Introduction

- People process sentences **incrementally** and build representations without delay as each word is read or heard. (Marslen-Wilson, 1975)

- Three alternative algorithms for building tree structures.
  - bottom-up parsing
  - top-down parsing
  - left-corner parsing (LC parsing)

Parsing algorithms

- **bottom-up parsing** $B \Rightarrow C \Rightarrow A$
  - conservative: never makes predictions

- **top-down parsing** $A \Rightarrow B \Rightarrow C$
  - liberal: can make an unbounded number of predictions

- **left-corner parsing**
  - limited predictions:
    - based on the left-corner node $B$, projects $A$
    - based on $A$, predicts $C$
LC parsing in Japanese

In Japanese, how does LC parsing construct the tree structure, when the sentence-initial NP input is nominative or dative?

Attachment preference in Japanese

"Taro" can be attached to either verb.

a. Taro-\text{-}ga hon-\text{-}o katta-\text{-}to itta.
   Taro\_nom book\_ACC bought\_that said

b. Taro-\text{-}ni hon-\text{-}o katta-\text{-}to itta.
   Taro\_dat book\_ACC bought\_that said

Attachment preferences:
When "Taro" is nominative $\rightarrow$ it attaches to the matrix verb "said"
When "Taro" is dative $\rightarrow$ it attaches to the embedded verb "bought"
(Miyamoto, 2007)

$\Rightarrow$ Only LC parsing can predict this preference.

In sum:

LC parsing has advantages over top-down parsing and bottom-up parsing.

- It correctly predicts the difficulty in center-embedded constructions
- It explains attachment preferences for sentence-initial NPs in Japanese
  (Miyamoto, 2007)

Purpose of this study

We report:
- Evidence supporting LC parsing as a better candidate to explain memory load in the comprehension of sentences in Japanese
- Evidence against processing models in which working-memory load is determined by dependency relations alone (e.g., the Dependency Locality Theory, DLT, Nakatani & Gibson, 2010).

LC parsing and working memory load

LC parsing should affect the memory load of sentence-initial NPs in Japanese.

Taro-ga/-ni long VP-internal adjunct V.
NP-nom/dat $\Rightarrow$ $\Rightarrow$ $\Rightarrow$

Working memory load in the VP-internal adjunct...

- In the NP-dat, it increases.
- In the NP-nom, it does not increase.

$\Leftrightarrow$ DLT: only considers dependencies needed to complete the sentence; it cannot explain this difference.
When the NP is **dative**, working memory load in the VP-internal adjunct **increases**.

When the NP is **nominative**, working memory load in the VP-internal adjunct **does not increase**.

**Reading-time experiment**

2. **Nominative NP condition**
   - Taro ga kôhî-no oisari Ginza-no oshigare-na kissaten de netaeita.
   - Taro, NOM coffee—GEN tasty Ginza—GEN stylish café—LOC slept.
   - Taro was sleeping in a stylish café in Ginza whose coffee is tasty.

b. **Dative NP condition**
   - Taro-ni kôhî-no oisari Ginza-no oshigare-na kissaten de atta.
   - Taro, DAT coffee—GEN tasty Ginza—GEN stylish café—LOC met.
   - (Someone) met Taro in a stylish café in Ginza whose coffee is tasty.

**Prediction**

Reading Times at the critical region (VP-internal Adjunct) should be...

**LC parsing:** Working memory load in the dat condition is bigger.
→ longer in 2b (dat condition) than in 2a (nom condition).

**Bottom-up parsing:** It does not predict upcoming heads.
→ both conditions are equally easy. (no difference)

**Top-down parsing:** It attaches incoming materials to TP1.
→ both conditions are equally costly. (no difference)

※Models like DLT predict **no difference** in the critical region.

**Method**

- **Participants**: 14 native Japanese speakers from the University of Tokyo.
- **Materials**: 24 sets of experimental items with 48 distractors presented in pseudo-random order.
- **Analysis**: Reading-time analyses were conducted for the sentences whose comprehension question was answered correctly.
- **Trimming**: Points beyond 5000 ms and below 150 ms were removed.
  - Then, points beyond 3 SD from the condition-region mean were eliminated (less than 2% eliminated).
Results

- In the Critical Region (Regions 2 to 6), reading times in the dative condition were statistically longer than in the nominative condition.

![Graph showing reading times comparison]

Discussion

- In the critical region (also when excluding region 2), reading times were statistically longer in the dative condition. ⇒ greater working memory load in the dative condition.
- As predicted by LC parsing. Against: bottom-up and to-down parsing. (predicted no difference) ⇒ LC parsing is the most plausible alternative.
- Keeping track of the number of syntactic head alone is not enough (contra DLT: Nakatani & Gibson, 2010).

On-going experiments

1) LC parsing predicts the same working memory load for dative NPs and accusative NPs. ⇒ A new experiment including an accusative condition.
2) What caused the long reading times? Was it really working memory load? ⇒ In ERP, memory load is associated with a sustained anterior negativity (e.g., in Japanese, Ueno & Kluender, 2003).

References