句を対象とした構成的な言い換えの生成*

加藤 修平 †　　藤田 篤 †　　佐藤 理史 †
† 名古屋大学大学院工学研究科
shuhei@sslab.nuee.nagoya-u.ac.jp, {fujita,ssato}@nuee.nagoya-u.ac.jp

1　はじめに

同じ命題的意味を持つ表現を**言い換え**と呼ぶ．自然言語処理の諸タスクにおいて，言い換えを生成したり認識したりする技術が求められている．しかし，与えられた表現に対して適切な言い換えを頑健に生成する実用的な機構はいまだ実現されていない．

文献 [2] では，日本語に存在する様々な種類の言い換えを，言い換えの影響が及ぶ範囲や変形の種類に基づいて分類している．このような現象の類型化は現象全体を俯瞰する上で不可欠だが，言及できていない問題もある．例えば，同じ統語構造を持つ句を言い換える場合でも，句全体の命題的意味が構成要素の意味から構成的に推定できるかどうか，構成要素間にどのような関係があるかによって，可能な言い換え方に違いがある．「名詞+格助詞+動詞」という構造を持つ句の言い換えを見てみよう．

(1) a. 慣用句：足を運ぶ ⇒ 立ち寄る
b. 換喩：漱石を読む ⇒ 漱石が書いた本を読む
c. 各構成語が独立に置換可能な句：
リスクが激増する ⇒ リスクが増える
d. 機能動詞構文：感動を与える ⇒ 感動させる
e. 機能動詞構文化可能な句：
肌を刺激する ⇒ 肌に刺激を与える
f. 主辞交替可能な句：
確認を急ぐ ⇒ 急いで確認する

(1a)〜(1c) の言い換えは，語あるいは句をまったく別の語句に置き換えている．このような表現間の同義性を計算するには，同義の表現対に関する膨大な知識が必要である [2]．一方，例 (1d)〜(1f) の例では，表現中の構成語の依存関係や品詞を変えることで，一方から他方を生成することができる．このような構成的言い換えは統語構造の変換をパターン化して網羅し，「急ぐ」–「急いで」のような語彙資源，および与えられた語を用いて様々な表現を生成する機構によって実現できると考えられ，言い換え生成の頑健性向上にも寄与すると期待できる．

我々は，句を一つの命題を表す言語表現，すなわち言い換え処理の単位とみなし，統語的な情報に基づいて構成的に言い換えを生成する機構の実装を進めている．本稿で句と呼ぶのは，具体的には，

「格要素名詞句+格助詞+述語文節」
という文法的形式に一般化できる範囲の表現

を指し，単語の置換や特定の動詞交替に限定せず，例 (2) に示すような現象も扱う．

(2) a. 集合場所を決める ⇒ どこに集合するかを決める
b. 再現可能性を調査する ⇒
再現できるかどうかを調べる
c. 法案の成立を阻止する ⇒
法案が成立しないようにする

これは，類義語辞典に相当する情報の動的生成に相当するため，我々は dynamic phrasal thesaurus と呼ぶ．

本研究に取り組むねらいは次の 2 点に集約される．

頑健な言い換え生成機構の実現： 句に対する様々な文法的形式の構成的言い換えを網羅し，頑健な言い換え生成機構を実現する．

言い換えの基本処理単位についての検討： 語の言い換えにおいては語義曖昧性解消が必須であり，実際は句程度の情報を参照している．明示的に句を対象として言い換えを実現することで語間の様々な相互作用を明らかにするとともに，句を言い換え処理の基本単位とすることの妥当性を検証する．

本稿では，以下，提案モデルにおける知識の定義と実装状況について述べる．

2　言い換え生成のための知識

格要素名詞句，述語文節とひとくくりにできる表現には，図 1 に示すような多様性がある．例 (1d)〜(1f) に示したような格要素名詞と述語文節の動詞の相互作用以外にも，複合名詞や複合動詞における複数の内容語の間の相互作用，構成語の品詞を変えるような言い換えが存在する．我々の目的はそれらの構成的な言い換えを生成する機構を実現することである．

例 (1) のような「名詞+格助詞+動詞」という構造を持つ表現を，以下，品詞に対応する記号を用いて「$N:C:V$」型の表現と呼ぶ．他の型の表現についても同様とする．ここで，(1d)〜(1f) の言い換えを，各々(3) のようにパターン化して表す．

---

*Dynamic Phrasal Thesaurus: A Generator of Synonymous Phrases.
Shuhei Kato†, Atsushi Fujita†, Satoshi Sato†
†Graduate School of Engineering, Nagoya University

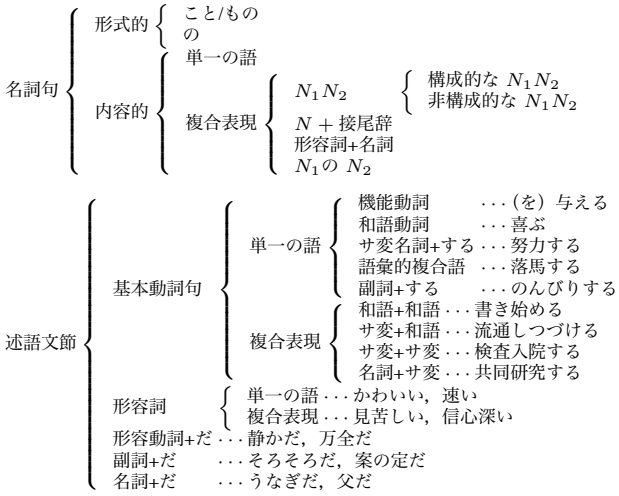

図 1: 名詞句と述語文節の表現の多様性

(3) d. $N:C:V \Rightarrow vp(N)$
e. $N:C:V \Rightarrow N:genCase():lvc(V)$
f. $N:C:V \Rightarrow adv(V):vp(N)$

矢印右側は，言い換え後の表現の大まかな形を，変換元の表現の構成要素を用いて記述したものである．品詞交替や，元の表現の構成語を用いて句を生成する処理は，$vp(N)$ や $adv(V)$ のような関数内で実現する．各知識の定義および記法について，以下で詳述する．

2.1 統語的変換パターン

(3) に示したような，表層的な情報に基づいて表現を構成要素に分解・再構成して言い換え表現を生成する知識を統語的変換パターンと呼ぶ．文法を以下に BNF で示す．

- <統語的変換パターン> ::= <左パターン>⇒<右パターン>
- <左パターン> ::= (<品詞記号>|<語表記>)+
- <品詞記号> ::= N, C, V, Adj, Adv
- <語表記> ::= (<ひらがな>|<カタカナ>|<漢字>)+
- <右パターン> ::= (<品詞記号>|<語表記>|<句生成関数>|<語彙関数>)+

品詞記号に記したものは，各々，名詞 (N)，格助詞 (C)，動詞 (V)，形容詞 (Adj)，副詞 (Adv) を表す．

2.2 句生成関数

与えられた語を組み合わせて，指定された文法形式に合致する様々な表現を生成する (3f) の $vp(N)$ のような関数を句生成関数と呼び，次の文法で記述する．

- <句生成関数> ::= <生成対象句>'('<品詞記号>*')'
- <生成対象句> ::= np, vp, lvc

句生成関数には，内容語句を生成するものと機能表現を生成するものの 2 種類がある．例えば，(3d)，(3f) の $vp(N)$ は名詞 1 つを引数とし動詞句を生成する関数，(3e) の $genCase()$ は格助詞を生成する関数である．

生成可能な表現が複数ある場合はすべてを出力するために ‘,’ で連結した文字列が返り値とする．ただし，表層生成時にいずれか一つを選択できる場合は，それらを ‘/’ で連結して返す．例を (4) に示す．

(4) $vp(N) \Rightarrow v(N):genVoice():genTense()$
例：vp(‘会議’) ⇒ ‘会議する’:
{ϕ,‘れる’/‘られる’,‘せる’/‘させる’}:{ϕ,‘た’}

矢印右側の表現が返り値を表す．$v(N)$ は次節で述べる語彙関数，$genVoice()$，$genTense()$ は各々，態，時制を表す接尾辞表現を生成する句生成関数である．

2.3 語彙関数

品詞交替などを司る (3f) の $adv(V)$ のような関数を語彙関数と呼び，次に示すように記述する．

- <語彙関数> ::= <生成対象語>'('<品詞記号>')'
- <生成対象語> ::= v, n, adv, adj, lv, wh

句生成関数と同様，ある引数に対して複数の表現を返り値とする場合は，それらを ‘,’ で連結したものを返す．語彙関数の実体は語と語の対応関係を書き下した辞書であり，見出語を定義域，本文を値域に見立てて用いる．現在までに実装したものの多くが品詞交替を扱う (5a) のような関数 [3] であるが，他にも，(5b) のように，内容語から機能的な表現を生成するものもこの形式で実現する．

(5) a. $v(N)$ 例：v(‘教え’) =‘教える’
b. $wh(N)$ 例：wh(‘原因’) ={‘なぜ’, ‘どうして’}

3 実装状況

3.1 言い換え生成のための知識

図 1 に示した様々な表現に対する統語的変換パターンを得るため，(3) に示した「$N:C:V$」型の統語的変換パターンを出発点として，パターンを作成済みの型と少しだけ違う型の表現を対象として知識を実装する．これにより，型に特有の変換パターンや，句生成関数で生成すべき表現をなるべく網羅的に列挙する．

最初に，(3) に基づいて「$N_1:N_2:C:V$」型の表現の言い換えを表す次のようなパターン群を得た．

(6) a. $N_1:N_2:C:V \Rightarrow$
$N_1:genCase():vp(N_2)$ (3d)
b. $N_1:N_2:C:V \Rightarrow vp(N_1,N_2)$ (3d)
c. $N_1:N_2:C:V \Rightarrow$
$np(N_1,N_2):genCase():lvc(V)$ (3e)
d. $N_1:N_2:C:V \Rightarrow adv(V):vp(N_1,N_2)$ (3f)
e. $N_1:N_2:C:V \Rightarrow np(N_1,N_2):C:V$ (新規)

(6c)，(6e) で $np(N_1,N_2)$ という句生成関数を導入した．これは，図 1 の『構成的な N_1N_2』を生成する関数で

あり，例えば，次のような表現を出力する．

(7) $np(N_1, N_2) \Rightarrow \{N_1, N_2, N_1 : の : N_2, N_1 : N_2,$
$vp(N_1) : N_2, wh(N_2) : vp(N_1) : か, \ldots\}$
例:np(‘出火’,‘原因’) $\Rightarrow$ {‘出火’,‘原因’,‘出火の原因’,
‘出火原因’, vp(‘出火’):原因’,‘なぜ:vp(‘出火’):か’, $\ldots$}

句生成関数および語彙関数は，Meaning-Text Theory（以下，MTT）[5] において定義されている語彙関数を参考に設計する．MTT では 60 種類以上の語彙関数が定義されているが，我々は，句の言い換えに必要なものから順次実装を進める．まずは上のような手順でパターンと関数を列挙し，その後，作例から帰納的に得たパターンを追加した．現在までに，表 1 に示す種類・規模の知識をシステムに実装済みである．

3.2 言い換えシステム

図 2 に言い換えシステムのブロック図を示す．システムの処理は次の 4 つのステップからなる．

1. 形態素解析： 入力文字列を形態素解析し，品詞などの情報を付与した形態素ノードの列に変換する．その際，文字列中にサ変名詞と「する」が連続して現れる場合はそれらをまとめて 1 つの動詞ノードにしておく．なお，現在は MeCab[1] を用いている．

2. 統語構造変換： 2 節で述べた知識を用いて，言い換え候補表現を生成する．まず，入力の形態素列に合致する統語的変換パターンによって言い換え表現の概形を生成する．次に，句生成関数，語彙関数を呼び出し，各言い換え候補表現を具現化する．この際，活用語は基本形を生成する．

3. 言い換え候補表現の展開： 言い換え候補表現中の選言（‘,’）を全て展開し，複数の言い換え候補表現を列挙する．

4. 表層生成： 構造変換後に，比較的容易な修正・選択処理 [1] を施す．具体的には，各言い換え候補表現について,「れる/られる」のように ‘/’ で区切られた要素が含まれる場合はいずれか 1 つを選択し，その後，活用語の活用形を修正する．例えば「確認する:れる/られる」は，まず「確認する:れる」とし，続けて「確認さ:れる」とする．

4 議論

4.1 言い換えの基本単位

分類語彙表 [4] や EDR 日本語単語辞書 [6] を用いれば，単一の語の置換による言い換えは実現できる．ただし，次の 2 つの問題が示すように，実際は語の周辺を参照する必要がある．

[1] http://mecab.sourceforge.jp/, ver. 0.91

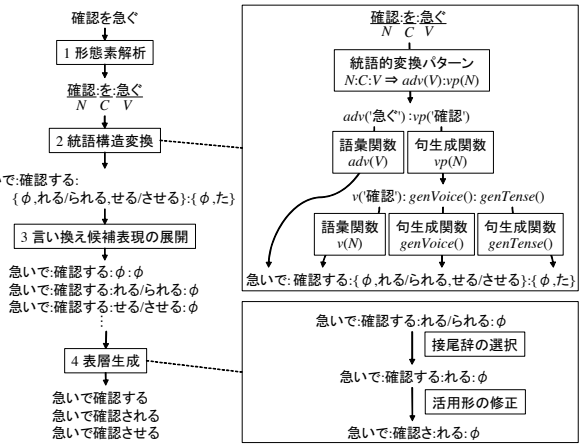

図 2: 言い換えシステムのブロック図

入力の語義曖昧性： 例えば,「薄い」という語は文脈によって「ひらべったい」と言い換えられる場合と,「淡い」と言い換えられる場合がある．このように，対象語の語義によって可能な言い換え方は異なるので，語義の曖昧性解消は欠かせないが，現在の語義曖昧性解消法の多くは周辺語の分布を教師データのそれと比較しており，句程度の内容を参照していることに等しい．

出力の文脈適合性： 語のみを同義表現に置き換えるだけでも，前後の文脈と統語的・意味的に整合がとれなくなる場合がある．複数の表現を生成して言語モデルで最適なものを選択する手法を，検証対象の境界を句とすることで改良できる可能性がある．

一方，句より長い節を超えるような言い換えを実現するには，節間の修辞的関係・因果関係を捉える必要がある．しかし，本質的にはそれらは節および句の意味処理（意味理解）に根ざしている．我々は，句の言い換えを柔軟に扱うことが，節や文の言い換えを扱う際の基本的な技術になると考える．これについては今後実験を踏まえて検証したい．

4.2 言い換えの構成性

これまでに実装済みの知識を用いてどれくらい多くの言い換えを生成できるかを調査した．まず，毎日新聞データ集 1991 年版によく出現する「$N_1 : N_2 : C : V$」型の表現に対して，人手で言い換えを付与し，211 件の事例を作成した．これらのうち我々のシステムは 34 件（16%）を生成できた．現状の再現率は非常に低いが，現在の枠組で，パターンと実装済みの辞書を拡張すれば 88 件（約 40%）を，さらに現在は未実装の同義語辞書を追加すれば 175 件（約 80%）を生成できる見込みである．事例に偏りがあったことは否定できないが，決して少なくない割合の言い換えが構成的に表現

表 1: 実装済みの知識

統語的変換パターン（6 つの型，37 パターン） （一部は用例に基づいて帰納的に作成）		「$N:C:V$」型: 3 パターン，「$N_1:N_2:C:V$」型: 10 パターン，「$N:C:V_1:V_2$」型: 10 パターン， 「$N:C:Adv:V$」型: 7 パターン，「$Adj:N:C:V$」型: 4 パターン，「$N:C:Adj$」型: 3 パターン
句生成関数	内容語句生成関数（5 種）	名詞句生成: $np(N_1,N_2)$，動詞句生成: $vp(N)$, $vp(N_1,N_2)$, $vp(V_1,V_2)$，機能動詞構文生成: $lvc(V)$
	機能表現生成関数（4 種）	格助詞の生成: $genCase()$，態を表す接尾辞の生成: $genVoice()$, 時制を表す接尾辞の生成: $genTense()$，アスペクトを表す接尾辞の生成: $genAspect()$
語彙関数	文法カテゴリ交替（11 種）	動詞への交替: $v(N)$, $v(Adj)$, $v(Adv)$, $v(An)$，名詞への交替: $n(V)$，機能動詞への交替: $lv(N)$ 副詞への交替: $adv(V)$, $adv(Adj)$, $adv(N)$，形容詞への交替: $adj(N)$, $adj(Adv)$
	その他の表現対（1 種）	名詞が表す 5W1H 情報: $wh(N)$

できると考えられ，それらの実装に取り組むことには意味があると言える．

4.3 実装すべき機能

言い換え生成に不可欠ないくつかの機能を，早急に実装する．優先度の高い順に示す．

係り受け情報の利用： 言い換え生成時に係り受け情報を用いていないため，「$Adv:N:C:V$」と「$N:C:Adv:V$」のように修飾語の順序が異なる表現の統語的変換パターンを適用できない．言い換えエンジン KURA [7] では形態素をノードとする依存構造を扱うことで，様々な言い換えを柔軟に処理している．これを参考に，我々も係り受け構造を扱えるようにシステムを改良する．

知識の管理： より多くの型，複雑な型の表現を扱うにつれて，知識の規模が大きくなり，管理が困難になる．これを防ぐために，句の一部に対する構成性を認めて，統語的変換パターンを句生成関数と同じように用いることを検討している．例えば，例 (8) に示すように，「$N_1:N_2:C:V_1:V_2$」型の統語的変換パターンから，句生成関数 $applyPat$ を介して「$N:C:V$」型の表現用のパターンを呼び出す．

(8) a. $N_1:N_2:C:V_1:V_2 \Rightarrow$
$N1:applyPat(N_2:C:V_1):V_2$

関数適用の制約と言語モデル： 例 (3)〜(7) に示したように，これまでに実装した知識は非常に単純である．とくに，統語構造変換パターンと句生成関数については，品詞に関する適用条件しか指定していないので，品詞が合致するだけですべての表現を過剰に生成してしまう．これを防ぐため，表層生成の後に言語モデルを用いて不適格な表現を棄却することはもちろん，必要に応じて知識の適用条件を記述できるように枠組を改良する．

辞書的知識の精緻化と大規模化： 語彙関数の実体として現在使用している辞書は，形態素解析用の単語辞書およびコーパスからヒューリスティックスを用いて自動構築したもの [3] であり，「見にくい」–「見つける」のようなノイズを含むため，人手によってノイズを取り除く作業は不可欠である．また，自他や授受の動詞対のような同品詞の派生語対，同義語辞書のうち，サ変名詞と和語動詞の対に関する辞書などを実装する．

これら以外にも，与えられた表現が構成であるか否かを判別する機構も必要だが，関数を適用するための制約や表現生成後の言語モデルの適用によって，淘汰できると期待できるため，上の機能の実現を優先する．

5 おわりに

いわゆる句と呼ばれる表現には図 1 に示したような多様性があり，それらを相互に生成するような機構が頑健な言い換え生成には不可欠である．これをふまえ，本稿では，「格要素名詞句+格助詞+述語文節」からなる句に対する多様な言い換えを自動生成するために，統語的変換パターン，句生成関数，語彙関数を用いたモデルを提案し，各知識の実装状況について述べた

今後は，言い換えの対象を広げつつ，4.3 項で列挙した各機能を実現する．また，句を言い換えの処理単位とすることの有効性を検証し，文章作成支援などへの応用を検討する．

本研究の一部は次の研究費の支援を受けている：科研費基盤研究 (A)「円滑な情報伝達を支援する言語規格と言語変換技術」（課題番号：16200009，代表：佐藤理史）および科研費若手研究 (B)「文法カテゴリ交替を裏付ける語彙特性の体系化と辞書記述」（課題番号：18700143，代表：藤田篤）．

参 考 文 献

[1] 藤田篤, 乾健太郎. 語彙・構文的言い換えにおける変換誤りの分析. 情報処理学会論文誌, Vol. 44, No. 11, pp. 2826–2838, 2003.

[2] 乾健太郎, 藤田篤. 言い換え技術に関する研究動向. 自然言語処理, Vol. 11, No. 5, pp. 151–198, 2004.

[3] 加藤直樹, 藤田篤, 佐藤理史. 語末の形態的特徴に基づく日本語派生語対の収集. 2007. (in this proceedings).

[4] 国立国語研究所（編）. 分類語彙表（増補改訂版）. 大日本図書, 2004.

[5] I. Mel'čuk. Lexical functions: a tool for the description of lexical relations in a lexicon. In L. Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pp. 37–102. John Benjamin Publishing Company, 1996.

[6] 日本電子化辞書研究所. EDR 電子化辞書仕様説明書. 1995.

[7] T. Takahashi, T. Iwakura, R. Iida, A. Fujita, and K. Inui. KURA: a transfer-based lexico-structural paraphrasing engine. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 37–46, 2001.