Is Japanese CCGBank empirically correct? A case study of passive and causative constructions

Daisuke Bekki
Ochanomizu University
bekki@is.ocha.ac.jp

Hitomi Yanaka
The University of Tokyo
hyanaka@is.s.u-tokyo.ac.jp

Abstract

The Japanese CCGBank serves as training and evaluation data for developing Japanese CCG parsers. However, since it is automatically generated from the Kyoto Corpus, a dependency treebank, its linguistic validity still needs to be sufficiently verified. In this paper, we focus on the analysis of passive/causative constructions in the Japanese CCGBank and show that, together with the compositional semantics of ccg2lambda, a semantic parsing system, it yields empirically wrong predictions for the nested construction of passives and causatives.

1 Introduction

The process of generating wide-coverage syntactic parsers from treebanks was established in the era of probabilistic context-free grammar (CFG) parsers in the 1990s. However, it was believed at that time that such an approach did not apply to linguistically-oriented formal syntactic theories. The reason was that formal syntactic theories were believed to be too inflexible to exhaustively describe the structure of real texts. This misconception was dispelled by the theoretical development of formal grammars and the emergence of linguistically-oriented treebanks.[1] In particular, Combinatory Categorial Grammar (CCG) (Steedman, 1996, 2000) and CCGbank (Hockenmaier and Steedman, 2005) gave rise to the subsequent developments of CCG parsers such as C&C parser (Clark and Curran, 2007) and EasyCCG parser (Lewis and Steedman, 2014), and proved that wide-coverage CCG parsers could be generated from treebanks in a similar process to probablistic CFG parsers.

This trend has also impacted research on Japanese syntax and parsers. Bekki (2010) revealed that CCG, as a syntactic theory, enables us to provide a wide-coverage syntactic description of the Japanese language. It motivated the development of the Japanese CCGBank (Uematsu et al., 2013), followed by Japanese CCG parsers such as Jigg (Noji and Miyao, 2016) and depccg (Yoshikawa et al., 2017).

The difficulty in developing the Japanese CCGBank lay in the absence of CFG treebanks for the Japanese language at that time.[2] While CCGbank was generated from the Penn Treebank, which is a CFG treebank, the only large-scale treebank available for Japanese was the Kyoto Corpus[3], which is a dependency tree corpus, from which Uematsu et al. (2013) attempted to construct a Japanese CCGBank by automatic conversion.

The syntactic structures of CCG have more elaborated information than those of CFG, such as argument structures and syntactic features. Thus, it is inevitable that a dependency tree, which has even less information than that of CFG, must be supplemented with a great deal of linguistic information. Uematsu et al. (2013) had to guess them systematically, which is not an obvious process, and ad-hoc rules had to be stipulated in many places to accomplish it, including the "passive/causative suffixes as $S\backslash S$ analysis," which we will discuss in Section 3.

Since CCGBank serves as both training and evaluation data for CCG parsers, syntactic descriptions

[1] To mention a few, the LinGO Redwoods treebank (Oepen et al., 2002) contains English sentences annotated with Head-driven Phrase Structure Grammar (HPSG) parse trees. The INESS treebank repository (Rosén et al., 2012) offer Lexical Functional Grammar (LFG) treebanks such as The ParGram Parallel Treebank (ParGramBank) (Sulger et al., 2013), which provides ten typologically different languages.

[2] Recently, a large-scale CFG treebank for the Japanese language is available as a part of NINJAL parsed corpus of modern Japanese https://npcmj.ninjal.ac.jp/, and there is also an attempt to generate a treebank of better quality by using it (Kubota et al., 2020). However, the questions of what is empirically problematic about the Japanese CCGBank and, more importantly, why it is, remain undiscussed. The importance of answering these questions as we do in this paper is increasing, given that attempts to generate a CCGBank from a dependency corpus such as Universal Dependency are still ongoing (cf. Tran and Miyao (2022)).

[3] https://github.com/ku-nlp/KyotoCorpus

Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023), pages 32 - 36
March 9-12, 2023 ©2023 Association for Computational Linguistics

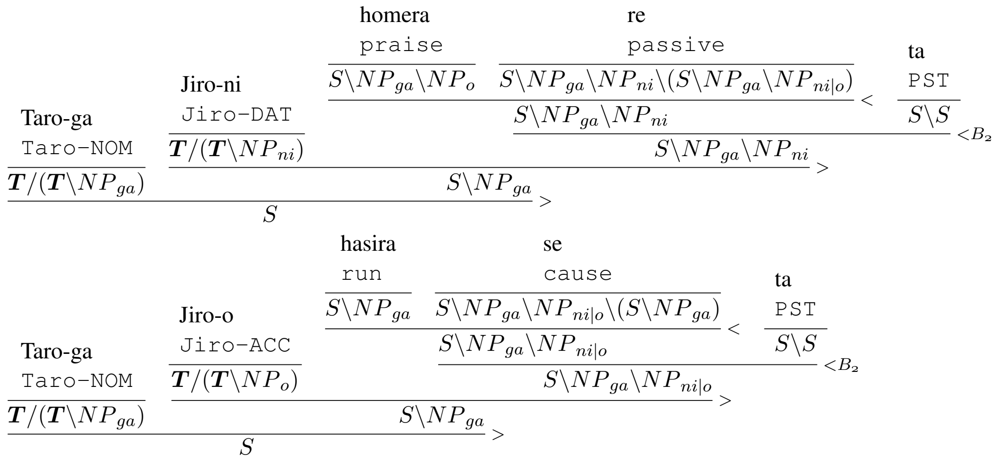
homera
praise
$S\backslash NP_{ga}\backslash NP_{o}$
re
passive
$S\backslash NP_{ga}\backslash NP_{ni}\backslash(S\backslash NP_{ga}\backslash NP_{ni|o})$
<
$S\backslash NP_{ga}\backslash NP_{ni}$
ta
PST
$S\backslash S$
$<B_2$
Jiro-ni
Jiro-DAT
$\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ni})$
$S\backslash NP_{ga}\backslash NP_{ni}$
>
Taro-ga
Taro-NOM
$\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga})$
$S\backslash NP_{ga}$
>
S
hasira
run
$S\backslash NP_{ga}$
se
cause
$S\backslash NP_{ga}\backslash NP_{ni|o}\backslash(S\backslash NP_{ga})$
<
$S\backslash NP_{ga}\backslash NP_{ni|o}$
ta
PST
$S\backslash S$
$<B_2$
Jiro-o
Jiro-ACC
$\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{o})$
$S\backslash NP_{ga}\backslash NP_{ni|o}$
>
Taro-ga
Taro-NOM
$\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga})$
$S\backslash NP_{ga}$
>
S

Figure 1: Syntactic structures of (1) and (2) in Bekki (2010)

in CCGBank set an upper bound on CCG parser performance, which inherit any empirical fallacies in CCGBank: thus the validity of the syntactic structures in CCGBank is important. However, little research from the perspective of formal syntax has been conducted regarding the adequacy of syntactic structures contained in treebanks.

This paper aims to assess the syntactic structures exhibited by the Japanese CCGbank from the viewpoint of theoretical linguistics. Specifically, we focus on the syntax and semantics of case alternation in passive and causative constructions in Japanese, a linguistic phenomenon analyzed differently in the standard Japanese CCG and CCGBank, and show that the syntactic analysis of the Japanese CCGBank contains empirical fallacies.

2 Passive and Causative Constructions in Japanese

We first present some empirical facts about Japanese passives and causatives and how they are described in the standard Japanese CCG (Bekki, 2010). *Ga*-marked noun phrases (henceforth NP_{ga}) in passive sentences correspond to *ni*-marked noun phrases (henceforth NP_{ni}) or *o*-marked noun phrases (henceforth NP_{o}) in the corresponding active sentences, which is expressed as (1) in the form of inferences.

(1) Taro-ga Jiro-ni homera-re-ta
Taro-NOM Jiro-DAT praise-passive-PST
⟹ Jiro-ga Taro-o home-ta
Jiro-NOM Taro-ACC praise-PST
(trans.) 'Taro is praised by Jiro.' ⟹ 'Jiro praised Taro.'

Next, NP_{ni} or NP_{o} in causative sentences correspond to NP_{ga} in the corresponding active sentences, which is also expressed in the form of inference as in (2).

(2) Taro-ga Jiro-{ni|o}
Taro-NOM Jiro-{DAT|ACC}
hasira-se-ta ⟹ Jiro-ga hasit-ta
run-causative-PST Jiro-NOM run-PST
(trans.) 'Taro made Jiro run.' ⟹ 'Jiro run.'

According to Bekki (2010), the syntactic structure of the left-side sentences of (1) and (2) are as shown in Figure 1.

For simplicity (omitting analysis of tense, etc.), let us assume that the semantic representations of *Taro-ga*, *Jiro-{ni|o}*, *homera*, *hasira*, and *ta* are respectively defined as $\lambda P.P(\mathbf{t})$,$\lambda P.P(\mathbf{j})$, $\lambda y.\lambda x.\lambda k.(e : ev) \times \mathbf{praise}(e, x, y) \times ke$, $\lambda x.\lambda k.(e : ev) \times \mathbf{run}(e, x) \times ke$, id by using event semantics (Davidson, 1967) with continuations (Chierchia, 1995) in terms of DTS (dependent type semantics) (Bekki and Mineshima, 2017), where id is the identity function and ev is the type for events.

Then the core of the analysis of case alternation is to define semantic representations of the passive suffix *re* and the causative suffix *se*, homomorphically to their syntactic categories, as

(3) Passive suffix *re*: $\lambda P.\lambda y.\lambda x.Pxy$

(4) Causative suffix *se*:
$\lambda P.\lambda y.\lambda x.\lambda k.Py(\lambda e.\mathbf{cause}(e, x) \times ke)$

In words, both suffixes *know* the argument structure of its first argument, namely, the verb. In

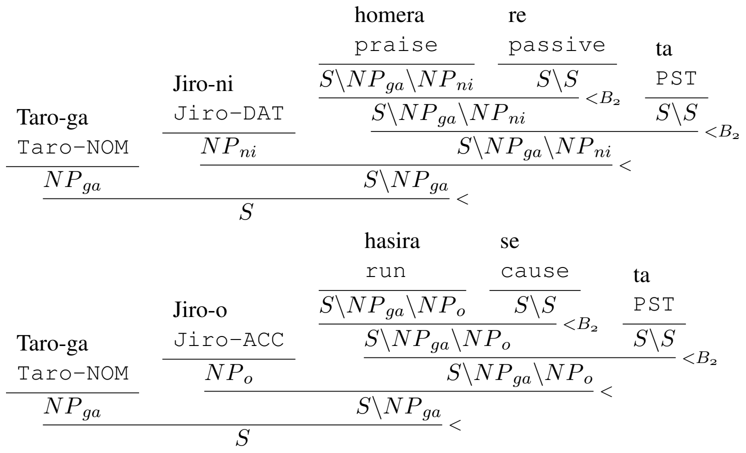

Figure 2: Syntactic structures of (1) and (2) in CCGBank

passive constructions, NP_{ga} corresponds to the NP_o or NP_{ni}, and NP_{ni} corresponds to NP_{ga}, in their active counterparts. In causative constructions, $NP_{ni|o}$ corresponds to NP_{ga} in their active counterparts. Assuming the event continuation k is replaced by the term $\lambda e.\top$ at the end of the semantic composition (where $\top$ is an enumeration type with only one proof term and plays the role of "true"), the semantic composition ends up in the following representations for the left-side sentences of (1) and (2), respectively.

(5) $(e : ev) \times \textbf{praise}(e, \textbf{j}, \textbf{t}) \times \top$

(6) $(e : ev) \times \textbf{run}(e, \textbf{j}) \times \textbf{cause}(e, \textbf{t}) \times \top$

These respectively entail the right-side sentences of (1) and (2), the semantic representations of which are (7) and (8) respectively, so the inferences (1) and (2) are correctly predicted.

(7) $(e : ev) \times \textbf{praise}(e, \textbf{j}, \textbf{t}) \times \top$

(8) $(e : ev) \times \textbf{run}(e, \textbf{j}) \times \top$

The validity of this analysis can be verified by inference data on various constructions including passives and causatives. In particular, causatives can be nested in passives in Japanese, as in (9).

(9) Jiro-ga Taro-ni
Jiro-NOM Taro-DAT
hasira-sera-re-ta $\Longrightarrow$ Taro-ga
run-causative-passive-PST Taro-NOM
Jiro-o hasira-se-ta
Jiro-ACC run-causative-PST
(lit.) 'Jiro was made run by Taro.' $\Longrightarrow$
'Taro made Jiro run.'

The constituent *hasira-sera-re* is the passivization of *hasira-sera*, the matrix predicate of the left-side sentence of (2) that is equivalent to the right-side sentence of (9), and thus also entails the right-side sentence of (2). The semantic representation of *hasira-sera-re* is obtained by a functional application of (4) to the semantic representation of *hasira*, followed by a functional application of (3), as follows.

(10) $\lambda y.\lambda x.\lambda k.$
$(e : ev) \times \textbf{run}(e, x) \times \textbf{cause}(e, y) \times ke$

Therefore, the semantic representation of the left-side of (9) is $(e : ev) \times \textbf{run}(e, \textbf{j}) \times \textbf{cause}(e, \textbf{t}) \times \top$, which is equal to (6), so it is correctly predicted to entail the left and right-sides of (2). Thus, the passive/causal analysis in Bekki (2010) robustly predicts and explains the process from syntactic structures to semantic representations, and inferences.

3 ccg2lambda and the $S\backslash S$ analysis

Analysis using a compositional semantic system ccg2lambda (Martínez-Gómez et al., 2016) relies on the syntactic structures output by the Japanese CCG parsers Jigg or depccg. As mentioned in Section 1, the output of these CCG parsers depends on Japanese CCGBank. In Japanese CCGBank, the lexical assignments for the left-side sentences of (1) and (2) are as shown in Figure 2, in which both the passive suffix *re* and the causative suffix *se* have the syntactic category $S\backslash S$.

Let us glance over how non-passive sentences in CCGBank are semantically analyzed. The semantic representation of the two-place predicate *homera* is given as follows (slightly simplified).

(11) $\lambda Q_2 Q_1 C_1 C_2 K.$

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

This appears to be considerably more complex than that of *homera* in the previous section. This is because in ccg2lambda, the relations between **Ag** (=Agent) and x_1, e and between **Th** (=Theme) and x_2, e are relativized by the higher-order variables C_1, C_2. After taking an NP_{ni} for Q_2 and an NP_{ga} for Q_1 to become the constituent of syntactic category S, ccg2lambda applies to it the function $\lambda S.S(\lambda xeT.(T(e) = x), \lambda xeT.(T(e) = x), id)$. This causes $\lambda xeT.(T(e) = x)$ to be assigned to C_1 and C_2 to specify $\textbf{Ag}(e) = x_1$ and $\textbf{Th}(e) = x_2$, and id to be assigned to K. Assuming $\lambda P.P(\textbf{t})$ and $\lambda P.P(\textbf{j})$ for the semantic representations of *Taro-ga* and *Jiro-ni*, the semantic representation of the right-side of (1) is obtained as (12), which is a standard neo-Davidsonian semantic representation (Parsons, 1990) for *Jiro praised Taro*.

(12) $\exists e(\textbf{praise}(e) \; \& \; \textbf{Ag}(e) = \textbf{j} \; \& \; \textbf{Th}(e) = \textbf{t})$

By contrast, in ccg2lambda the semantic representation of *re* is overwritten by the *semantic template* as

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

V is instantiated by the semantic representation of the adjacent transitive verb (=*homera* in this case). The semantic representation of *homera-re* thus becomes

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

That is, the semantic roles received by C_1, C_2 in (11) are discarded, and instead C_1 is given **Th** and C_2 is given **Ag**. By applying $\lambda P.P(\textbf{t})$, $\lambda P.P(\textbf{j})$, and $\lambda S.S(\lambda xeT.(T(e) = x), \lambda xeT.(T(e) = x), id)$ sequentially, the left-side of (1) becomes

(15) $\exists e(\textbf{praise}(e) \; \& \; \textbf{Ag}(e) = \textbf{j} \; \& \; \textbf{Th}(e) = \textbf{t})$

Because this is the same as (12), the inference (1) is correctly predicted. Similarly, for causative suffixes, given a semantic template

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

the semantic representation of *hasira-se* is obtained as follows.

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

Thus, the left-side of the (2) will be

(18) $\exists e(\textbf{run}(e) \; \& \; \textbf{Cause}(e) = \textbf{t} \; \& \; \textbf{Ag}(e) = \textbf{j})$

which entails $\exists e(\textbf{run}(e) \& \; \textbf{Ag}(e) = \textbf{j})$, the right-side of (2), so the inference (2) is also correctly predicted.

However, this analysis produces incorrect predictions for nesting: the semantic representation of *hasira-sera-re* is obtained by applying (13) to (17), which ends up in (19).

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

Notice that (19) is identical to the one obtained by applying passive suffix *re* directly to *hasira* (i.e., *hasira-re*). From here, neither the right-side of (9) = (2) nor the left-side of (2) is implied.

This error occurs because the passive suffix globally assumes that the first argument is given **Th** and the second argument is given **Ag**. On the contrary, the nesting example shows that the passive suffix connects the first and the second arguments in matrix with the second and the first argument of the verb, respectively. Capturing this behaviour of the passive suffix requires the verb's first and second arguments to be accessible from the syntax and the semantics of the passive suffix, which means the syntactic category of the passive suffix should be exactly $S\backslash NP_{ga}\backslash NP_{ni}\backslash(S\backslash NP_{ga}\backslash NP_{ni|o})$.

4 Conclusion

In this paper, we showed that the syntactic analysis of Japanese CCGBank together with the semantic analysis of ccg2lambda produces false predictions for passive and causative nesting, which means that the current syntactic analysis of passive/causative constructions in Japanese CCGBank does not have a semantic support that correctly predicts the inferences such as (1), (2), and (9). In other words, the claim that *re* and *se* have the syntactic category $S\backslash S$ cannot be maintained. Since the standard analysis described in Section 2 correctly explains all of those inferences, the burden of proof is clearly on the CCGBank side.

An important implication of this paper is that there is a need for outreach to the linguistic community, where not all linguists regard a treebank as an output of a linguistic analysis. We suggest that we should treat treebanks as outputs of some linguistic analyses and try to provide counterexamples in this way in order to keep treebanks and also the subsequent development of syntactic parsers sound from the linguistic perspective.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. This work was partially supported by JST CREST Grant Number JPMJCR20D2, Japan, and JSPS KAKENHI Grant Number JP20K19868, Japan.

References

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

Daisuke Bekki and Koji Mineshima. 2017. Context-passing and underspecification in dependent type semantics. In *Modern Perspectives in Type Theoretical Semantics*, Studies of Linguistics and Philosophy, pages 11–41. Springer.

Gennaro Chierchia. 1995. *Dynamics of Meaning*. Chicago Press.

Stephen Clark and James. R. Curran. 2007. Widecoverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Donald Davidson. 1967. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, Pittsburgh.

Julia Hockenmaier and Mark J. Steedman. 2005. CCG-bank LDC2005T13. Linguistic Data Consortium.

Yusuke Kubota, Koji Mineshima, Noritsugu Hayashi, and Shinya Okano. 2020. Development of a general-purpose categorial grammar treebank. In *the 12th Language Resources and Evaluation Conference*, pages 5195–5201. European Language Resources Association.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000. Association of Computational Linguistics.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A computational semantics system. In *the Association of Computational Linguistics (ACL2016)*, pages 85–90.

Hiroshi Noji and Yusuke Miyao. 2016. Jigg: A framework for an easy natural language processing pipeline. In *the 54th Association of Computational Linguistics*, pages 103–108.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher D. Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO redwoods treebank: Motivation and preliminary applications. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. The MIT Press, Cambridge MA.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29. Istanbul, Turkey.

Mark J. Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press, Cambridge.

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

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The Pargram Parallel Treebank. In *the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*, pages 550–560.

Tu-Anh Tran and Yusuke Miyao. 2022. Development of multilingual CCG treebank via universal dependencies conversion. In *the 13th Conference on Language Resource and Evaluation (LREC2022)*, pages 5220–5233.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage japanese CCG resources. In *the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1042–1051. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pages 277–287.