文章の類似性の近似計算

An Approximate Sentence Matching Algorithm

井宮 淳　関矢哲也　西田文彦*　市川 熹
Atsushi IMIYA　Tetsuya SEKIYA　Fumihiko NISHIDA　Akira ICHIKAWA
千葉大学工学部情報工学科
Dept. of Information and Computer Sciences, Chiba University

Abstract This paper proposes an on-line matching algorithm for spoken sentences which runs in real time. Our new algorithm is divided into two stages. At first, we transform a time-signal into a tree of the sentence by detcting local minima of short-time Fourier spectrum along the time axis. We introduce a real time parallel algorithm based on randamized Hough transform for this transformation. Secondly, we transform trees to regular trees by adding dummy nodes to express trees as lists. Thus, editing distance of trees is computed recursively by using a lists matching technique.

1. まえがき

文章は単語の集合の要素を文法規則に従ってある概念を表すように並べた配列である．個々の単語の音の強弱・上がり下がりであるアクセント成分と，文章全体の音の変化に関する情報であるイントネーション成分などとの2つの情報から成るプロソディが，発話された文章には含まれている．プロソディは発話文章の言葉としてのまとまりを表現しており，文章全体の構造はプロソディに反映されていると考えることができる．したがって，プロソディを利用すれば，発話文章の信号列から概念としてある程度まとまった部分を記号列として抽出できることがわかる．

プロソディの主要な情報は物理的には発話音声の周波数分布の時間変化に現れる．また，発話音声の周波数分布の時間変化が，区分線形関数でよく近似できることが知られている [1]．そこで以下では，まず確率化 Hough 変換 [2] による直線抽出を利用して周波数の時間変化の変曲点を抽出し，発話文章の信号列から概念としてある程度まとまった部分を実時間で抽出する手法について述べる．ついで，抽出された情報を基に発話文章から文章の木を生成し，木構造を利用しては発話された文章間の類似性を判定する高速算法を提案する．

2. 発話文章の記号化と構造化

本章では，多重解像度解析 [3] によって音声信号の短時間フーリエスペクトルのピーク値の時間変化の変曲点を抽出し，時間軸に沿って音声信号を分割することによって，音声信号から文章の木を実時間で抽出する算法を導く．

2.1. 発話文章の記号化

発話された文章を表す時間信号を $f(t)$ とする．$f(t)$ の短時間フーリエ変換 $F(\omega, t)$ は，

*現在: 東京工業大学大学院総合理工学研究科

$$F(\omega,t)=\frac{1}{w}\int_{-\infty}^{\infty}w(\tau-t)f(\tau)e^{-i\omega\tau}d\tau \quad (1)$$

で与えられる．ここで，$w(t)$ は窓関数と呼ばれ，正の定数 a に対して

$$w(-\tau) = w(\tau) \quad (2)$$
$$w(\tau) = 0,\ |\tau|>a \quad (3)$$

を満たす正値関数であり，規格化定数 wは

$$w=\int_{-a}^{a}w(t)dt \quad (4)$$

で与えられる．

周波数軸上での通過帯域をΩの中での短時間フーリエパワーのピーク値の時間変化 $p(t)$ は

$$p(t)=\max_{\omega\in\Omega}(\,|F(\omega,t)|\,) \quad (5)$$

で与えられる．

関数 $p(t)$ は区間線形関数によって良く近似できることが知られている [1]．そこで，

$$p(t)=a_it+b_i,\ t\in I_i \quad (6)$$

と置くことにする．ただし，

$$I_i=[t_i,t_{i+1}],\ t_1=0 \quad (7)$$

である．直線の傾き a_iは区間 I_iにおいて発話された音声の時間的な減衰・増幅を表す定数である．したがって，区間 I_iに対応する単語列 s_iが一塊の概念を表すことになる．

s_iを決定するためには，$p(t)$ の形状と変曲点とを時系列情報から決定する必要がある．ここでは，付録に示す確率化 Hough 変換 [2] を利用した以下の算法によって区間と直線の傾きとを同時に抽出する．ただし，$p(t)$ の変曲点を$\boldsymbol{p}_i=(t_i,\omega_i)^T$ とする．

1. $i:=1$, m, α, βを設定.
2. $0<t<\tau$である点の集合を $P(\tau)$ とする．
3. (ω,t) : $t>\tau$と $P(\tau)$ の点とで，並列に Hought 変換を計算する．
4. $a-b$平面上のカウンターの最大値を (a_i,b_i) とする．
5. A_iから離れた場所に，m 個以上 (a_τ,b_τ) が出現すれば，

$$P(\tau):=P(\tau)\setminus(P(\tau)\bigcup L(a_1,b_1;\varepsilon))$$

として，(a_i,b_i) と$\boldsymbol{p}_i$とを出力し，3へもどる．
そうでなければ，終了．

ただし，Hough 変換の投票領域を

$$A_i=[a_i-\alpha,a_1+\alpha]\times[b_i-\beta,b_1+\beta] \quad (8)$$

とした．

2.2. 発話文章の構造化

さて，$f(t)$ から $p(t)$ を介して抽出される区間列 $\{I_i\}_{i=1}^n$ に対応する記号列を$\boldsymbol{s}$

$$\boldsymbol{s}=\langle s_1,\ s_2,\cdots,s_m\rangle \quad (9)$$

とする．ここでは，$\boldsymbol{s}$から木構造を逐次抽出する算法を示す．

区間

$$I_{ij}=\bigcup_{k=i}^{j}I_k \quad (10)$$

において，変曲点の変位の値

$$\delta(k)=|\omega_{k+1}-\omega_k| \quad (11)$$

があまり変化しない場合は，$s_is_{i;1}\cdots s_j$ を一塊の記号列と考えることにする．したがって，処理

1. γを設定
2. $\boldsymbol{s}=\langle\quad\rangle$, $i:=1$.
3. $i<j$に対して，$\delta(i+1)/\delta(i)>\gamma$ ならば，

$$\boldsymbol{s}_{ij}=\langle s_i,\ s_{i+2},\cdots,s_j\rangle$$

とする．

4. $\boldsymbol{s}:=\langle\boldsymbol{s},\boldsymbol{s}_{ij}\rangle$,
$i:=j+1$ として，3 にもどる．

によって，$\boldsymbol{s}$をいくつかの部分配列の配列

$$\boldsymbol{s}=\langle\boldsymbol{s}_{1i},\boldsymbol{s}_{i+1\,j},\cdots,\boldsymbol{s}_{k+1\,n}\rangle \quad (12)$$

として表すことができる．

$\gamma_i > \gamma_{i+1}$ に対する部分配列をそれぞれs^i, $\{s_j^{i+1}\}_{j=1}^n$とする．このとき，

$$s^i = \langle s_1^{i+1}, s_2^{i+1}, \cdots s_n^{i+1} \rangle \tag{13}$$

となっていれば，

$$s^i \prec s_j^{i+1}, j = 1, 2, \cdots, n \tag{14}$$

と書くことにする．2項関係$\prec$は順序構造をなし，s_j^{i+1}がs^i の部分列で無い場合にはこの順序関係は成立しない．したがって，s^iを根とし，s_j^{i+1}を枝とする木を考えることができる．すなわち，$\gamma_1 > \gamma_2 > \cdots \gamma_m$ によって抽出される配列s_j^iの要素を，関係$\prec$によって統合すれば，記号列の間に木構造を決定できる．

抽出される木構造の葉だけに記号を残した木を音声信号 $f(t)$ の決める発話文章の木と呼ぶことにする．なお，各γ_iに対する処理は同時に総て並列に実行することができる．したがって，音声信号からオンラインで文章の木を抽出できることになる．この処理は，γ_iを媒介変数として変数ωに対して $p(t)$ を多重解像度解析 [3] して $p(t)$ の山や谷を抽出することになっている．

5段の枝別れのある2分木の葉の総数は32である．各分岐点での枝別れの数が多くなれば，30程度の葉を持つ木の段数は 5 より少なくなる．一方，人間は，30 以上の言語要素のある文語文章は理解に時間がかかると言われている．発話文章にもこの経験則を適用して考えれば，最大で5段の枝別れのある文章の木を発話文章から決定すれば良いことになる．したがって，5個程度の異なる定数γ_i に対して，並列に上の算法を実行すれば発話文章から文章の木を生成できることになる．図1に多重解像度解析による文章の木の抽出の概念を示す．

3. 文章の類似性の近似計算

本章では，2. において音声信号から決定した文章の木を利用して文章間の類似性を高速に判定する算法を提案する．

3.1. 木の間の近似変換

部分木を a_i, 枝点のラベルの集合をΣとすれば，木は

$$a_i ::= l | a_1 a_2 \cdots a_n,\ l \in \Sigma \tag{15}$$

と表すことができる．

2. の算法において抽出される文章の木では，一般にそれぞれの枝点からでる枝の数は一定ではない．そこで，木Tの部分木の中で最大の枝分かれの数を $m \geq 1$ とし，規則

$$a_i ::= a_i + +d \text{ s.t. } |d| = m - |a_i| \tag{16}$$

によってダミーの葉のリスト dを付加すれば，総ての木を正則木に変換することができる．ただし，$a + +b$ はリスト a の後尾へのリスト b の付加を表し，$|a|$ はリスト a の要素の数である．図2に正則木への変換の概念を示す．

木Tを上の変換によって正則木に変換した木をT^mとする．そして，A, Bの間の距離を

$$D(A, B) = D(A^m, B^n) \tag{17}$$

とする．木を正則木に変換にすることによって総ての部分木の間の距離の計算を，同じ規模の配列の間の距離の計算によって計算できる．

さらに，文章の木には根がある．そこで，文章の木の頂点に根に 0, 根の i 番目の子に i, 根の i番目の子の j番目の子に ij, によって枝点に番号を付けることにする [4].

そして以下では

T :木 T全体を表す.

T_i :T の i によって示される頂点を根とした部分木. ただし, $T = T_0$.

t_i :T の i によって示される頂点の要素. t_i の j 番目の子の要素は t_{ij} で表すことができる。

$n_i(T)$:T の i によって示される頂点とその子:

$$n_i(T) = t_i[T_{i1}\ T_{i2}\ \cdots\ T_{im}]$$

とする.

さて，以上の変換の後で木の距離を計算するための基本変換として，子部分木の挿入，削除，並べ換え，枝点要素の置換，の4種類の変換を考えることにする [5]. これらの変換の具体的な操作は以下のようになる．

頂点 $n_i(T)$ の要素 t_i と子部分木 T_{ik} の間に別の木 S を挿入する場合,

$$n_i(T) = t_i[T_{i1}\ T_{i2}\ \cdots\ T_{ik}\ \cdots\ T_{im}] \quad (18)$$

の T_{ik} のところに、別の木 S を挿入すると,

$$\begin{aligned} n_j(S) &= t_j[\cdots\ T_{ik}\ \cdots] && (19)\\ n_i(T) &= t_i[T_{i1}\ T_{i2}\ \cdots \\ &\qquad T_{i(k-1)}\ S\ T_{i(k+1)}\ \cdots\ T_{im}] && (20) \end{aligned}$$

となる. ただし S には、

$$n_j(S) = t_j[\cdots\ *\ \cdots] \quad (21)$$

となる j が必ず 1 つだけ存在するものとする. また $*$ は特別な記号で. 挿入時に T_{ik} が移動する位置を示す. また, 木の根に対する挿入は行なわないものとする.

$i < k < j$ のとき. T の t_{ik} から t_{ij} までを削除し, 削除した部分木を S とする. これは,

$$n_i(T) = t_i[T_{i1}\ T_{i2}\ \cdots\ T_{ik}\ \cdots\ T_{im}] \quad (22)$$

であるとき,

$$\begin{aligned} S &:= T_{ik} \\ n_i(T) &= t_i[T_{i1}\ T_{i2}\ \cdots \\ &\qquad T_{i(k-1)}\ T_{ij}\ T_{i(k+1)}\ \cdots\ T_{im}] \end{aligned} \quad (23)$$

とし, S の t_{ij} にあたる頂点の要素を $*$ にする. ただし, 挿入の場合と同様に, 木の根に対する削除は行なわないものとする.

頂点 $n_i(T)$ において,

$$\begin{aligned} n_i(T) &= t_i[T_{i1}\ T_{i2}\ \cdots \\ &\qquad T_{ij}\ \cdots\ T_{ik}\ \cdots\ T_{im}] \end{aligned} \quad (24)$$

であるとき,

$$\begin{aligned} n_i(T) &= t_i[T_{i1}\ T_{i2}\ \cdots \\ &\qquad T_{ik}\ \cdots\ T_{ij}\ \cdots\ T_{im}] \end{aligned} \quad (25)$$

によって j番目の子部分木と k 番目の子部分木とを入れ換える.

$$n_i(T) = t_i[T_{i1}\ T_{i2}\ \cdots\ T_{im}] \quad (26)$$

であるとき,

$$n_i(T) = x[T_{i1}\ T_{i2}\ \cdots\ T_{im}] \quad (27)$$

によって頂点 $n_i(T)$ の要素 t_i を別の値 x で置き換える.

次に, 根に近い部分から以下の処理を行えば 2 つの木の間のおおまかな異差を求めることができる.

1. 根およびその子木を注目する.
2. 2 つの木の注目部分同士を比較し, その違いを打ち消す適当な変換を一方の木に適用する.
3. 幅優先探索の順に従って, 次の注目部分に移動する.
4. 探索終了まで 2,3 を繰り返す.

木の構造の異差に注目すれば, 根に近い部分の小さな構造の違いは, 全体の構造の違いに対して大きな影響を持つ. 逆に, 葉に近い部分の異差は全体の構造に対して小さな影響しか与えない. したがって, 上の処理のように幅優先探索の順に根に近い部分から木の間の異差を探索することで, おおまかに木の異差を求めることができる.

3.2 木構造間の近似距離

木 A, Bに 3.1. の処理を適用して A を Bに変換するために手順 2 で決定される変換の列を

$$S = [s_1, s_2, \cdots, s_n] \quad (28)$$

とする. まず, 1 つ 1 つの変換に対して

$$|s_k| = \begin{cases} 1 & s_k\text{が置換} \\ |\text{挿入木}| & s_k\text{が挿入} \\ |\text{削除木}| & s_k\text{が削除} \\ \text{並べ換えの数} & s_k\text{が並べ換え} \end{cases} \quad (29)$$

によって変換のコストを定義する. さらに, 変換 s_kが, 根からの道の長さ l_kの部分でおこれば, 変換 s_kの構造コスト $d(s_k)$ を

$$d(s_k) = \frac{1}{1+l_k}|s_k| \tag{30}$$

とする．そして，

$$D(A,B) = \sum_{i=1}^{n} d(s_k) \tag{31}$$

によって，AからBへの距離を定義する．

式(31)の尺度は，

$$D(A,B) = 0 \text{ iff } A = B \tag{32}$$

$$D(A,B) = D(B,A) \tag{33}$$

を満たすが，一般には三角不等式はみたさない．

4. むすび

本論文では，発話音声の周波数分布の時間変化が，区分線形関数でよく近似できることを利用して，発話文章の信号列から，概念としてある程度まとまった部分を実時間で抽出する手法について述べた．ついで，抽出されて情報を基に，発話文章から文章の木を生成し，木構造を利用して文章間の類似性を判定する高速算法を提案した．

本研究の一部は文部省からの科学研究費補助金，旭硝子財団，ならびに，電気通信フロンティア研究開発によるものである．

文献

[1] Komatsu, A., Oohira, E., and Ichikawa, A.: "Spontaneous speech understanding based on cooperative problem-solving," IEICE Trans. Vol. E.74, pp.1845-1853, 1991.

[2] Bergen, A. R. and Shvayter, H.: "A probabilistic algorithm for computing Hough Transform," Journal of Algorithms, Vol.12, pp.639-656, 1991.

[3] Lindeberg, T.: "Scale-space for discrete signals," IEEE Transactions on PAMI, Vol. RAMI-12, pp.234-254, 1990.

[4] A. K. Joshi and Y. Schabes, "Tree-adjoining grammars and lexicalized grammars," pp.409-431, in M. Nivat and A. Pedelski eds. *Tree automata and languages*, Elsevier Science Publishers; Amsterdam, 1992.

[5] K. Zhang, D.Shasha, and J. T. Wang "Approximate tree matching in presence of variable length don't cares," Journal of Algorithms, Vol. 16, pp.33-66, 1994.

付録

関数uを

$$u(\tau) = \begin{cases} 1, & \tau = 0 \\ 0, & \tau \neq 0 \end{cases} \tag{34}$$

とする．ただし，変数τはスカラでもベクトルでも良いことにする．さて，直線上の相異なる2点に対して，

$$a = \frac{y_i - y_j}{x_i - x_j}, \quad b = \frac{x_i y_j - x_j y_i}{x_i - x_j} \tag{35}$$

を得る．そこで，

$$n(a,b) = \sum_{(\xi,\eta)\in P} u(a-\xi, b-\eta) \tag{36}$$

に対して，

$$H((x_i,y_i)^T) = (a,b) \text{ if } n(a,b) > \alpha \tag{37}$$

とすれば，複数の直線を同時に決定できる．以上の処理をHough変換という．そして，標本点の探査を確率的に行うものを確率化Hough変換という．

測定値に誤差や雑音がある場合には，適当に固定した正の定数をεとし，

$$l(a,b) = \{(x,y)^T | y = ax + b\} \tag{38}$$

とする．ここで，

$$L(a,b;\varepsilon) = \bigcup_{(\alpha,\beta)\in l(a,b)} \{(x,y)^T | (x-\alpha)^2 + (y-\beta)^2 \leq \varepsilon\} \tag{39}$$

とすれば，$L(a,b;\varepsilon)$は$l(a,b)$を中心とする幅εの帯である．そこで，

$$s(a,b) = \{(x,y)^T | (x,y)^T \in l(a,b), S \bigcap L(a,b) \neq \emptyset\} \tag{40}$$

を求める直線分とする．

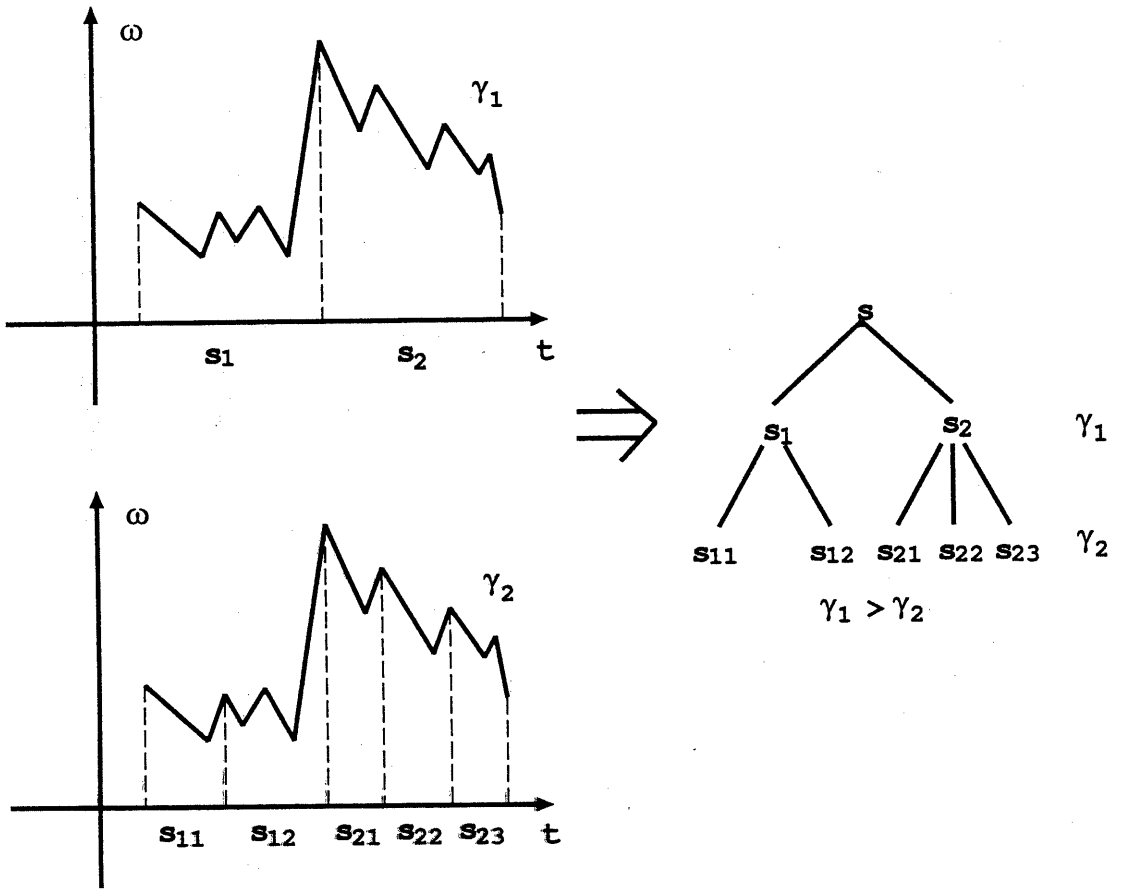

図1

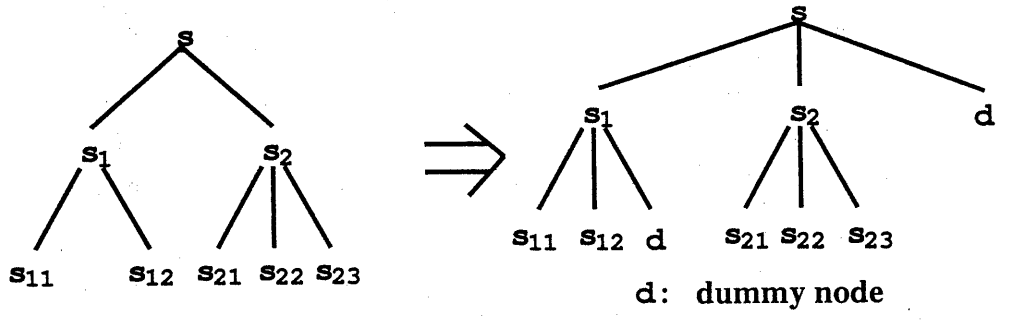

図2

図1 プロソディからの文章の木生成の概念図

図2 ダミーノードの付加による木の正則木への変換