Formal Approaches to Japanese Linguistics 8
Mie University: Tsu, Mie Japan
February 18, 2016

A Labeling-based Account of Japanese Imperatives*

Jason Ginsburg[1] Naomi Ogasawara[2]
[1]Osaka Kyoiku University [2]University of Aizu

*This work is supported by a Japan Society for the Promotion of Science Grant-in-Aid for Scientific Research: Classification and Linguistic Analysis of Natural Disaster Evacuation Calls (#26370490)

1. Introduction

- Problems of Projection (POP) (Chomsky 2013, 2015)
  - Core syntactic operations are connected with the need for Syntactic Objects (SOs) to be labeled.
- We explain how a POP-based computer model constructs the derivations of basic imperative sentences in Japanese.
- We examine a real-world application of this model.
  - Can a model of syntax have applications for disaster warnings?

2. Core assumptions

Chomsky (2013, 2015):

- The phase heads v* and C have uninterpretable phi-features uPhi.
  - uPhi are inherited by T from C.
  - uPhi are inherited by a verbal root V from v*.
- A Labeling Algorithm determines the label of a syntactic object (SO) by finding prominent features that are capable of labeling (e.g., phi-features).

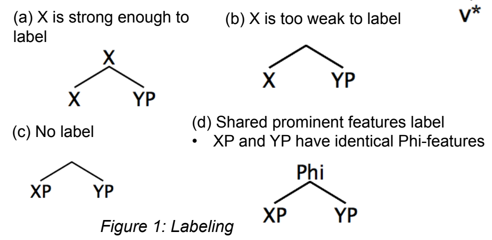
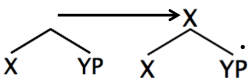


Figure 1: Labeling

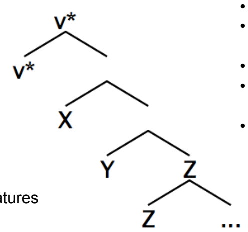
- An SO can be labeled via strengthening.
  - Strengthening refers to when an SO that is initially too weak to be labeled obtains prominent features that are capable of labeling.



- X inherits features that are capable of labeling
- Strengthened X labels

Figure 2: Labeling via strengthening

Proposals (cf. Ginsburg To Appear):
(1) Uninterpretable features are passed onto a complement that is too weak to label (based on Chomsky 2013, 2015).
(2) Feature inheritance (cf. Fong, 2014) leads to unified instances of a feature on multiple Syntactic Objects.



- X and Y are weak
- uPhi of v* are passed from v* to X
- uPhi are passed from X to Y
- uPhi on v*, X, and Y are unified
- If any instance of uPhi is checked, all unified instances of uPhi are checked

Figure 3: Feature transfer and unification

- Given an unlabeled {XP, YP} structure, if XP moves out, then the label of YP becomes the label.

(a) unlabeled {XP, YP} structure (b) labeled by 'read+v*' because 'Tom' has moved

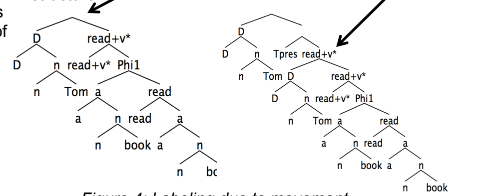
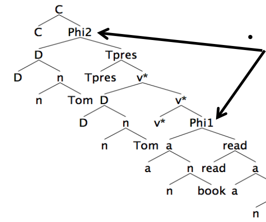


Figure 4: Labeling due to movement

- In languages such as English, T and a verbal root must be labeled via Phi-features shared with a remerged SO.



- Must be labeled via shared Phi-features

Figure 5: Shared Phi-features are necessary for labeling projections of T and verbal root

Proposal:
(3) in Japanese, T and verbal roots are labeled via strengthening.
- **Strengthening does not require phi-features that are shared with a remerged SO.**

3. Target Derivation

(4) ***Minna-san-wa hinan-shite kudasai***
Everyone-Top evacuate-do please
Everyone, please evacuate.
(Adapted from a Nemuro, Hokkaido evacuation call)

(5) Proposals about (4):
- ***shite kudasai* 'please do' contains a V-V serial verb construction (cf. Nishiyama 1998) consisting of two verbal roots that Merge with a single v*.**
  - ***-te* is part of the verbal projection (cf. Sugita 2009).**
- **v* assigns a subject theta-role**
  - **Both verbal roots, *shite* and *kuda*, essentially have a single subject.**

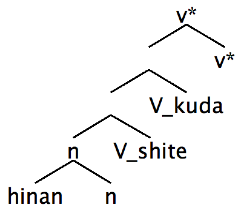
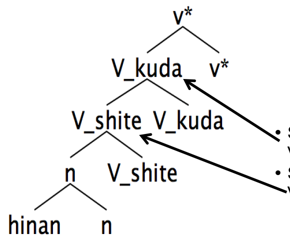


Figure 6: v*P of (4) before labeling

- n Merges with the root *hinan* 'evacuation' and n labels.
  - n is strong enough to label.
- Verbal root V_shite (*shite* 'do') is Merged.
- Verbal root V_kuda (*kuda* + *sai* = 'please') is Merged.
  - verbal roots are too weak to label
- The phase head v* is Merged.
  - uPhi of v* are passed to V_kuda.
  - uPhi are passed from V_kuda to V_shite.
  - v*, V_kuda, V_shite contain unified uPhi.



- Strengthened V_kuda labels
- Strengthened V_shite labels

Figure 7: v*P of (4) after labeling

- The uPhi features on v* Agree with the nominal *hinan* 'evacuation'.
  - uPhi on v* are checked by the valued phi-features of *hinan* 'evacuation'.
  - *hinan* 'evacuation' obtains Case.
  - unified uPhi on V_kuda and V_shite are checked.
  - V_kuda and V_shite are strengthened.
- The Labeling Algorithm finds the checked phi-features on the strengthened V_kuda and V_shite.
  - V_kuda and V_shite label.

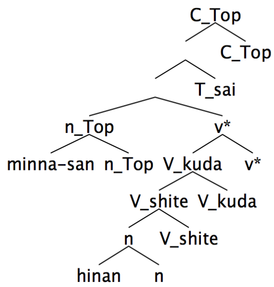
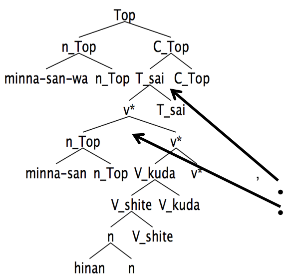


Figure 8: CP of (4) before labeling

- The subject *minna-san* 'everyone' is Merged with the v* projection.
  - unlabeled {XP, YP} structure
- Merge T_sai.
  - *–sai* in *kudasai* 'please' is a T head.
  - T_sai is too weak to label.
- C_Top is Merged.
  - C_Top is a C phase head with a Topicalization feature Top.
- uPhi from C_Top are inherited by T_sai.

Proposal:
(6) Movement of an SO occurs to create a structure that can be labeled for semantic reasons.



- Strengthened T_sai labels
- v* labels because subject has moved

Figure 9: CP of (4) after labeling

- Agree(C_Top,minna-san)
  - The subject remerges with C_Top, in accord with (6).
    - Movement creates a structure that can be labeled by a shared Topic feature.
  - The subject has a Top feature that checks a uTop feature on C_Top.
  - The subject obtains Case from C_Top.
    - Top surfaces as the topic particle *wa* 'Top'.
  - The unified uPhi on T_sai are checked.
- The Labeling Algorithm labels the strengthened T_sai.
- The lower v* projection is labeled by v* because the subject has moved out.
- The subject and C_Top are labeled via shared Top features.

4. Cost

- This computer model automatically computes the cost of core operations in a derivation.
  (7) Cost:
  - **Merge Cost:** add 1 for Merge of X and Y.
  - **Feature Inheritance Cost**: add 1 for inheritance of features from X by Y, regardless of the number of features involved.
  - **Feature Checking Cost:** add 1 for checking of features on X by features of Y, regardless of the number of features involved.
- Costs calculated for the derivation of (4)
  - ***Merge Cost*: 11; *Feature Inheritance Cost:* 3; *Feature Checking Cost:* 5**
- Computing the cost of more complex constructions can be done instantaneously and accurately by a computer.
- If the computational cost of a sentence can be connected with the actual cognitive burden of processing a sentence, then this type of model could be useful for determining optimal expressions for disaster situations, etc.

5. Conclusions

- We've shown how:
  - this model automatically generates a Japanese imperative construction.
  - this model calculates cost of a derivation.
- Research questions for future work:
  - Can this model automatically generate a wider variety of imperative constructions in Japanese?
  - What is the most accurate way to calculate cost of a derivation?
  - How best can information about cost be used?
  - Can cost can be linked to cognitive processing load, as measured in psycholinguistics experiments?
- There may be real-world applications for this type of model, especially if cost can be linked to cognitive processing load.

References:
Chomsky, Noam. 2013. Problems of projection. *Lingua* 130: 33-49.
Chomsky, Noam. 2015. Problems of projections: Extensions. In *Structures, strategies and beyond: Studies in honour of Adriana Belletti*, ed. by E. D. Domenico *et al*, pp. 1-16 pp. 1-16.
Fong, Sandiway. 2014. Unification and efficient computation in the Minimalist Program. In Lowenthal, F. & L. Lefebvre (eds.), *Language and recursion*, 129-138. New York: Springer.
Ginsburg, Jason. To appear. Modeling of Problems of Projection: A non-countercyclic approach. *Glossa*.
Nishiyama, K. 1998. V-V compounds as serialization. *Journal of East Asian Linguistics* 8:183-222.
Sugita, M. 2009. *Japanese –te iru and –te aru: The aspectual implications of the stage-level and individual level distinction*. CUNY Phd thesis.