人文科学とコンピュータ研究会　47－6
（2000．7．21）

歌集間における表現特徴の自動抽出
―部分文字列の生起頻度にみる―

竹田 正幸† 福田 智子‡ 南里 一郎*

† 九州大学大学院システム情報科学研究院 ‡ 福岡女学院大学 * 純真女子短期大学

要旨. 本稿では，文学作品，とくに和歌集から表現特徴を抽出する問題を扱う．すなわち，二人の歌人(作家)による作品を対象に，一方に比較的よく表れるが他方には表れにくい表現を特徴として取り出す．この問題は，テキストデータに対する最適パターン発見問題において，抽出するパターンを部分文字列パターンに制限したものと捉えることができる．このための有効な方法の一つとして，テキスト中に表れるパターンを「良さ」の尺度に従って降順に整列したリストを作成し，その上位部分を人間であるエキスパートの手によって吟味する，という方式が考えられる．しかし，日本語テキストには，単語間に明示的な区切りがないため，部分文字列は，単語や単語列の無意味な断片であることが多く，エキスパートの作業負担は小さくない．そこで，その負担をいかにして軽減し，作業支援を行なうかが，成功の鍵を握っているといってよい．本稿では，(1) リスト中の冗長性を除くため，ある同値関係のもとで各同値類の最長文字列である代表元(これを主要部分文字列とよぶ)のみを扱うこと，(2) 興味のある文字列の「近傍」をその生起のコンテクストと同様にして自由に閲覧するための方法を提案する．この方法を用いて，私家集からの表現特徴の抽出を行なった．得られた結果は，個々の歌人のもつ，これまで見過ごされてきた一面の発見へつながるものと期待できる．

Discovering Characteristic Expressions from Collections of Classical Japanese Poems

Masayuki Takeda† Tomoko Fukuda‡ Ichirō Nanri*

† Department of Informatics, Kyushu University

‡ Fukuoka Jo Gakuin College * Junshin Women's Junior College

Abstract. We attempt to extract characteristic expressions from literary works. That is, our problem is, given two sets of literary works written by two writers, to find expressions that distinguish the sets. It is considered as a special case of *the optimal pattern discovery* from textual data, in which only *the substring patterns* are considered. One reasonable approach is to create a list of substrings arranged in the descending order of their *goodness*, and to examine a first part of the list by a human expert. Since there is no word boundary in Japanese texts, a substring is often a fragment of a word or a phrase. How to assist the human expert is a key to success in discovery. In this paper, we propose (1) to restrict to the *prime* substrings in order to remove redundancy from the list, and (2) a way of browsing the neighbor of a focused string as well as its context. Using this method, we report successful results against two pairs of anthologies of classical Japanese poems. The extracted expressions provide a researcher clues for further investigation.

1 はじめに

文学作品の表現分析は，文学研究におけるもっとも基本的な方法の一つである．とりわけ古典和歌の場合，いわゆる「梅に鶯」に象徴されるように，歌語の組合せに規範があり，「ことば」の選択に強い規制が働いている．したがって，個々の歌人たちが，それらの規範を学びつつ，いかに自らの表現を獲得していったのかという問題は，和歌文学研究において，きわめて重要であるといえよう．そこでまず，ある歌人が，誰の歌の，どのような表現を採り入れているかという，表現の授受関係を，見出してみたくなる．

著者らは，このような考えに立って，古典和歌を対象に類似歌の自動抽出法を開発した[10]．そして，この手法を用いて，これまで指摘のなかったいくつかの和歌について，その表現の影響関係を明らかにすることに成功し，そこに，和歌文学研究における興味深い問題点を発見，解決することができた．

- 三十六歌仙に数えられる歌人，藤原兼輔の代表歌「人の親の心は闇にあらねども子を思ふ道にまどひぬるかな」が，実は，『古今集』歌の替え歌であるという事実を発見した．これまで，「子を思う親の情の率直な吐露」とのみ紹介されてきたこの歌の陰にある，兼輔の作歌の手の内が，明らかになった．

- 成立年代未詳とされてきた謎の家集『為忠集』が，実は，十五世紀 (室町時代前期) の歌人のものであることを，正徹の家集『草根集』との類似歌抽出によって，指摘した．鎌倉中期頃の家集かと指摘されてから四半世紀が経ち，その間，研究者の勘でのみ囁かれていた成立年代を確定したことになる．

このように類似歌抽出を行なう一方で，著者らはさらに，以下のような場合の表現分析の方法を，新たに案出する必要性を感じる．すなわち，歌人Aと歌人Bが，親と子，師匠と弟子といった，近い間柄にあるときである．この場合，歌人Aの影響 (あるいは指導) を歌人Bが受け，類似歌を多く詠むことは容易に推測されるため，それらの多くは，既に研究者の手によって指摘されていることも少なくない．とすると，こういった歌人の家集間に見出される類似歌を，その上に追加していくよりむしろ，その差異を明らかにすることの方が，表現分析において，重要な観点になるであろう．もし，ある表現を，歌人Aが頻繁に使用した反面，歌人Bはほとんど (もしくは全く) 用いなかったとしたら，それがまた，和歌文学研究の糸口になるかもしれないのである．そこで，本稿では，二人の歌人の家集から，その表現の頻度の違いをもとに，表現の特徴・差異を抽出することを目指す．これは，古典和歌の表現分析をする上で，著者らがこれまで行なってきた，表現の授受関係を見出すための類似歌抽出と，表裏一体をなすものである．

日本語で書かれたテキストには，単語間に明示的な区切りがないため，単語単位の生起頻度統計をとるためには，単語分割，あるいは品詞分解の作業が必要となる．この作業は容易ではなく，大変な時間と労力とを要する．たとえば，村上ら [13] は，源氏物語における助動詞の生起頻度に着目した計量分析を試みているが，源氏物語の品詞分解を行ない，タグ付きコーパスを作成する作業には，実に8年を要したという．古典和歌の場合，事情はさらに深刻で，ひとつの単語 (もしくは単語の一部) に複数の意味を担わせる「掛詞」が多用される．この技法は，単語やその切れ目の曖昧さ (ambiguity) をいわば逆手にとった技法といえる．したがって，このような場合に単語分解を一意に決定することは不可能であり，むしろ，一意に決定すべきではない．

最近になって，一部の和歌文学研究者は，単語を単位とした生起頻度統計の代わりに，テキストの部分文字列の生起頻度統計を用いることの有効性に気づき始めている．テキストの部分文字列は，必ずしも，単語やひとまとまりの単語列ではなく，多くはその一部であるが，そのことを割り引いても，利点が大きいのである．たとえば，近藤 [12] は，n グラム統計を用いた表現分析手法を提唱し，古今和歌集を対象に，歌人の性差による表現上の差異を報告している．ここで用いている n グラム統計は，n を固定できないために，本質的には部分文字列統計と同一である．近藤の研究は，日本古典文学作品の表現分析に，文字列統計手法を適用する門戸を開いたものといえる．

しかしながら，m を和歌の長さの総和とすると，すべての部分文字列の数は $O(m^2)$ である．古今集程度の比較的小さい歌集 (1,111 首) ですら，研究者が一つ一つを吟味するには，この数はあまりにも大きい．この理由により，近藤は吟味する部分文字列を，(1) 長さが3から7までであり，(2) 2回以上生起し，かつ，(3) 男性歌人だけが用いたものに限定している．本稿では，このような制限は不要であることを，文字列に関する組合せ的性質に基づいて，示すことになる．

テキストデータから「良い」パターンを見つけ出す問題は，近年関心を集めており，理論と実用の両面から研究されている [9, 7, 3, 11]．本稿で扱う問題は，この問題の特別な場合とみなすことができる．すなわち，パターンを，$*w*$ という形式をした部分文字列パターンに制限したものである．ここで，$*$ は任意の文字列と合致するワイルドカードであり，w は非空な文字列とする．いま，パターンの「良さ」に関する適切な尺度が与えられたとする．良いパターンを抽出するための有効な一つの方法として，(1) すべての部分文字列についてその「良さ」を算出し，(2) 部分文字列を「良さ」の順に並べたリストを作成し，(3) そのリストの先頭部分を調べる，という方式が考えられる．(1),(2) の二つのステップはテキストマイニングアルゴリズムによって実行されるが，(3) は，人間のエキスパートによる労力を要する．このステップは，知識発見プロセスの中で，データマイニングの後処理として必要な「得られたパターンの解釈・評価」[6] のステップに対応する．このステップをいかに行なうかが，成功の鍵を握っているといってよい．

解決すべき問題は，以下のように要約できる．

1. 人手でチェックすべき文字列のリストが冗長であること，すなわち，文字列としては異なるが，同じ位置に生起する (したがって生起頻度の等しい) 文字列が多く含まれること．
   S をテキストの有限集合とする．S 中のテキストの部分文字列の個数は $O(\|S\|^2)$ である．ここに，$\|S\|$ は，S 中のテキスト長の総和である．これは，S に対する接尾辞木 [4] の分岐節点に対応する部分文字列のみを考えることにより，$O(\|S\|)$ に抑えることができるが，その場合でも，冗長性は依然として残る．
2. リスト中の文字列は，しばしば，既に表れたものとほとんど同一であること．
   チェック中の文字列は，既出の別の文字列の一

部分でありうる．さらに，作業者がチェック中の文字列のコンテクストを見た際には，その文字列の**超文字列** (superstring) を見ており，それは，それ以降，リスト内で出くわしうる．しかし，二つの文字列が互いに，超文字列–部分文字列の関係にあるとしても，それらが同じ頻度で表れるのでなければ，同一視することはできない．したがって，この関係をユーザに明示的に示す必要がある．

3. **伝統的に用いられる KWIC (KeyWord In Context) は，チェック中の文字列の頻度が大きい場合には，効率的でないこと．**
研究者らは，単語の頻度だけでなく，その実際の用例に興味があるため，コンテクストを眺める作業は基本的かつ重要である．しかも，単語を単位とした統計を用いる場合と異なり，リスト中の文字列の多くは単語や単語列の断片であって，一見無意味な文字列であることが多い．このため，その断片を部分として含む単語や単語列を求めるために，その断片の前後にどのような文字列が続くのかを各生起ごとに調べる作業が必須である．したがって，この作業を効率的に行なう方法を開発することは，作業効率の向上の面から極めて重要である．

本稿では，上に述べた三つの問題に対する効果的な解を示す．まず，第1の問題に対して，扱う文字列を，S の主要部分文字列 (prime substrings) に限定する．ここで，主要部分文字列とは，左右両方向に頻度が変わらない限りにおいてできるだけ拡張したものをいうが，より形式的には，以下のようになる．S を非空なテキストの有限集合とする．部分文字列 x のすべての生起が，その直前に文字列 α を伴い，かつ，直後に文字列 β を伴っているとき，部分文字列 x は文字列 $\alpha x\beta$ を**含意する** (*imply*) という．ここで，α, β は空でもよい．部分文字列 x によって含意される文字列 $\alpha x\beta$ が文字列 x の**拡張**であるとは，文字列 α と文字列 β が最長のときをいう．部分文字列 x がその拡張であるとき，x は**主要**であるという．われわれの提案は，この主要な部分文字列だけを扱うことにより，作業の無駄を省こうというものである．

S の接尾辞木の分岐節点は，S の部分文字列の右拡張に対応することにふれておかなければなるまい．ここで，文字列 $x\beta$ が部分文字列 x の**右拡張**であるとは，β が，x が $x\beta$ を含意するような最長のものであるときをいう[1]．すなわち，右方向にだけ延長し，左方向には延長しない部分文字列が考慮されているのである．そのような部分文字列の個数は $O(\|S\|)$ であるが，一方，主要な部分文字列の個数も同じく $O(\|S\|)$ である．よって，O 記法の陰に隠れた定数を無視する限り，主要な部分文字列を考える価値はないようにみえる．主要な部分文字列という概念を最初に導入したのは，1984年の Blumer ら [2] であったが，現在までほとんど注目を受けていないのは，おそらくこの理由によるものであろう．しかし，主要な部分文字列の個数は，実際には，右延長の個数と比べるとかなり小さい．この違いは，特徴的表現の候補としての部分文字列の各々を吟味しなければならない研究者にとっては，まさに，死活問題である．

第2の問題に対しては，主要な部分文字列の間の超文字列–部分文字列関係をうまく図示する方法が必要である．一つの方法として，この超文字列–部分文字列関係は半順序関係であるから，半順序付けられた有限集合を図示するためのハッセ図を，グラフドローイングの技法を用いて描画することがある．しかし，グラフを何らかの制約を満たすように描画する問題の計算量は一般に大きく，ここで描画したいハッセ図のサイズはかなり大きいため，計算量の面から問題がある．さらに，文学研究者らは半順序関係やハッセ図に必ずしも親しくない．一方，ここでは，グラフの巨視的ビューは必要ではなく，局所的ビューこそが必要である．すなわち，いま着目している節点の近傍 (*neighbor*) を自由に閲覧できることが重要なのである．

第3の問題の中で述べたように，文学研究者らは部分文字列の生起のコンテクストに興味をもつ．後に述べるように，着目した部分文字列の「近傍」を閲覧することは，部分文字列のコンテクストを閲覧することと密接に関係している．そこで，第2・第3の問題の両方のために，着目した部分文字列の左と右のコンテクストをそれぞれ，木構造として図示することを提案する．この木の各節点は，ある部分文字列でラベルづけられている．研究者は，これら二つの木の節点を自由に横断することができる．コンテクストの木構造によるビューは，KWIC による伝統的なビューよりも効率的である．もちろん，KWIC と同様，着目した文字列を含むすべての文 (もしくは文の一部) が表示され，研究者の関心が別の文字列に移れば，表示する文の集合もそれにつれて変更できるような機能も備えておくべきである．

提案した方法を用いることにより，著者らは，二つの家集からその差異を特徴づける文字列を抽出し，これを人手で調べていくことで，特徴的表現の発見に成功した．すなわち，西行の『山家集』と慈円の『拾玉集』，また，藤原定家の『拾遺愚草』とその息子為家の『為家集』をそれぞれ比較し，その差異となる特徴表現を得たのである．

西行と慈円は，ともに歌僧であるとはいえ，身分や境涯は極めて対照的であった．だが，摂関家の子弟の身で僧籍に入り，俗世間の権力と関わりを断てない立場の慈円は，晩年の西行に，隠遁の志を打ち明けるなど，和歌だけではなく生き方にまで，大きな影響を受けたことが知られている．

また，為家は，歌の家として名高い御子左家の嫡流として，父定家から，厳しい指導を受けた．定家の歌の中には，為家の手本になったものも，少なからずある．

これら二組の歌集は，それぞれ，類似歌が存する必然性を備えている．そのような歌集間において，

[1] 一方，DAWG [4] は，各状態が部分文字列の左拡張に対応するような有限状態機械である．

逆に，表現の差異が抽出されたことで，それが，個々の歌人のもつ，見過ごされてきた一面の発見へとつながる可能性は，じゅうぶんに期待できる．

2 主要部分文字列

この章では，主要部分文字列の形式的定義をあたえ，その性質のいくつかを述べる．

2.1 準備

Σ を有限アルファベットとする．Σ^* の要素を文字列 (*string*) とよぶ．文字列 x が文字列 y の部分文字列 (*substring*) であるとは，文字列 u, v が存在して $y = uxv$ となるときをいう．このとき，文字列 y は文字列 x の超文字列 (*superstring*) であるという．文字列 u の長さを $|u|$ で表す．空文字列を ε で表す．すなわち，$|\varepsilon| = 0$ である．$\Sigma^+ = \Sigma^* - \{\varepsilon\}$ とおく．文字列 u の第 i 番目の文字を $u[i]$ で表す $(1 \leq i \leq |u|)$．文字列 u の部分文字列で，位置 i で始まり位置 j で終わるものを，$u[i:j]$ で表す．ただし，$1 \leq i \leq j \leq |u|$ とする．便宜上，$j < i$ なる i, j に対しては，$u[i:j] = \varepsilon$ と定めておく．文字列 w の部分文字列全体の集合を $Sub(w)$ で表す．文字列の集合 S に対して，$Sub(S) = \bigcup_{w \in S} Sub(w)$ とおく．S の文字列の長さの総和を $\|S\|$ で表し，S の濃度を $|S|$ で表す．文字列 u の文字を逆順に並べた文字列を u^R で表し，文字列の集合 S に対して，$S^R = \{u^R \mid u \in S\}$ とする．

2.2 主要部分文字列の定義

定義 1 $S = \{w_1, \ldots, w_k\} \subset \Sigma^+$ とする．$Sub(S)$ 中の任意の x に対して

$$Beginpos_S(x) = \left\{ \langle i, j \rangle \,\middle|\, \begin{array}{l} 1 \leq i \leq k, 0 \leq j \leq |w_i|, \\ x = w_i[j : j + |x| - 1] \end{array} \right\},$$

$$Endpos_S(x) = \left\{ \langle i, j \rangle \,\middle|\, \begin{array}{l} 1 \leq i \leq k, 0 \leq j \leq |w_i|, \\ x = w_i[j - |x| + 1 : j] \end{array} \right\}$$

とする．$x \notin Sub(S)$ なる任意の x については，$Beginpos_S(x) = Endpos_S(x) = \emptyset$ とする．

例えば，$w_1 = ababc$, $w_2 = abcab$ としよう．このとき，$Beginpos_S(a) = Beginpos_S(ab) = \{\langle 1,1\rangle, \langle 1,3\rangle, \langle 2,1\rangle, \langle 2,4\rangle\}$, $Endpos_S(b) = Endpos_S(ab) = \{\langle 1,2\rangle, \langle 1,4\rangle, \langle 2,2\rangle, \langle 2,5\rangle\}$ である．

以後，集合 S を略し，単に $Beginpos$, $Endpos$ のように書くことにする．

定義 2 x と y を Σ^* の任意の文字列とする．$Beginpos(x) = Beginpos(y)$ であることを $x \equiv_R y$ で表し，$Endpos(x) = Endpos(y)$ であるときに $x \equiv_L y$ で表す．関係 $\equiv_R$ と $\equiv_L$ はともに同値関係である．文字列 $x \in \Sigma^*$ の $\equiv_R$ に関する同値類を $[x]_{\equiv_R}$ で表し，$\equiv_L$ に関する同値類を $[x]_{\equiv_L}$ で表す．S の部分列でないすべての文字列から成る同値類を退化 (*degenerate*) 同値類とよび，それ以外の同値類を非退化 (*nondegenerate*) であるという．

もし，$S = \{ababc, abab\}$ ならば，$[a]_{\equiv_R} = \{a, ab\}$, $[c]_{\equiv_L} = \{c, bc, abc\}$ である．

$\equiv_R$ の定義より，もし，x と y が $\equiv_R$ のもとで同一の非退化同値類に属するならば，x が y の接尾辞であるか，その逆であるかのいずれかが成り立つ．したがって，$\equiv_R$ の非退化同値類は，それぞれ，唯一つの最長要素をもつ．同様の議論が $\equiv_L$ に対しても成立する．

定義 3 $Sub(S)$ の任意の x に対して，$[x]_{\equiv_R}$ と $[x]_{\equiv_L}$ 中の最長要素を，それぞれ，$\overrightarrow{x}$ と $\overleftarrow{x}$ で表す．

$Sub(S)$ の任意の文字列 x に対して，$\overleftarrow{x} = \alpha x$, $\overrightarrow{x} = x\beta$ となる文字列 α, β が，それぞれ，一意的に存在する．上の例でいえば，$\overleftarrow{\varepsilon} = \overrightarrow{\varepsilon} = \varepsilon$, $\overrightarrow{a} = ab$, $\overleftarrow{b} = ab$, $\overleftarrow{c} = abc$, $\overrightarrow{c} = c$, $\overrightarrow{cd} = cab$, $\overleftarrow{ca} = abca$ となる．

定義 4 $Sub(S)$ の任意の x に対して，$\overleftrightarrow{x} = \alpha x \beta$ とする．ここで，α と β は，$\overleftarrow{x} = \alpha x$ と $\overrightarrow{x} = x\beta$ を満たす文字列である．

上の例でいえば，$\overleftrightarrow{\varepsilon} = \varepsilon$, $\overleftrightarrow{a} = ab$, $\overleftrightarrow{ab} = ab$, $\overleftrightarrow{c} = abc$, $\overleftrightarrow{cd} = abcab$ である．

定義 5 文字列 x と y が S に関して等価であるとは，

1. $x \notin Sub(S)$ かつ $y \notin Sub(S)$, もしくは，
2. $x, y \in Sub(S)$ かつ $\overleftrightarrow{x} = \overleftrightarrow{y}$

であるときをいい，このことを $x \equiv y$ と表す．同値関係 $\equiv$ のもとで，x の属する同値類を $[x]_{\equiv}$ で表す．

$Sub(S)$ の任意の x に対して，文字列 $\overleftrightarrow{x}$ は，$[x]_{\equiv}$ の最長要素であることに注意されたい．直感的には，$\overleftrightarrow{x} = \alpha x \beta$ であることは，以下のことを意味している．

- S において x が生起するときはいつでも，その直前に α があり，かつ，その直後に β がある．
- 文字列 α と β は，そのような文字列の中で最も長いものである．

さて，これで主要部分文字列を定義するための準備が整った．

定義 6 $Sub(S)$ 中の x が主要 (*prime*) であるとは $\overleftrightarrow{x} = x$ であるときをいう．

補題 1 (Blumer et al. (1987)) 同値関係 $\equiv$ は，関係 $\equiv_L \cup \equiv_R$ の推移的閉包である．

この補題からただちに，任意の $x \in Sub(S)$ について $\overleftrightarrow{x} = \overleftrightarrow{(\overrightarrow{x})} = \overleftrightarrow{(\overleftarrow{x})}$ であることがわかる．

2.3 主要部分文字列の性質

S 中の文字列のすべての部分文字列全体の個数は $O(\|S\|^2)$ である．もし，$\overrightarrow{x} = x$ となるような部分文字列 x だけに限定すれば，個数を $O(\|S\|)$ に抑えることができる．実際，接尾辞木は，この性質を利用したデータ構造である．同様に，DAWG は，任意の部分列 x を $\overleftrightarrow{x}$ と同一視することにより，その領域を $O(\|S\|)$ に抑えた有限状態機械である．S の主要部分文字列の個数も，同じく $O(\|S\|)$ であるから，主要部分文字列だけを考えることの利点はないかのように見えるだろう．しかし，現実の応用においては，ユーザは主要でない部分文字列をチェックしないでよいわけであるから，これは大きな利点である．次の補題は，より厳密なバウンドをあたえている．

補題 2 (Blumer et al. (1987)) $\|S\| > 1$ と仮定せよ．同値関係 $\equiv_R (\equiv_L)$ のもとで，非退化同値類の個数は，高々 $2\|S\| - 1$ である．また，同値関係 $\equiv$ のもとでの非退化同値類の個数は高々 $\|S\| + |S|$ である．

この補題から，$\overrightarrow{x} = x$ となる部分文字列 x の個数，あるいは，$\overleftrightarrow{x} = x$ となる部分文字列 x の個数は，いずれも，高々 $2\|S\| - 1$ であり，一方，S の主要部分文字列の個数は高々 $\|S\| + |S|$ であることがわかる．

さて，主要部分文字列全体の集合に自然な半順序関係 '$\succeq$' を導入しよう．

定義 7 $Prime(S)$ を S の主要部分文字列全体の集合とする．すなわち，$Prime(S) = \{\overleftrightarrow{x} \mid x \in Sub(S)\}$ である．x, y を $Prime(S)$ の任意の要素とする．x が y の部分文字列であるとき，かつ，そのときに限り，$x \succeq y$ と書く．

すなわち，半順序関係 '$\succeq$' は，$Prime(S)$ 上の超文字列–部分文字列関係そのものである．対 $\langle Prime(S), \succeq \rangle$ は，以下のような半順序集合である．

- 最大要素は ε である．
- 極小要素は S 中のすべての文字列である．

$x \succeq y$ かつ $x \neq y$ であることを，$x \succ y$ と表すことにしよう．

定義 8 $Prime(S)$ の任意の x, y に対して，$x \succ y$ であり，かつ，$x \succ z \succ y$ となるような z が $Prime(S)$ に存在しないとき，$x \triangleright y$ と書き，y を x の**直接後続者** (*direct successor*) といい，x を y の**直接先行者** (*direct predecessor*) という．

ハッセ図は有限半順序集合を図示したものとしてよく知られている．半順序集合 $\langle Prime(S), \succeq \rangle$ に対するハッセ図は，$V = Prime(S)$ を頂点の集合とし，$E = \{(x, y) \in V \times V \mid x \triangleright y\}$ を辺の集合とする DAG $H(S) = (V, E)$ である．

補題 3 ハッセ図 $H(S)$ は，非空な文字列の有限集合 S から，$O(\|S\|)$ 時間・領域で構成できる．

証明． 対称コンパクト DAWG(symmetric compact DAWG) [2] とよばれるデータ構造から直接得ることができるが，これは $O(\|S\|)$ 時間・領域で構成可能である．∎

3 特徴的部分文字列の抽出

文字列データから「良い」パターンを得る方法として，以下のような方法が考えられる．

1. パターンの「良さ」に関する適切な尺度を選択する．
2. テキスト中に表れるパターンを「良さ」の順に整列したリストを作成する．
3. 当該分野の専門家が，このリストの先頭から調べていって，有用なパターンを得る．

著者らの経験では，2 で得られたパターンの多くは，無意味であったり，自明なものであるなど，必ずしも有用でないことが多い．したがって，当該分野の専門家がいかに 3 を行なうかが，成功の鍵を握っているいえる．このことは，また，Fayyad ら [6] が，知識発見のプロセスにおいて，データマイニングの後処理である「解釈・評価」を重要なステップとしてあげていることとも符合する．

本稿では，扱うパターンを部分文字列パターン (*substring patterns*) とよぶものに制限してパターンの抽出を考える．すなわち，$*$ を任意の文字列に合致するワイルドカード，w を非空な文字列としたとき，$*w*$ という形式のパターンのみを考えるのである．いま，パターンの「良さ」は，その頻度にのみ依存して決まるものと仮定すれば，文字列 w を主要部分文字列に限定することができる．これにより，ステップ 3 における作業の無駄を省くことができる．

しかし，主要部分文字列に限定したとしても，リスト中の文字列は，互いに無関係ではあり得ず，超文字列–部分文字列の関係にあることが多い．この関係をユーザに明示的に示すため，指定したパターンの近傍を自由に閲覧するための機構をもたせる必要がある．

3.1 主要部分文字列上の半順序関係の閲覧

S の主要部分文字列間の超文字列–部分文字列関係をユーザにどのように見せるかについて考えよう．一つの方法は，グラフドローイングの技法 [1] を用いてハッセ図を画面上に描画することであろう．しかし，何らかの美的観点からくる制約条件を満たすようにグラフを描画する問題の計算量は一般に大きい．ここで描画したいハッセ図はサイズが大きいため，多大な計算量を要する．これに加えて，ハッセ図は，文学研究者になじみにくいものである．そもそも，ここでは巨視的ビューが必要なのではなく，局所的なビューがあればよい．要するに，現在着目

している文字列の近傍，すなわち，直接先行者や直接後続者をみることができればよいのである．主要部分文字列 x の直接先行者は x の真の部分文字列であるから，直接先行者の数は相対的には少ない．しかし，直接後続者の数は膨大に成りうる．

文学研究者らは，現在着目している主要部分文字列の(直接)後続者がみたいというよりも，その文字列のコンテクスト，すなわち，左右それぞれにどのような文字列が続くのか，ということに興味がある．左コンテクストと右コンテクストを，以下のように定めよう．

定義 9 $Sub(S)$ の要素 x に対して，

$$\begin{aligned} LC(x) &= \{u \mid u, v \in \Sigma^* \text{ and } uxv \in S\}, \\ RC(x) &= \{v \mid u, v \in \Sigma^* \text{ and } uxv \in S\}. \end{aligned}$$

集合 $LC(x)$ と $RC(x)$ を，それぞれ，x の S に関する**左コンテクスト** (*left context*)，**右コンテクスト** (*right context*) とよぶ．

定義 10 $Sub(S)$ の任意の x に対して，以下のようにおく．

$$\begin{aligned} ILC(x) &= \{\alpha \in \Sigma^* \mid \alpha x = \overleftarrow{ax} \text{ for some } a \text{ in } \Sigma\}, \\ IRC(x) &= \{\beta \in \Sigma^* \mid x\beta = \overrightarrow{x\beta} \text{ for some } a \text{ in } \Sigma\}. \end{aligned}$$

集合 $ILC(x)$ と $IRC(x)$ を，それぞれ，x の S に関する**直接左コンテクスト** (*immediate left context*) と**直接右コンテクスト** (*immediate right context*) とよぶ．

もし，$x = \overrightarrow{x}$ ならば，集合 $RC(x)$ は以下のような木構造をなす．

- 根は x である．
- すべての節点 y に対し，$y = \overrightarrow{y}$ が成り立つ．
- 節点 y の子は，$y \cdot IRC(y)$ の要素である．
- 節点 y から節点 $\overrightarrow{ya}$ への辺のラベルは，$\overrightarrow{ya} = ya\beta$ となる文字列 $a\beta$ である．

このように，節点 y から出る辺のラベルの集合は，$IRC(y)$ に等しい．$RC(x)$ に対する木は S に対する接尾辞木の部分木で根が x であるものと等しい．この木を文字列 x の *RC*-木とよぶ．同様に，もし $x = \overleftarrow{x}$ ならば，集合 $LC(x)$ は木構造をなすが，この木は，S^R に対する接尾辞木の部分木で x^R を根とするものに等しい．この木を x の *LC*-木とよぶ．

次の補題は，主要部分文字列に対して，その直接左コンテクストと直接右コンテクストを考えることが，その直接後続者を考えることと等価であることを示している．

補題 4 x を主要部分文字列とする．このとき，x の *RC*-木における子と *LC*-木における子の集合は，x の直接後続者の集合に等しい．

証明. 任意の文字列 $x, y \in Prime(S)$ に対して，$x \triangleright y$ が成り立つための必要十分条件が，ある文字 $a \in \Sigma$ があって $\overleftarrow{ax} = y$ または $\overrightarrow{xa} = y$ となることであることを示せばよい．∎

以上のことから，ハッセ図を図示する代わりに，上記の木構造を表示することにする．

RC-木と *LC*-木の節点は，主要部分文字列でないことがある．すなわち，これらの木において，$x = \overrightarrow{x}$ もしくは $x = \overleftarrow{x}$ であるが，必ずしも $x = \overleftrightarrow{x}$ ではないのである．最近，Maaß [8] は，文字列 T に対する**接辞木** (*affix tree*) を構築するための線形時間オンラインアルゴリズムを示したが，この接辞木は，T に対する接尾辞木と T^R に対する接尾辞木 (すなわち，T に対する接頭辞木) の両方を兼ね備えたものである．このアルゴリズムを複数テキストに拡張することにより，*RC*-木と *LC*-木のためのデータ構造を得ることができる．

3.2 文学作品からの発見のためのシナリオ

文学作品から表現特徴を得るためのシナリオとして，以下のような手順を考えよう．

1. テキスト文字列の二つの集合 *Pos* と *Neg* を選ぶ．
2. テキスト部分文字列の「良さ」の尺度を選ぶ．この尺度は，各文字列の集合 *Pos* と *Neg* における頻度にのみ依存するものとする．
3. $S = Pos \cup Neg$ の主要部分文字列を，「良さ」の順に並べたリストを作成する．
4. 以下の手続きを繰り返して，表現特徴を得る．
   (a) リスト中のある主要文字列 x に着目し，その頻度等をしらべる．
   (b) x の *LC*-木と *RC*-木の節点を閲覧し，ある $\gamma \in \Sigma^+$ に対して $x' = \overleftarrow{\gamma x}$ または $x' = \overrightarrow{x\gamma}$ となるような x' を選ぶ．
   (c) もし x' に興味をもてば，(a) に戻って $x := \overleftrightarrow{x'}$ とする．

このシナリオに沿ってテキストマイニングを行なうには，以下の関数が必要である．

- 与えられた文字列 $x \in Sub(S)$ に対してその *Pos* 中および *Neg* 中における生起頻度を返す関数 $Freq_{Pos}(x)$ と $Freq_{Neg}(x)$．
- 与えられた文字列 $x \in Sub(S)$ に対して文字列 $\overleftarrow{x}$，$\overrightarrow{x}$，$\overleftrightarrow{x}$ をそれぞれ返す関数 $LeftExt(x)$，$RightExt(x)$，$Ext(x)$．
- 与えられた文字列 $x \in Sub(S)$ に対してその直接左コンテクストと直接右コンテクストをそれぞれ返す関数 $ILC(x)$ と $IRC(x)$．
- 与えられた文字列 $x \in Prime(S)$ に対してその直接先行者のリストと直接後続者のリストをそれぞれ返す関数 $DirectPred(x)$ と $DirectSucc(x)$．

補題 5 上の関数は，$O(\|S\|)$ 時間・領域を用いて構成でき，$O(|x|)$ 時間で応答する (答の出力に要する時間を除く).

4 実験

西行と慈円の家集，および，藤原定家とその息子為家の家集を，それぞれ比較し，その差異となる表現特徴の抽出を試みた．また，参考のために，散文作品の例として，『源氏物語』全五十四帖のうち，宇治十帖とそれ以外についての比較も行なった．

4.1 「良さ」の尺度

Pos と Neg をテキストの集合とし，$S = Pos \cup Neg$ とおく．w を任意の文字列とする．Pos と Neg 中の文字列のうち，文字列 w を部分文字列として含むものの個数を，それぞれ，P_1 と N_1 で表す．$P_0 = |Pos| - P_1$，$N_0 = |Neg| - N_1$ とおく．目標関数 $G(w; Pos, Neg)$ を以下のように定める．

$$G(w; Pos, Neg) = \frac{P_1+N_1}{|S|} \cdot \psi\left(\frac{P_1}{P_1+N_1}\right) + \frac{P_0+N_0}{|S|} \cdot \psi\left(\frac{P_0}{P_0+N_0}\right).$$

ここに，$\psi(r)$ は不均衡度関数 (*impurity function*)[5] であり，ここでは，エントロピー関数，すなわち，

$$\psi(r) = -r\log r - (1-r)\log(1-r)$$

を用いた．与えられた Pos と Neg に対して $G(w; Pos, Neg)$ の値を最小にする文字列 w を「最良」の文字列と考え，$S = Pos \cup Neg$ の主要部分文字列をこの値の昇順に整列したリストを作成した．

4.2 使用したテキスト

次の古典文学作品に対して実験を行なった．

(A) 西行 (1118-1190) の『山家集』と慈円 (1155--1225) の『拾玉集』．慈円が，歌僧として，西行に強い影響を受けたことはよく知られている．実際，慈円は，西行の歌を踏まえた歌を，少なからず作っている．

(B) 藤原定家 (1162--1241) の『拾遺愚草』と藤原為家 (1198--1275) の『為家集』．定家は，歌人として，また，歌学者として，日本古典文学史上，最も偉大な足跡を残す人物である．その息子為家は，父定家の歌と歌論の両方から，直接的に影響を受けている．

(C) 『源氏物語』．全五十四帖のうち，末尾の宇治十帖については，作者が紫式部ではないとする説がある．そこで，ここでは，宇治十帖とそれ以外について，比較を行なった．

表 1 は，(A), (B), (C) のテキストに対して，三つの同値関係 $\equiv_R$, $\equiv_L$, $\equiv$ のもとでの非退化同値類の個数を示している．この結果から，主要部分文字列に制限することで，人間が調べるべき部分文字列の候補の数が劇的に低減することがわかる．このように，主要部分文字列の概念を導入することによって，部分文字列統計に基づくテキスト分析が，きわめて現実的な手段となるのである．

4.3 得られた表現特徴

表 2 に，主要部分文字列を「良さ」の順に並べた上位 20 個を (A),(B) について示す．

得られたリストの上位部分から，たとえば，以下のような事項が見て取れる．

- 『拾玉集』には，「のりのみち」「まことのみち」「のりのはな」という仏教語が，16～20 首まとまってみられるが，『山家集』にはない．
- 「～をいかにせむ」(多くは「身をいかにせむ」) という表現は，『山家集』にはなく，『拾玉集』に 21 例ある．
  このような表現が，慈円の歌に見られ，西行のそれにないということは，彼らの仏教との関わり方，ひいては身の処し方の相違が，和歌表現にあらわれているようで，興味深い．
- 『為家集』には，「おいのねさめ」「おいのなみた」という表現が多数用いられているのに対し，『拾遺愚草』には用例がない．
  為家が，こういった「老い」に関わる表現を多く用いているからといって，彼が父定家より長寿であったわけではない (定家は 80 歳，為家は 78 歳で没)．そもそも，『新編国歌大観』全体を見ても，「おいのねざめ」「おいのなみだ」の用例が最も多く見出されるのが，『為家集』なのである[2]．したがって，これらの表現は，彼の和歌表現を絶対的に特徴づけるものであると言えよう．なお，為家の晩年には，歌の家の相続をめぐる争いが起こり，結果，次の代には二条・京極・冷泉の三家が分立するという事態になった．為家という，ひとりの跡取りを決め得た定家とは，対照的である．
- 『源氏物語』から得た主要文字列には，「いかにもいかにも」「しきわさかな」といった，宇治十帖の作者，もしくは登場人物の言葉遣いの癖らしきものもみられた．だが，ほとんどの文字列が，人名，官職名，地名などの固有名詞であって，物語の舞台や登場人物の違いによるところが大きく，いわゆる発見とはおよそ程遠い．これらの文字列の多くは，$P_1 = 0$ または $N_1 = 0$ となるものを除去すれば除くことができるが，

[2] 角川書店『歌ことば歌枕大辞典』(平成 11 年 5 月) には，「老いの寝覚め」が立項されている (三村晃功氏) が，この点についての言及はない．

表 1: 同値関係 $\equiv_R$, $\equiv_L$, $\equiv$ のもとでの非退化同値類の数の比較

テキスト					非退化同値類の個数		
Pos	*Neg*	$\|S\|$	$\|\|S\|\|$	$\|Sub(S)\|$	$\equiv_R$	$\equiv_L$	$\equiv$
山家集 (1,552 首)	拾玉集 (5,803 首)	7,355	229,728	2,817,436	259,576	265,238	65,149
拾遺愚草 (2,985 首)	為家集 (2,101 首)	5,086	158,290	1,989,446	183,358	185,987	46,288
字治十帖以外	字治十帖	54	859,796	1,493,709,707	1,182,601	1,181,439	251,343

表 2: 二組の私家集から得た上位 20 の主要部分文字列

(A) 山家集 vs 拾玉集

順位	G	P_1	N_1	部分文字列
1	0.5115	26	352	はるの
2	0.5116	38	422	るの
3	0.5120	919	2822	て
4	0.5120	18	277	みよ
5	0.5124	41	28	ここち
6	0.5124	44	33	こち
7	0.5127	60	495	そら
8	0.5128	7	166	みよし
9	0.5128	26	297	のそら
10	0.5130	3	122	すみよ
11	0.5131	7	152	みよしの
12	0.5132	52	422	のそ
13	0.5132	3	114	すみよし
14	0.5133	7	147	のりの
15	0.5133	101	178	まし
16	0.5134	487	2281	よ
17	0.5134	53	418	ゆふ
18	0.5134	8	151	つかせ
19	0.5135	0	67	のこる
20	0.5135	117	722	はる

(B) 拾遺愚草 vs 為家集

順位	G	P_1	N_1	部分文字列
1	0.6668	14	103	おい
2	0.6685	2	63	おいの
3	0.6719	7	55	をくら
4	0.6731	246	74	そら
5	0.6731	16	63	をく
6	0.6735	4	38	にける
7	0.6736	109	168	こそ
8	0.6738	3	34	くらやま
9	0.6738	54	107	いの
10	0.6739	3	33	をくらやま
11	0.6740	0	23	おいのね
12	0.6741	69	7	なかめ
13	0.6744	166	47	のいろ
14	0.6745	2	27	らのやま
15	0.6747	0	19	おいのねさめ
16	0.6747	88	16	そての
17	0.6748	292	115	いろ
18	0.6751	54	6	きえ
19	0.6751	8	35	くらや
20	0.6751	67	106	にけ

同時に，ほかの重要な文字列をも除いてしまう危険がある．したがって，散文作品に対しては，有効なフィルタリング手法を考える必要がある．

5 おわりに

本稿で示した表現特徴抽出法は，和歌に対しては，研究者の興味を引く結果を出すことができたが，散文に対しては問題を残した．ここでは，主要部分文字列すべてを表現特徴の候補としたが，散文の場合，何らかのフィルタリングが必要と思われる．文献[11]において，著者らは，「ふし」とよぶ特徴パターンを和歌から抽出することを試みたが，これは，パターンの定数部分を，付属語列に限定したものであった．散文の場合に適したフィルタリング手法をもとめることは今後の課題である．

参考文献

[1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

[2] A. Blumer, J. Blumer, D. Haussler, R. Mcconnell, and A. Ehrenfeucht. Complete inverted files for efficient text retrieval and analysis. *J. ACM*, Vol. 34, No. 3, pp. 578–595, 1987. Preliminary version in: STOC'84.

[3] A. Brāzma, E. Ukkonen, and J. Vilo. Discovering unbounded unions of regular pattern languages from positive examples. In *Proc. 7th International Symposium on Algorithms and Computation (ISSAC'96)*, pp. 95–104, 1996.

[4] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.

[5] L. Devroye, L. Gy orfi, and G. Lugosi. *A Probablistic Theory of Pattern Recognition*. Springer, 1997.

[6] U. M. Fayyad, G. P.-Shapiro, and P. Smyth. From data mining to knowledge discovery: an overview. In *Advances in Knowledge Discovery and Data Mining*, pp. 1–34. The AAAI Press, 1996.

[7] R. Fujino, H. Arimura, and S. Arikawa. Discovering unbounded and ordered phrase association patterns for text mining. In *PAKDD2000*, LNAI 1805, pp. 281–293. Springer-Verlag, 2000.

[8] M. G. Maaß. Linear bidirectional on-line construction of affix trees. In *Proc. 11th Ann. Symp. Combinatorial Pattern Matching*, 2000. (to appear).

[9] S. Shimozono, H. Arimura, and S. Arikawa. Efficient discovery of optimal word-association patterns in large databases. *New Gener. Comput.*, Vol. 18, No. 1, pp. 49–60, 2000.

[10] K. Tamari, M. Yamasaki, T. Kida, M. Takeda, T. Fukuda, and I. Nanri. Discovering poetic allusion in anthologies of classical Japaneses poems. In *Proc. 2nd Int. Conf. Discovery Science*, LNAI 1721, pp. 128–138. Springer-Verlag, 1999.

[11] M. Yamasaki, M. Takeda, T. Fukuda, and I. Nanri. Discovering characteristic patterns from collections of classical Japanese poems. *New Gener. Comput.*, Vol. 18, No. 1, pp. 61–73, 2000. Preliminary version in: DS'98, LNAI 1532.

[12] 近藤みゆき. n グラム統計を用いた文字列分析による日本古典文学の研究. 千葉大学『人文研究』, No. 29, pp. 187–238, 2000.

[13] 村上征勝, 今西祐一郎. 源氏物語の助動詞の計量分析. 情報処理学会論文誌, Vol. 40, No. 3, pp. 774–782, 1999.