「語」を超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

吉川 正人 (慶應義塾大学大学院)

1. はじめに

近年、PC やインターネットの普及に伴い、電子データを利用した言語分析が以前よりもはるかに容易になったことから、電子化された言語データベース、所謂「電子コーパス」に基づく実証的な研究が言語学内外で盛んになってきている。しかしながら、後に述べるように、少なくとも「文法」の記述・分析においては、現在行われている手法では方法論的な限界と理論的な問題点が存在するということが指摘できる。

本稿では、そのような問題を解決する手段として、「語」という単位を超えた、「超語(彙)(super-lexical)」の単位に基づいた分析を行う必要性を論じ、その手段として有効と思われる方法論を提示する。具体的には、コーパス C を構成する表現群 $E = \{e_1, e_2, \ldots e_n\}$における個々の表現 e_i を 1) m 個の分節に分節化(segmentation)し、2) 1~ m 個の分節を網羅的かつ再帰的に変項化することで可能な全パターンの集合 P を得、3) そこから何らかの方法で有用なパターンを発見する、という方法を紹介する。ただし、現段階ではその方法論を利用した実証研究やケーススタディを実践しているわけではない。従って本稿の目的はそのような方法論をとることの理論的な意義を議論することである。

2. 従来のコーパス分析の問題点

本節では、従来のコーパスを用いた文法分析で主流となっていると思われる 2 つの分析手法を紹介し、その方法論的限界と理論的な問題点とを指摘する。ただし、スペースの関係上具体的な研究内容を紹介することはできないため、興味のある向きは、参考文献に直接あたっていただきたい。

2.1. 背景

チョムスキー(Chomsky 1957)以来、文法とはある言語 L において「適格 (well-formed)」な文の全集合を一義的に定義(もしくは「生成 (generate)」)し、「不適格 (ill-formed)」な文を厳密に排除するような規則もしくは原理の体系である、という見方が理論言語学の基本想定となった。Chomsky (1957, 1965)の議論では、具体的な語彙項目の(確率的/統計的な)連続(e.g., マルコフ過程)として文法を捉えることは不可能であり、文法を構成するのはより抽象的な単位(e.g., 品詞, 書き換え規則, 句構造)であるとされた。

このような背景から、コーパスという実際の具体的な言語データに基づいて議論を行うコーパス言語学は、特に文法研究においては、理論言語学とあまり親和性を持たずに、時に互いを批判しあいながら(e.g., Leech 1992), 独自の発展を遂げてきたと言える。実際、Leech (1992)は、コーパス言語学は理論と言うより方法論であり、理論言語学とは一線を画すという趣旨の発言をしている(Leech 1992: 105-107)

しかしながら、近年、コーパス言語学の中から文法理論を構築しようとする動きがみられるようになってきている(e.g., Hunston & Francis 2000; Sinclair & Mauranen 2006)。この背

景には恐らく、大規模コーパス(e.g., *British National Corpus*, BNC)の登場や PC およびインターネットの普及に伴い、以前では考えられなかったような規模のデータが容易に入手でき、また、容易に分析ができるようになったということがあろう。

2.2. コーパスに基づく文法理論

コーパス言語学の範囲内で文法理論を構築するということは、コーパスのデータからある言語の文法特性を十分に記述し切る、ということであると言える。この時、コーパス言語学には(少なくとも)以下二つの方法論が許される:

(1) a. 何らかの文法項目(e.g., 構文)に合致する例を(主に)調査者の直感に基づいて選別し、その選別したデータから(統計量などを用いて)一貫した特徴/構造が指定できることを示す;
b. ある程度の規模を持つデータを何らかの方法(e.g., 一定の範囲内での語彙の共起頻度を測定する)で定量的/統計的に分析し、その結果が文法特性の指定に有効であることを示す。

(1)a は予め記述対象をトップダウンに限定し、得られたデータからボトムアップに構築した結果との整合性を計る方法であり、(1)b は完全にボトムアップにデータを分析し、得られた結果が有用なものであるかどうかを評価するという方法論である。

多くのコーパス言語学的な文法分析(e.g., Biber, Conrad & Reppen 1998)では(1)a の方法が取られている。例えば Stefanowitsch & Gries (2003)による「共起構文分析 (collostructional analysis)」では、何らかの構文(e.g., 二重目的語構文)の実例を収集し、そこに現れやすい/にくい語彙を生起頻度を利用した統計量で測定する、といった手法がとられる。一方、Hunston & Francis (2000)の「パターン文法 (Pattern Grammar)」や Sinclair & Mauranen (2006)の「線形単位文法 (Linear Unit Grammar)」などは、(1)b の方法論を取る分析であると言える。前者は様々な語の生起環境(= パターン)を特定しようという試みであり、後者は与えられたデータを有意味な単位に分割(= チャンキング)していき、記述的に妥当な表現集を蓄積しようとするものである。

2.3. 問題

しかしながら、このような方法論には明らかな限界がある。第一に、語を超えた単位のパターンを「発見」するのが困難であるということが挙げられる。そもそも、(1)a の方法論を取る分析では初めからこの目的を放棄している。(1)b の方法論を取る上述のパターン文法や線形単位文法では、有用なパターンや表現を発見することはできても、網羅性を達成するのは実に困難である。それ自体は(時間と根気さえあれば解決するという意味で)本質的な問題ではないかもしれないが、最も深刻なのは、そのような方法論を用いて得られた記述が、必要な記述量の何パーセント程度を網羅しているのか、つまり、達成率/網羅率が計算できない、ということにある。言い換えれば、ゴールが見えないということである。

2.4. 「全体」優先の議論

このような問題提起を不思議に感じる者もいるかもしれない。恐らくその場合、念頭にあるのは以下のような想定であろう:

(2) a. 文法の基本構成単位は「語 (もしくは形態素)」である (大前提);
b. 語より大きな単位は、語から構成されている (「構成性原理」の想定);
c. 従って、語の用法が特定できれば、文法の記述は完結する (小結論);
d. あるコーパス C に生起する語のタイプの集合は有限の集合として特定できる;
e. 従って、「語」と言う単位に基づいた調査を行えば網羅的に文法記述が可能である (結論)

しかし、このような考えは誤りである可能性が高い。

Sinclair (1991)の示した「イディオム原則(Idiom Principle)」、即ち、他の条件が同じならば、全体は常に部分に優先するという原則の存在は、このことを強く物語っている。[1] また、理論言語学では、80 年代末から「構文文法 (Construction Grammar)」と呼ばれる文法理論が提唱されて、Goldberg (1995)や Croft (2001)などを経て、一つの大きな潮流を築き上げている。構文文法では、文法を構成する要素には、「構文」という単位が存在し、それが語に還元できない独自の意味を持っているという主張がなされる。例えば英語の「二重目的語構文 (Ditransitive Construction: e.g., (3)a)」には“X causes Y to receive Z”といった意味が備わっており、例えばそれを構成する動詞(e.g., *kick*)にはその意味を還元することができない、と考える(Cf. (3)b)。

(3) a. He gave me the book.
b. He kicked me a ball.

さらには、例えば Alison Wray (Wray 1999, 2002; Wray & Perkins 2000)による「定型表現 (formulaic language)」の議論、即ち、「言語の本質は定型性にある」という主張も同一線上にあると言える。彼女はまた、太古の言語は未分化な「全体的 (holistic)」な特徴を持つ複雑な伝達機能を果たす表現であり、言語の進化は「分析 (analysis)」の過程である、とする議論も展開している (Wray 2000)。

このような議論から、語が一次的な基本構成単位であり、文を語から構成される派生的・二次的な単位と看做す発想は、言語の重要な性質の多くを捉え切れない可能性が明らかになる。そうなれば、当然の帰結として、コーパス言語学も、このような言語の性質をうまく捉えられるような、語のような構成単位を前提としない、「全体」をそのまま扱える分析手法を確立する必要があるということになる。

3. 「語」を超えた単位に基づいた分析へ

本節では、前節の議論で明らかになった問題を解決する方法論及びそれを支える理論として、「パターンラティスモデル (Pattern Lattice Model: PML)」(黒田・長谷部 2009; Kuroda 2009)を紹介する。以下ではまず、PLM の導入の全段階として、「語」を超えた単位に基づく分析の難しさについて述べる。

---

[1] 「イディオム原則」とは、ある語 (e.g., *kick*)が何らかのイディオム (e.g., *kick the bucket*)の一部として現れた場合、イディオムとしての解釈が常に優先されるということである。勿論、イディオム = 全体の解釈が「優先」されるだけであって、外部要因(e.g., 先行文脈)によって部分に注目する解釈 (e.g., 「バケツを蹴る」)が得られることはある。このように、部分から構成されていると見る解釈が可能であることは、「自由選択原則 (Open Choice Principle)」という原則によって捉えられる。

3.1. 方法論的問題

前節で述べたように、(1)b のようなデータ駆動(data-driven)型の分析手法は網羅性という観点で困難を伴う。一方、網羅性の達成が容易な「語」という単位に訴えた分析法は、理論的な問題を孕む。このジレンマの背景には、明らかに、語を超えた「超語(彙) (super-lexical)」の単位を認定する手段がない、ということがある。

変項を含まない完全に語彙的に特定されたパターンのみを考えるならば、任意の n に対して(単語)n-gram を作成するという方法は存在する。単語 n-gram とは、対象のテキストから n 語の連続を網羅的に抽出したものである。例えば、(4)a の文からは、$n = 2$ の場合(4)b の 2-gram (= *bigram*)の集合が、$n = 3$ の場合(4)c の 3-gram (= *trigram*)の集合が抽出される。

(4) a. Keio has a proud history as Japan's very first private institution of higher learning, which dates back to the formation of a school for Dutch studies in 1858 in Edo (now Tokyo) by founder Yukichi Fukuzawa. [2]

b. Keio has, has a, a proud, proud history, history as, as Japan's, Japan's very, very first, first private, private institution, institution of, of higher, higher learning,, learning, which, which dates, dates back, back to, to the, the formation, formation of, of a, a school, school for, for Dutch, Dutch studies, studies in, in 1858, 1858 in, in Edo, Edo (now, (now Tokyo), Tokyo) by, by founder, founder Yukichi, Yukichi Fukuzawa

c. Keio has a, has a proud, a proud history, proud history as, history as Japan's, as Japan's very, Japan's very first, very first private, first private institution, private institution of, institution of higher, of higher learning,, higher learning, which, learning, which dates, which dates back, dates back to, back to the, to the formation, the formation of, formation of a, of a school, a school for, school for Dutch, for Dutch studies, Dutch studies in, studies in 1858, in 1858 in, 1858 in Edo, in Edo (now, Edo (now Tokyo), (now Tokyo) by, Tokyo) by founder, by founder Yukichi, founder Yukichi Fukuzawa

しかしこれでは当然ながら変項を含むパターンは扱えず、必然的に記述対象は限定される。n-gram ベースで単一の変項を含むパターンを生成するアルゴリズムは存在する(e.g., kfNgram, http://www.kwicfinder.com/kfNgram/kfNgramHelp.html)が、それでも網羅性は部分的である。[3]

3.2. パターンラティスモデル (PLM)

以上の困難を克服するアルゴリズムとして、「パターンラティスモデル (Pattern Lattice Model, 以降 PLM)」を紹介する。PLM は、黒田・長谷部 (2009)によって提案された超語(彙)パターンの生成アルゴリズム及びその構造のモデルである。PLM では、ヒトの言語知識が具体的な事例と以下で定義する事例の「索引」としての「パターン」からなると考える。そしてそのパターンの集合は階層的なネットワーク構造であるラティス構造を成すと考えられる。

ある表現の事例 e からパターンの集合 $P(e)$を生成するアルゴリズムは以下のように定式

---

[2] この文は慶應義塾大学ホームページ英語版の紹介ページ (http://www.keio.ac.jp/english/about_keio/introduction.html)から引用した。

[3] kfNgram では、ユーザーが指定した任意の n に対し生成した n-gram の集合から、n-gram の頻度リストを生成し、その中から n - 1 語が共通するものを一つにまとめ、非共通部分を変項とするパターンを生成する。

化される (黒田・長谷部 2009: 670-671):

(5) a. 事例 e を適切な分割モデル M によって分節化(segmentation)し、その結果を $T(e) = [t_1, t_2, \ldots t_n]$ とする;
b. $T(e)$の要素 t_i を再帰的に単一の変項 X で変項化し、得られたパターンのべき集合を $P(e)$とする

(5)b の変項化の産物をパターンと定義する。例えば、(6)a の文を(6)b のように分節化した場合、(6)c のパターンが生成される("_"は変項を表わす):

(6) a. I am a boy.
b. [I, am, a, boy]
c. I am a boy, _ am a boy, I am a _, I am _ boy, I _ a boy, _ am a _, _ am _ boy, _ _ a boy, I am _ _, I _ a _, I _ _ boy, _ am _ _, _ _ a _, _ _ _ boy, I _ _ _, _ _ _ _

分節化のモデルはそれ自体独立に既定される必要があるが、さしあたっては、英語への適応の場合はおおよそ「単語分節 (word-segmentation)」と同一視して構わないと思われる。

このようにして得られたパターンの集合は、以下の is-a 関係を持つ半順序集合、パターンラティス $PL(e)$を構成する (黒田・長谷部 2009: 671):

(7) $P(e)$に含まれる p_i, p_j において、パターン p_i の n 番目の要素 $p_i[n]$ とパターン p_j の n 番目の要素 $p_j[n]$ との関係で，i) $p_i[n] = p_j[n]$ であるか，ii) $p_j[n]$ が変項ならば，$[p_i$ is-a $p_j]$ である

e = "I am a boy" (=(6)a)の場合の $PL(e)$を図示すると図 1 のようになる。図 1 のラティスは Pattern Lattice Builder (http://www.kotonoba.net/rubyfca/ からダウンロード可能)を用いて作製した。図 1 では最上部が頂点 (TOP)、最下部が底 (BOTTOM)となっている。

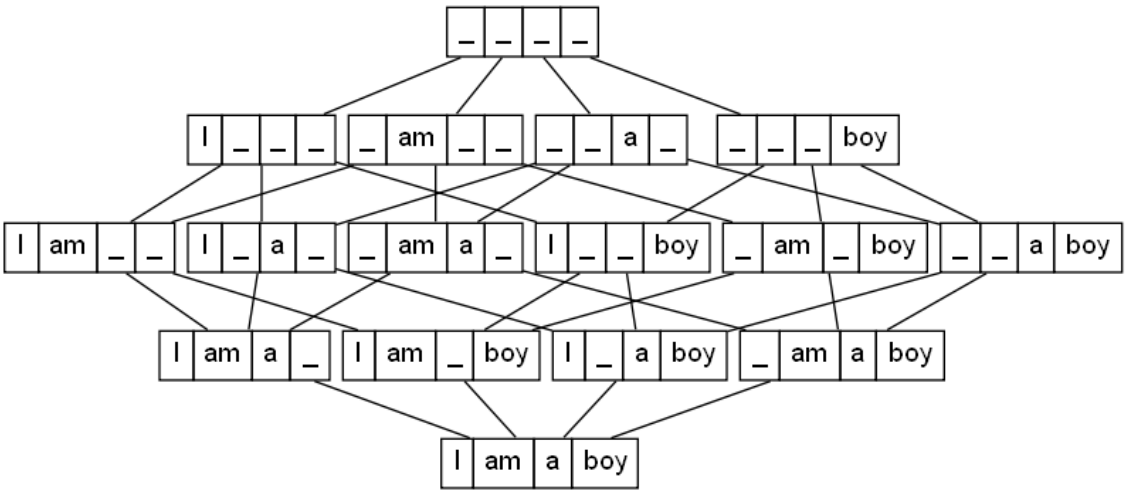

図 1: "I am a boy"のパターンラティス (Pattern Lattice Builder を用いて作製)

パターンラティスの階層を「ランク (rank)」と呼ぶ。ランクは、非変項 = 定項である分節の数で定義される。一般に、n 個の分節から成る事例に対してはランク n からランク 0 のパターンが生成される。ラティスの頂点は常にランク 0 = 全て変項からなるパターンであり、ラティスの底はランク n = 事例である (黒田・長谷部 2009: 671)。

以上は単一事例 e に対するラティス $PL(e)$の生成プロセスであったが、コーパス全体の表現群 $E = \{e_1, e_2, \ldots e_n\}$に対して $PL(E)$を生成するには、個々の PL を統合する必要がある。この際、互いに「長さ」=「分節数」の異なる事例 e_i, e_j のラティス $PL(e_i)$と $PL(e_j)$は互換性を持たないため、この互換性を保証するために以下のような処理を行う(黒田・長谷部 2009: 671):

(8) 任意の連続した l 個の変項列 X と連続した l–1 個の変項列 X'について，[X' is-a X]とすることで異なる長さのパターンを統合する (「変項の再帰的単純化」)

ただし、現行の PLM の実装である Pattern Lattice Builder (以下 PLB)では、(8)の処理は

(9) 連続する変項を単一の変項に置換する

という処理に置き換えられている。これは技術的な問題による対処であるが、本稿の目的は PLM をコーパスに基づく文法研究に活用する意義を議論することであり、PLM の理論的な検討ではないため、以下ではこの代替処理を想定した議論を進める。

図 2 に E = {*I am a boy*, *I am a student*, *I was a boy*}の場合の $PL(E)$を提示する。図 2 では頂点が左端に、底が右端(ただし底は図には現されていない)に位置され、図 1 を反時計回りに 90 度回転した形となっている。図 2 では、パターンの重要度に応じた色づけが成されている。重要度とは、現行の PLB では、それぞれのランクにおける標準化された頻度、即ち、「z スコア (z-score)」を用いて計算されている。ランク k のパターン p_i の z スコア $z(p_i)$は、p_i の頻度を $f(p_i)$、ランク k の頻度の平均を $f(k)$、頻度の標準偏差を $s(k)$として、

$$z(p_i) = \frac{f(p_i) - f(k)}{s(k)}$$

で計算される。ただし、このようなパターンの重要度の評価尺度は、それ自体独立に議論される必要のあるものであり、z スコアによる計算は暫定的なものであることは断っておく。[4]

3.3. PLM の有用性

PLM の利点は、言うまでもなくその網羅性にある。調査対象のコーパス C の全表現 $E = \{e_1, e_2, \ldots e_n\}$に対し $PL(E)$を作成する、もしくは、単に $P(E)$を生成することで、有用な可能性のある語を超えた単位のパターンの全集合を得ることができる。勿論、得られた $P(E)$のパターン全てが重要/有用であるわけでなく、どのパターンが有用であるかを評価する基準が独立に必要であるが、それでも、パターンの全集合を盲目的に(=理論等によるバイアス抜きに)得られる手法であるというのは大いに評価すべき点である。逆に考えれば、$P(E)$さえ得られれば、複数の重要性の評価基準を $P(E)$に適応することで、様々な観点から有用なパターンを抽出できる、ということを意味する。

[4] z スコアを用いること、及びこのように z スコアを計算することの問題点として考えられることとして、少なくとも以下の 2 点が挙げられる: 1) 計算に標準偏差を用いているが、パターンの頻度の分布が正規分布になっている保証がない; 2) 平均・標準偏差を計算する際の母集団が同一のランクに位置するパターンの集合となっているが、この妥当性が明らかではない。

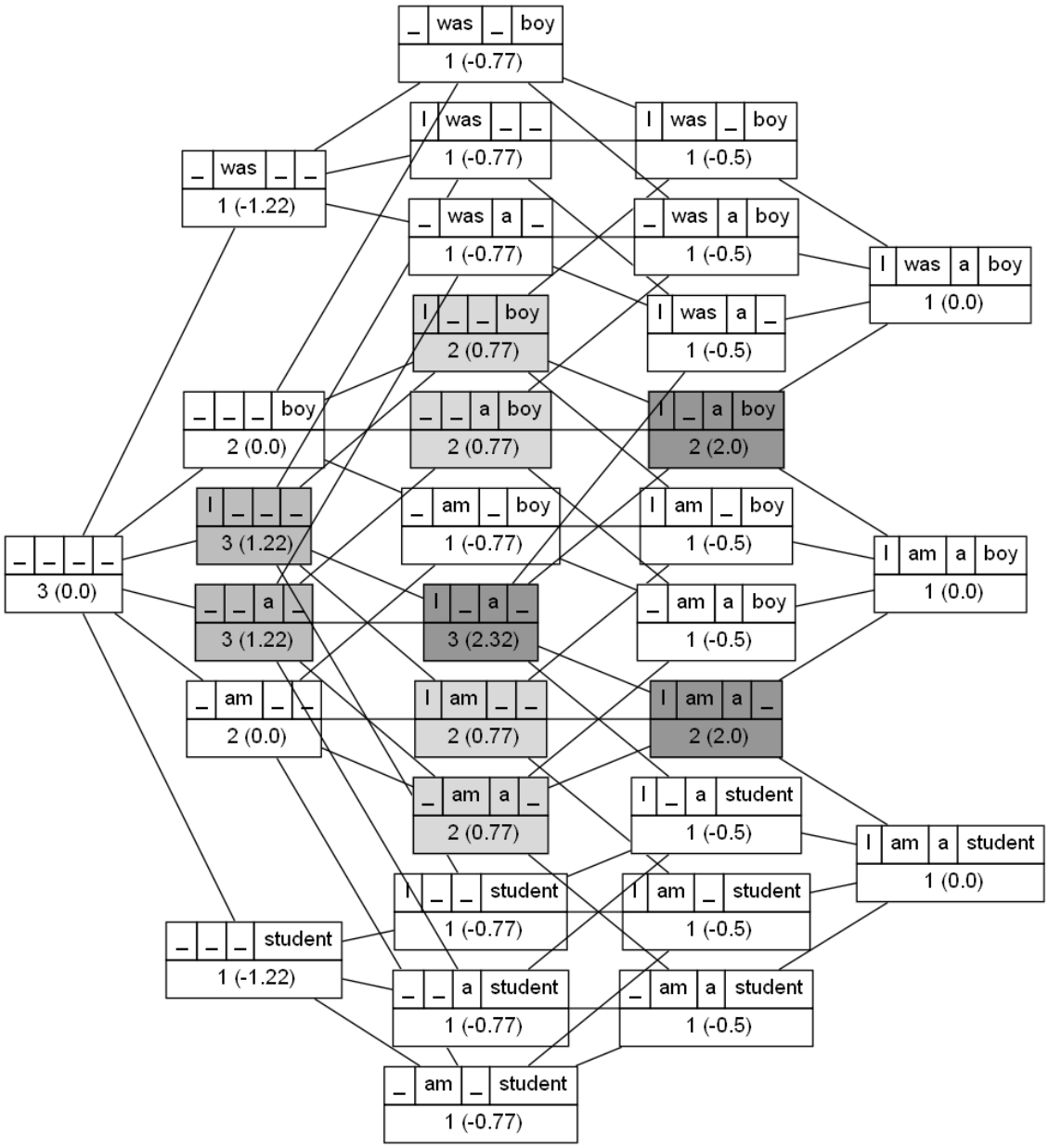

図 2: E = {*I am a boy*, *I was a boy*, *I am a student*}の $PL(E)$ (数値は頻度、括弧内の数値は z スコア)

3.4. PLM の難点

しかしながら、現時点では PLM には、コーパスの分析に適用するにはあたってはいくつかの難点を指摘できる。第一に、パターンを盲目的に生成するという上述の利点が、大規模データを対象とした場合、得られるパターン集合が処理が困難なほど膨大になるという難点を生んでしまう。対象となるコーパスの規模によっては、相当の性能を持つ PC でなければそもそもパターンを生成し切ることができない可能性もある。第二に、コーパスのデータは、特に書きことばの場合、一文の長さ(= 語数)がある程度長いものが多いが、そうなると、一文に対してパターンを生成する段階でかなりの処理負荷がかかり、実処理と

して非現実なものとなってしまう可能性が高い。[5]

一点目の問題は深刻であるが、ひとまず適応するコーパスの規模を小規模のものに限定し、断片的な記述を積み重ねるという方法で対処は可能である。二点目に関しては、以下のような処理を行うことで対処できる:

(10) 語数が閾値 l (e.g., 7 語)を超える文 s に関しては、s から $n = l$ となる n-gram を作成し、得られた n-gram の一つ一つに対しパターン集合を生成したものを統合する

3.5. PLM の活用案

筆者は現在、試験的に PLM のアルゴリズムをいくつかの独自の改定を加えた上で複数のコーパスに対して適用し、分析を行っている途上である。そのうちの一つは、*Brown Corpus* への適用である。[6] 実際の PLM の適用にあたっては、(9)の単純化を採用し、(10)をやや改定した処理を実施している。[7] それは以下のようなものである:

(11) 語数 m が閾値 l (e.g., 7 語)を超える文 $s = \{w_1, w_2, \ldots, w_m\}$ に関しては、

a. w_1 から $w_{l\text{-}1}$ の連続の末尾に変項を一つ追加したパターン $I_{\text{init}} = [w_1, w_2, \ldots, w_{\text{l-}1}, \_\ ]$ を作成する;

b. $w_{m\text{-}l+1}$ から w_m までの連続の先頭に変項を一つ追加したパターン $I_{\text{end}} = [\ \_, w_{m\text{-}l+1}, w_{\ m\text{-}l+2}, \ldots, w_m]$ を作成する;

c. w_2 から $w_{m\text{-}1}$ の連続に対し、$n = l\text{-}2$ となる n-gram を作成し、その先頭と末尾に変項を追加したパターン(群)I_{mid} を作成する;

d. I_{init} , I_{mid} , I_{end} それぞれに対しパターン集合 $P(I_{\text{init}})$, $P(I_{\text{mid}})$, $P(I_{\text{end}})$ を生成する;

e. s のパターン集合 P(s)を $P(I_{\text{init}})$, $P(I_{\text{mid}})$, $P(I_{\text{end}})$ の和集合 $= P(I_{\text{init}}) \cup P(I_{\text{mid}}) \cup P(I_{\text{end}})$ と定義する

例えば、$l = 7$ の場合、(12)a の文に対しては(12)b の I_{init}、(12)c の I_{end}、(12)d の I_{mid} 群が作成され、それがそれぞれパターン生成の入力となる。

(12) a. Yesterday my sister and I went to the park.
b. [Yesterday, my, sister, and, I, went, _]
c. [_, and, I, went, to, the, park]
d. {[_, my, sister, and, I, went, _], [_, sister, and, I, went, to, _], [_, and, I, went, to, the, _]}

Brown Corpus への適用にあたっては、l を 7 に定め、結果、11,688,086 パターンを得た。このうち、3.2 節で述べた方法でパターンの z スコアを計算し、$z \geq 1$ のパターンを「有用なパターン (good patterns)」とし、これを抽出した。結果、124,969 のパターンを得た。[8] 現

---

[5] ちなみに、*Brown Corpus* においては、非単語記号を取り除いた場合の平均文長は 17.66 語である。

[6] *Brown Corpus* は、1960 年代に Henry Kucera と W. Nelson Francis によって編纂されたアメリカ英語の電子コーパスであり、正式名称を *The Standard Corpus of Present-Day Edited American English* と言う。規模は約 100 万語 (正確には 1,014,312 語)である。参考: http://icame.uib.no/brown/bcm.html

[7] パターンの生成には、スクリプト言語 Python (ver. 2.5.2; windows 版)を用いて独自に作成したプログラムを利用した。

[8] 黒田航氏 (情報通信研究機構 NICT)との私信により、実際の z スコアの計算は、母集団を同一ラ

時点ではここまでの処理しか行っていないが、今後は、得られたパターンから変項の値を取得した上で、様々な文法現象の記述に有効であることを実証していく予定である。

現時点で言えることは、このような前処理と事後処理を行うことで、100 万語程度の規模のデータ対しては現実的な処理が可能となるということである。勿論問題はこの後であり、得られたデータからどう有益な議論を展開するか、ということは未知であるが、これだけでも、実際のデータ分析への適用にあたっての指針を示すことはできたかと思われる。

大規模コーパス(1000 万~1 億語規模)への適用にあたっては、例えばそのコーパスを構成するジャンル (もしくはサブコーパス)毎にパターンを生成するといった手法が考えられる。実際、例えば話しことばと書きことば、新聞と小説では有効に利用されているパターンのタイプが異なっているということは容易に考えられる。従って、ジャンル毎に生成したパターンを基にジャンル間の比較を行うと言った研究手法も有効と思われる。

4. 結語

本稿では、従来のコーパスに基づく文法研究における方法論的な限界と理論的な問題点を指摘し、その解決案として、パターンラティスモデル (PLM)を適用した分析の有用性を議論した。PLM の利点は、1) 盲目的かつ網羅的に、2) 語を超えた単位 = 超語(彙)単位のパターンの生成できるという点であり、いくつかの難点は指摘できるが、扱うデータや分析の目的に合わせて適切に前処理・事後処理を行えば、非常に有効に応用利用のできるモデルであると言える。

本稿で実際の分析を提示し、その有用性を実例で示すことができなかったのは残念であるが、少なくともこれまで実施の困難だった分析を可能にするモデルであることは示すことができたと思われる。

4.1. 展望

現在、前節で紹介した *Brown Corpus* への適用の他、言語習得の分析目的に作られた Childes データベース (MacWhinney 2000)内のコーパスに対しても分析を行っている。習得データは大規模コーパスのような規模はなく、その点で PLM の適用も容易である。大規模(均衡)コーパスはある程度の一般性・代表性を確保するためには有用であるが、ちょうどそれと正反対に、ある特定の個人の言語産出のパターンを記述するといった目的には不向きである。その点、習得データは、特定の幼児の言語産出を対象に記述することが可能であり、言語発達のプロセスをパターンの発達のプロセスと見て、習得データを対象に PLM を適用した分析を行うことは有意義であると言える。

4.2. 注記

ちなみに、2.3 節でパターン文法や線形単位文法の問題点を指摘したが、それはこれらの理論/方法論が不適切である、もしくは、誤ったものである、ということは意味しない。むしろ、少なくとも筆者の見る限り、この二つの理論は非常に優れた理論であると言える。実際、PLM をコーパス分析に適用しても、「何が重要か」という基準を別に与え、得られた

ンクのパターン集合ではなく、同一ランクであり、かつ、長さ = 分節数が同一であるパターンの集合としている。

パターン集合から重要なパターンを選定するプロセスなしには、あまり有意義な記述は行えない。その意味では、コーパス分析への PLM の適用は線形単位文法のような手法と相補的に組み合わせて用いることが必要となろう。

参考文献

Biber, D., Conrad, S., & Reppen, R. 1998. *Corpus linguistics: Investigating language structure and use*. Cambridge; New York: Cambridge University Press.

Chomsky, N. 1957. *Syntactic structures*. The Hague: Mouton

——. 1965. *Aspects of the theory of syntax*. Cambridge, MA.: MIT Press.

Croft, W. 2001. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford: Oxford University Press.

Goldberg, A. E. 1995. *Constructions: A construction grammar approach to argument structure.* Chicago; London: University of Chicago Press.

Hunston, S., & Francis, G. 2000. *Pattern grammar: A corpus-driven approach to the lexical grammar of English*. Amsterdam: John Benjamins.

Kuroda, K. 2009. Pattern lattice as a model of linguistic knowledge and performance. *Proceedings of The 23rd Pacific Asia Conference on Language, Information and Computation.* (available at http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/papers/kuroda-paclic23-paper.pdf)

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』, 670-673. (http://yohasebe.com/files/documents/NLP2009Paper1.pdf から入手可能)

Leech, G. 1992. Corpora and theories of linguistic performance. In Startvik, J. (ed.) *Directions in corpus linguistics: Proceedings of Nobel Symposium 82, Stockholm, 4-8 August 1991* (pp. 105-122). Berlin: Mouton de Gruyter.

MacWhinney, B. 2000. *The CHILDES Project: Tools for analyzing talk*. Mahwah: Lawrence Erlbaum Associates.

Sinclair, J. 1991. *Corpus, concordance, collocation*. Oxford: Oxford University Press.

Sinclair, J., & Mauranen, A. 2006. *Linear unit grammar: Integrating speech and writing*. Amsterdam: John Benjamins.

Stefanowitsch, A., & Gries, S. Th. 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics, 8*, 209-243.

Wray, A. 1999. Formulaic language in learners and native speakers. *Language Teaching, 32*, 213-231.

——. 2000. Holistic utterances in protolanguage: The link from primates to humans. In Knight, C., Studdert-Kennedy, M. & Hurford, J. (eds.) *The evolutionary emergence of language: Social function and the origins of linguistic form* (pp. 285-302). New York: Cambridge University Press.

——. 2002. *Formulaic language and the lexicon*. Cambridge: Cambridge University Press.

Wray, A., & Perkins, M. 2000. The functions of formulaic language: An integrated model. *Language & Communication, 20*, 1-28.