社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

係り受け木を用いた日本語文書の重要部分抽出

伊藤 潤† 酒井 哲也†† 平澤 茂一†

† 早稲田大学 理工学部 経営システム工学科 〒169-8555 東京都新宿区大久保3-4-1
†† (株) 東芝 研究開発センター 知識メディアラボラトリー 〒212-8582 川崎市幸区小向東芝町1
E-mail: †{jun,hirasawa}@hirasa.mgmt.waseda.ac.jp, ††tetsuya.sakai@toshiba.co.jp

あらまし 日本語の文は，係り受け関係をもとに木構造（係り受け木）で表すことができる．係り受け木の部分木の表す文は，係り受け関係が保存されるため一般に正しい文となる．本稿では，文書を拡大係り受け木として表し，そのノード，エッジに重みを与える．そして，重要部分抽出問題を「拡大係り受け木の部分木のうち評価値を最大にする木を探索する問題」として定式化し，その最適化問題を解くアルゴリズムを示す．その後，提案手法による要約システムを実装し，作成された要約文を人手による採点と原文との類似度で評価を行った．

キーワード 文書自動要約，重要部分抽出，係り受け木，最適部分木の探索

Japanese Text Extraction using the Dependency Tree

Jun ITO†, Tetsuya SAKAI††, and Shigeichi HIRASAWA†

† Department of Indudtrial and Management System Engineering
School of Science and Engineering Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
†† Knowledge Media Laboratory, Toshiba Corporate R&D Center
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki-shi, 212-8582 Japan
E-mail: †{jun,hirasawa}@hirasa.mgmt.waseda.ac.jp, ††tetsuya.sakai@toshiba.co.jp

Abstract A Japanese sentence can be expressed as a tree structure (dependency tree) based on dependency relations. Since a subtree of a dependency tree preserves the dependency relations of the original tree, it generally represents a correct sentence on its own. In this paper, a document is expressed as an extended dependency tree, in which weights are assigned to its nodes and edges. Moreover, the problem of extracting important text fragments is formalized as that of "searching for a subtree that maximizes a certain score from subtrees of the extended decision tree". We implemented such a summarization system and performed evaluations based on manual assessment as well as comparison with original texts.

Key words automatic summarization, text extraction, dependency tree, search of the optimal partial tree

1. はじめに

近年，高度情報化社会の到来とともに電子化されたテキストデータが氾濫しており，ユーザが効率的に必要な情報にアクセスするのを支援する技術が求められている．その中でも，要点の迅速な把握を支援するテキスト自動要約技術が重要性を増してきており，様々な観点から活発に研究が行われている[1],[2].

文書自動要約の手法として，伝統的に，テキスト中の重要な文を抜き出す重要文抽出法が用いられてきた[3],[1]. 1990年代に入り，研究の方向性が多様化し，文中の重要箇所を抽出することによる要約手法（重要部分抽出）や言い換えや書き換えを用いた要約手法などが研究されている[1],[2].

本稿では，文節単位での重要部分抽出によって自動要約を実現する．日本語文の構造は係り受け関係の木構造で表すことができ，その部分木は，係り受け関係が保存されるため，文法的に正しい文となっている．本稿では，文書全体をひとつの係り受け木で表して拡大係り受け木とし，そのノード，エッジに重みを与えることによって，重要部分抽出問題を拡大係り受け木の部分木の最適化問題として捉える文書要約のモデルを提案する．また，この最適化問題を効率的に解くアルゴリズムを示す．そして，この要約システムを実装して要約文を作成し，主観評価，客観評価によって評価する．

＜文＞　昔々、あるところにおじいさんとおばあさんが住んでいました。
＜係り受け木＞

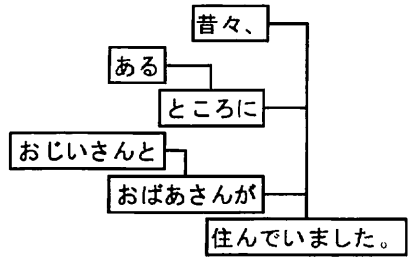

図 1　係り受け木の例

＜文＞　昔々、おばあさんが住んでいました。
＜係り受け木＞

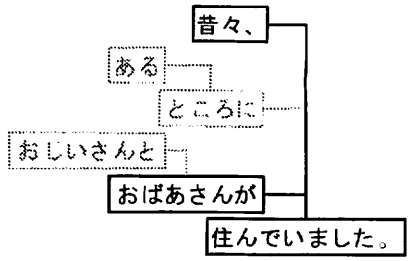

図 2　根ノードを含む部分木

2. 従来研究

2.1 係り受け関係と文書の構造

日本語の文は，より細かい単位として文節に分解できる．文節 s とは，意味を壊さない程度にできるだけ細かく区切った言葉の単位である．文節間には，主語・述語の関係，修飾・非修飾の関係といった係り受け関係が存在する．係り受け関係には次のような性質がある．

- 文末以外の文節は，必ず同じ文内の他の 1 つの文節に係る．
- 係り元の文節より係り先の文節が後ろにある．
- 係り受け関係は交差しない．(文節列 $s_1s_2s_3s_4$ において，s_1 は s_3 に係り s_2 は s_4 に係るということはない)

そのため，文節をノード，係り受け関係をエッジで表すと，1 つの文は文節の出現順を保持したまま，文末の文節を根ノードとする木構造で表すことができる．これを係り受け木という．係り受け木の例を図 1 に示す．

この係り受け関係を自動的に同定する研究は盛んに行われている．2001 年に公開された係り受け解析器 CaboCha[5] は，サポートベクターマシンを利用することによって係り受け解析を 90%程度の精度で行うことができ，現在も継続的に改良が行われている．

以降，文書は文節の系列 $s_1s_2\cdots s_i\cdots s_I$ であるとする．文節（ノード）s の係り先の文節（s の親ノード）を $p(s)$ と表し，s を係り先とする k 番目の文節（s の k 番目の子ノード）を $c(s,k)(k=1,2,\cdots,K)$ と表す．また，s の表す文字列を $h(s)$，その文字数を $n(s)$ とする．なお，文節（ノード）の個数 I は，後で述べる仮想ノードの挿入のため，定数ではない．

2.2 関連研究

係り受け関係を利用して文書要約を行う研究として，以下のようなものが挙げられる．

石井ら [6] は，単語の中心性を表す「後続する助詞による重要度」と係り受け解析によって得られた文末の述語からの距離から求められる「文節の深さによる重要度」を用いて重要文抽出を行う手法を提案している．この手法は重要文抽出の手法であり，少ない文字数に多くの情報を盛り込むには文単位の抽出では不十分である．

一方小黒ら [7] は，重要部分抽出問題を，「原文から，文節重要度と文節間係り受け整合度の総和が最大になる部分文字列を選択する」問題として定式化し，それを解くための効率的なアルゴリズムを提案した．この手法は内容に関する重要度と係り受けの整合度を並列に扱うため，内容に引きずられて係り受け関係を間違って解釈してしまう恐れがある．

3. 文書の構造と抽出のモデルの提案

3.1 係り受け木とその部分木

係り受け木の部分木のうち，文末の文節である根ノードを含む部分木は，係り受け関係の係り先の語が必ず保存される．そのため，この部分木は，元の係り受け木と比べ情報の欠落はあるものの，日本語の文として正しい文となっており，元の文とも意味が矛盾しない．そのため，この部分木の表す文は，元の文の要約のひとつであると考えられる．

図 1 の係り受け木の根ノードを含む部分木の例を図 2 に示す．

3.2 拡大係り受け木と要約部分木

係り受け木は，1 つの文の係り受け関係を表したものであるが，以下のように各文の係り受け木を結合することにより，文書全体を 1 つの係り受け木として取り扱う．この文書全体を表す結合された係り受け木を拡大係り受け木と呼び，dt と表す (図 3)．拡大係り受け木の作成手順を以下に示す．

[拡大係り受け木の作成]

（1）　各段落の末尾に当たる位置（段落の最後の文節の次の位置）に，段落末を表す仮想ノードを挿入する．

（2）　各文末のノード（文の係り受け木の根ノード）から，その文が属する段落末を表す仮想ノードへの係り受け関係を設定し，エッジを張る．

（3）　文書の末尾に当たる位置（最終段落の段落末である仮想ノードの次の位置）に，文書末を表す仮想ノードを挿入する．

（4）　各段落末のノード（段落の係り受け木の根ノード）から，文書末を表す仮想ノードへの係り受け関係を設定し，エッジを張る．　□

以後，仮想ノードは文字列を持たない（$n(s)=0$）文節として扱う．また，文書に章立てがある場合には，同様に章末，節末の仮想ノードを挿入してエッジを張ることにより，文書の構造を係り受け木で表す．

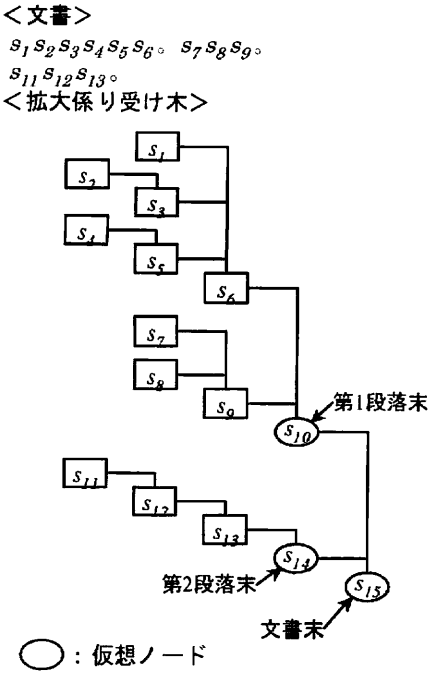

図 3　拡大係り受け木

拡大係り受け木においても 3.1 節と同様の議論が成り立ち，拡大係り受け木の根ノード（文書末を表す仮想ノード）を含む部分木の表す文章は，この文書の要約であると考えられる．そこで，以下のように要約部分木を定義し，要約部分木の表す文章を要約文の候補とする．

「定義 1」　**（要約部分木）**

拡大係り受け木の部分木のうち，根ノード（文末を表す仮想ノード）を含む部分木を要約部分木と呼ぶ．　□

3.3　重みの設定

要約部分木の良し悪しを計る評価値を求める際の要素として，文節であるノードに語彙重みを，係り受け関係であるエッジに係り受け重みを設定する．以下に語彙重み，係り受け重みの定義を示すが，この定義は他にも考えられ，第 4 章で示すアルゴリズムはこの定義にはよらない．

3.3.1　語彙重み

語彙重み mw は，その文節が文書全体にとってどの程度中心的な内容であるかを表した重みである．そこで，文節 s の頻度情報をもとに語彙重みを設定する．

語彙重み mw は，その文節に含まれる内容語である形態素 w_j を品詞による重み付けをして平均をとったものとする．品詞重みの値は，辞書によって与える．

「定義 2」　**（語彙重み）**

文節 s の語彙重み $mw(s)(\geq 0)$ を以下の式で与える．

$$mw(s) = \begin{cases} \frac{1}{J}\sum_j tf(w_j)\cdot tw(w_j) & J>0 \text{ のとき} \\ 0 & J=0 \text{ のとき} \end{cases} \quad (1)$$

w_j ： 文節 s に含まれる内容語である形態素 $(j=1,2,\cdots,J)$

$tf(w_j)$ ： 形態素 w_j の文書全体中の出現回数

$tw(w_j)$ ： 形態素 w_j の品詞重み (>0)

□

3.3.2　係り受け重み

係り受け重み dw は，係り先の文節にとって，係り元の文節が文法的にどの程度必要であるかを表した値である．本来は，文を意味レベルで解析し，係り元の文節がどういった格であるかに基づいて決めるべきものである．しかし，格を認定するためには複雑な処理が必要となるため，簡便法として，文節末の形態素の品詞や活用形，文節に含まれる機能語（主に助詞）から，辞書によって重みを与える．

「定義 3」　**（係り受け重み）**

文節 s からその親ノードである文節 $p(s)$ への係り受け重み $dw(s)(>0)$ は，例に示すような係り受け重み辞書によって与える．

（係り受け重み辞書の例）

- 文節 s の末尾が格助詞「が」 $\rightarrow dw(s)=1$
- 文節 s の末尾が動詞の連用形 $\rightarrow dw(s)=0.8$　□

また，係り先の文節が段落末等を表す仮想ノードである場合，位置情報による重みを係り受け重みとして設定する．文書が新聞記事である場合，先頭付近にある文は文書の中心的な内容を示している可能性が高い[8]．そこで，先に出現するノード（文，段落）ほど係り受け重みの値を大きくする．

「定義 4」　**（仮想ノードに対する係り受け重み）**

文節 s の親ノード $p(s)$ が文末等を表す仮想ノードであり，s は $p(s)$ の k 番目の子ノード $c(p(s),k)$ であるとき，s から $p(s)$ への係り受け重み $dw(s)$ は，

$$dw(s) = \frac{1}{k} \quad (2)$$

とする．　□

3.4　要約部分木の評価値

語彙重み，係り受け重みから，文節 s の重要度を与える．文節 s の重要度 $x(s)$ は，s の語彙重み $mw(s)$ と，s から文書末を表す仮想文節（根ノード）s_I へのパスにあたる係り受け関係（エッジ）の係り受け重み $dw(s)$ の全ての積とする．

[定義 5]　**（文節の重要度）**

文節 s の重要度 $x(s)$ を次式で与える．

$$x(s) = mw(s)\cdot dwa(s) \quad (3)$$

ここで，$dwa(s)$ は，文節 s の総係り受け重み（s から根ノードへのパスにあたるエッジの係り受け重みの積）

$$dwa(s) = \begin{cases} dw(s)\cdot dwa(p(s)) & s \neq s_I \text{ のとき} \\ 1 & s = s_I \text{ のとき} \end{cases} \quad (4)$$

である．　□

要約部分木 et の評価値は，要約部分木に含まれる文節 s_i の重要度 $x(s_i)$ の総和とする．

[定義 6] **(要約部分木の評価値)**

要約部分木 et の評価値 $z(et)$ を次式で与える.

$$z(et) = \sum_{s} x(s) \cdot f(et, s) \tag{5}$$

$$f(et, s) = \begin{cases} 0 & et \text{ が } s \text{ を含まないとき} \\ 1 & et \text{ が } s \text{ を含むとき} \end{cases} \tag{6}$$

□

3.5 重要部分抽出問題

以上より，重要部分抽出問題を，制限文字数を満たす要約部分木のうち評価値の最も高いもの（最適要約部分木）を求める問題として定義する.

[定義 7] **(重要部分抽出問題)**

$$\arg\max_{l} z(et_l) \tag{7}$$

$$\text{ただし } n(et_l) \leqq N \tag{8}$$

$n(et_l)$: 要約部分木 et_l の文字数

N : 要約したい文字数 □

4. 要約作成アルゴリズム

4.1 アルゴリズムの方針

これ以降，ある部分木 st の根ノードを $r(st)$ と表す．また，木 t のノード s を根とする部分木のうち，s とそのすべての子孫ノード，それらを結ぶエッジからなる部分木のことを t の s 以下の部分木と表す.

拡大係り受け木 dt のある部分木 st（要約部分木でなくてもよい）を考える．もし dt の $r(st)$ 以下の部分木 st' が文書全体である文書があったとき，st は st' の要約部分木であると捉えることができる．そこで，st の評価値 $z(st)$ を，同様に (5) 式で与えるものとする．ただし，(4) 式の代わりに

$$dwa(s) = \begin{cases} dw(s) \cdot dwa(p(s)) & s \neq r(st) \text{ のとき} \\ 1 & s = r(st) \text{ のとき} \end{cases} \tag{9}$$

を用いる.

ここで，文節 s を根とするある部分木 $st_m (r(st_m) = s)$ があり，st_m の $c(s,k)$ 以下の部分木を $st_{m,k} (r(st_{m,k}) = c(s,k))$ とすると，$z(st_m)$ は次のように漸化式で計算できる.

$$z(st_m) = \sum_{k} (z(st_{m,k}) \cdot dw(c(s,k))) + mw(s) \tag{10}$$

そこで，葉ノードから順に，そのノードを根とする全ての部分木 st_m を列挙してその評価値 $z(st_m)$ を計算していき，最終的に文書末ノード $s_l (= r(dt))$ を根とする部分木として挙げられたものが要約部分木である．その過程の各段階で，評価値 $z(st_m)$ を最大にできない部分木（より多い文字数でより低い評価値となる部分木）を候補から削っていく.

以下に，アルゴリズムの流れを示す.

[アルゴリズムの流れ]

(1) 係り受け解析結果から拡大係り受け木を作り，各ノード，エッジに語彙重み $mw(s)$，係り受け重み $dw(s)$ を設定する.

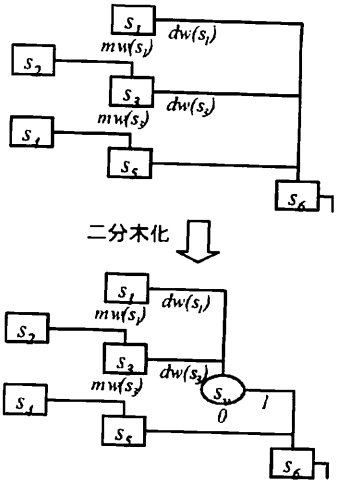

図 4 二分木化

(2) 拡大係り受け木を組み替えて，二分木化する（4.2 節参照）.

(3) 先頭のノード s_1 から順にそのノードを根とする部分木（解候補）を列挙する．(4.3 節参照）.

(4) 文書末のノード s_l の解候補から，定義 7 に基づき解候補を選択し，それを要約文として出力する. □

4.2 拡大係り受け木の二分木化

処理の簡素化と計算量削減のため，拡大係り受け木を二分木になるよう再構築する.

ノード s が，子ノードを 3 個以上持つ ($K \geqq 3$) とき，2 番目の子ノード（文節）$c(s,2)$ の次の位置に仮想ノード s_w を挿入し，次のように係り受け関係をつなぎ変え，重みを設定する．(図 4)

[二分木化]

(1) $c(s,1)$ の係り先を s_w にする.

(2) $c(s,2)$ の係り先を s_c にする.

(3) s_w の係り先を s にする.

(4) s_c の語彙重み $mw(s_c) = 0$

(5) s_w から s の係り受け重み $dw(s_w) = 1$ □

この処理を，拡大係り受け木が二分木になるまで繰り返す.

4.3 解候補の探索

文節 s を根とする部分木 $st_m (r(st_m) = s)$ が，s 以下の部分木の解候補である．このステップでは，s 以下の部分木の解候補を列挙し，明らかに劣っている解候補（他のある解候補より文字数が多く評価値が低い解候補）を削除する．またここでは，各 N に対して唯一の要約文を提示するシステムを考えているため，全く同じ文字数，評価値の解候補も一方を削除する.

これ以降，部分木 st_m である解候補 a_m を，st_m の評価値 $z(st_m)$，st_m の表す文字列 $h(st_m)$，$h(st_m)$ の文字数 $n(st_m)$ の組

$$a_m = (z(st_m), h(st_m), n(st_m)) \tag{11}$$

で表す．また文節 s は，文節 s を根ノードとする解候補 a_m の集合 $A(s)$ を持つ.

先頭のノードから順に，s の子ノードの数に応じて場合分けし，解候補集合 $A(s)$ を構築する．

(1) s が葉ノードの場合

解候補は，s がある場合とない場合なので，

$$A(s) = \{(0, \phi(\text{空文字列}), 0), (mw(s), h(s), n(s))\} \quad (12)$$

とする． □

(2) s が 1 個の子ノードを持つ場合

s の子ノード $c(s,1)$ の解候補集合 $A(c(s,1))$ のすべての解候補 $a_{m'} = (z(st_{m'}), h(st_{m'}), n(st_{m'}))$ において，

$$z(st_m) = z(st_{m'}) \cdot dw(c(s,1)) + mw(s)$$
$$h(st_m) = h(st_{m'})h(s)$$
$$n(st_m) = n(st_{m'}) + n(s)$$

となる $a_m = (z(st_m), h(st_m), n(st_m))$ を s の解候補集合 $A(s)$ に加える．ここで，$h()h()$ は，文字列の連結を表す．

また，s が仮想ノードでない場合，$a_m = (0, \phi, 0)$ を解候補集合 $A(s)$ に加える． □

(3) s が 2 個の子ノードを持つ場合

s の子ノード $c(s,1), c(s,2)$ の解候補集合 $A(c(s,1)), A(c(s,2))$ の解候補 $a_{m'} = (z(st_{m'}), h(st_{m'}), n(st_{m'}))$，$a_{m''} = (z(st_{m''}), h(st_{m''}), n(st_{m''}))$ のすべての組み合わせにおいて，

$$z(st_m) = z(st_{m'}) \cdot dw(c(s,1)) + z(st_{m''}) \cdot dw(c(s,2)) + mw(s)$$
$$h(st_m) = h(st_{m'})h(st_{m''})h(s)$$
$$n(st_m) = n(st_{m'}) + n(st_{m''}) + n(s)$$

となる $a_m = (z(st_m), h(st_m), n(st_m))$ を s の解候補集合 $A(s)$ に加える．また，s が仮想ノードでない場合，$a_m = (0, \phi, 0)$ を解候補集合 $A(s)$ に加える．

次に，s の解候補集合 $A(s)$ 中の解候補 a_m を，$n(st_m)$ の昇順にソートする．その後，隣接する解候補 a_m, a_{m+1} が $z(st_m) \geq z(st_{m+1})$ ならば，a_{m+1} は a_m より劣っているので a_{m+1} を $A(s)$ から取り除く．また，$z(st_m) < z(st_{m+1}) \wedge n(st_m) = n(st_{m+1})$ ならば，a_m は a_{m+1} より劣っているので a_m を $A(s)$ から取り除く． □

最終的に，文書末のノード s_I の解候補集合 $A(s_I)$ は，各要約率での最適要約部分木の集合となっている．

4.4 計 算 量

解候補の探索における各ステップの計算量は，(1) のとき $O(1)$，(2) のとき $O(|A(c(s,1))|)$，(3) のとき $O(|A(c(s,1))| \cdot |A(c(s,2))|)$ となる．全体の計算量は途中でどれだけ解候補を削除できるか次第であるが，全く削除できない場合最大計算量は $O(2^I)$ となる．一方，一般的な多くの要約手法は多項式オーダーで計算できる．

5. 評価実験

5.1 実験データ

毎日新聞 95 年記事[9]から，文章のみからなる（箇条書きなどの部分がない）500~600 文字程度の記事 40 文書をランダムに選び，以下の 4 手法で要約率 20%,40% の要約文を作成した．

表 1 「内容」の評価の結果

要約率	(a)	(b)	(c)	(d)
20%	39.3	35.3	36.7	32.7
40%	59.2	59.8	59.4	51.4

表 2 「読みやすさ」の評価の結果

要約率	(a)	(b)	(c)	(d)
20%	91.9	80.4	82.6	46.5
40%	89.5	84.0	84.1	52.5

(a) LEAD 手法[8] によるベースラインシステム（文抽出）

(b) A 社ワープロソフトの要約機能（文抽出）

(c) B 社統合検索ソフトの要約機能（文抽出）

(d) 提案手法

それぞれの要約文を主観評価，客観評価で比較する．

5.2 主観評価

5.2.1 評価方法

原文である新聞記事と各要約文を被験者に読んでもらい，「内容」と「読みやすさ（可読性）」の 2 つの観点から採点してその得点で比較する．

「内容」の評価は，要約文が，原文書である記事が伝えようとしている情報をどの程度含んでいるかを 100 点満点で評価してもらった．「読みやすさ」の評価は，何を指しているか分からない指示語・代名詞は無いか，つながっていない接続詞は無いか，意味不明の言葉は無いか，文章は読み返す必要はなくスムーズに読めるかといった観点から，要約文を 100 点満点で評価してもらった．いずれの評価も原文を 100 点の基準としてもらった．

被験者は大学生，大学院生で，1 記事あたり 5 人から回答を得た．

5.2.2 結　　果

各要約手法の「内容」の評価の平均点を表 1 に，「読みやすさ」の評価の平均点を表 2 に示す．提案手法 (d) はいずれも最も悪い結果となった．

5.3 客観評価

5.3.1 評価方法

原文と要約文の類似度を余弦類似度で測り，その値を比較する．要約中に原文の情報がどの程度保存されているかを形態素レベルで評価する．

原文，要約文とも文書を単語頻度ベクトル $\boldsymbol{d} = (tf(w_1), tf(w_2), \cdots, tf(w_D))$ で表す．原文と要約文の余弦類似度は以下の式で与える．

$$sim(\boldsymbol{d}, \boldsymbol{d}') = \frac{\boldsymbol{d}\boldsymbol{d}'^{\mathrm{T}}}{\sqrt{\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}}\sqrt{\boldsymbol{d}'\boldsymbol{d}'^{\mathrm{T}}}} \quad (13)$$

$\boldsymbol{d}$ ： 原文の単語頻度ベクトル

$\boldsymbol{d}'$ ： 要約文の単語頻度ベクトル

5.3.2 結　　果

原文と要約文の類似度の 40 文書における平均を表 3 に示す．提案手法 (d) は，要約率 20%,40% とも 4 手法の中で最も原文との類似度が高かった．

表3 原文と要約文の類似度

要約率	(a)	(b)	(c)	(d)
20%	0.603	0.653	0.652	0.684
40%	0.761	0.812	0.808	0.818

6. 考　　察

「内容」の評価では，提案手法はよい結果を得られなかった．3.3 で述べた重みの設定方法や品詞重み辞書，係り受け重み辞書の値の与え方は性能に大きく影響を与える部分であるが，現段階では十分に検討を重ねていない．これらの重みをうまく設定することができれば，良い要約文を生成できることが期待できる．

「読みやすさ」の評価も提案手法は良い結果を得られなかった．提案手法の作成した要約文は，ある文の文末付近の文節のみが抽出されて意味不明の文となってしまうことが少なくなく，「読みやすさ」の評価点を大きく下げてしまう傾向にあった．一方，比較対象とした要約システムはいずれも重要文抽出の手法であるため，不完全な文が抽出されることはなく，「読みやすさ」は高得点となった．

提案手法では，「今回の事件は広範な組織的な事件で大変難しい事件だった。」という部分が「事件は事件で事件だった。」と要約されたり，「新方式は松下電器産業、ソニーなど国内外の電機、テープメーカー十七社の賛同を得ており、さらに業界に呼び掛けて規格標準化を目指す。」が「新方式は得ており、目指す。」と要約されたりするなど，根に近い部分のみが抽出されて文の主題部が抜け落ちてしまう例が多く見られた．今回は自明な高頻度語の語彙重みを低くするといった処理は行っていなかったが，文書集合を仮定して idf 値を導入することにより自明な高頻度語の影響を小さくすることを考えたい．また，人手による要約文では原文中のある係り受け関係の何%が要約文にそのまま保存されるかという確率をその係り受け関係の係り受け重みとするなど，意味のある重み付けをできるように改善していきたい．さらに，意味解析によってある文節がその文内でどういう役割にある分節かを同定できるようになれば，より適切な係り受け重みの設定が可能になるであろう．

「原文との類似度」の評価では，提案手法が最も良い結果となった．提案手法は抽出の過程で形態素の tf 値を用いているため，tf 値を元に計算する余弦類似度では良い結果であったと思われる．比較対象とした手法 (b),(c) は具体的な手法は公表されていないが，これらの手法も何らかの形で tf 値を使っている可能性があり，比較的良い結果となった．一方 LEAD 手法は tf 値を考慮しない手法であるので，原文との類似度は比較的低い値となった．

表4に提案手法での要約例を示す．原文の主題部分が抜け落ちてしまっているところは見受けられるが，係り受け関係は保存されているため，日本語として文法的には誤りのない文章が作成できている．

表4 要 約 例

<原文（438 文字）>

経済企画庁が十八日発表した一九九五年四―六月期の国民所得統計速報によると、国内総生産（ＧＤＰ、季節調整済み）の成長率は、物価変動の影響を除いた実質で前期（一―三月期）比〇・八%、年率で三・一%の伸びとなった。前期に阪神大震災の影響で冷え込んだ個人消費が反動で伸びたことに加え、民間設備投資が前期比ベースで四期連続プラス、前年同月比ベースで九一年十一―十二月期以来十四期ぶりにプラスに転じたため。同庁の小林惇事務次官は七―九月期に関して「消費、投資ともに本格回復が見られない」と慎重に分析。政府が二十日に決定する経済対策の重要性を強調した。

物価の動きを示すＧＤＰデフレーターは前年同期比マイナス一・二%と四期連続で低下、マイナス幅も一―三月期の〇・八%から拡大、デフレ色が強まっていることを示した。政府の九五年度経済成長見通しは実質二・八%。これを達成するには、九五年七―九月期以降四半期ベースで平均一・五%の伸びを確保する必要があり、「常識的には難しい」（小林事務次官）情勢だ。

<要約文（要約率 30%，131 文字）>

国内総生産（ＧＤＰ、季節調整済み）の成長率は、物価変動の影響を除いた実質で前期（一―三月期）比〇・八%、年率で伸びとなった。前期に影響で冷え込んだ個人消費が伸びたことに加え、プラスに転じたため。物価の動きを示すＧＤＰデフレーターは低下、マイナス幅も拡大、示した。

7. おわりに

日本語の文書を拡大係り受け木で表し，語彙重み，係り受け重みの 2 種類の重みを与えることによって，重要部分抽出問題を要約部分木の最適化問題として解くモデルを提案した．また，その問題を効率的に解くアルゴリズムを提案した．そして，主観評価，客観評価によりその性能を評価した．

主観評価ではあまりよい結果は得られなかったが，重みにどのような値をどうやって与えれば良い要約につながるかはまだ十分な検討を重ねておらず，今後の課題としたい．

文　　献

[1] 奥村学，難波英嗣："テキスト自動要約に関する研究動向（巻頭言に代えて）"，自然言語処理，Vol.6, No.6, pp.1–26, 1999.

[2] I. Mani ："Automatic Summarization", John Benjamins Publishing Company, 2001.

[3] H. P. Luhn ："The Automatic Creation of Literature Abstracts", IBM Journal of Research and Development, Vol.2, No.2, pp.159–165, 1958.

[4] 伊藤潤，石田崇，後藤正幸，平澤茂一，"文間の単語共起類似度を用いた重要文抽出手法"，FIT2002 一般講演論文集，No.2, pp.83–84, 2002.

[5] 工藤拓，松本裕治："チャンキングの段階適用による日本語係り受け解析"，情処学論，Vol.43, No.6, pp.1834–1842, 2002.

[6] 石井弘志，林日華，古郡延治："単語の中心性に基づくテキスト自動要約システム"，情処研報，NL142-12, pp.83–90, 2001.

[7] 小黒玲，尾関和彦，張玉潔，高木一幸："文節重要度と係り受け整合度に基づく日本語文簡約アルゴリズム"，自然言語処理，Vol.8, No.3, pp.3 18, 2001.

[8] R. Brandow, K. Mitze, L. F. Rau ："Automatic Condensation of Electronic Publications by Sentence Selection", Information Proceeding and Management, Vol.31, No.5, pp.675–685, 1995.

[9] 毎日新聞社："CD-毎日新聞'95 データ集"，日外アソシエーツ．