

# 比較文の計算意味論の構築 (試論)

窪田悠介

国立国語研究所

## 概要

- ▶ 「係り受け」以上に深い構造を自動解析するシステムを構築したい
- ▶ その手始めとして、日本語の比較文の意味解析に取り組む
- ▶ このポスターでは、その研究の進捗を報告する

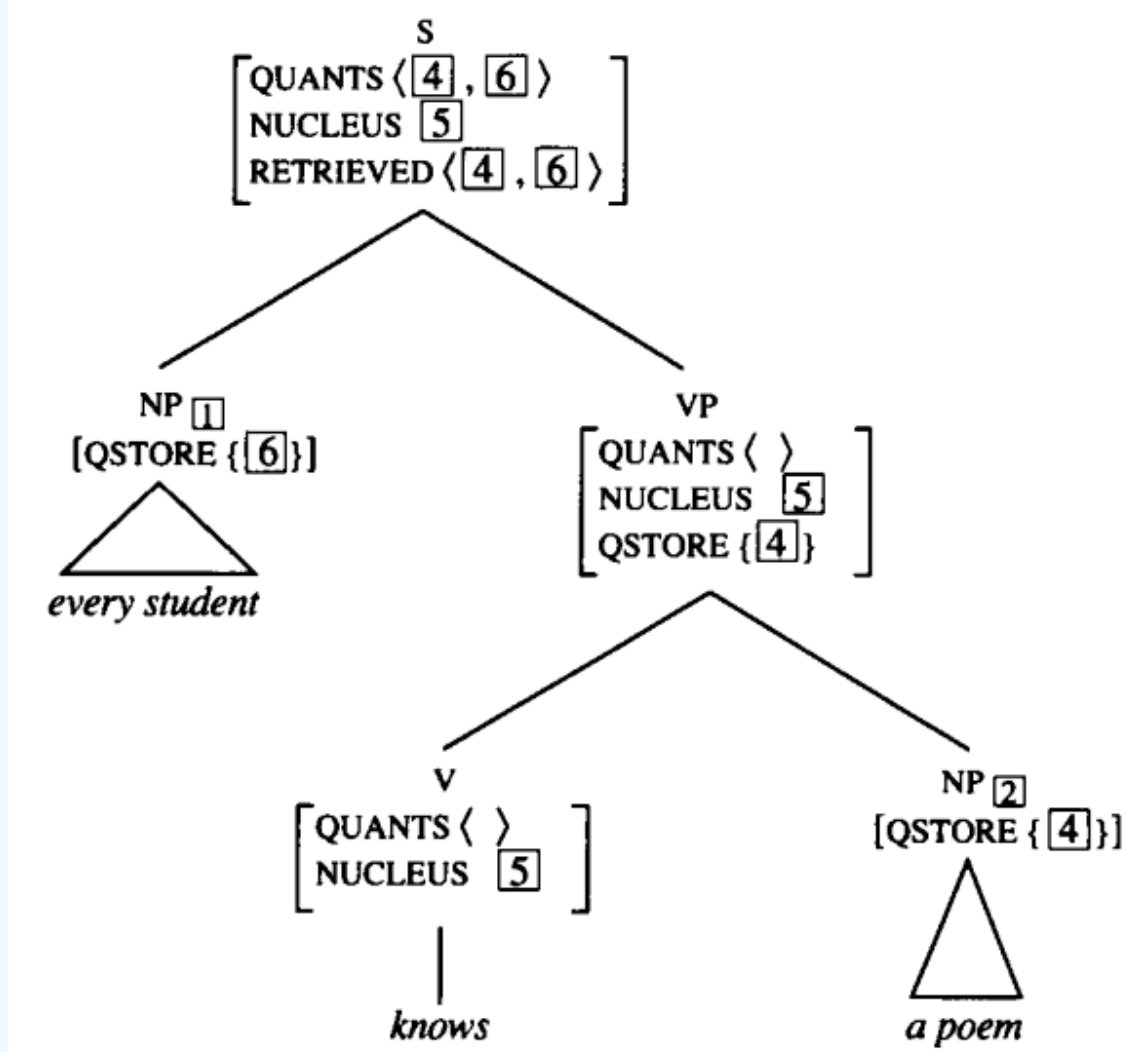
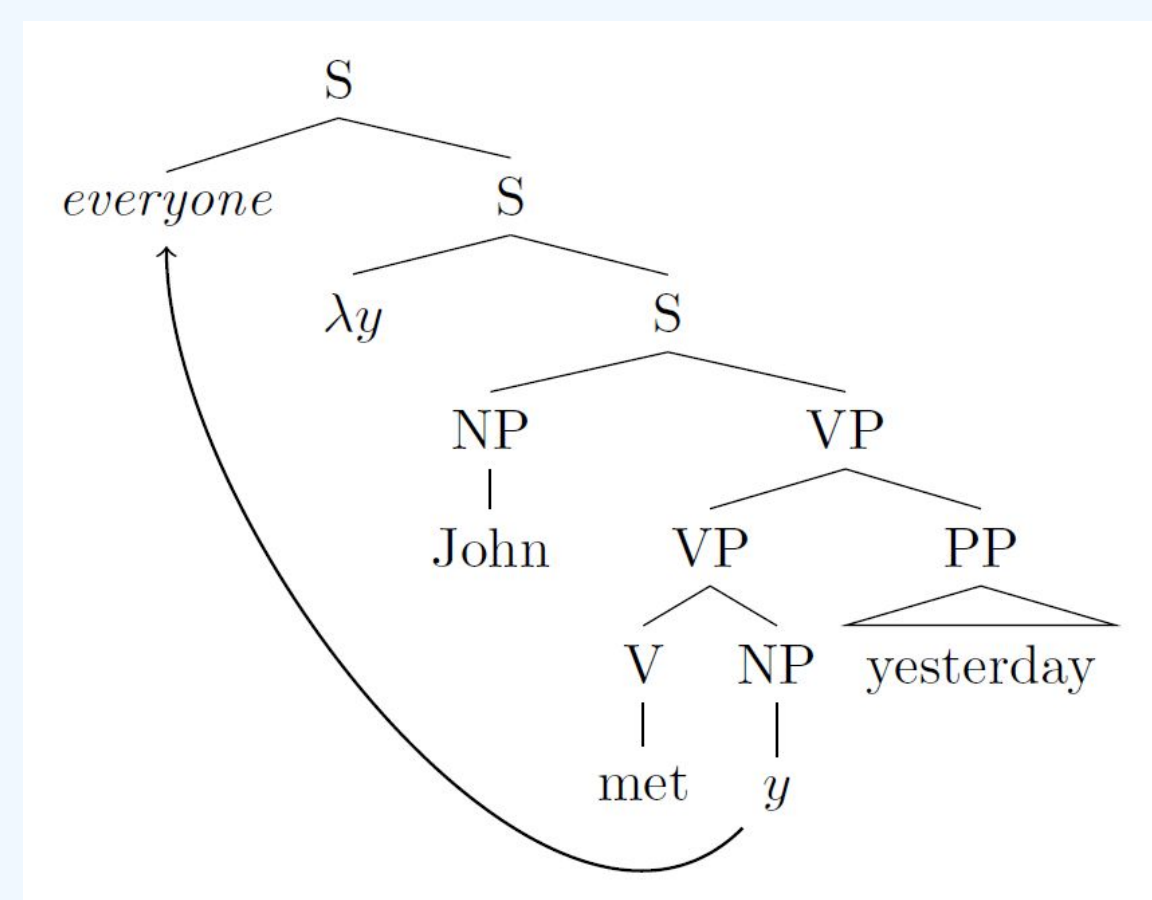
## 統語変換とは?

### 統語変換

- (1) a. Which language does John speak \_\_?  
b. Which language do you think [John speaks \_\_]?
  - (2) a. ジョンは 何語を 話せるの?  
b. 君は [ジョンは 何語を 話せる] と思う?
- ▶ 英語はwh句が表層で動くが、日本語(など)ではcovertに動く (Huang 1982)

### covert movementの例

### 移動を使わない分析 (HPSG)



### 疑問:

- ▶ そもそも何故自然言語には統語変換があるのか?
- ▶ 本当に統語変換などというものはあるのか?
- ▶ 何を示したら統語変換がある (あるいはない) と言えるのか?

## コードの抽象化 (メタプログラミング)

- ▶ 構造の抽象化を行うことの利点は自然言語の分析に限ったことではない
- ▶ 例: Lisp (最も古いプログラミング言語の一つ)
- ▶ Lispのコードはただの構文木

```
(defun treep (exp)
  (or (symbolp exp)
      (numberp exp)
      (and (symbolp (car exp))
           (let ((l (mapcar #'treep (cdr exp))))
             (and 1 (not (member nil l)))))))
```

- ▶ マクロ (構文木の変換) を使うと 複雑なプログラムを簡潔に書ける

## コードの抽象化 (自然言語の場合): 概念的基盤

### 提案: 統語変換は自然言語におけるマクロ展開

- ▶ 自然言語の統語構造をプログラムと見なす
- ▶ プログラムを実行すると文の意味解釈が得られる
- ▶ 表層の文構造 (≈ 係り受けの分析) はマクロを使って書かれたプログラム ⇒ これを直接計算するのではなく、マクロ展開してから意味計算する
- ▶ 人間の脳は神経回路からなる超高性能並列計算機
- ▶ マクロ表記のさらなる略記 (あいまい性を含む) を実時間で難なく解釈できる

### 問 (あるいは謎):

- ▶ 何故自然言語はそこまでコンパクトな設計になっているのか?
- ▶ ウェットウェア (脳の構造) の制約?
- ▶ 進化?
- ▶ 言語の社会的機能?

## コードの抽象化 (自然言語の場合): 実際にやってみる

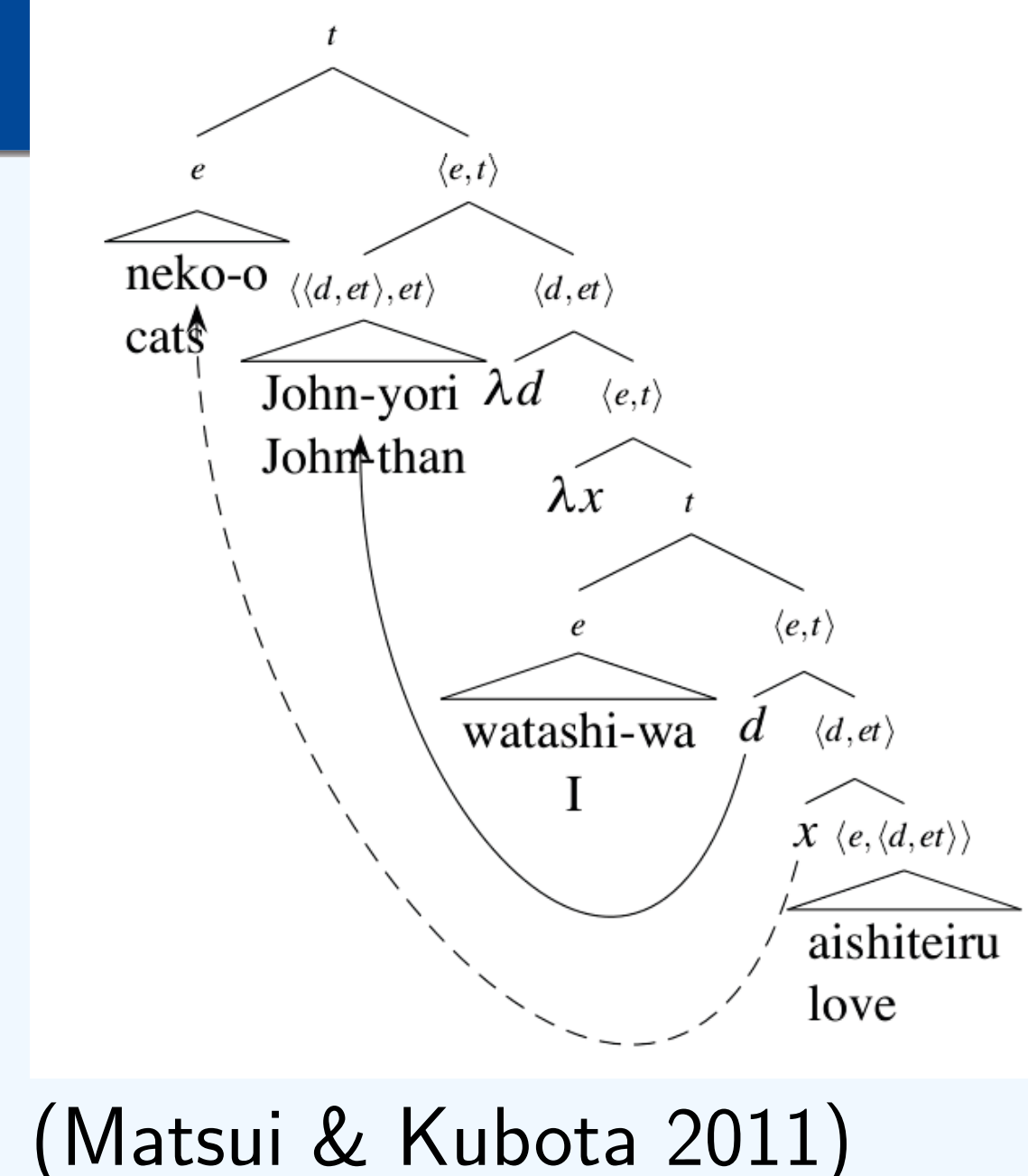
材料: (いくつかは借り物、いくつかは自前で作ったもの)

- ▶ 概念的背景: 生成文法の統語論・意味論研究
- ▶ 理論的基盤: ハイブリッドCG (Kubota & Levine 2020)
- ▶ 言語資源: ABCツリーバンク (+ 統語変換アノテーションの上乗せ)
- ▶ 構文解析器: depccg
- ▶ 実装用の言語: Common Lisp (と、その他プログラミング環境)

## 実際にやってみる (題材)

題材: 日本語の比較構文

- (3) 花子は太郎よりポチが好きだ。



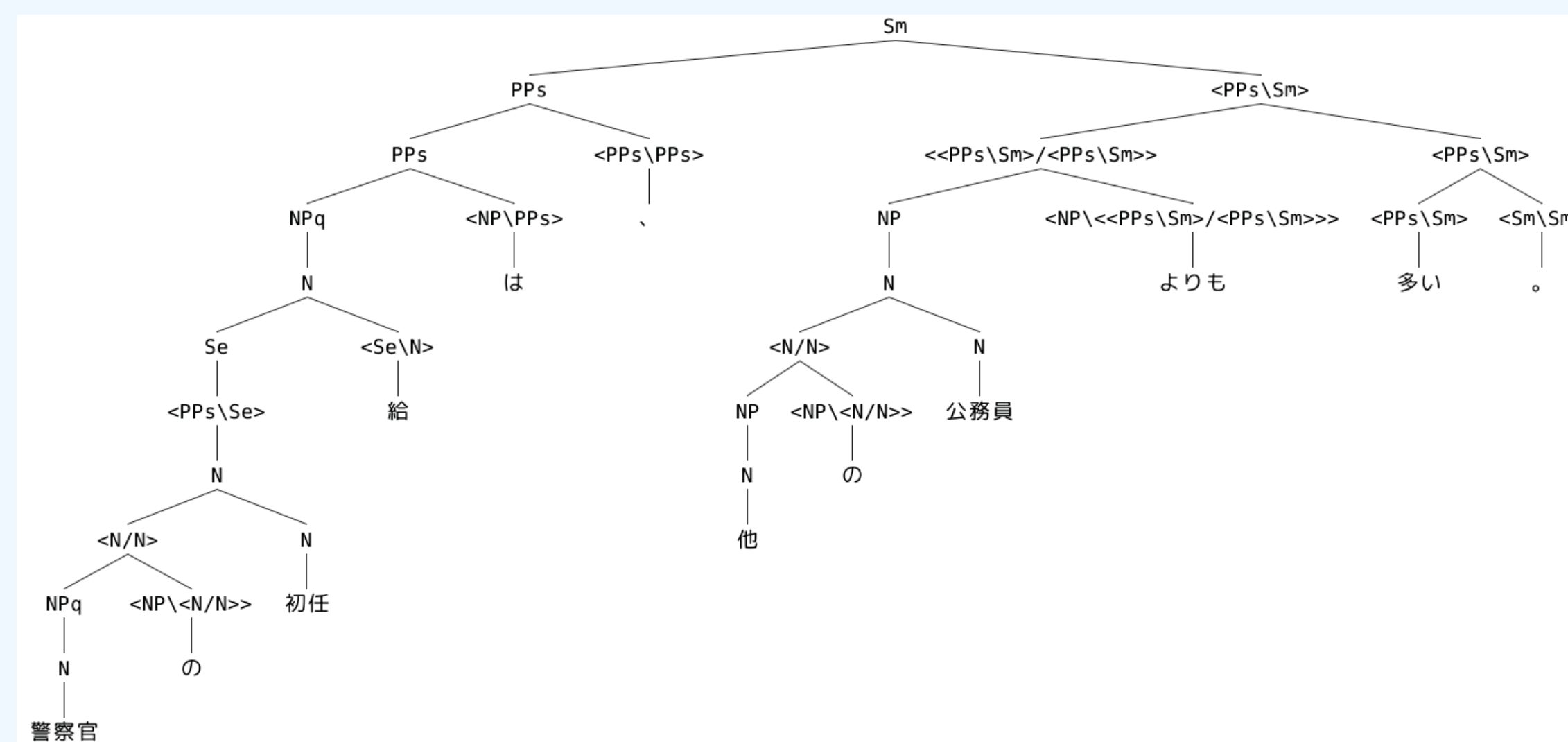
(Matsui & Kubota 2011)

## 実際にやってみる (途中経過)

比較構文のアノテーション (とりあえず人手で付与)

- (4) a. correct: 警察官<sub>cont</sub> の初任給は 他<sub>prej</sub>の公務員<sub>deg</sub> よりも 多い<sub>deg</sub>  
b. incorrect: 警察官の初任給<sub>cont</sub> は 他<sub>prej</sub>の公務員<sub>deg</sub> よりも 多い<sub>deg</sub>

degccg+ABCのパーズ結果 (自動解析)



### 意味表示

- (5) (((よりも ((の 他) 公務員))  
(λg (λx ((g 多い)  
(は (給 ((の x) 初任)))))))  
警察官)
- (6) max d[多い(初任給(警察官))(d)] > max d[多い(初任給(他(公務員)))(d)]

### 結論:

- ▶ 比較構文のアノテーション (≈ 高次の係り受け) が正しければ、正しい意味表示を導出できる

## 今後の課題

- ▶ 「高次の係り受け」を当てるタスクを機械学習で実装する
- ▶ 推論の計算のタスク (など) につないで出口の観察可能性を担保する

## まとめ

係り受け + 「統語変換」 = シンプルなコード + マクロ表記

