From Keyaki to ABC

A treebank conversion project

Yusuke Kubota[1] Koji Mineshima[2]

[1]University of Tsukuba

[2]Ochanomizu University

November 4, 2017
NPCMJ Kobe Meeting

Overview

Goal

- Describe an ongoing project of converting the Keyaki Treebank [Butler et al., 2017] to a categorial grammar (CG) treebank.

Roadmap

- Background
- Outline of the treebank conversion process
- Parser demo
- Remaining issues and challenges

Overview

Goal

- Describe an ongoing project of converting the Keyaki Treebank [Butler et al., 2017] to a categorial grammar (CG) treebank.

Roadmap

- Background
- Outline of the treebank conversion process
- Parser demo
- Remaining issues and challenges

Background

ccg2lambda

[Mineshima et al., 2015, Martínez-Gómez et al., 2016, Mineshima et al., 2016]

- Syntactic parser (CCG) + semantic inference system (HOL prover) for solving inference problems.
- Potentially offers a new, powerful methodology for formal semantics research.

Hybrid Type-Logical Categorial Grammar

[Kubota, 2015, Kubota and Levine, 2016, Kubota and Levine, 2017]

- A version of CG that can be thought of as a formalization of the core component of the minimalist syntax.
- Incorporates and improves on a number of major analytic ideas from the mainstream syntactic theory.

Common (larger) goal:

- An attempt to bridge the gap between theoretical linguistics and computational linguistics/NLP.

Background

ccg2lambda

[Mineshima et al., 2015, Martínez-Gómez et al., 2016, Mineshima et al., 2016]

- Syntactic parser (CCG) + semantic inference system (HOL prover) for solving inference problems.
- Potentially offers a new, powerful methodology for formal semantics research.

Hybrid Type-Logical Categorial Grammar

[Kubota, 2015, Kubota and Levine, 2016, Kubota and Levine, 2017]

- A version of CG that can be thought of as a formalization of the core component of the minimalist syntax.
- Incorporates and improves on a number of major analytic ideas from the mainstream syntactic theory.

Common (larger) goal:

- An attempt to bridge the gap between theoretical linguistics and computational linguistics/NLP.

Background

ccg2lambda

[Mineshima et al., 2015, Martínez-Gómez et al., 2016, Mineshima et al., 2016]

- Syntactic parser (CCG) + semantic inference system (HOL prover) for solving inference problems.
- Potentially offers a new, powerful methodology for formal semantics research.

Hybrid Type-Logical Categorial Grammar

[Kubota, 2015, Kubota and Levine, 2016, Kubota and Levine, 2017]

- A version of CG that can be thought of as a formalization of the core component of the minimalist syntax.
- Incorporates and improves on a number of major analytic ideas from the mainstream syntactic theory.

Common (larger) goal:

- An attempt to bridge the gap between theoretical linguistics and computational linguistics/NLP.

Things still lacking

ccg2lambda: A linguistically adequate parser

- The analyses implemented in the system are hard to understand for ordinary linguists.
- Currently still unclear whether this work is 'mere formalization' of pencil-and-paper formal semantics or something more.

Hybrid TLCG: An efficient parser

- Since the theory is complex (as it's essentially a formalization of the 'derivational' architecture of grammar), there is as yet no efficient parser comparable to state-of-the-art CCG parsers.
- Without a robust parser, the possibilities of an explicit, formalized grammar are very limited.

Common next step:

- We both need a good CG treebank.

Things still lacking

ccg2lambda: A linguistically adequate parser

- The analyses implemented in the system are hard to understand for ordinary linguists.
- Currently still unclear whether this work is 'mere formalization' of pencil-and-paper formal semantics or something more.

Hybrid TLCG: An efficient parser

- Since the theory is complex (as it's essentially a formalization of the 'derivational' architecture of grammar), there is as yet no efficient parser comparable to state-of-the-art CCG parsers.
- Without a robust parser, the possibilities of an explicit, formalized grammar are very limited.

Common next step:

- We both need a good CG treebank.

Things still lacking

`ccg2lambda`: A linguistically adequate parser

- The analyses implemented in the system are hard to understand for ordinary linguists.
- Currently still unclear whether this work is 'mere formalization' of pencil-and-paper formal semantics or something more.

Hybrid TLCG: An efficient parser

- Since the theory is complex (as it's essentially a formalization of the 'derivational' architecture of grammar), there is as yet no efficient parser comparable to state-of-the-art CCG parsers.
- Without a robust parser, the possibilities of an explicit, formalized grammar are very limited.

Common next step:

- We both need a good CG treebank.

Desiderata

Linguistic adequacy

- incorporate sound linguistic analyses of major syntactic phenomena in Japanese, e.g.,
  - quantification (including floated quantifiers)
  - argument sharing in (syntactic) complex predicates
- transparent syntax-semantics interface

Versatility

- can be easily converted to different grammatical theories:
  - CCG
  - Hybrid TLCG/'movement'-based syntax
  - HPSG/LFG
- can be used as a learning dataset for parsers

(Somewhat) larger goal

- facilitate comparison of different theories based on
  - explicit formalization
  - large-scale attested data

Desiderata

Linguistic adequacy

- incorporate sound linguistic analyses of major syntactic phenomena in Japanese, e.g.,
  - quantification (including floated quantifiers)
  - argument sharing in (syntactic) complex predicates
- transparent syntax-semantics interface

Versatility

- can be easily converted to different grammatical theories:
  - CCG
  - Hybrid TLCG/'movement'-based syntax
  - HPSG/LFG
- can be used as a learning dataset for parsers

(Somewhat) larger goal

- facilitate comparison of different theories based on
  - explicit formalization
  - large-scale attested data

Desiderata

Linguistic adequacy

- incorporate sound linguistic analyses of major syntactic phenomena in Japanese, e.g.,
  - quantification (including floated quantifiers)
  - argument sharing in (syntactic) complex predicates
- transparent syntax-semantics interface

Versatility

- can be easily converted to different grammatical theories:
  - CCG
  - Hybrid TLCG/'movement'-based syntax
  - HPSG/LFG
- can be used as a learning dataset for parsers

(Somewhat) larger goal

- facilitate comparison of different theories based on
  - explicit formalization
  - large-scale attested data

Building a CG Treebank from a PSG Treebank

Previous work [Hockenmaier and Steedman, 2007, Uematsu et al., 2013, Moot, 2015]

	original corpus	CG variant	Language
H&S	Penn Treebank	CCG	English
Uematsu et al.	Kyoto Corpus	CCG	Japanese
Moot	French PSG Bank	TLCG	French

Challenges for current work

- Keyaki Treebank contains rich linguistic information, such as:
  - grammatical relations
  - quantification (including floated quantifiers)
  - fine-grained distinction of empty elements (trace, pro, PRO, exp, arb)
- We don't want a CCG treebank or a TLCG treebank; we want both.

Building a CG Treebank from a PSG Treebank

Previous work [Hockenmaier and Steedman, 2007, Uematsu et al., 2013, Moot, 2015]

	original corpus	CG variant	Language
H&S	Penn Treebank	CCG	English
Uematsu et al.	Kyoto Corpus	CCG	Japanese
Moot	French PSG Bank	TLCG	French

Challenges for current work

- Keyaki Treebank contains rich linguistic information, such as:
  - grammatical relations
  - quantification (including floated quantifiers)
  - fine-grained distinction of empty elements (trace, pro, PRO, exp, arb)
- We don't want a CCG treebank or a TLCG treebank; we want both.

ABC Grammar as an 'inter-language'

ABC Grammar

= AB Grammar + (Harmonic) Function Composition

≈ PSG + (a little bit of) 'syntactic movement'

- Can be thought of as a convenient 'inter-language' mediating a PSG treebank and different types of CG treebanks
- So, we don't mean to propose it as a serious linguistic theory (just like an interlanguage isn't a real language); it's only a step toward an adequate linguistic theory

Main advantages:

- simple and easy to understand
- can already capture many important linguistic generalizations
- not too parochial ('let's forget about the battle between CCG and TLCG for the time being')

ABC Grammar as an 'inter-language'

ABC Grammar

$$= \text{AB Grammar} + \text{(Harmonic) Function Composition}$$

$$\approx \text{PSG} + \text{(a little bit of) 'syntactic movement'}$$

- Can be thought of as a convenient 'inter-language' mediating a PSG treebank and different types of CG treebanks
- So, we don't mean to propose it as a serious linguistic theory (just like an interlanguage isn't a real language); it's only a step toward an adequate linguistic theory

Main advantages:

- simple and easy to understand
- can already capture many important linguistic generalizations
- not too parochial ('let's forget about the battle between CCG and TLCG for the time being')

Some linguistic analyses in ABC Grammar

AB grammar

$$\dfrac{\dfrac{\text{John}}{\text{NP}} \quad \dfrac{\dfrac{\text{read}}{(\text{NP}\backslash\text{S})/\text{NP}} \quad \dfrac{\text{PTQ}}{\text{NP}}}{\text{NP}\backslash\text{S}}}{\text{S}}$$

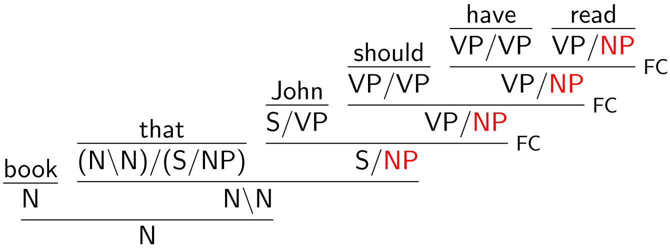
Function Application:

$$\text{A}/\text{B} \quad \text{B} \quad \Rightarrow \quad \text{A}$$

$$\text{B} \quad \text{B}\backslash\text{A} \quad \Rightarrow \quad \text{A}$$

Some linguistic analyses in ABC Grammar

wh-movement (in English)

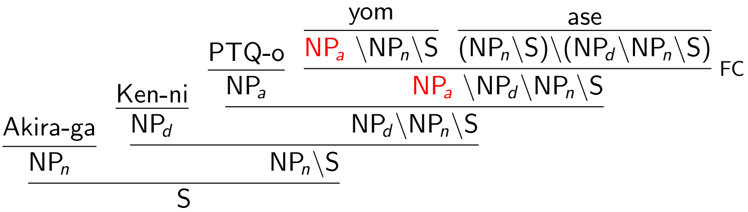


Function Composition:

$A/B \quad B/C \quad \Rightarrow \quad A/C$

Some linguistic analyses in ABC Grammar

Causative in Japanese



Function Composition:

$A\backslash B \quad B\backslash C \quad \Rightarrow \quad A\backslash C$

This is sort of like

- argument transfer / argument composition (in LFG, HPSG)
- head movement (in GB)

Some linguistic analyses in ABC Grammar

Causative in Japanese

$$\dfrac{\dfrac{\text{Akira-ga}}{NP_n} \quad \dfrac{\dfrac{\text{Ken-ni}}{NP_d} \quad \dfrac{\dfrac{\text{PTQ-o}}{NP_a} \quad \dfrac{\dfrac{\text{yom}}{NP_a \backslash NP_n \backslash S} \quad \dfrac{\text{ase}}{(NP_n \backslash S) \backslash (NP_d \backslash NP_n \backslash S)}}{NP_a \backslash NP_d \backslash NP_n \backslash S}\,\text{FC}}{NP_d \backslash NP_n \backslash S}}{NP_n \backslash S}}{S}$$

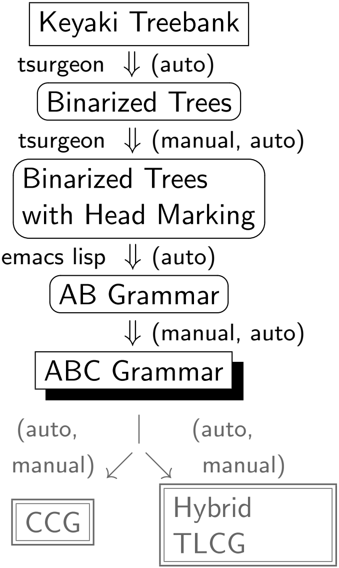
Function Composition:

$$A \backslash B \quad B \backslash C \quad \Rightarrow \quad A \backslash C$$

This is sort of like

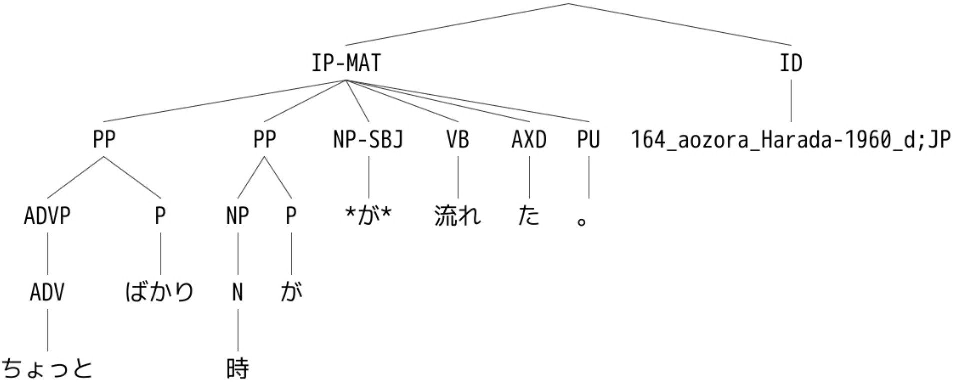
- argument transfer / argument composition (in LFG, HPSG)
- head movement (in GB)

Conversion process



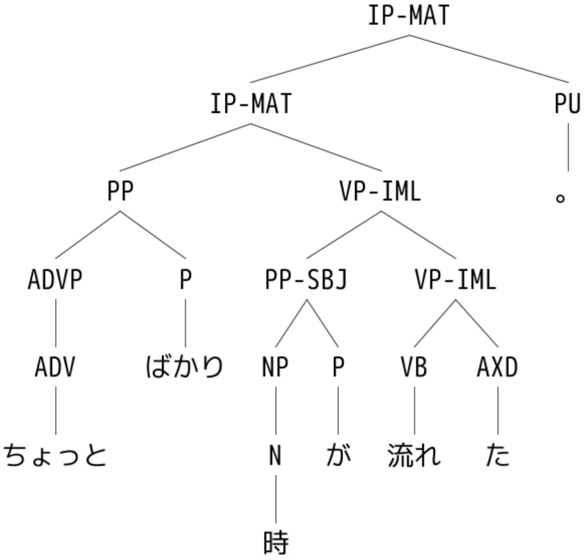
From Keyaki to AB

Keyaki tree:



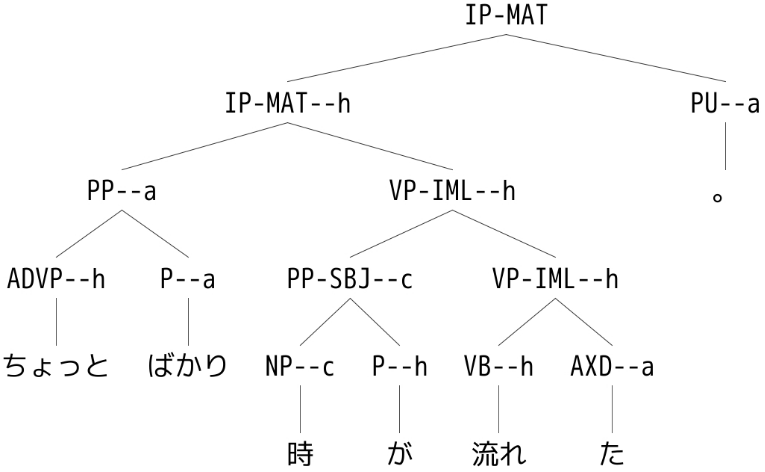
From Keyaki to AB

Binarized tree:



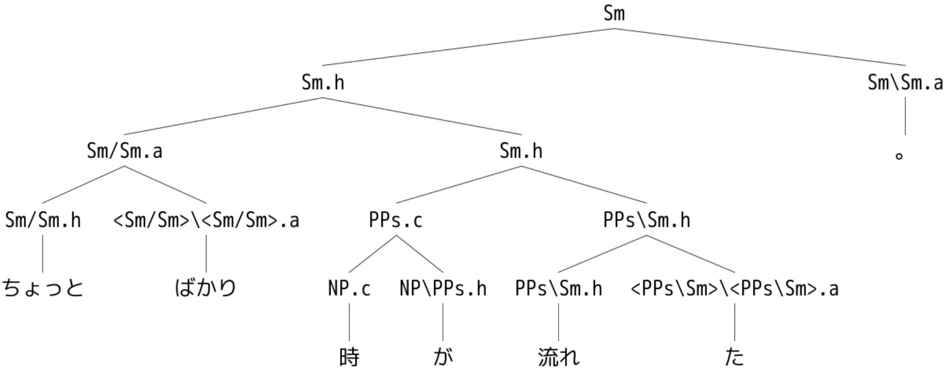
From Keyaki to AB

Head-dependent marking:



From Keyaki to AB

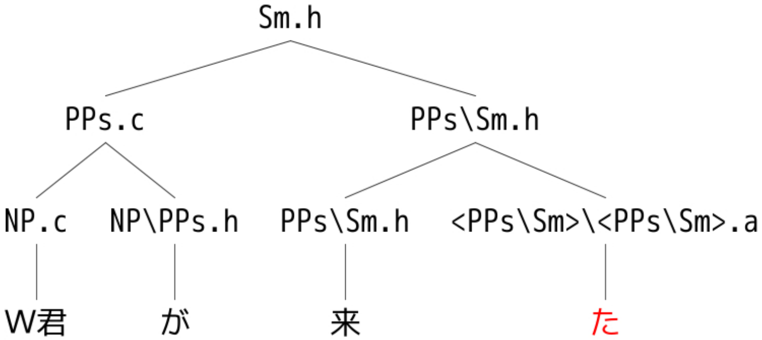
AB tree:



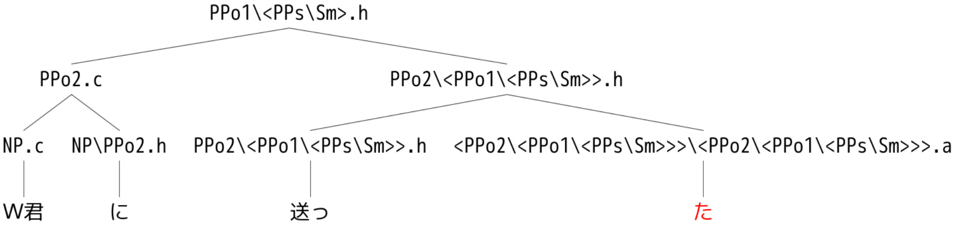
Why not stop here?

- AB grammar is like PSG without movement
- So, at this point, the treebank looks like:
  - GB syntax without movement
  - HSPG without the SLASH feature, argument composition
  - LFG without f-structure
- More specifically, there's massive lexical redundancy ☹

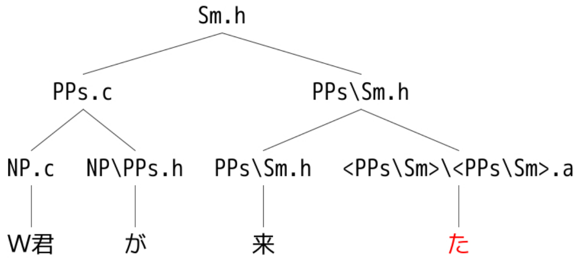
From AB to ABC

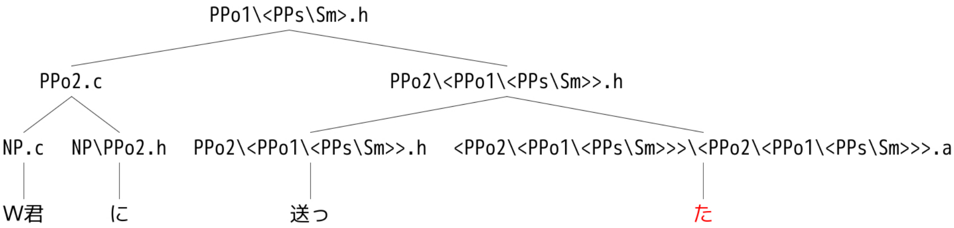


From AB to ABC



From AB to ABC





From AB to ABC

Same category for *ta* suffices if we have Function Composition:

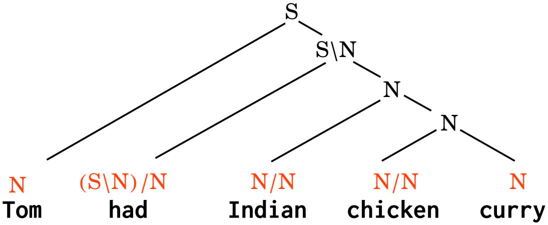
ki	ta		
$PP\backslash S$	$S\backslash S$	$\Rightarrow$	$PP\backslash S$
okut	ta		
$PP\backslash PP\backslash PP\backslash S$	$S\backslash S$	$\Rightarrow$	$PP\backslash PP\backslash PP\backslash S$

Demo

- This part is joint work with Masashi Yoshikawa (NAIST)
- CCG Parser: depccg [Yoshikawa et al., 2017]
  `https://github.com/masashi-y/depccg`
- Training data: a pilot version of AB grammar treebank converted from NPCMJ (10K sentences)
- Interface with ccg2lambda [Mineshima et al., 2015]
  `https://github.com/mynlp/ccg2lambda`
- Features:
  - Compositional semantics
  - Automatic theorem proving

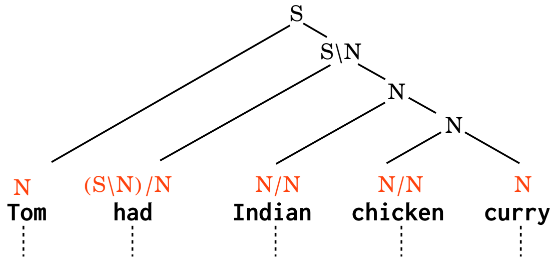
Combinatory Categorial Grammar (CCG)

- Rich supertags, a small set of rules
- Supertagging is almost parsing (Bangalore and Joshi, 1999)
  - Given the supertags, the tree structure below is unique under normal form.



Supertag-factored model [Lewis and Steedman, 2014]

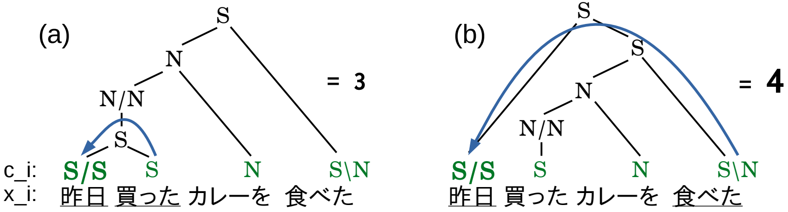
- The probability of a tree is the product of **supertag** probabilities
- CCG Parsing:
  - Find the best supertag sequence that forms a tree
    - → Efficient A* search is possible



$$P(\mathbf{y}) = P_{\text{tag}}(N)\ P_{\text{tag}}(S\backslash N/N)\ P_{\text{tag}}(N/N)\ P_{\text{tag}}(N/N)\ P_{\text{tag}}(N)$$

Limitation of supertag-factored model

- The same list of supertags can result in more than one tree.
- The model cannot decide which one is better.



Supertag & Dependency Factored Model [Yoshikawa et al., 2017]

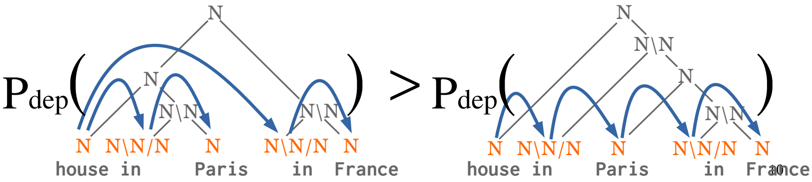
- The probability of a CCG tree is the product of the probabilities of the **supertags** and **dependency structure**

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{c_i \in \boldsymbol{y}} P_{tag}(c_i|x_i) \prod_{h_i \in \boldsymbol{y}} P_{dep}(h_i|x_i)$$

- What if there are two trees from the same supertags?

  → Choose one with **the higher scoring dep. structure**

- **KEY**: a simpler dependency model still allows efficient A* decoding



Some issues and challenges

1. ‘controlled’ PRO; cf. ID 147
2. argument vs. adjunct; cf. ID 51
3. renyookei, -*te* form; cf. ID 147

Butler, A., Yoshimoto, K., Hiyama, S., Horn, S. W., Nagasaki, I., and Kubota, A. (2017).
The Keyaki Treebank Parsed Corpus, version 1.0.
http://www.compling.jp/Keyaki/, accessed 2017/07/26.

Hockenmaier, J. and Steedman, M. (2007).
CCGbank: A corpus of CCG derivations and dependency structures extracted from the penn treebank.
Computational Linguistics, 33(3):355–396.

Kubota, Y. (2015).
Nonconstituent coordination in Japanese as constituent coordination: An analysis in Hybrid Type-Logical Categorial Grammar.
Linguistic Inquiry, 46(1):1–42.

Kubota, Y. and Levine, R. (2016).
Gapping as hypothetical reasoning.
Natural Language and Linguistic Theory, 34(1):107–156.

Kubota, Y. and Levine, R. (2017).
Pseudogapping as pseudo-VP ellipsis.
Linguistic Inquiry, 48(2):213–257.

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

Martínez-Gómez, P., Mineshima, K., Miyao, Y., and Bekki, D. (2016).
ccg2lambda: A compositional semantics system.
In *Proceedings of ACL 2016 System Demonstrations*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

Mineshima, K., Martínez-Gómez, P., Miyao, Y., and Bekki, D. (2015).
Higher-order logical inference with compositional semantics.
In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Mineshima, K., Tanaka, R., Martínez-Gómez, P., Miyao, Y., and Bekki, D. (2016).
Building compositional semantics and higher-order inference system for a wide-coverage Japanese CCG parser.
In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Austin, Texas. Association for Computational Linguistics.

Moot, R. (2015).
A type-logical treebank for French.
Journal of Language Modelling, 3(1):229–264.

Uematsu, S., Matsuzaki, T., Hanaoka, H., Miyao, Y., and Mima, H. (2013).
Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources.
In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1042–1051.

Yoshikawa, M., Noji, H., and Matsumoto, Y. (2017).
A* CCG parsing with a supertag and dependency factored model.
CoRR, abs/1704.06936.