(m, n) 文字列方式による英単語の自動分節とその評価†

浅倉 秀三††

英単語を自動分節する (m, n) 文字列方式を提案し，この方式の評価を行う．ただし，任意の字間（文字と文字との間）に先行する m 文字列と続く n 文字列からなる文字列を (m, n) 文字列とよぶ．この文字列はその字間で切ってよいまたはよくないという情報を有する．その情報を利用して自動分節する方式を (m, n) 文字列方式とよぶ．清書した文書の各行末に余った空白の総数を数え，その総数が x である任意の方式の分節率 $E_{(x)}$ を $E_{(x)}=[(c-x)/(c-b)]\times 100(\%)$ で定義した．ここで，b は人間が可能な限り切った場合の総数，c はまったく切らない場合の総数である．(m, n) 文字列方式の能力を評価するため，m, n をいろいろに変えて例文につき $E_{(x)}$ を実測してみた．(4, 4) 文字列方式が最高値 91.8% を示した．一方，辞書引き方式（辞書の見出し語を参照する方式）でも $E_{(x)}$ を実測してみた．その値は 79.0% であった．これらの方式に単純な語尾変化を処理する機能を付加した．そのとき，その値は，それぞれ 95.5%，91.1% に至った．実用性に関する検討では，(4, 4) 文字列方式は所要の記憶容量の点などから簡便な方式でないこと，一方，辞書引き方式はすでに辞書をもつシステムに付加して利用できるという点から実用的な方式であること，また，簡単な分節規則だけで切ってみたら，その分節率は 31.8% であり，これは簡便で実用的な一方式となること，などが明らかになった．

1. まえがき

英語文書の作成を援助する目的で，編集・清書プログラムの開発がなされてきた[1)~6)]．今後，端末機器やパーソナルコンピュータの普及に伴い，このプログラムはますます有用になると考えられる．このプログラムに，行末の単語を必要に応じて切り，ハイフンでつなぐ機能を付加することが提案された[2),4),6),7)]．

行末の単語を切ることは可能な限り避けるべきことである[9)]．また，最近では，ピッチ可変の印字機も普及し，単語を切ることに対する重要性は薄らいできている．しかし，確実に切る簡便な方式や既存のシステムに容易に付加して利用できる方式があれば，それらは実用的な自動分節方式として意義がある．

Knuth[6)] は，接尾辞，接頭辞，連続する子音字などに対するたいへん凝った分節規則とこの規則に合致しない約 350 語を並べた辞書とを用意し，それらを利用して自動分節する方式を示した．一方，この方式がまれに間違って切った単語やとくに切りたい単語に対しては，校正のときなどに人間がそれらの文字列中の切ってよい字間（文字と文字との間）に特別な記号を挿入し，その記号を利用して自動分節する対話的な方式を示した．ただし，後者は文献 2) の方式とほぼ同様の方式である．

筆者[7)] は，文字列が有するマルコフ性（任意の 1 文字はそれに先行する文字列の影響を受けること）を利用して自動分節する方式を示した．

本論文では，(m, n) 文字列方式を提案し，この方式の評価を行う．ただし，任意の字間に先行する m 文字列と続く n 文字列からなる文字列を (m, n) 文字列とよぶ．この文字列はその字間で切ってよいまたはよくないという情報を有する．その情報を利用して自動分節する方式を (m, n) 文字列方式とよぶ．

一方，辞書引き方式（辞書の見出し語を参照する方式）の評価も行う．

以後，m 文字列と n 文字列との字間が，音節の切れ目である文字列を音節の切れ目である (m, n) 文字列とよび，音節の切れ目でない文字列を音節の切れ目でない (m, n) 文字列とよぶ．また，辞書のすべての見出し語においてその字間が必ず音節の切れ目である文字列を，切ってよい (m, n) 文字列とよぶ．

また，音節の切れ目，空白をそれぞれ (・)，(␣) で表す．ただし，図 4 ではそれぞれ (☆)，(♯) で表す．

2. (m, n) 文字列方式

文字列中で切ってよい字間とは，音節の切れ目である．言い換えると，音節と音節との間である．任意の字間が音節の切れ目であると判定するには，そこに先行する文字列と続く文字列はそれぞれ 1 音節をなす文字列であるということが判明すればよい．したがっ

† Automatic Hyphenation of English Words by an (m, n)-Letter Sequence Algorithm and Its Evaluation by SYUZO ASAKURA (Faculty of Engineering, Chubu Institute of Technology).

†† 中部工業大学工学部自然系列情報工学

て，本論文の(m, n)文字列方式は，文献7)の方式*と比べ．続くn文字列を利用するという点で，より合理的な方式であると考えられる．

(m, n)文字列方式を具体的に言い表すと，まず，切ってよい(m, n)文字列の表を作成し，用意しておく．単語を切る必要が生じたとき，切りたい字間に先行するm文字列と続くn文字列をその単語から取り出す．その(m, n)文字列をその表中に捜し，それがあるときに限りそこで切る，ということになる．

この方式には，どの程度まで細かく切ることができるか，切ってよい(m, n)文字列の表がどの程度の大きさであるか，という不明な点がある．とくに後者は，文字列を利用する方式が辞書引き方式より優れているといわれる点[8]（所要の記憶容量が少なくて済むことや表の更新および操作が容易であること）との関連をもつ重大な問題点である．

2.1 細かい分節とその限界

文献7)と同じ例で説明する．辞書の見出し語が次のようにあるとする**．

⋮
tel•e•gram
tel•e•graph
te•leg•ra•phy
⋮

たとえば，telegram ならびに telegraph をeとgとの間で切りたい場合，文献7)の方式はこの2単語をそこで切ることができなかった．しかし，(m, n)文字列方式では，次に示すように，m, n を適切に選べば切ることができる．

(4, 3)～(4, 6)文字列方式のとき，その3見出し語から，切ってよい(4, 3)～(4, 6)文字列を**表1**(a)のように得る．また，この2単語から取り出す文字列を表1(b)に示す．表1から，(4, 3)文字列方式はこの2単語を切らない，(4, 4)や(4, 5)文字列方式は telegram だけを切る．(4, 6)文字列方式はこの2単語を切る，ということになる．したがって，m, n を大にすれば，文字列のもたらす情報が増し，さらに細かく切ることを期待できる．

しかし，たとえば telegrams をeとgとの間で切りたい場合，(4, 4)～(4, 6)文字列方式において，この単語から取り出す文字列は，それぞれ telegram, telegrams, telegrams␣ である．したがって，(4, 4)文字列方式は切る*，(4, 5)や(4, 6)文字列方式は切らない**，ということになる．m, n を大にすることは，一方ではこのような欠点を生む***．

2.2 準　備

切ってよい(m, n)文字列の表を作成する手順を示す．そのあと，辞書の見出し語から実際に作成した(m, n)文字列の表に対する検討を行う．

2.2.1 切ってよい(m, n)文字列の表の作成

その手順を次に示す．

1° 辞書の分節されている見出し語の文字列中から音節の切れ目である(m, n)文字列を残らず取り出し，表を作成する．

2° 同じ見出し語の文字列中から音節の切れ目でない(m, n)文字列を残らず取り出す．これらの文字列が1°で作成した表中にあれば，表中のそれらの文字列を削除する．

3° 終了．

2.2.2 (m, n)文字列の表に対する検討

文献9)の見出し語を対象にした．ただし，ハイフンでつないだ語（たとえば，right-hand などの単語）や略語などは除いた．見出し語の長さの分布を**図1**に，1音節をなす文字列の長さの分布を**図2**に示す．

音節の切れ目である(m, n)文字列と切ってよい

表1 切ってよい(m, n)文字列と単語から取り出す文字列
Table 1 An example of divisible (m, n)-letter sequences and letter sequences taken out from the words.

(a) 切ってよい(m, n)文字列
(a) Divisible (m, n)-letter sequences.

(m, n)	m　n
(4, 3)	
(4, 4)	tele•gram
(4, 5)	tele•gram␣
(4, 6)	tele•gram␣␣ tele•graph␣

tele•e•gram, tel•e•graph, te•leg•ra•phy から得る

(b) 単語から取り出す文字列
(b) Letter sequences taken out from the words.

(m, n)	m　n
(4, 3)	tele gra tele gra
(4, 4)	tele gram tele grap
(4, 5)	tele gram␣ tele graph
(4, 6)	tele gram␣␣ tele graph␣

上段の文字列は telegram から，下段の文字列は telegraph から取り出す

* そこでは，先行する文字列と続く1文字から切ってよいまたはよくないという情報を得ていた．

** これは，te•leg•ra•phy のeとgが切れないことによって，tel•e•gram や tel•e•graph をeとgとの間で切ることができなくなる例であった．

* これは，辞書の見出し語にない単語を(m, n)文字列方式が切る例である．

** 切ってよい(m, n)文字列の表にない文字列には，切ってよくない文字列だけでなく，このように辞書の見出し語になかったから現れない文字列がある．間違って切ることを防ぐため，その表にない文字列はすべて切らないことにしている．

*** 派生語（たとえば，接尾辞 -ly, -ment, -ness などや接頭辞 non- などがついた単語）に対してもこのようなことは起こる．

(m, n)文字列を調査した．これらの文字列の数を**表2**に示す．この表において，文字列の長さを長くするに従い，あるいは，長さが同じなら m, n のどちらか一方へ著しく偏らせないときほど，

（1） 音節の切れ目である (m, n) 文字列は多くなる．

（2） 切ってよい (m, n) 文字列は多くなる．

（3） 切ってよい (m, n) 文字列の数と音節の切れ目である (m, n) 文字列の数との比は大になる．

という傾向が見られる．（1），（2），（3）の傾向を示す原因は，それぞれ次の（1）′，（2）′，（3）′のように考えられる．

（1）′ たとえば，n を $n+1$ にすると，最後の文字が空白である文字列は1種類だけの文字列を，空白で

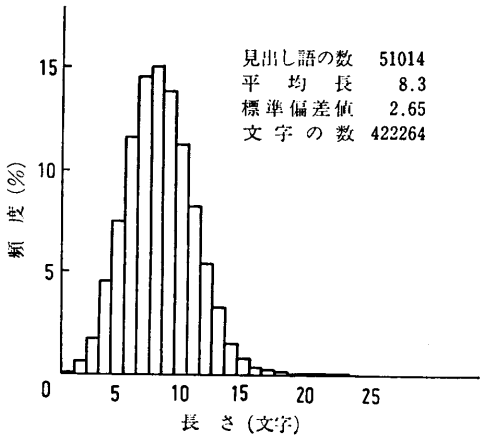
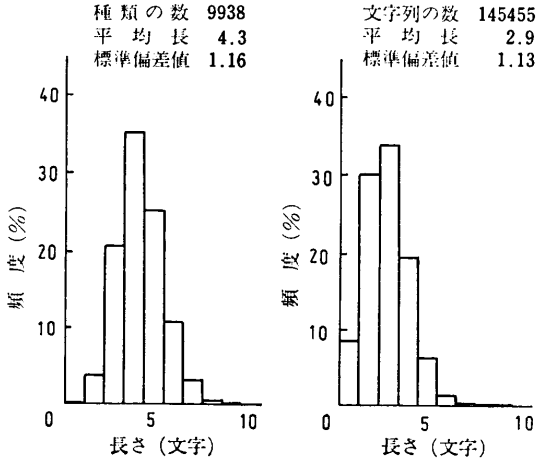


図 1 見出し語の長さの分布
Fig. 1 Lengths of the entries in a dictionary.



（a） それぞれの文字列の頻度を1とする
（a） Frequency of each kind weighted with one.
（b） 出現頻度を重みとする
（b） Weighted with occurrence frequency.

図 2 音節をなす文字列の長さの分布
Fig. 2 Lengths of syllables as sequences of letters.

表 2 音節の切れ目である (m, n) 文字列と切ってよい (m, n) 文字列の調査
Table 2 Investigation of (m, n)-letter sequences.

n \ m	0	1	2	3	4	5	6	7	8	9	10
10											28097 22412
9										27353 21686	41420 37239
8									25909 20285	39858 35675	51773 49419
7								23662 18201	37397 33199	49233 46871	62316 61084
6							20545 15162	33975 29788	45644 43204	58602 57284	69162 68585	75771 75438	79777 79502
5						15905 10509	28984 24719	40408 37804	53127 51625	63500 62735	70049 69522	74027 73557	
4					9164 4182	21713 16898	32774 29698	45091 43191	55092 53884	61459 60470	65331 64398		
3				2744 554	12366 7319	23670 19406	35901 33086	45506 43474	51667 49861	55381 53636	57585 55843		
2			304 3	3347 1145	12209 7206	25717 20493	35847 31617	41939 38094	45476 41611	47565 43601	48618 44640		
1			537 60	4051 1296	14617 7106	25679 16860	32288 23484	35917 27018	38004 28762	39055 29574	39596 30024		
0			349 7	3297 316	11583 2978	18924 8315	22733 11870	24712 13811	25686 14607	26179 14838	26411 15013		

上段：音節の切れ目である (m, n) 文字列の数
下段：切ってよい (m, n) 文字列の数

ない文字列は最大27種類までの文字列を，形態上作るように見える．したがって，長くするに従い，その文字列は多くなる．一方，偏らせたときには偏らせないときより前者の文字列の占める割合は大である．長くしてもこの文字列が原因になり，文字列の多くなる程度は少ない．したがって，その長さが同じなら偏らせないときほど，その文字列は多くなる．

(2)' 切ってよい(m, n)文字列は，音節の切れ目である(m, n)文字列から音節の切れ目でない(m, n)文字列を除いた文字列である．したがって，(1)' と同様の理由である．

(3)' m文字列の長さやn文字列の長さが1音節をなす文字列の長さを越えると，それらの文字列が有する規則性は減じてしまう[10]．このことから，音節の切れ目である(m, n)文字列はそれを取り出した単語に固有な文字列となる．当然のこととして，その文字列は，他の単語中に見られず，削除されない．このような文字列が多くなることによって，この比は大になる．

また，(2)の傾向はより細かく切ることができることを意味する[7]．(3)の傾向は(3)' で述べた理由から，次のⅰ)，ⅱ)のようなおそれがあることを意味すると考えられる．

ⅰ) 切ってよい(m, n)文字列が多くなっていても，実際にはほとんど参照されない文字列のそれに占める割合も大になっており，より細かく切ることを期待できない．

ⅱ) 辞書の見出し語にない単語を切ることができなくなる．

2.2.3 分節機能

分節する機能の手順を次に示す．

1° 単語を切る必要が生じる．
2° ハイフンをつけてちょうど行末がそろうような字間を決める．
3° そこに先行するm文字列と続くn文字列を取り出す．その(m, n)文字列を切ってよい(m, n)文字列の表中に捜す．
3.1° ある．そこで切る．
3.2° ない．そこの前にある文字列の長さを調べる．
3.2.1° 2以上である．そこを1文字分前へ移し，3° へもどる．
3.2.2° 2未満である．その単語を次行へ送る．
4° 終了．

3. 辞書引き方式

辞書の見出し語を参照する方式は，間違ったつづりの検出・訂正など文字列処理によく使われてきた．この方式は他の方式より能力では優れている．しかし，所要の記憶容量が大きいことや辞書の見出し語にない単語に対処できないことなどの欠点がある．

分節する機能の手順を次に示す．

1° 単語を切る必要が生じる．
2° ハイフンをつけてちょうど行末がそろうような字間を決める．
3° 辞書の見出し語にその単語を捜す．
3.1° ある．その字間の前に音節の切れ目を捜す．
3.1.1° ある．そこに最も近い切れ目で切る．
3.1.2° ない．その単語を次行へ送る．
3.2° ない．その単語を次行へ送る．
4° 終了．

4. 評 価

アメリカ合衆国の独立宣言と自然言語処理に関する論文を，(m, n)文字列方式および辞書引き方式で切って清書してみた．また，分節規則だけで切って清書してみた．一方，評価の基準として，人間が可能な限り切って，まったく切らないで，清書してみた．これらの実験の結果を示す．辞書引き方式でも文献9)を利用した．ただし，この場合には，ハイフンでつないだ語や略語なども含めた．それで，辞書の見出し語の数は53,932となった．清書した英文の単語の数は9,002であった．

4.1 分節規則

次に述べる規則(1)～(6)をプログラムに書き下した．まったく切らない場合には規則(1)，(2)だけを，切る場合には規則(1)，(2)と規則(3)～(6)* とを使った．

(1) 単語と単語との間の空白やピリオド，コンマなど区切り記号の後の空白は1空白とする．

(2) 1行当りの文字数は65とする．

(3) 単語を語頭の1文字や語尾の2文字で切らない．

(4) ハイフンでつないだ語は，ハイフンの直後だけで切られる．

(5) ダッシュによって見かけ上連結されている2

* 本論文では，文献9)，11)に示されているこれらの分節規則だけを利用した．

表3 実験の結果
Table 3 Results of the experiment.

(a) (m,n) 文字列方式
(a) (m,n)-letter sequence algorithms.

n \ m	0	1	2	3	4	5	6	7	8	9	10
10											953 1932 64.3
9										953 1911 65.8	953 1798 73.5
8									953 1916 65.4	953 1788 74.2	953 1696 80.5
7								953 1904 66.3	953 1766 75.7	953 1666 82.6	953 1680 81.6
6							953 1904 66.3	953 1760 76.2	953 1655 83.4	953 1658 83.2	953 1634 84.8	953 1648 83.8	953 1667 82.5
5						953 1895 66.9	953 1717 79.1	953 1607 86.7	953 1574 88.9	953 1576 88.8	953 1595 87.5	953 1621 85.7
4					954 2032 57.5	953 1690 81.0	953 1578 88.7	953 1539 91.3	953 1533 91.8	953 1554 90.3	953 1600 87.1
3				956 2293 39.5	954 1942 63.6	952 1673 82.1	952 1554 90.3	952 1566 89.5	952 1551 90.5	952 1593 87.6	952 1593 87.6
2			957 2406 31.8	957 2288 39.9	953 1919 65.2	953 1719 79.0	953 1659 83.1	953 1652 83.6	953 1629 85.2	953 1656 83.3	953 1658 83.2
1		957 2400 32.2	955 2263 41.6	954 1993 60.1	954 1900 66.5	953 1822 71.9	953 1838 70.8	953 1826 71.6	953 1822 71.9	953 1822 71.9	
0	957 2406 31.8	957 2357 35.1	956 2258 41.9	954 2173 47.8	954 2178 47.4	954 2185 46.9	954 2169 48.0	954 2178 47.4	954 2176 47.6		

上段：行の数
中段：空白の総数
下段：分節率（%）

(b) 辞書引き方式
(b) Dictionary lookup algorithm.

(c) まったく切らない場合，分節規則だけで切る場合，可能な限り切る場合
(c) Not divided at all, divided by only rules for division, and divided as precise as possible.

辞書引き方式	まったく切らない	分節規則だけで切る	可能な限り切る
953	964	957	952
1,719	2,868	2,406	1,413
79.0	0	31.8	100

単語の後の単語を切る必要が生じたときは，ダッシュの直後で切る．

（6） 下記の文字列が語尾である単語をその文字列中で切る必要が生じたときは，その文字列の直前で切る．

-able, -ible ; -cial, -sial, -tial ; -cion, -sion, -tion ; -gion ; -ceous, -cious, -tious ; -geous.

4.2 分節率

清書に要した行の数と各行末に余った空白の総数を数えた．また，その総数がxである任意の方式の分節率 $E_{(x)}$ を次式のように定義した．

$$E_{(x)}=\frac{c-x}{c-b}\times 100(\%) \qquad (1)$$

ここで，bは人間が可能な限り切った場合の総数，cはまったく切らない場合の総数である．

行の数，空白の総数，分節率を**表3**に示す．また，(m,n)文字列方式の分節率を等高線の図の形にまとめて，**図3**に示す．図3における下限の値32%は，表3(c)から明らかなように，分節規則が有する分節率である．

5. 考察と検討

実験の結果についての考察を行う．そのあと，分節率の向上や実用性についての検討を行う．

5.1 (m,n) 文字列方式に対する考察

文献7)の方式と本論文の(1,1)～(5,1)文字列方式は手続き上では同じである．前者の分節率と後者の分節率は，利用する文字列の長さを長くするに従い，その値が大になるという傾向において一致している．しかし，後者の(1,1)や(2,1)文字列方式の分節率は分節規則によって，また(2,1)～(5,1)文字列方式の分節率は切ってよい(m,n)文字列が多くなったこと* や分節規則によって，前者の分節率より高くなった．

(4,4)文字列方式が最高の分節率を示した．このことから，(4,4)文字列は分節することに関する必要な情報をもたらす長さの文字列であり，また，辞書の見出し語にない単語を分節することについても適切な長さの文字列であると考えられる．この長さ4は，図2から明らかなように，1音節をなす文字列の平均的な長さとほぼ一致する．

一方，利用する文字列の長さをこれより長くしても，それに伴い，切ってよい(m,n)文字列が多くなっていても，分節率は逆に低下している．これは，2.2.2項のⅰ），ⅱ）で指摘したおそれが現実に起こったからである．

5.2 辞書引き方式に対する考察

表3(b)に示した分節率は，推測される値より低い

* 切ってよい(m,n)文字列は，本論文で利用した辞書の見出し語がより多かったこと（文献7）では約17,000，本論文では約51,000）から，多くなった．

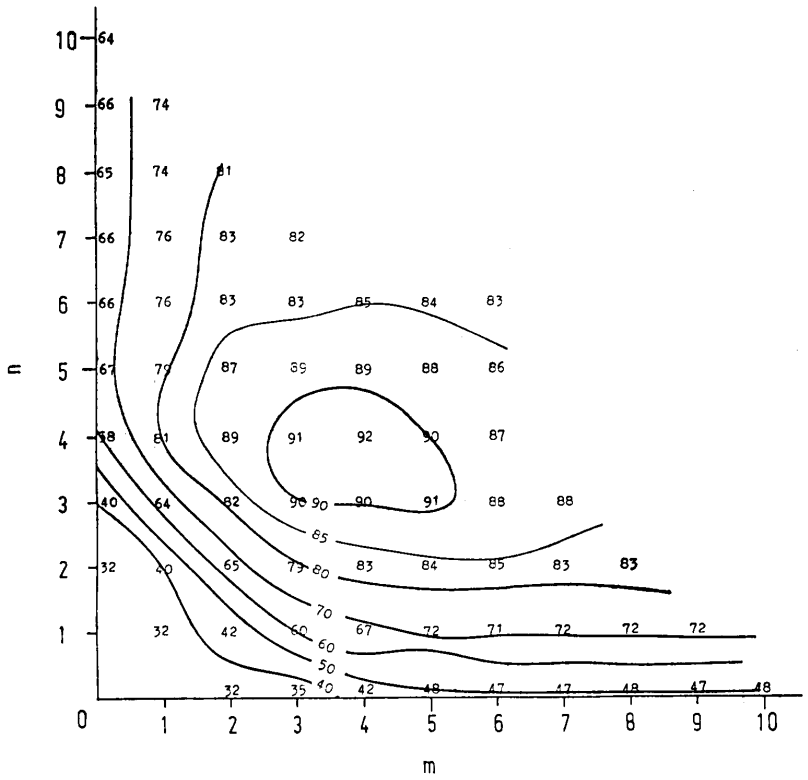


図 3 (m, n) 文字列方式の分節率の等高線図
Fig. 3 Contour figure of the powers of the (m, n)-letter sequence algorithms.

表 4 見付けた単語の割合
Table 4 Ratios of words looked up in a dictionary to all words in the runoff text.

(a) 種類の数
(a) Number of kinds.

種類の数 (種類)	1,483
見付けた数(種類)	1,094
見付けた割合(%)	73.8

(b) 出現頻度を重みとする
(b) Weighted with occurrence frequency.

単語の数 (語)	9,002
見付けた数 (語)	7,801
見付けた割合(%)	86.7

値である．この原因として，利用した辞書の見出し語にない，規則的に語尾変化した単語，派生語および特殊な単語に対処できなかったことが考えられる．清書した英文の単語を辞書の見出し語に捜し，それの有無を調べた(**表 4**)．この結果から，辞書引き方式の分節率はほぼ妥当であると考えられる．

5.3 検　討

表 4 の調査において辞書の見出し語にない単語の大部分は，規則的に語尾変化した単語であった．それで，(3, 3)文字列方式，(4, 4)文字列方式，辞書引き方式に単純な語尾変化を処理する機能を付加した．この機能は，これらの(m, n)文字列方式では切ってよい

表 5 単純な語尾変化の処理
Table 5 Processing for some simple inflections.

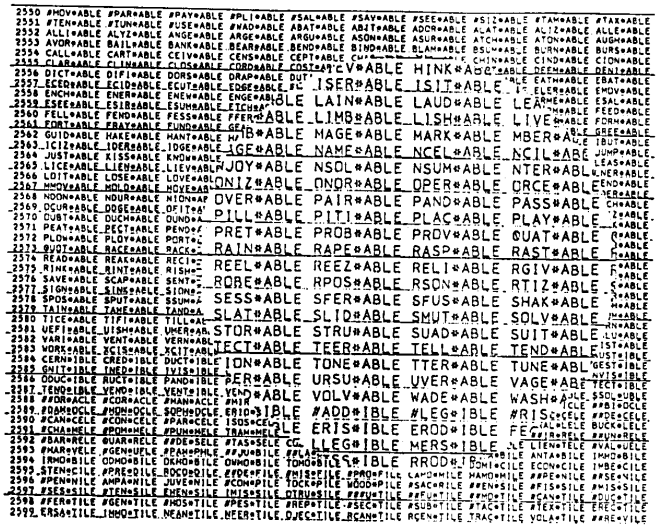
語 尾	置き換える文字	語 尾	置き換える文字
s	␣	d	␣
es	␣␣	ed	␣␣
ies	y␣␣	r	␣
ing	␣␣␣	er	␣␣
ing	e␣␣	st	␣␣
ings	␣␣␣␣	est	␣␣␣
ings	e␣␣␣		

(m, n)文字列の表に捜している文字列を見いだせないときに，辞書引き方式では辞書の見出し語に捜している単語を見いだせないときに，その単語（切りたい単語）に対して働く．**表 5** にその処理の内容を示す．その単語の語尾が左欄の文字列と一致するとき，それを対応する右欄の文字列と置き換える．また，二つ以上の文字列と一致するとき，次から次へと置き換える．ただし，この置き換えは，見いだしたとき，あるいは試す文字列がなくなったときに終る．この機能を使って，表 4 と同じ調査を行った(**表 6**)．表 4 と表 6 との比較から，分節率では，辞書引き方式には約 10% の

向上を期待できる．一方，(m, n)文字列方式は語尾変化した単語を切ることもあり，これにはそれほどの向上を期待できない．再び，同様に清書してみた(**表7**)．(3, 3)文字列方式の分節率，(4, 4)文字列方式の分節率，辞書引き方式の分節率は，それぞれ0.5，3.7，12.1%向上して，90.8，95.5，91.1%に至った．切ってよい(4, 4)文字列の表の一部分を**図4**に，清書した文書の一部分を**図5**に示す．

(4, 4)文字列方式や辞書引き方式における所要の記憶容量を概算すると，その大きさは，それぞれ長さ8と切ってよい(4, 4)文字列の数53,884から約430Kバイト，53,932見出し語に要する約540Kバイト(音節の切れ目を表す中黒も1文字と数える)となる．この点については，(4, 4)文字列方式は辞書引き方式よりやや優れている．

しかし，実用性の面では，(4, 4)文字列方式などは，これほどの大きさでかつ他に転用できない表を必要とすることなどから，簡便な方式であるとは言えない．分節率に関しては，(4, 4)文字列方式などは確



*：音節の切れ目，#：空白

図4 切ってよい(4, 4)文字列の表の一部分

Fig. 4 A partial list of divisible (4, 4)-letter sequences.

かに興味のある方式である．しかし，図5(b)では，明らかに切りすぎである．したがって，実用的な方式でないと考えられる．一方，辞書を新たに用意して，辞書引き方式を利用することは，所要の記憶容量の点などからあまり実用的でない．しかし，すでに辞書をもつシステムに辞書引き方式を付加して利用することは，実用的であると考えられる．

He has affected to render the Military independent of and superior to the Civil power.--He has combined with others to subject us to a jurisdiction foreign to our constitution, and unacknowledged by our laws; giving his Assent to their Acts of pretended Legislation:--For quartering large bodies of armed troops among us:--For protecting them, by a mock Trial, from punishment for any Murders which they should commit on the Inhabitants of these States:--For cutting off our Trade with all parts of the world:--For imposing Taxes on us without our Consent:--For depriving us in many cases, of the benefits of Trial by Jury:--For transporting us beyond Seas to be tried for pretended offences:--For abolishing the free System of English Laws in a neighbouring Province, establishing therein an Arbitrary government, and enlarging its Boundaries so as to render it at once an example and fit instrument for introducing the same absolute rule into these Colonies:--For taking away our Charters, abolishing our most valuable Laws and altering fundamentally the Forms of our Governments:--For suspending our own Legislatures, and declaring themselves invested with power to legislate for us in all cases whatsoever.--

(a) 可能な限り切る

(a) Divided as precise as possible.

67 HE HAS AFFECTED TO RENDER THE MILITARY INDEPENDENT OF AND SUPE-
68 RIOR TO THE CIVIL POWER.--HE HAS COMBINED WITH OTHERS TO SUBJECT
69 US TO A JURISDICTION FOREIGN TO OUR CONSTITUTION, AND UNACKNOWL-
70 EDGED BY OUR LAWS; GIVING HIS ASSENT TO THEIR ACTS OF PRETENDED
71 LEGISLATION:--FOR QUARTERING LARGE BODIES OF ARMED TROOPS AMONG
72 US:--FOR PROTECTING THEM, BY A MOCK TRIAL, FROM PUNISHMENT FOR
73 ANY MURDERS WHICH THEY SHOULD COMMIT ON THE INHABITANTS OF THESE
74 STATES:--FOR CUTTING OFF OUR TRADE WITH ALL PARTS OF THE WORLD:--
75 FOR IMPOSING TAXES ON US WITHOUT OUR CONSENT:--FOR DEPRIVING US
76 IN MANY CASES, OF THE BENEFITS OF TRIAL BY JURY:--FOR TRANSPORT-
77 ING US BEYOND SEAS TO BE TRIED FOR PRETENDED OFFENCES:--FOR ABOL-
78 ISHING THE FREE SYSTEM OF ENGLISH LAWS IN A NEIGHBOURING PROV-
79 INCE, ESTABLISHING THEREIN AN ARBITRARY GOVERNMENT, AND ENLARGING
80 ITS BOUNDARIES SO AS TO RENDER IT AT ONCE AN EXAMPLE AND FIT IN-
81 STRUMENT FOR INTRODUCING THE SAME ABSOLUTE RULE INTO THESE COLO-
82 NIES:--FOR TAKING AWAY OUR CHARTERS, ABOLISHING OUR MOST VALUABLE
83 LAWS AND ALTERING FUNDAMENTALLY THE FORMS OF OUR GOVERNMENTS:--
84 FOR SUSPENDING OUR OWN LEGISLATURES, AND DECLARING THEMSELVES IN-
85 VESTED WITH POWER TO LEGISLATE FOR US IN ALL CASES WHATSOEVER.--

(b) (4, 4)文字列方式に語尾変化処理機能を付加(行印字機による出力)

(b) By (4, 4)-letter sequence algorithm plus the function for simple inflections.

図5 清書した文書の一部分(アメリカ合衆国 独立宣言)

Fig. 5 Partial results of the experiment.

6. むすび

行末の単語を必要に応じて切る，(m, n)文字列方式

表6 見付けた単語の割合(語尾変化処理の機能を付加)

Table 6 Ratios of words looked up in a dictionary using the processing for some simple inflections to all words in the runoff text.

(a) 種類の数 (a) Number of kinds.		(b) 出現頻度を重みとする (b) Weighted with occurrence frequency.	
種類の数 (種類)	1,483	単語の数 (語)	9,002
見付けた数(種類)	1,377	見付けた数 (語)	8,642
見付けた割合(%)	92.9	見付けた割合(%)	96.0

表7 実験の結果(語尾変化処理の機能を付加)

Table 7 Results of the experiment using the processing for some simple inflections.

	(3, 3) 文字列方式と語尾変化処理	(4, 4) 文字列方式と語尾変化処理	辞書引き方式と語尾変化処理
行の数	952	952	952
空白の総数	1,547	1,479	1,543
分節率(%)	90.8	95.5	91.1

および辞書引き方式の分節率と問題点に関して次のような結論を得た.

(m, n)文字列方式は,利用する文字列の長さを長くするに従い,分節率が高くなると考えられる.しかし,必要以上に長くすると,辞書の見出し語にない単語を切ることもあるという長所が損われ,その値は逆に低下することが明らかになった.(4, 4)文字列方式が最高の分節率を示した.人間が可能な限り切る場合の分節率を100%,まったく切らない場合の分節率を0%としたとき,その値は91.8%であった.その長さ4は,1音節をなす文字列の長さと関係がある.また,このときを含め,十分な分節率を示す(m, n)文字列方式では,切ってよい(m, n)文字列の表がかなり大きくなるという欠点があることも明らかになった.

辞書引き方式は,可能な限り切る場合に最も近い分節率を示すと考えられる.しかし,単純な語尾変化に対処できず,その値は79.0%であった.

(4, 4)文字列方式や辞書引き方式に単純な語尾変化を処理する機能を付加して,分節率の向上を試みた.その値は,それぞれ95.5, 91.1%に至った.残りの4.5%や8.9%が生じたおもな原因は,利用した辞書の見出し語にない派生語によるものであった.

実用性の面では,(m, n)文字列方式には所要の記憶容量がかなり大きいという問題点がある.一方,すでに辞書をもつシステムに辞書引き方式を付加して利用することは実用的であると考えられる.また,簡単な分節規則だけで切ってみたら,その分節率は30%を越えることが明らかになった.したがって,それらの分節規則に接頭辞や他の接尾辞に対する規則を追加すれば,実用的な一方式が実現できると考えられる.

謝辞 ご指導いただく慶応義塾大学理工学部相磯秀夫教授,ご助言いただいた電子技術総合研究所五十嵐実子女史,植村俊亮氏,坂本義行氏に感謝する.本論文において,方式などの名称に対する示唆や全体にわたって有益なご教示いただいた査読委員に感謝する.

参考文献

1) Mashey, J. R. and Smith, D. W.: Documentation Tools and Techniques, Proc. 2nd International Conference of Software Engineering, pp. 177-181 (1976).
2) Thompson, C. and Garland, S.J.: *DTSS RUNOFF*** *Reference Manual*, p. 42, Kiewit Computation Center, Dartmouth College (1977).
3) Kernighan, B. W., Lesk, M. E. and Ossanna, J. F., Jr.: UNIX Time-Sharing System: Document Preparation, *Bell Syst. Tech. J.*, Vol. 57, No. 6, pp. 2115-2135 (1978).
4) 井田哲雄:英語文書清書プログラム ROFF(2), 東京大学大型計算機センターニュース, Vol. 9, No. 9, pp. 61-65 (1977).
5) 石田晴久:コンピュータによる英語論文の編集・清書法,東京大学大型計算機センターニュース, Vol. 10, No. 7・8, pp. 38-41 (1978).
6) Knuth, D. E.: *TEX and METAFONT: New Directions in Typesetting*, Digital Press (1979).
7) 浅倉秀三:英語論文の清書における英単語の自動分節に関する1統計的文法,情報処理学会論文誌, Vol. 21, No. 1, pp. 38-44 (1980).
8) Riseman, E. M. and Hanson, A. R.: A Contextual Postprocessing System for Error Correction Using Binary *n*-Grams, *IEEE Trans. Comput.*, C-23, pp. 480-493 (1974).
9) Stein, J.(ed.): *The Random House Dictionary*, paperback ed., p. 1070, Ballantine Books, New York (1978).
10) 浅倉秀三:英文字列中のマルコフ性の起源とそれが及ぶ範囲について,電子通信学会論文誌, Vol. 61-D, No. 12, pp. 933-939 (1978).
11) Turabian, K. L.: *A Manual for Writers of Term Papers, Theses, and Dissertations*, 3rd ed., revised, p. 164, The University of Chicago Press, Chicago (1972).

(昭和56年7月2日受付)
(昭和57年9月6日採録)