまうことへの警鐘である。そもそも理論言語学においては理論は提示・反証・修正の繰り返しであるから、どの時点においても「合意済みの理論」というものは存在しない、という当たり前の事実を再確認しておきたい。

本論文はツリーバンクを言語学的分析と見做して反証する試みであり、そのような方法論の例示でもある。今後、理論言語学のコミュニティから、本論文で行われたようなツリーバンク内の分析への反論が試みられ、それを受けたツリーバンク開発者による分析の修正、それに対するさらなる反論、という研究の流れが現れることを期待したい。

謝辞 本研究の一部は、JST CREST JPMJCR20D2、JSPS 科研費 JP20K19868 の支援を受けたものである。

参考文献

Ades, A. E. and M. J. Steedman. (1982) “On the Order of Words”, **Linguistics and Philosophy** 4, pp.517–558.

Bekki, D. (2010) **Nihongo-Bunpoo-no Keisiki-Riron - Katuyootaikei, Toogohantyuu, Imigoosei - (trans. ’Formal Japanese Grammar: the conjugation system, categorial syntax, and compositional semantics’)**. Tokyo, Kuroshio Publisher.

Bekki, D. (2014) “Representing Anaphora with Dependent Types”, In the Proceedings of N. Asher and S. V. Soloviev (eds.): **Logical Aspects of Computational Linguistics (8th international conference (LACL2014), Toulouse, France, June 2014 Proceedings), LNCS 8535**. pp.14–29, Springer, Heiderburg.

Bekki, D. (2021) “Proof-Theoretic Analysis of Weak Crossover”, In the Proceedings of **Logic and Engineering of Natural Language Semantics 18 (LENLS18)**. pp.75–88.

Bekki, D. and K. Mineshima. (2017) “Context-passing and Underspecification in Dependent Type Semantics”, In: **Modern Perspectives in Type Theoretical Semantics**, Studies of Linguistics and Philosophy. Springer, pp.11–41.

Clark, S. and J. R. Curran. (2007) “Widecoverage efficient statistical parsing with CCG and log-linear models”, **Computational Linguistics** 33(4), pp.493–552.

Hockenmaier, J. and M. J. Steedman. (2005) “CCGbank LDC2005T13”. Linguistic Data Consortium.

Kubota, Y., K. Mineshima, N. Hayashi, and S. Okano. (2020) “Development of a General-Purpose Categorial Grammar Treebank”, In the Proceedings of **the 12th Language Resources and Evaluation Conference**. pp.5195–5201, European Language Resources Association.

Lewis, M. and M. Steedman. (2014) “A* CCG Parsing with a Supertag-factored Model”, In the Proceedings of **the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. pp.990–1000, Association of Computational Linguistics.

Mineshima, K., P. Martínez-Gómez, Y. Miyao, and D. Bekki. (2015) “Higher-order logical inference with compositional semantics”, In the Proceedings of **Conference on Empirical Methods in Natural Language Processing (EMNLP2015)**. pp.2055–2061.

Noji, H. and Y. Miyao. (2016) “Jigg: A framework for an easy natural language processing pipeline”, In the Proceedings of **the 54th Association of Computational Linguistics**. pp.103–108.

Steedman, M. J. (1996) **Surface Structure and Interpretation**. Cambridge, The MIT Press.

Steedman, M. J. (2000) **The Syntactic Process (Language, Speech, and Communication)**. Cambridge, The MIT Press.

Tran, T.-A. and Y. Miyao. (2022) “Development of Multilingual CCG Treebank via Universal Dependencies Conversion”, In the Proceedings of **the 13th Conference on Language Resource and Evaluation (LREC2022)**. pp.5220–5233.

Uematsu, S., T. Matsuzaki, H. Hanaoka, Y. Miyao, and H. Mima. (2013) “Integrating Multiple Dependency Corpora for Inducing Wide-coverage Japanese CCG Resources”, In the Proceedings of **the 51st Annual Meeting of the Association for Computational Linguistics**, Vol. 1. pp.1042–1051, Association for Computational Linguistics.

Yoshikawa, M., H. Noji, and Y. Matsumoto. (2017) “A* CCG Parsing with a Supertag and Dependency Factored Model”, In the Proceedings of **the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)**. pp.277–287.