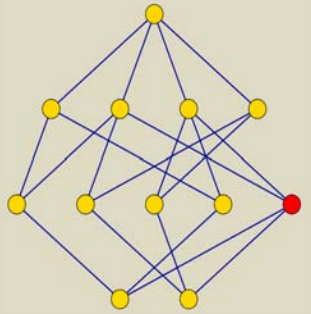
LC研@国立国語研究所
2010年11月26日



文事例の超語彙索引付けに基づく構造記述の理論と実践

パターン束モデルの基礎原理と応用 + α

吉川 正人 machayoshikawa@dream.com
慶應義塾大学大学院/日本学術振興会特別研究員

0. 始める前に

自己紹介

☼ こんにちは、吉川 正人 (よしかわ まさと) です。

- 所属など
  - 慶應義塾大学文学研究科英米文学専攻後期博士課程2年
  - 指導教授: 井上 逸兵 (相互行為の社会言語学, Gumperzian)
- 研究テーマ
  - 学部 (卒論): 翻訳論
    - 定訳表現と共起表現に基づく翻訳モデルを提案
  - 修士 (修論): 「徹底した」用法基盤主義による統語論
  - 現在 (博論): 修論の続き (詳しくはこれから)
  - 将来 (生涯研究?): 社会統語論 (Sociosyntax)

I.　はじめに

今日の目的

- パターン束モデル (PLM) の宣伝
  - 「何をやってるのかよく分からない」と言われがち
    - 基礎原理と応用可能性を提示することで概要と有効性を示す
    - PLM = 言語知識を構成するスキーマの計算理論
      - ただし現時点では生成的な側面は扱いきれていない
      - これはPLを外部操作によって「動かす」ことで達成される?
  - PLMの基礎原理とそのコーパス解析への応用の概説
    - ただし応用はちゃんと評価できるレベルまでは至っていない
    - 有効な評価手順・手法があればご教示頂きたい
      - 特に統計に関して

発表の構成

- 2節: 統語論とは何か
  - 統語論の目的と従来の理論の問題を指摘
- 3節: パターン束モデルの紹介
  - 「徹底した用法基盤モデル」
  - パターンの生成アルゴリズム
- 4節: パターン束モデルの応用
  - 実データ分析への適用にあたっての問題
- 5節: 事例研究
  - JCSS2010での発表の修正版
- 6節: パターン束モデルの将来 (時間があれば)

2. 統語論とはなにか

統語論の目的

- 統語論 as 文構造論
  - 「文」という単位の構造を指定する構造記述のモデル
    - 類似の文には同一の構造指標を与える
      = 文の(部分一致に基づく) 体系的な類型化
    - 未知の文の構造も指定可能なものが望ましい
- 統語論 as 文形成論
  - 「文」という単位を作り出す生成アルゴリズム
    - ただしこの前提には文を構成する「部品」の指定が不可欠
    - この「部品」を提供するのが構造記述
  - 「実質的」なものと「比喩的」なものがあると言える
    - 後者は文構造論と大差ない

従来の統語論

☼登場人物

- 構成単位 (= 部品)
  - ▶ 多くの場合は語 (words)
- (範疇) ラベル
  - ▶ 多くの場合品詞 (Part-of-speech)
- 構成規則
  - ▶ 書き換え規則 (rewrite rule): 生成文法の標準理論,
    併合(Merge): 極小主義生成文法,
    素性の単一化 (unification): HPSG など,
    スキーマの合成 (schema-composition): 認知文法,
    「構文」や雛型への「当てはめ」, etc.

本当に構成単位は「語」か? [1]

☼ 語の文脈依存性

- イディオム原則 (Sinclair 1991)
  - ▶ 他の条件が同じならばイディオムの解釈が常に優先される
- 多義語のパラドクス (Taylor 2003)
  - ▶ 理論的には大問題 ⇔ 人は難なく処理可能
    ➔ 結局語はその使用実態の寄せ集めではないか？
- 「生成的」な語彙意味 (e.g., Pustejovsky 1995)

(1) a. He baked potatoes.

b. He baked cake.

(2) a. I begin reading the book.

b. I begin the book. (reading? selling? writing?)

本当に構成単位は「語」か? [2]

☼語に還元できない意味

- 構文文法 (Construction Grammar) の登場
  - ▶ あらゆる単位で成立する (形式, 意味) の対としての構文

(3) a. He sneezed the napkin off the table.

b. She topamased him something.

(Goldberg 1995 より一部改変)

  - ▶ 抽象的な「文の型」= 項構造構文と意味のペアを想定
    - E.g., [Subj Verb Obj_1 Obj_2]/[X CAUSES Y TO RECEIVE Z]
- 急進構文文法 (Radical Construction Grammar)の主張
  - ▶ 構文から独立した「語」など存在しない (e.g., Croft 2005)

では、構成単位は何か?

- 「構文」と言えば済むか?
  - 「十分」だが「必要」ではない
  - [具体-抽象]の軸と[単純-複雑]の軸の混同
    - 語 = (具体, 単純); 構文 = (抽象, 複雑); 品詞 = (抽象, 単純)
  - 抽象性・複雑性を二値で捉えている
    - 連続値であってしかるべき

	具体	抽象
単純	語	品詞
複雑	??	構文

- では?
  - 求む: 段階的な抽象性・複雑性を表現可能な形式素
    - E.g., "She _ him something" のようなパターン
      - Cf. 連語 (multiword expressions)

語から「パターン」へ

- 例: *John gave her something.* (= s)
  - p_1 = (John, ___, her, something)/
    "John CAUSES her TO RECEIVE something"
  - p_2 = (___, gave, her, something)/
    "X causes her to receive something"
  - p1・p2 = s
    - ただし: この二つだけである理由は無い
- p_1, p_2のような変項と語の連続 = パターン
  - 文事例の分節化に基づく変項化によって定義される
    - 詳しくは次節

3. パターン束モデルの基礎

パターン束モデル(PLM)とは

- パターン束モデル (Pattern Lattice Model, PLM)
  - 黒田・長谷部 (2009) によって提案された
    - 徹底した用法基盤モデル(Extremely Usage-based Model) の体現
      - ヒトの言語記憶を事例記憶とその索引の体系と看做す
      - 索引の体系を表現するのがパターン束
  - パターン
    - 事例 e の任意の分節モデル *T* による分節化 *T*(e) に基づく分節の再帰的変項化の産物
  - PLM関係の研究
    - Kuroda 2009; 長谷部 2009; 吉川 2010a, 2010b
    - WS@ JCLA 11:「徹底した用法基盤モデルの展開」

徹底した用法基盤モデル

- 用法基盤モデル (Usage-based Model)の徹底版
  - 黒田 (2007) で提案された記憶ベースの言語モデル
    - Robert Port の "Rich Phonology" (e.g., Port 2007) を土台に考案
    - 言語知識を膨大な言語事例 (exemplers) の集積と看做す
      - 事例は言語形式 *f* とその使用された状況 s の対 (*f*,*s*)
    - 事例想起に索引(indices)が有効利用されていると想定
- 索引 = スキーマ
  - スキーマは記憶の実体ではなくただの案内役と考える
    - 記憶の実体はあくまで事例
  - 様々な事例基盤の記憶・言語モデルと整合する(?)

パターンの生成アルゴリズム

☼ 例: e = *John hit Mary.* / *T* = 単語分節

i. *T*(e) = [John, hit, Mary]

ii. *T*(e) の分節を一つずつ変項 (“__”)に置換

= {(__, hit, Mary), (John, __, Mary), (John, hit, __)}

iii.iiを全ての分節が変項になるまで再帰的に適用

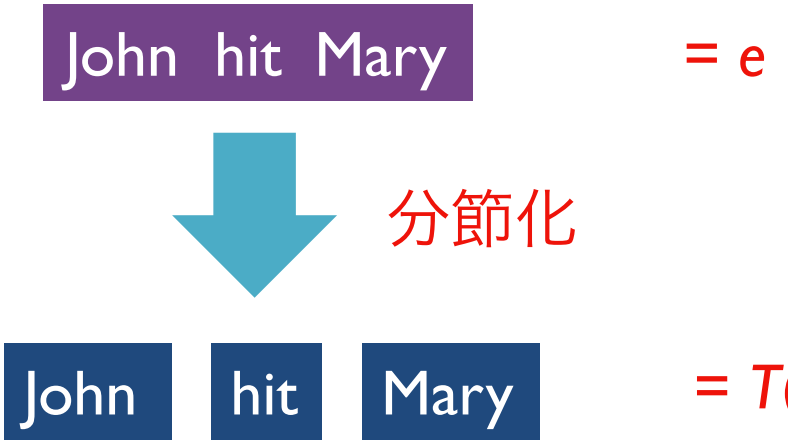
= {(__, hit, Mary), (John, __, Mary), (John, hit, __),
(__, __, Mar), (__, hit, __), (John, __, __)(__, __, __)} = *P*(e)

- *T* はPLMとは独立に指定される
  - ▶ 複数の分節モデルが並列に働いていると考えてもよい

パターンの定義の詳細 [1]

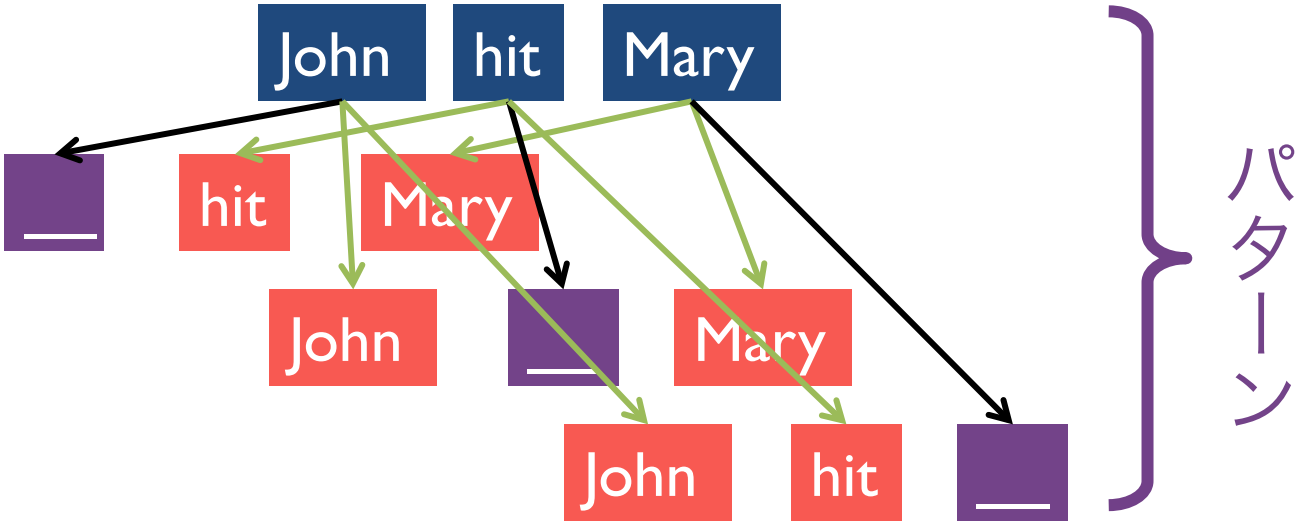
☼ 分節化

- 任意の事例 *e* を分節モデル*T* で分節化
  ➔ 分節列*T*(e)を得る
  - *e* の例: *John hit Mary.*
  - *T* の例: 単語分節
  - *T*(e)の例: [John, hit, Mary]



パターンの定義の詳細 [2]

- パターンの定義の続き
  - *T*(e) の分節を一つずつ変項 *X* によって置換
  - この産物をパターンと定義



  - この工程を全分節が変項化されるまで再帰的に適用
    - 得られた産物 = e のパターン集合 *P*(e)

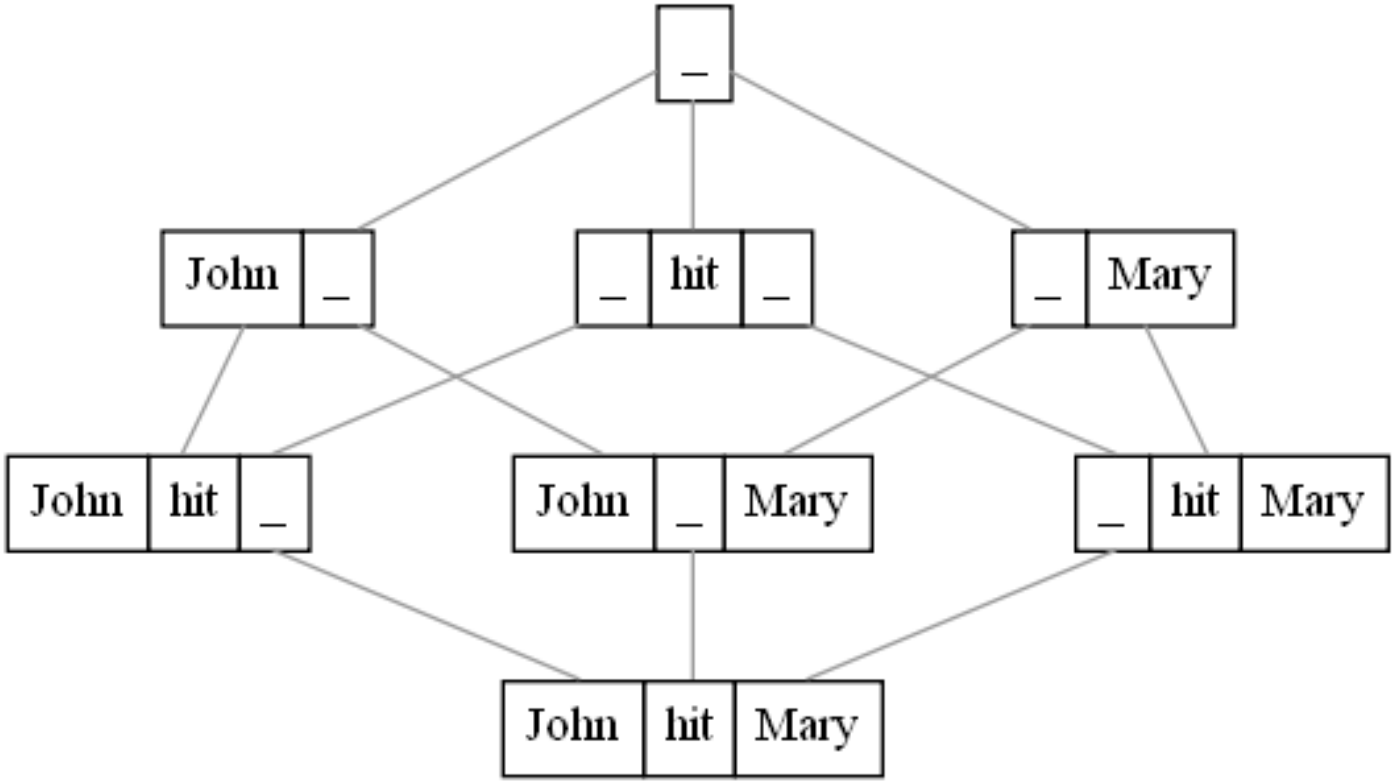
いくつかの付記

- パターン間の関係性
  - 事例 e から生成されたパターン集合 $P(e)$について
    - $p_i, p_j \in P(e)$ に [p_i matches p_j] が成り立つ場合 [p_j is-a p_i]
      - e.g., (John, __, __) matches (John, __, Mary)
        ⇒ (John, __, Mary) is-a (John, __, __)
  - is-a 関係は推移律を満たす
    - p_i is-a p_j, p_j is-a p_k → p_i is-a p_k
- 変項の縮約
  - 連続する変項を単一の変項に縮約する簡略化
    - e.g., (John, __, __, __) → (John, __)
      - 主に技術的な理由による (問題もないわけではない)

パターン束 (PL) とは

- 継承関係 (is-a) の定義されたパターンの集合 *P*
  - is-a 関係を順序とした半順序集合
    - e.g., e = *John hit Mary.* のパターン束 *L*(e)
      - *L*(e) = {(John, hit, Mary),
        (John, hit, __), (John, __, Mary), (__, hit, Mary)
        (John, __), (__, hit, __), (__ Mary)
        (__)}
    - is-a 関係
      - (John, hit, Mary) is-a (John, __)
        (__, hit, Mary) is-a (__, hit, __)
        (__, Mary) is-a (__)
      - 最大限 = (__), 最小限 = (John, hit, Mary) ➔ 束

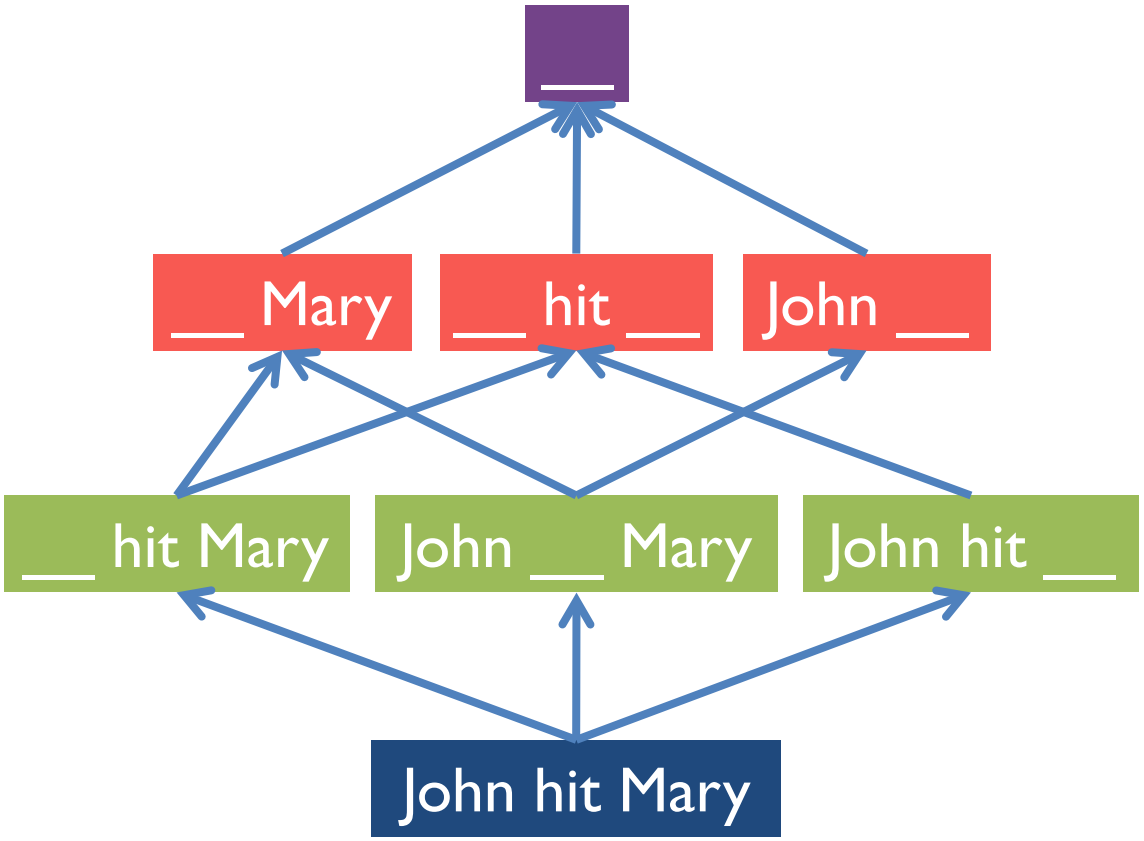
パターン束の例



Pattern Lattice Builder (http://www.kotonoba.net/rubyfca/) で作成
(被覆関係のみを明示)

パターン束のハッセ図

☼ *L*(e) (ただし e = *John hit Mary*.)



… 頂

… 語彙パターン

… 超語彙パターン

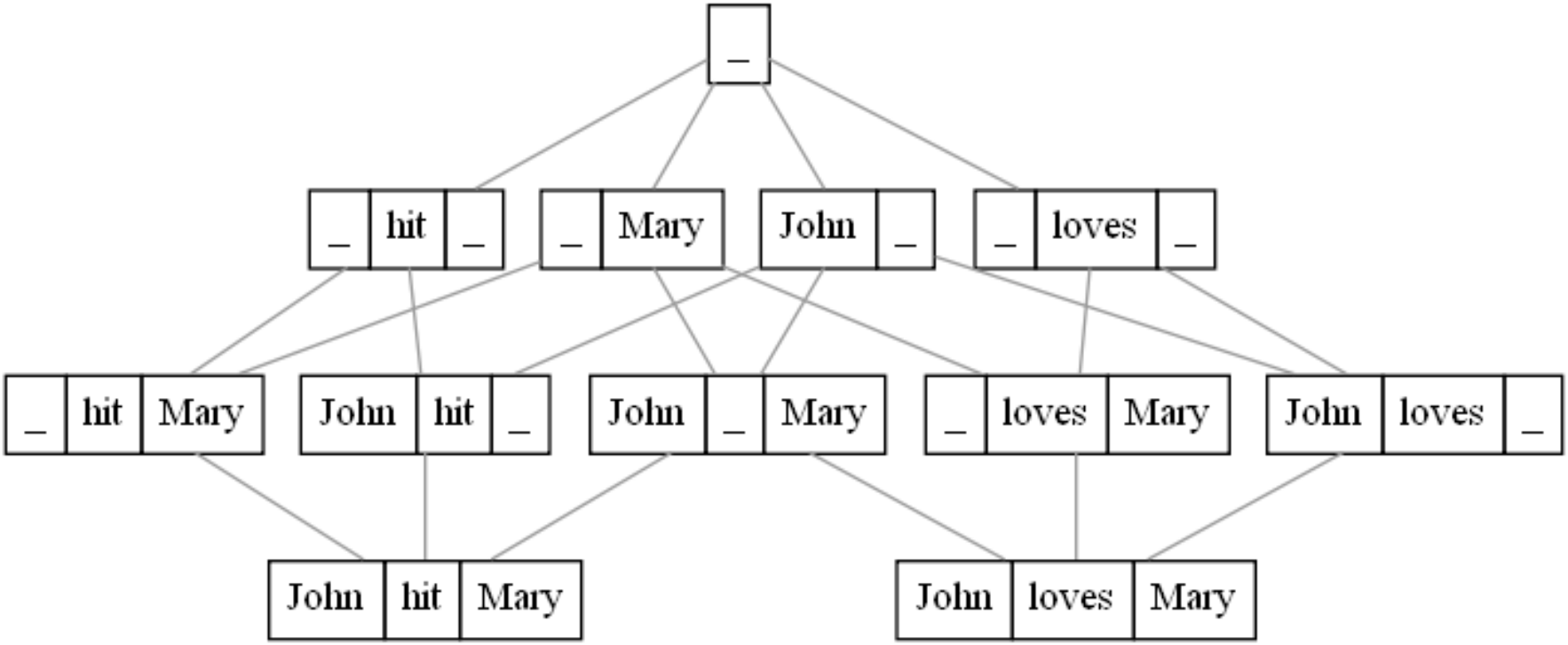
… 底 = 事例

パターン半束

- 単一事例から生成したPL = 束

- 複数事例の束を結合したPL ≠ 束
  - 複数事例の場合下限が存在しない場合がある
  - 上限は必ず「頂 (*top*)」= (⎯) になる = 存在する
- 従って
  - 複数事例からなるパターン束 = パターン半束
  - 上限のみ存在するので上半束

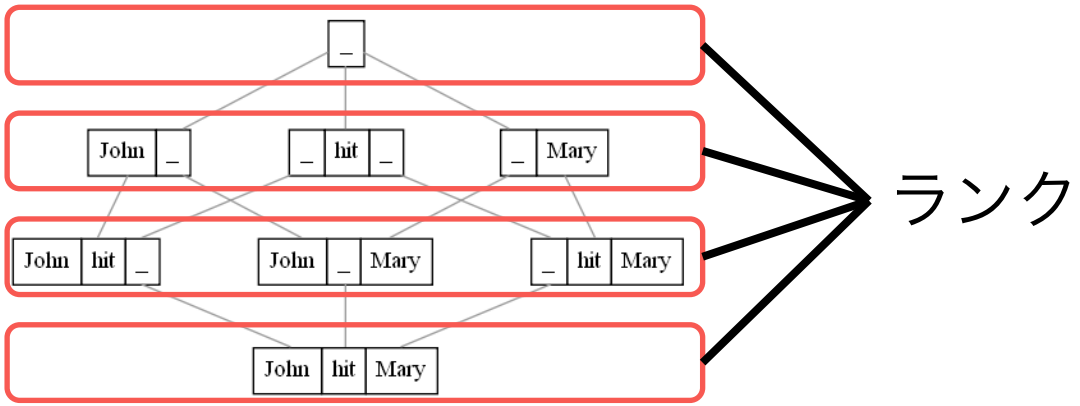
パターン半束の例


_
_ hit _
_ Mary
John _
_ loves _
_ hit Mary
John hit _
John _ Mary
_ loves Mary
John loves _
John hit Mary
John loves Mary

パターン束とランク

☼ PLの特性

- 変項 (“__”) ではない分節 (= 定項) の数が同一の場合
  ➔ 絶対にis-a関係になることはない
  - e.g., (John, __) と (__, Mary)
- 定項数が同一のパターンの集合が一つの「層」を成す
- この「層」を「ランク (rank)」という
  - パターン p のランク $r(p)$ = $|\{s \mid s \neq \text{変項}\}|$ (ただし s = 分節)



4.　パターン束モデルの応用

パターン束モデル (PLM) の効用

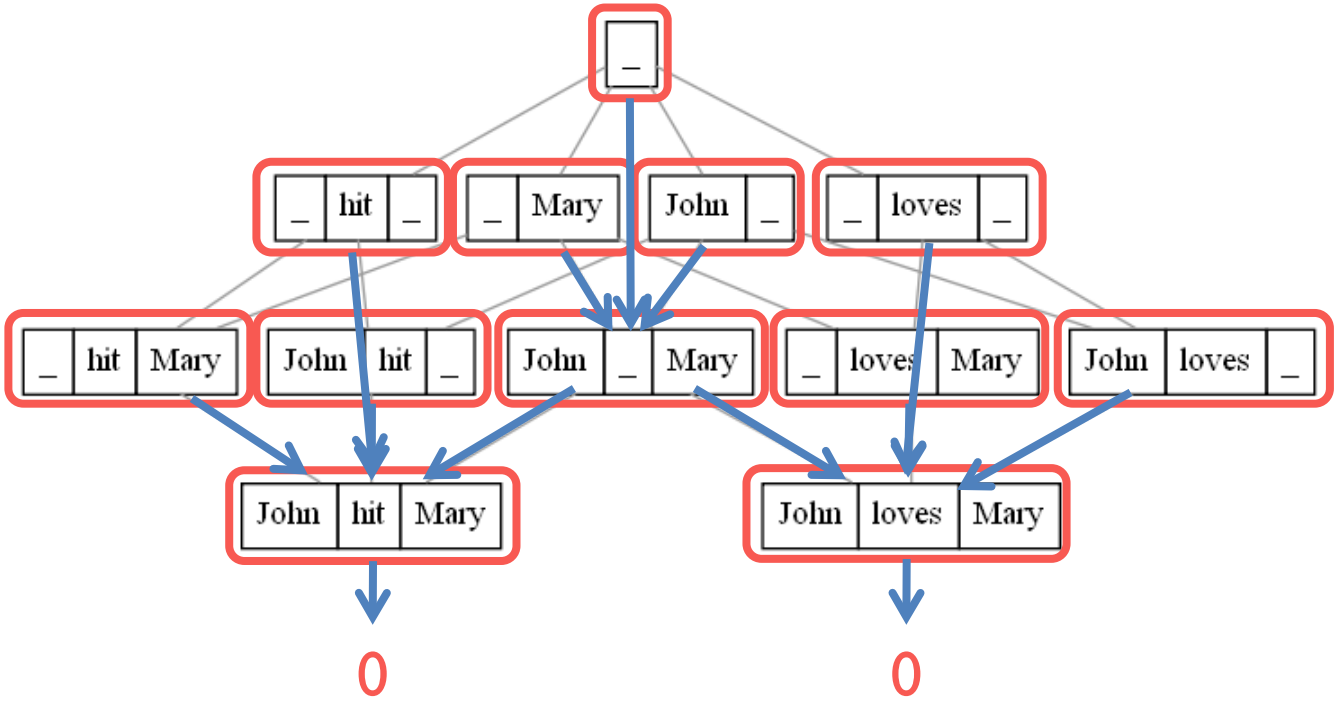
- PLMによる実データの分析 (吉川 2010a)
  - 分節化 (と前処理) だけである種の構造を記述可能
    - 文構造 (≈ 統語構造) の記述には通常理論が必要
      = 記述の妥当性が理論に依存する
    - 「語」を超えた単位の網羅的記述装置は存在しない
      - Cf. N-gram
  - PLMは統語構造・構文の「発見」に有益
    - 注意: PLそれ自体が統語構造ではない
- しかし
  - 現状 有益な記述結果は未蓄積
    - 主に技術的制約による

ノードの最適化

- パターン束モデルのパターン生成アルゴリズム
  - 事例 e の任意の分節化に基づく分節の再帰的変項化
    ➔ 網羅的に可能なパターンの全集合を生成
  - 効果:　入れ知恵無しにパターンの候補を生成可能
    副作用: 不要なパターンの生成 (過剰生成)
  - 効果は無くさず不要なパターンの生成を抑制したい!
- そこで: ノードの最適化
  - アルゴリズム: 子ノードが1つしかないノードを削除
    - Delete x from L iff $|\{ y \in L \mid y \text{ is-a } x \}| = 1$
    - 実際はこれをランクごとに「下から」順次適用

最適化アルゴリズムの大枠

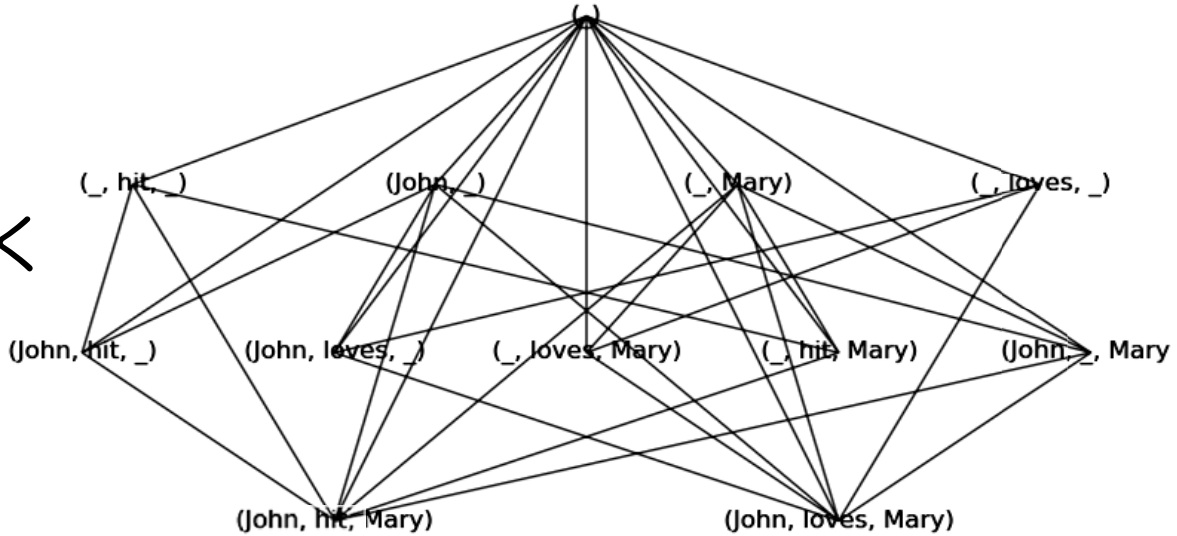
☼ 例: E = {*John hit Mary, John loves Mary*}

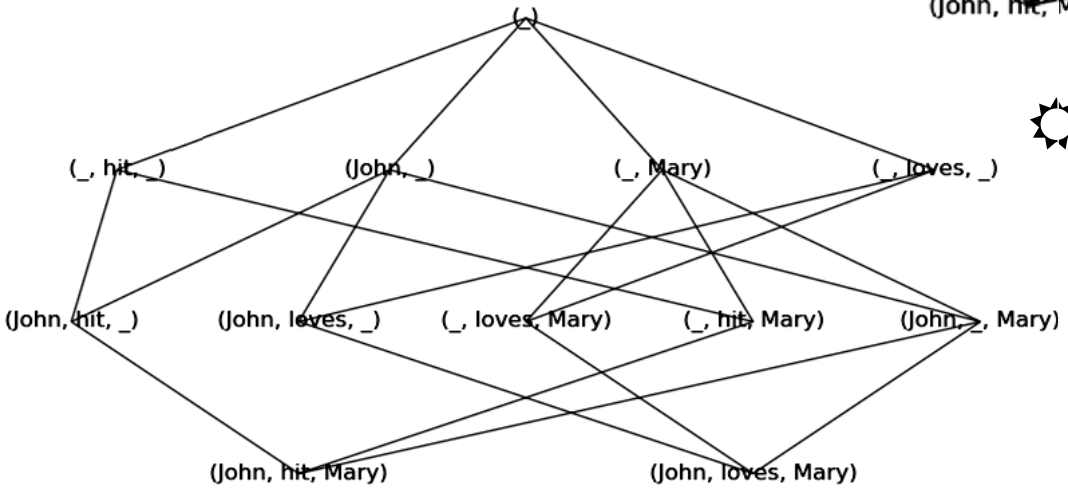


- 具体的なノードを優先し冗長なノード・エッジを削除
  - Cf. イディオム原則 (Sinclair 1991), Langackerの議論 (1987など)

パターン束の例[最適化前]

- 冗長なエッジを残したグラフ →
  - 自己ループは除く

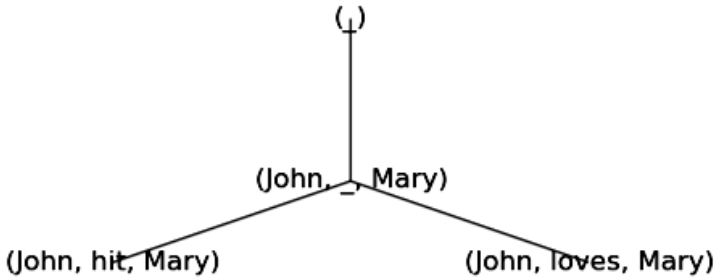




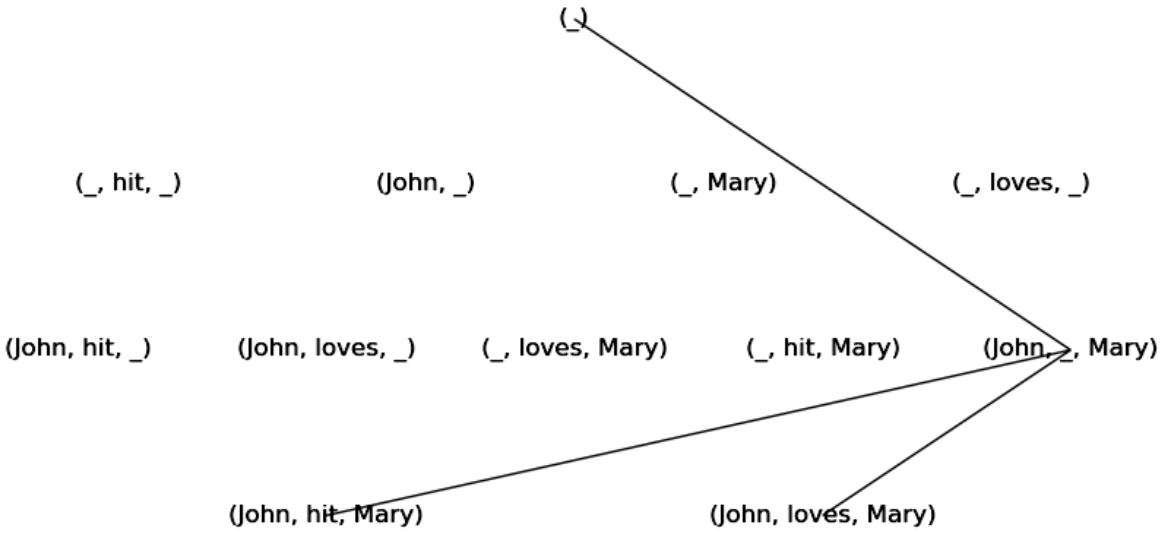
- ←おなじみのグラフ
  - 被覆関係のみを明示

パターン束の例[最適化後]

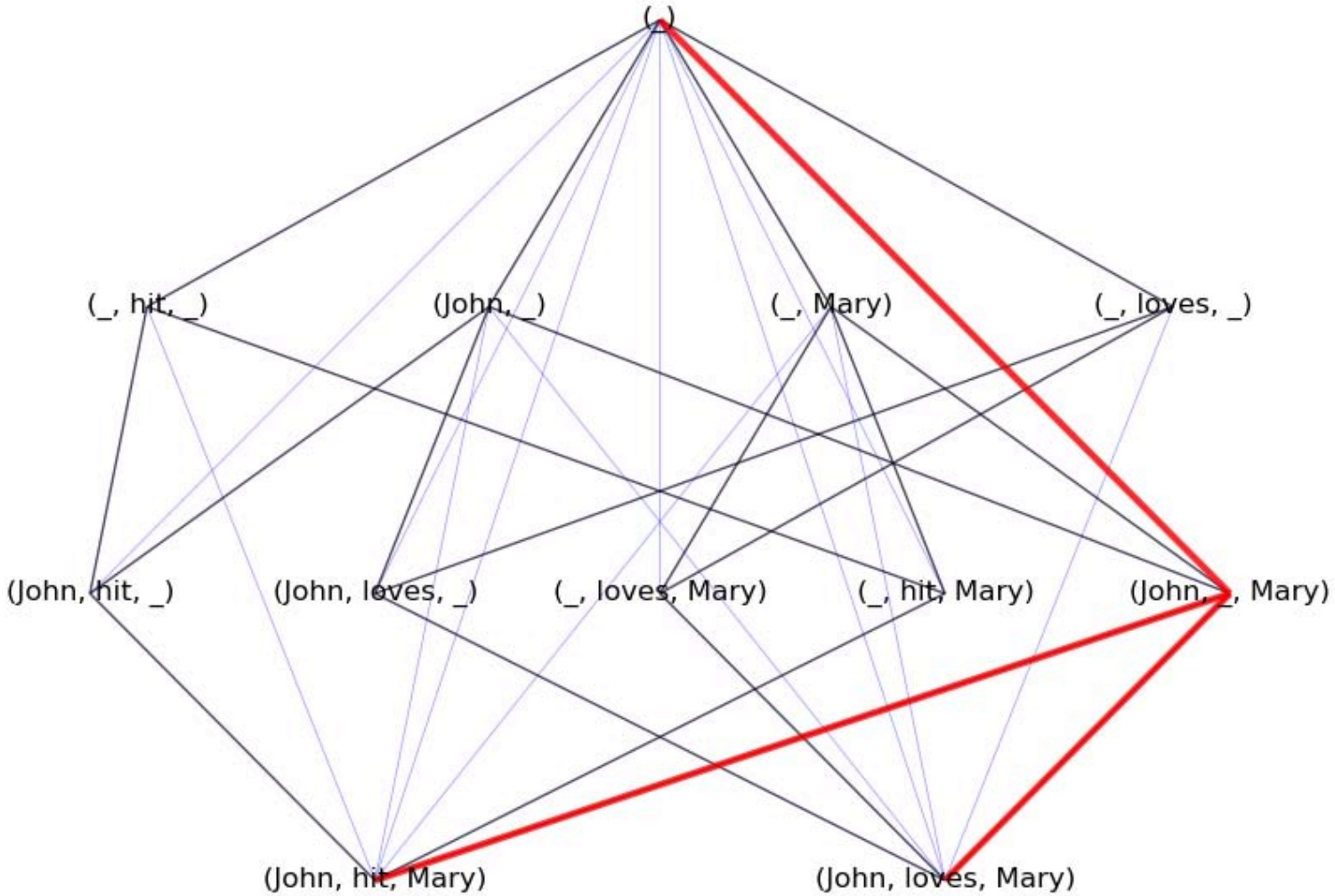
- ノードを刈込(pruning)



- 冗長なノードを削除
  - ただし頂 (top) は保存

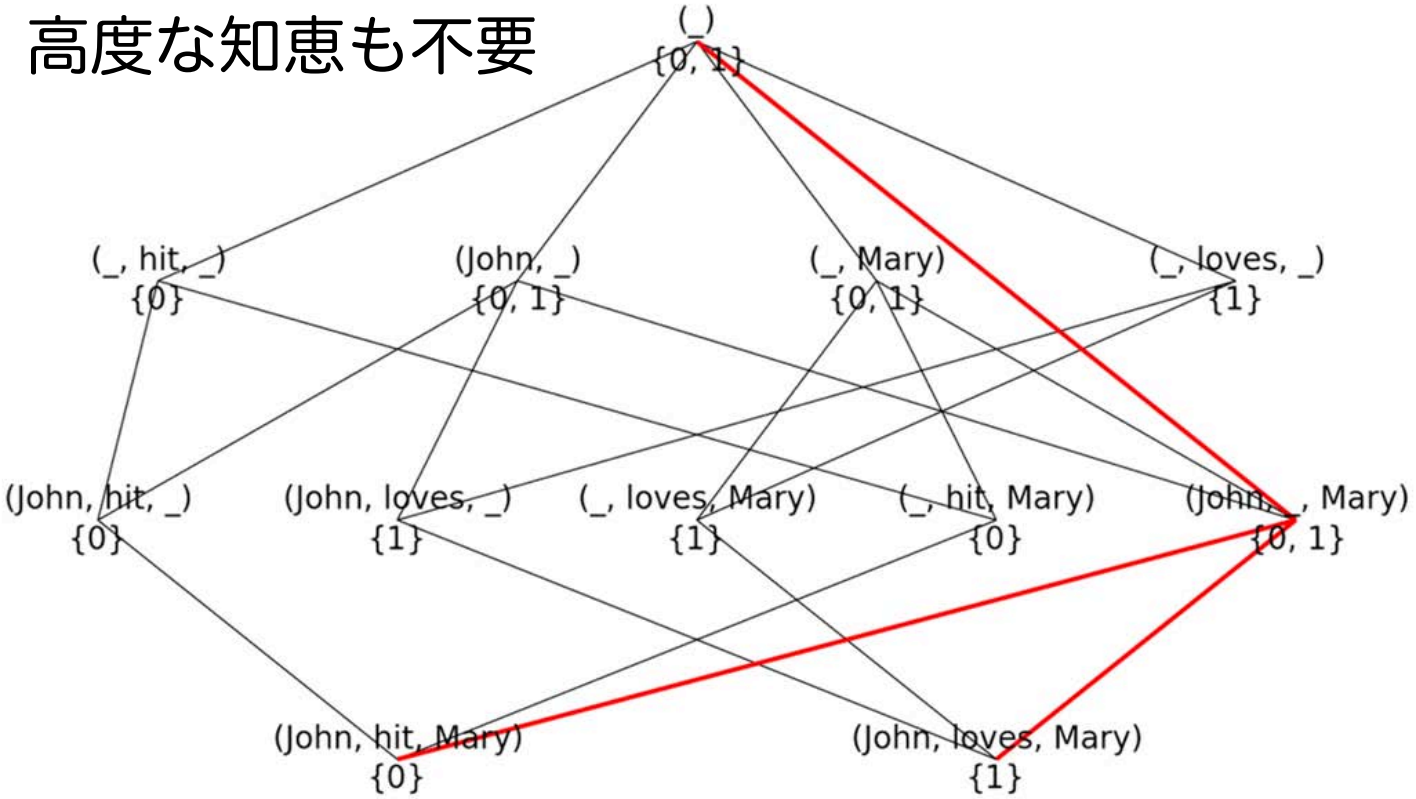


3種のグラフ(重ねて表示)



なぜこのような最適化?

- この最適化のメリット
  - 事例に対する説明力が低下しない
    = 事例に対する網羅性 を保持
  - 高度な知恵も不要



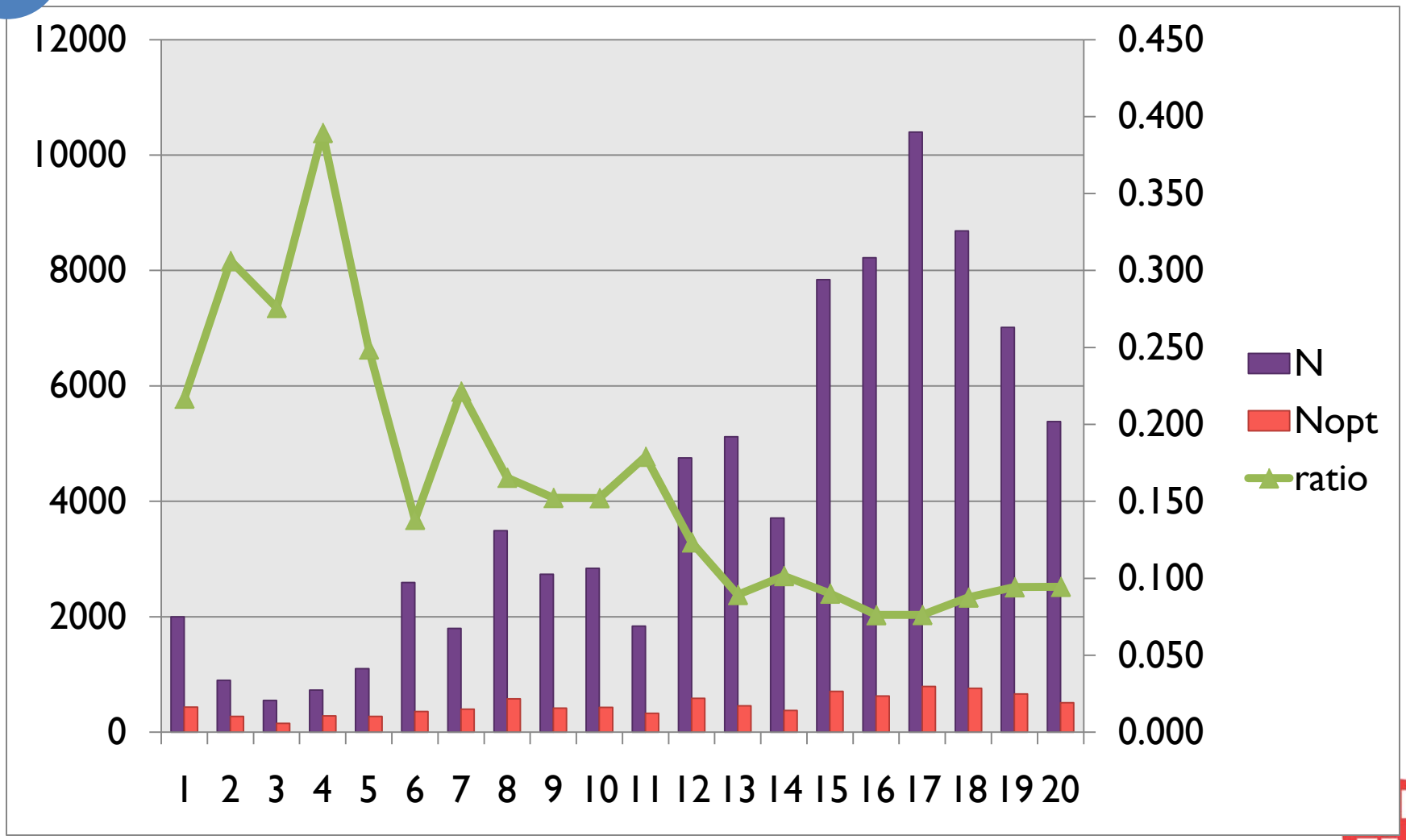
最適化の効用

- Brownコーパス in CHILDES (MacWhinney 2000)
  - Adam, Eve, Sarah の三幼児の対話データを収録
    - Eveのデータ (20ファイル) からパターン束を生成
    - 全パターン数, 最適化後のパターン数:

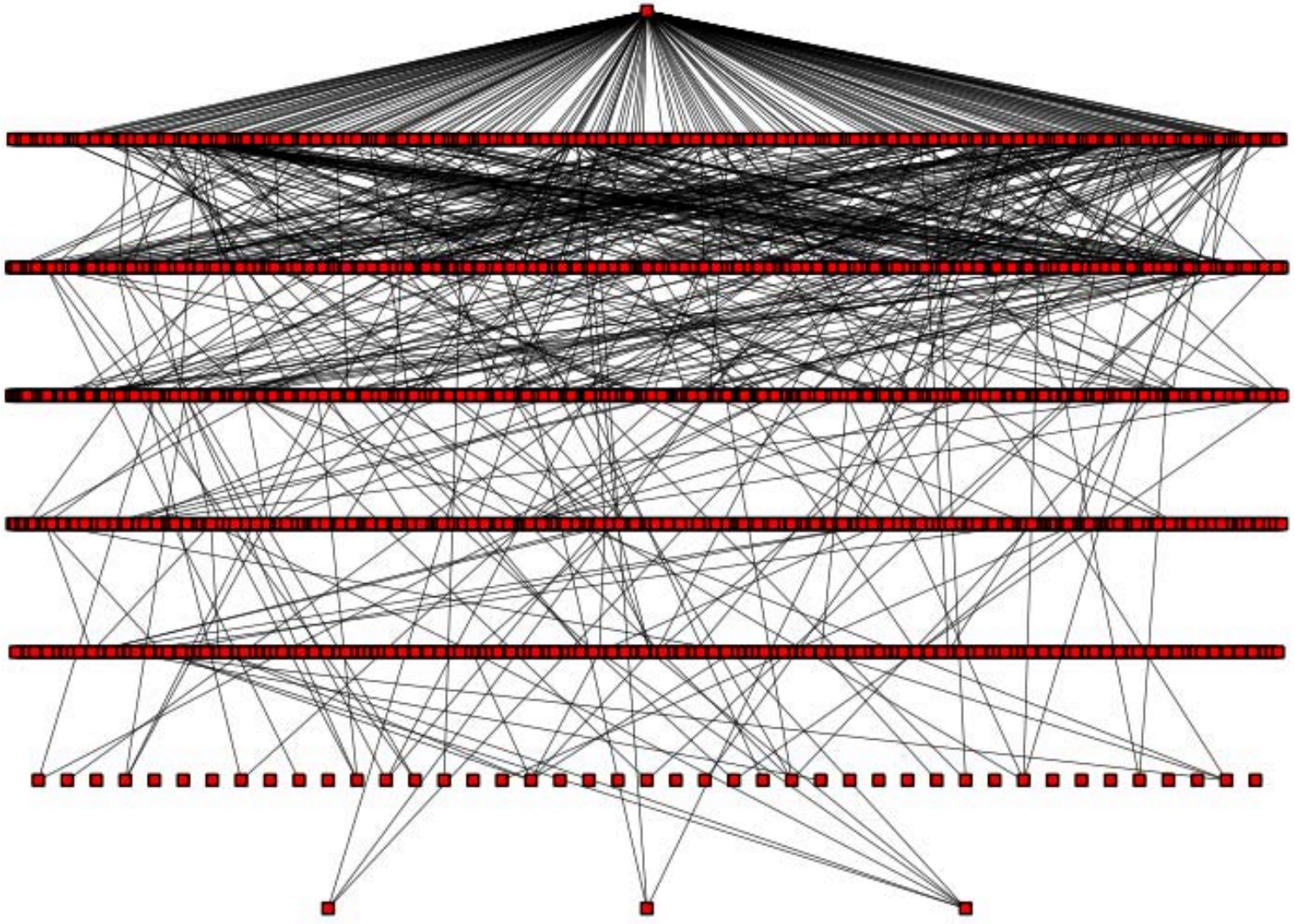
id	N	N_{opt}	Ratio
1	2001	434	0.217
2	901	276	0.306
3	551	152	0.276
4	729	284	0.390
5	1101	274	0.249
6	2592	358	0.138
7	1797	398	0.221
8	3493	578	0.165
9	2737	417	0.152
10	2838	432	0.152

id	N	N_{opt}	Ratio
11	1837	329	0.179
12	4754	587	0.123
13	5118	457	0.089
14	3711	377	0.102
15	7843	707	0.090
16	8219	626	0.076
17	10398	793	0.076
18	8687	761	0.088
19	7012	661	0.094
20	5383	509	0.095

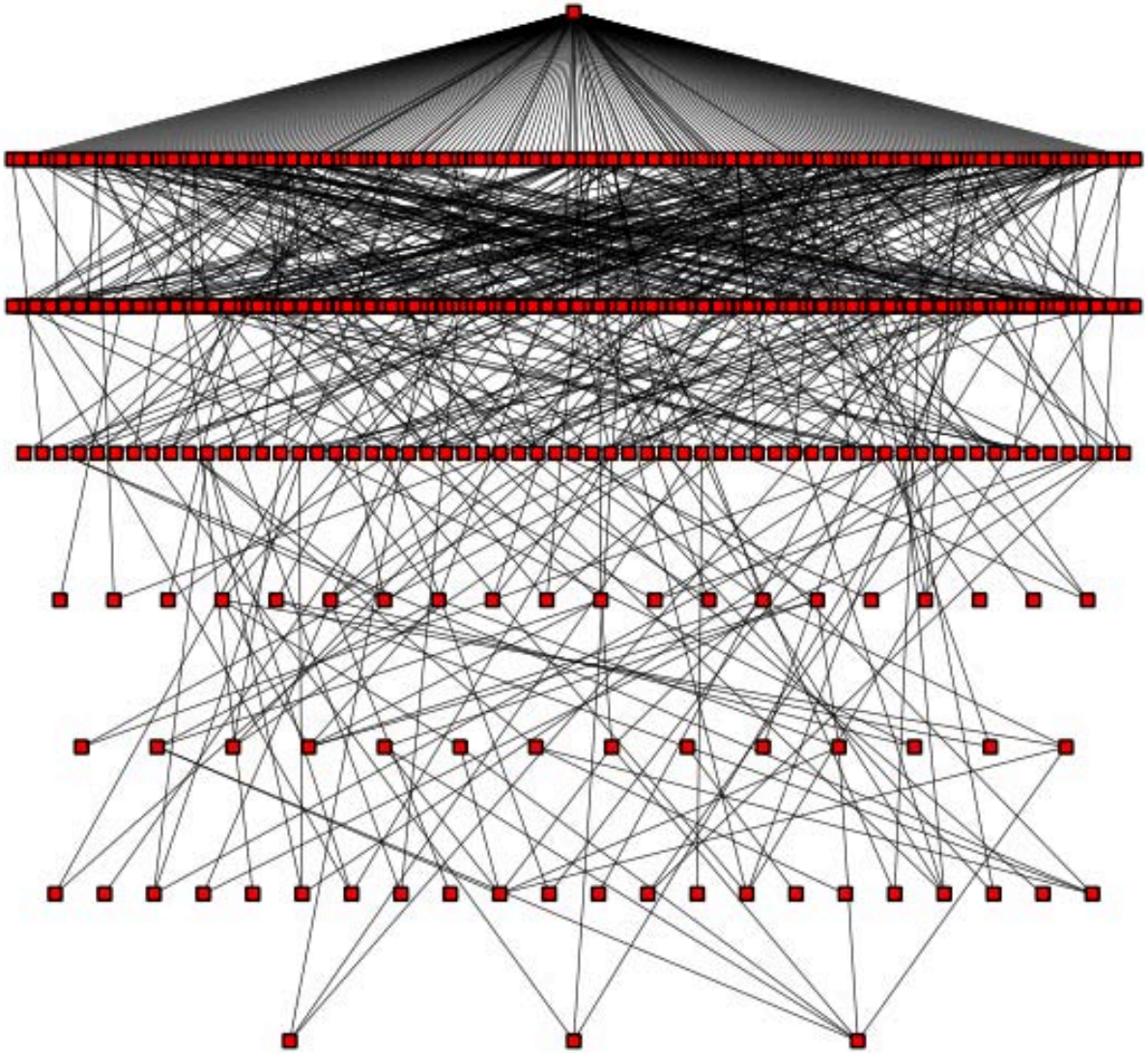
削減率の推移


12000
10000
8000
6000
4000
2000
0
0.450
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.050
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
N
Nopt
ratio

PL from Eve[file1]



PL from Eve[file1] (pruned)



ノードのエントロピー $H(v)$

- パターンの数だけ計ってもあまり面白くない
  - 個々のパターンの持つ「性質」にまで踏み込みたい
  - これまで見てきたパターンの性質 = 生産性
    - そのパターンがどれだけ多様な事例と結びついているか
- 生産性の算定法
  - 変項におけるシャノンのエントロピー(H)を使用
    - 詳しくは吉川 (2010b)
    - しかしこれは問題アリ (日高昇平氏 (北陸先端大学) の指摘)
  - そこで:
    - 「ノードのエントロピー」$H(v)$ というものを考えた

シャノンのエントロピー (*H*)

- シャノンのエントロピー (平均情報量) とは
  - 確率空間上の起こり得る事象の確率に基づく情報量
  - 確率空間 *X* のエントロピー*H*(*X*):

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

    - ただし$p(x)$ はX 内の事象 x の生起確率
  - 例: コインを投げて表が出るか裏が出るか
    - 起こり得る事象: {表が出る, 裏が出る} ➔ 確率分布 = (0.5, 0.5)
    - エントロピー
      - $- (0.5 * \log_2 0.5 + 0.5 * \log_2 0.5) = - (-1*0.5)*2 = -(-0.5)*2 = 1$

$H(v)$ の産出法

- 各ノード (= パターン)における確率空間
  - どれくらいの確率でどの子ノードと接続されるか
    = 上位パターンが下位パターンで実現される確率
    = 継承率 $p_s(v, s_i)$
    - ただし $1 < i \leq n$ (nはノード v の子ノードの数)
  - $p_s(v, s_i)$ の計算法

$$p_s(v, s_i) = \frac{f(s_i)}{f(v)}$$

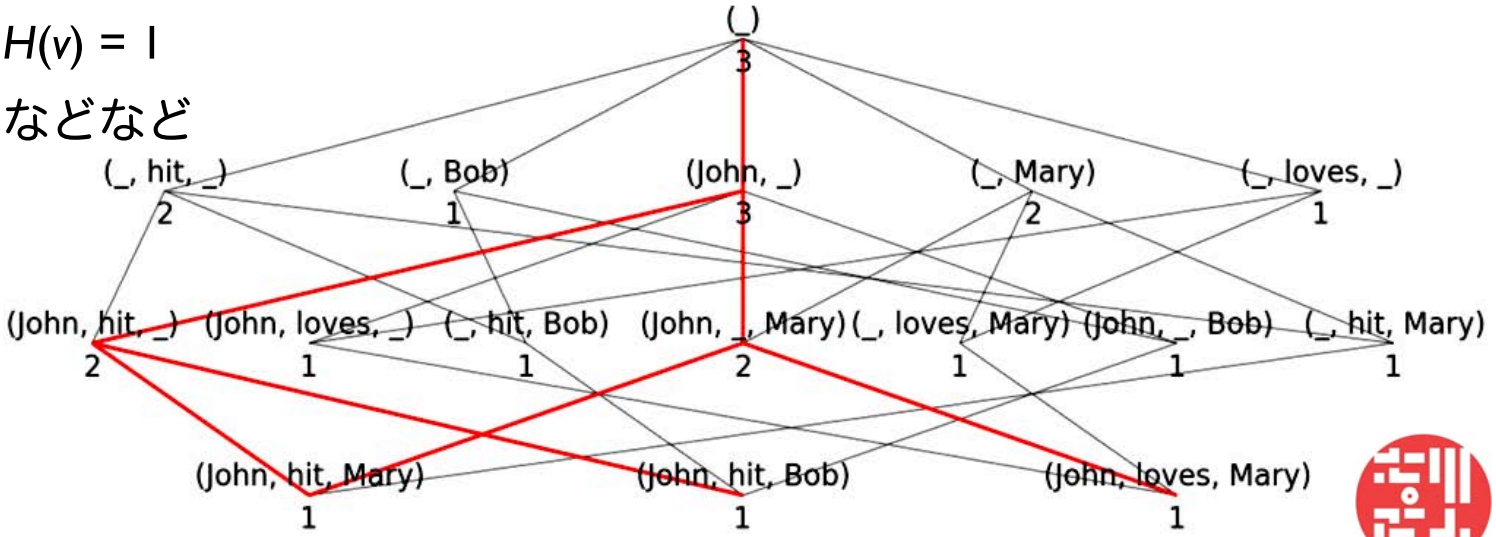
(ただし$f(v)$ はvの頻度)

- 従って

$$H(v) = -\sum_{i=1}^{n} p_s(v, s_i) \log_2 p_s(v, s_i)$$

例

- E = {*John hit Mary, John hit Bob, John loves Mary*}
  - 最適なノードのみを対象にする
    - (John, ___): 子ノード = {(John, hit, ___), (John, ___, Mary)}
      - $H(v)$ = -(log2 (2/3)* (2/3) + log2 (2/3) * (2/3)) = -(0.58…* 0.66…)*2
        = -(-0.38998…)*2 = 0.77995… ≈ 0.78
    - (John, hit, ___): 子ノード = {(John, hit, Mar), (John, hit, Bob)}
      - $H(v)$ = I
      - などなど



注意など

- 要するに
  - v = (John, __) の場合の $p_s(v, s_i)$
    - (John, __)が使用された際
      - 同時に子ノード((John, hit, __), (John, __, Mary)) が実現される確率
- 確率は普通足すと1になるが…
  - v について $p_s(v, s_i)$ を合計すると1を超える場合がある
    - 先の “(John, __)” はその好例
      - 子ノード (John, hit, __), (John, __, Mary) が(John, hit, Mary)で交差
  - 確率が1を超える = エントロピーが目減りする
    - これによって間接的に事象の非排他性を取り込む
      - Cf. コイン投げなら裏と表が同時に出ることはない

H(*v*)で何が図れるか

- 例えば:
  - 語彙パターン の*H*(*v*) < 超語彙パターン の*H*(*v*)
    ➔ 構文効果!? イディオム性??
  - 軸スキーマや動詞島(Tomasello 2003)の発見も可能!?
    - この検証に関しても今後の課題
- 現状
  - 自宅設置の高性能PCでBrown Corpusの解析を実施中
    - 結果は好ご期待
  - CHILDESの方のBrownコーパスはいろいろ試行錯誤中

ちょっとした例

- Eve[file 19] のPL
  - 子ノードの方がエントロピーの高かったペア上位10組

super pattern	sub pattern	H_{super}	H_{sub}	ratio
(_, want, _)	(I, want, _)	1.66861809	3.260491008	1.954007
(you, not, _)	(you, not, a, _)	0.811278124	1.584962501	1.953661
(_, little, one)	(_, a, little, one)	0.811278124	1.584962501	1.953661
(_, bread, _)	(_, bread, _, butter, _)	0.779950001	1.5	1.9232
(_, bread, _)	(_, bread, and, _)	0.779950001	1.5	1.9232
(_, not, a, _)	(you, not, a, _)	0.884358713	1.584962501	1.792217
(_, not, a, _)	(_, not, a, man)	0.884358713	1.584962501	1.792217
(I, _, some, _)	(I, want, some, _)	1.251629167	2	1.597917
(put, _, pencil, _)	(put, you, pencil, _)	0.779950001	1	1.282133
(put, _, pencil, _)	(put, _, pencil, in, there)	0.779950001	1	1.282133

簡易PLパーサーの実装

- PythonでPLベースの簡易なパーサーを実装
  - パーサーのインスタンスにコーパスデータを読み込む
  - 任意の文 *s* に対し
    - 1) $L(s) = \{p_1, p_2, \ldots p_n\}$を生成
      2) 全てのパターンに対し変項“__”を“.+?”に書き換え
      3) 正規表現検索で対応する事例を収集
      4) マッチしたパターンのみからなるパターン束$L^*(s)$を最適化
      5) 残ったパターンとそのis-a関係をグラフで表示
- 課題
  - 遅い
  - 現時点ではあまり大規模なコーパスを読み込めない
  - ノードのエントロピーは計算できていない

5. 事例研究

言語習得の「学習」説

- 「用法基盤モデル(Usage-based Model)」では
  - 言語の習得:
    - 具体的な一語文 (Holophrases)
    - スロットを含んだ二語文 (e.g., 軸スキーマ, 島構文)
    - 抽象的な構文 (e.g., 二重目的語構文, 中間構文)

    と進む段階的なプロセスを想定 (Tomasello 2003)
- このプロセスの検証
  - Borensztajnら (2009)
    - が: これ↑は実は所謂認知言語学的研究ではない
      - 二股枝分かれしか許さない木構造からなる統語表示を想定

学習説の問題

☼UBMなどの「学習」モデルを取るなら

- 生得的な構造・範疇ラベルの想定ができない
- 刻一刻と変化する知識を記述するツールが必要
- が:今のところそのようなツールはない
  - ▶ そもそも明示的な記述を支える理論がない(次スライド)

発達研究とPLM

- 表示の問題は表面的
  - 本当の問題 = 認知言語学における計算理論の不在
    - スキーマは計算的に扱える対象のはず
      - コネクショニズム研究など見れば明らか(e.g., Elman 1991)
  - 「木構造解析 ≠ 唯一の方法」を示すべき
- そこで
  - 「スキーマの計算理論」の候補 = パターン束モデル
  - Borensztajnら (2009)の検証をPLMベースに再現

  することを目指す

調査概要

- データ
  - Brownコーパス (Brown 1973) in CHILDESデータ (MacWhinney 2000)
- 方法
  - PLMに基づきパターンを作成
  - パターンの生産性をシャノンのエントロピで算定
    - 年齢経過に伴うエントロピーの増加を検証
- 目的
  - 用法基盤の習得プロセスの実証的な検証
  - PLMの有効性のアピール

仮説

- PLMの想定から以下の仮説が導かれる:
  - 幼児の統語発達 = PLMの定義する「パターン」の発達
  - 補足: パターンの発達 =
    - 利用可能なパターンの総数増加
    - 個々のパターンの生産性上昇
- 仮説の予測
  - PLMのパターン生成アルゴリズムを用いて幼児の発話からパターンを生成した場合，その総数及び生産性は年齢を経る毎に上昇していくはず
  - 以下でこれを検証

データ

- 使用したデータはBorensztajnら (2009)と同様
  - CHILDES(MacWhinney 2000)内のBrownコーパス (Brown 1973)
  - 3幼児 (Adam, Eve, Sarah) のデータをそれぞれ3分割
    - データ量が均一になるように分割
    - 発達の段階や幼児間の年齢の対応などは考慮せず
- 補足: CHILDESとは
  - Child Language Data Exchange System の略
  - 幼児と養育者等との自然な会話のデータベース
    - 様々な言語による様々なコーパスの集積

データの詳細

	Files	Age	#sent.	MLU	vocab.	t/t
Adam						
P1	1-16	2:3-2:11	11,184	1.83-2.90	1,407	.056
P2	17-32	2:11-3:6	11,578	2.44-4.06	2,010	.053
P3	33-48	3:6-4:5	9,071	3.63-4.97	2,006	.055
Eve						
P1	1-7	1:6-1:9	3,485	1.53-2.28	669	.102
P2	8-14	1:9-2:0	3,395	2.51-3.22	785	.083
P3	15-20	2:1-2:3	3,535	2.60-3.41	958	.087
Sarah						
P1	1-45	2:3-3:2	11,693	1.48-2.70	1,389	.063
P2	46-90	3:2-4:1	8,384	2.23-3.70	1,706	.075
P3	91-135	4:1-5:0	8,525	2.98-4.86	1,944	.071

(Borenzstajinら (2009:Table1) を一部改編)

手順 [1]

- 前処理
  - 幼児の発話のみを抽出
  - ポーズ, 言いさし・重複 を含む発話を除外
    - Borensztajnら (2009) に倣った
- 3幼児 × 3データそれぞれからPLを作成
  - ただし: 長い発話 (8語以上)には以下の処理を実施
    - 発話 $u = [w_1, w_2, \ldots, w_n]$ (n > 7)に対し以下の$l_{init}, l_{end}, l_{mid}$を作成
      - $l_{init} = [w_1, w_2, w_3, w_4, w_5, w_6, \_\_]$, $l_{end} = [\_\_, w_{n-5}, w_{n-4}, w_{n-3}, w_{n-2}, w_{n-1}, w_n]$
      - $u^* = [w_2, w_3, \ldots, w_{n-1}]$ から 5-gram $= G = \{g_1, g_2, \ldots, g_m\}$ を作成
      - $l_{mid} := \{ \_\_ + g_i + \_\_ \mid 1 \leq i \leq m\} = \{[\_\_, w_2, w_3, w_4, w_5, w_6, \_\_], \ldots\}$
    - $P(u) = P(l_{init}) \cup P(l_{mid}) \cup P(l_{end})$ とする (吉川 2010a: 211)

手順 [2]

- パターン (ノード) の最適化
  - 先述のアルゴリズムでパターンを最適化 = L_{opt}
    - ノードのエントロピーを算出
  - 頻度 1 のパターンを削除 = L^*_{opt}
- エントロピーの平均値H_{ave}を算出

$$H_{ave} = \frac{\sum_{p \in L} H(p)}{\left| L^*_{opt} \right|}$$

(ただし|X| = X のパターンの異なり数)

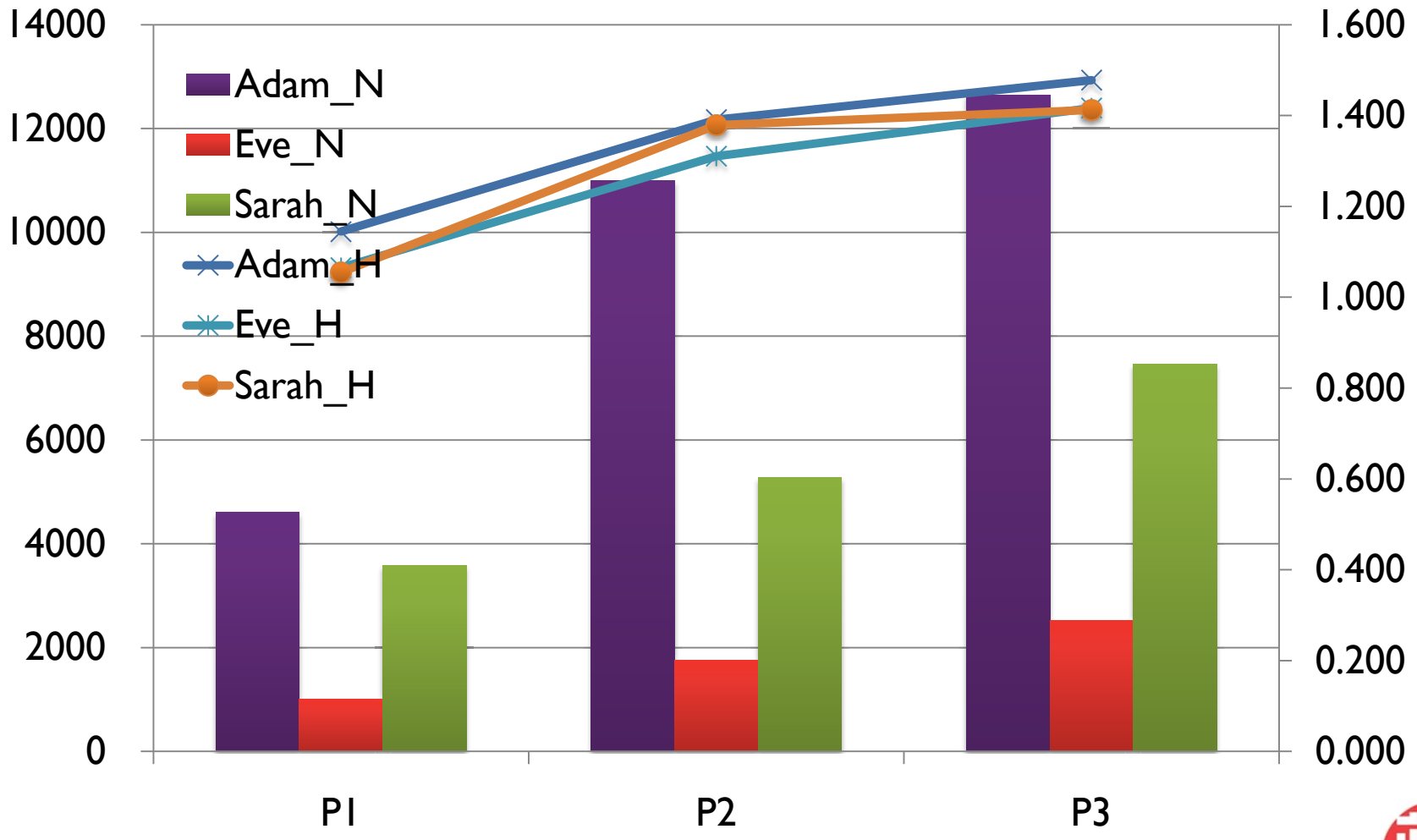
  - パターンの頻度 = 延べ数は無視
    - 日高昇平氏 (北陸先端大学) につっこまれたがよしとする

結果 [1]

☼どの幼児もパターン数・H_{ave}共に増加

	$\|L^*_{opt}\|$	H_{ave}
Adam		
P1	4604	1.14447800902
P2	11002	1.39089797092
P3	12650	1.47766220949
Eve		
P1	1000	1.06479053276
P2	1764	1.31060837109
P3	2515	1.41659346137
Sarah		
P1	3588	1.05450427761
P2	5277	1.37869506232
P3	7457	1.41225519364

$|L^*_{opt}|$ (= N) と H_{ave} (= H)



結果 [2]

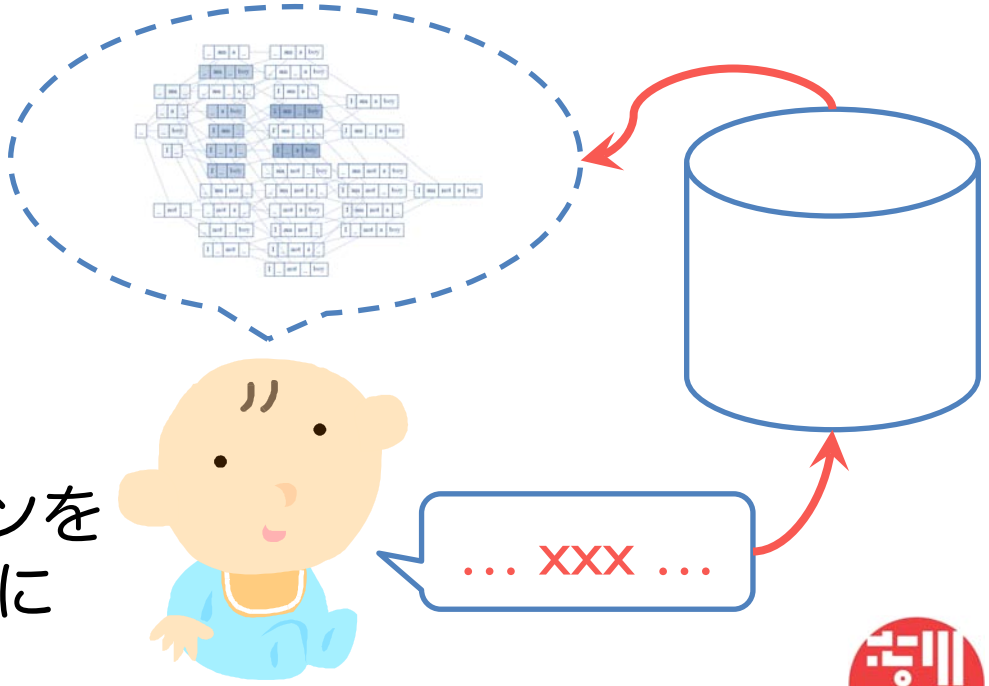
- H_{ave}の増加は有意か?
  - Wilcoxonの順位和検定で差を検定
    - ただし自由度がとんでもないので手法としてどうかとも思う
      - 何か良い方法がないかご教示頂きたい
    - Python の “Rpy2” モジュール経由でR の関数を使用
  - 結果:
    - 軒並み低い p 値
    - Sarah の P2-P3 間以外 0.1%水準以下で有意
      - Sarah のP2-P3感は $p \approx 0.06$ なので5%水準で有意傾向くらい
  - 詳しくは次スライド

Wilcoxon rank-sum test

	W	*p*
Adam		
P1-P2	21578871.5	4.31389649895851 28e-50
P2-P3	65664007.5	1.5063902128014479e-14
P1-P3	23320754.0	1.3440122166095003e-93
Eve		
P1-P2	767478.0	7.153084061382414 6e-09
P2-P3	2051425.5	1.742628223819061 8e-05
P1-P3	1007219.5	3.880645745187603 8e-21
Sarah		
P1-P2	7570812.5	3.4466859453722055e-60
P2-P3	19298939.5	0.05853546503556 2909
P1-P3	10419603.5	1.5870343107419802e-82

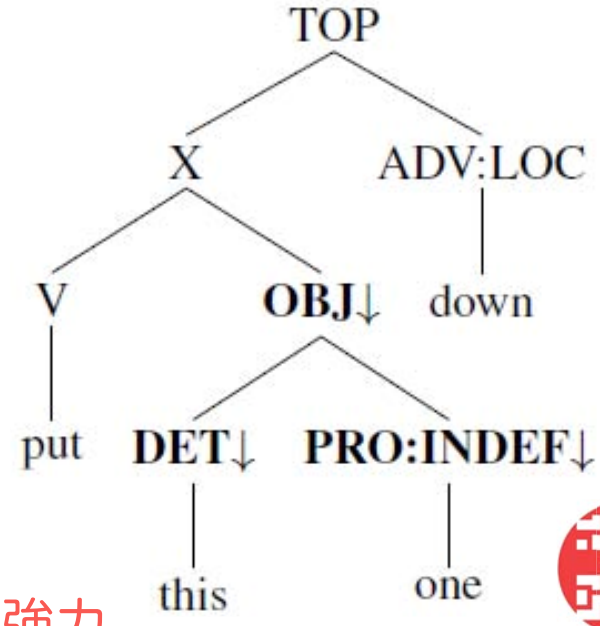
考察 [I]

- 今回のパターン生成方法より
  - 得られたパターン = 幼児の産出データからの逆算
    - 幼児の入力データから得た模擬学習結果ではない
- 従って
  - 今回の結果
    = 年齢を経る毎に生産性の高いパターンを利用した発話を行うようになる
    ≠生産性の高いパターンを知識として持つようになる



考察 [2]: 先行研究との比較

- Borensztajnら (2009)の手法
  - 「(確率的) 木代入文法 ((Probabilistic) Tree-Substitution Grammar)」を背景に持つDOPによる解析
    = 二股枝分かれしか許さない木構造による解析
  - 明らかに構文の「深さ」を過大評価している例アリ
    - Borensztajn et al. (2009: 183)
      - 三又でもよければ ノード"X"は不要
        ➔ 階層が一つ減る
    - この点で不備あり!?
  - さらに
    - 年齢経過に伴う抽象度増加の検定
      はなし: 単なる数量調査のみ
    - 本研究の方が統計的妥当性のある分強力



6. パターン束モデルの将来

抑制系の記憶システム

- いろいろ考えてみると
  - 用法基盤モデルはいろいろと成果を上げている
  - ただどうしても解決が難しい問題:
    - ある表現 X が「なぜ言えないのか」の説明が困難
      - Stefanowitsch (2008) 等統計に訴えた分析もあるが妥当性は未知
      - 大規模な統計処理を行っていると考えるには処理時間が短すぎる
    - 「否定証拠不在」の問題はやはり根深い
- なんとかならないか?
  - おそらく鍵は「抑制」
  - 不要な記憶の想起を抑制 → 結果的に否定証拠を得る

抑制系の記憶システムの例

- EMILE (月元 2007)
  - エピソード記憶ベースの記憶モデル
    - 計算機への実装によるシミュレーション実験実施済み
  - 「抑制」機構を積極的に取り入れる
    - というより「抑制に基づく」記憶検索モデル
- 従来の記憶検索モデルとの対比
  - 従来: 文字通り「検索」
    - 「記憶」を検索クエリを頼りに探索し見つけ出す
  - EMILE: グローバルマッチング・生成
    - 想起対象は見つけない、作り出す!

EMILE の原理

☼基本原理 (アルゴリズム)

- エピソード (= エングラム) **e** 同士の「側抑制」
  - **e** を多次元ベクトルで表現 (各次元の値: {-1, 1})
  - エングラム $\mathbf{e}_i$, $\mathbf{e}_j$ の類似度 S_{ij} をベクトルの内積 $\mathbf{e}_i \cdot \mathbf{e}_j$ で表現
  - $\mathbf{e}_i$ の活性値 A_i = プローブ(手掛かり刺激)との類似度 S_{ip} を3乗
    - 類似度はエングラムの総数で割り区間[-1, 1]に標準化される
  - $\mathbf{e}_i$ の「脱活性」値 D_i →
  - 「非転送率」 τ_i も計算

$$D_i = \left(\frac{\sum_{j \neq i}^{n} \left| S_{ij} A_i A_j \right|}{\sum_{j \neq i}^{n} \left| S_{ij} \right|} \right)^{\alpha}$$

- 総活性化値 → $I = \sum_{i=1}^{n} \tau_i^t (1 - D_i) A_i$

EMILEは何をしているのか

☼要するに

- 「何か」が欠けた情報を補完する
  - e.g., 名前からその人の人となりを思い出す
- ただし: 想起 = 生成
  - 記憶していたもの「そのもの」が想起されるとは限らない
  - 過去の経験の「合成」が返される
- 「合成」の際にエングラム間の「差」を最大化する
  - わずかな「似ているもの」のみを活性化し他を脱活性化
    - ただし全てのエングラムが生成に貢献する
  - 脱活性化の履歴は「非転送率」として蓄積される → 忘却

EMILEの示唆

☼表現の「非」認可のメカニズム

- 入力表現(プローブ)と似ている事例同士の競合の結果
  - 活性値 (= プローブとの類似度) が高い = 脱活性値も高い
    = 他の事例の活性化を抑制 (足をひっぱる)
  - プローブとの類似度がわずかに強いものが勝つ
- 以下のようなパターンもありうる
  - プローブとそれほどでもないが多少似ている事例が勝つ
    - よく似ているもの同士が足を引っ張り合い漁夫の利状態になる
  - 特殊な構文使用 などはこのような原理で説明可能?
    - e.g., He sneezed the napkin off the table.

EMILEとPLM

- 言語記憶の事例 e を特徴ベクトルで表現
  - 形式面に関しては「パターンの実現」のパターン
    - e.g., あるパターンと結びついていれば1, そうでなければ0
  - EMILEの計算法がそのまま適用可能
    - 実装を検討中
    - ただし値をどうするかは問題
      - EMILEのように{-1, 1}とするとほとんどが -1 となる
        = ほとんどの事例間の類似度が非常に高くなってしまう
      - かといって{-1, 0, 1} とすると-1と0の値の設定法が問題

課題

- PLM(とその応用)には課題が山積
  - 分節モデルをどうするか
    - 現状ではとりあえず単語分節
      - 形態構造は扱えない, 単語分節自体の妥当性に依拠, 習得の問題
  - 大規模コーパスの分析は到底不可能
    - 有意義な一般化を得るのは困難か
  - 抽象構文 (e.g., SV O_1 O_2) の指定が困難
    - 少なくとも直接は不可能
    - 解決策: 複数のノードを束ねる「架空のノード」を導入

謝辞と参考文献

謝辞 (五十音順)

- 井上 逸兵 氏 (慶應義塾大学)
- 伊澤 宜仁 氏 (慶應義塾大学大学院)
- 上野 良子 氏 (慶應義塾大学大学院)
- 黒田 航 氏 (京都工芸繊維大学, 早稲田大学)
- 佐治 伸郎 氏 (慶應義塾大学)
- 伝 康晴 氏 (千葉大学)
- 長谷部 陽一郎 氏 (同志社大学)
- 日高 昇平 氏 (北陸先端大学)

参考文献

- Borensztajn, G., Zuidema,W., & Bod, R. 2009. Children's grammars grow more abstract with age: Evidence from an automatic procedure for identifying the productive units of language. *Topics in Cognitive Science, 1 (1)*, 175–188.
- Brown, R. 1973. *A first language:The early stages*. Cambridge, MA.: Harvard University Press.
- Croft,W. 2005. Logical and typological arguments for radical construction grammar. In Östman, J.-O., & Fried, M. (eds.) *Construction Grammars: Cognitive grounding and theoretical extensions* (pp. 273-314). Philadelphia: John Benjamins.
- 長谷部陽一郎. 2009. 計算的手法を用いた構文習得の可能性（研究ノート） 『言語文化（同志社大学言語文化学会）』 12, 395-420.
- Kuroda, K. 2009. Pattern lattice as a model for linguistic knowledge and performance. *Proceedings of the 23rd PACLIC* (pp. 278–287).
- 黒田航・長谷部陽一郎. 2009. Pattern Lattice を使った(ヒトの) 言語知識と処理のモデル化. 言語処理学会第15 回大会発表論文集(pp. 670–673).
- Langacker, R. 1987. *Foundations of cognitive grammar Vol. 1: Theoretical prerequisites*. Stanford: Stanford University Press.
- 黒田航. 2007. 徹底した用法基盤主義の下での文法獲得:「極端に豊かな事例記憶」の仮説で描く新しい筋書き. 『言語』, 36 (11), 26-34.
- MacWhinney, B. 2000. *The CHILDES project:Tools for analyzing talk*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Port, R. 2007. How words are stored in memory?: Beyond phones and phonemes. *New ideas in psychology, 25,* 143-170.
- Pustejovsky, J. 1995. *The generative lexicon*. Cambridge: MIT press
- Stefanowitsch, A. 2008. Negative entrenchment: A usage-based approach to negative evidence. *Cognitive Linguistics, 19,* 513–531
- Taylor, J. 2003. Polysemy' s paradoxes. Language Sciences 25, 637–655.
- Tomasello, M. 2003. *Constructing a language: A usage-based theory of language acquisition.* Cambridge, MA.:Harvard University Press.
- 月元敬. 2007. 『抑制に基づく記憶検索理論の構成』東京: 風間書房
- 吉川正人. 2010a. 「語」を越えた単位に基づくコーパス分析に向けて: パターンラティスモデル(PLM) とその有用性. 『藝文研究』 98, 221–207.
- ——. 2010b. パターンの生産性に見る統語発達: パターン束モデルに基づく習得プロセスの検証. 日本認知科学会第27 回大会発表論文集 (pp. 235-241).

ご清聴有難うございました。