人口知能と対話技法 24－1
（1982. 1. 28）

最長一致法と文節数最小法について

吉村賢治　日高達　吉田将
（九州大学　工学部）

1. まえがき

電子計算機による自然言語理解の過程は，システムが持っている言語的情報の適用の仕方からいくつかの型に分類できる．図-1(a)はシステムが持っている形態論的情報，構文論的情報，意味論的情報を同時に使用する例であり，図-1(b)はこれらの情報を段階的に使用する例である．(a)は人間が行っている方法であるが，解析の途中で不十分な意味的情報を強く反映させることは，誤った解析に導く原因になることが考えられる．また，入力文が仮名又はローマ字でべた書きされた日本語文の場合，解の探索空間は非常に大きくなり，これに対して意味解析などの複雑な処理を行うことは能率の低下をもたらすと考えられる．従って，システムの持つ意味論的情報が不十分な場合には，少なくとも形態素解析や構文解析などの統語論的解析と意味論的解析を分離した段階的な構成が望まれる．

以下，段階的な構成として図-1(b)の方式を用いた場合について考察する．このとき，各解析段階における解の探索空間の変化を図-2に示す．集合S_1は形態論的制約を満たす解の集合であり，集合S_2は構文論的制約を満たす解の集合である．一般に，日本語文の形態素解析に用いられる形態論的制約は，正規文法のクラスに属する規則で表現される．この規則は明確である反面正しい日本語文に対する制約が弱い．そのため，入力文として仮名又はローマ字表記のべた書き日本語文を用いた場合，集合S_1は多くの誤った解析を含む大きな集合となる．このような条件のもとで効率の良い解析を行うためには，集合S_1の尤度（もっともらしさの度合）

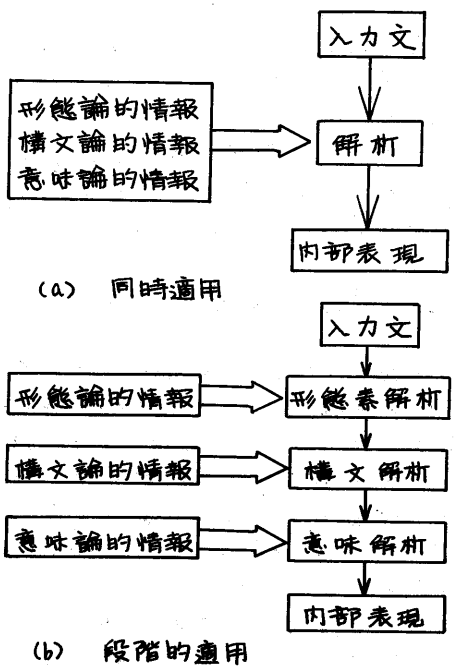

図-1 情報の適用例

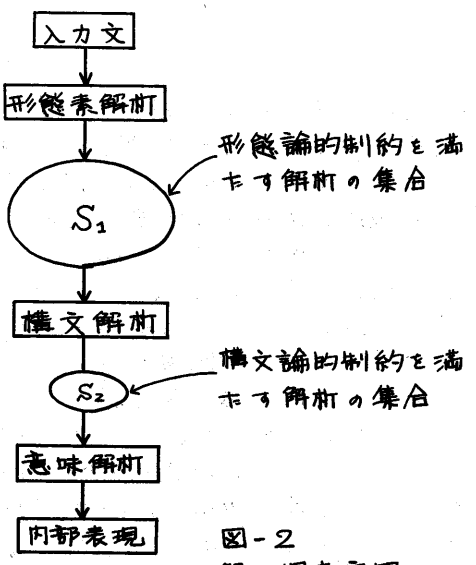

図-2
解の探索空間

が高い解析結果から構文解析を適用する必要がある．

本稿では，集合S_1の各要素に対してその尤度を評価するヒューリスティック情報である最長一致法と文節数最小法について考察し，両者の比較をするために行った実験について報告する．

2. 最長一致法

従来，このヒューリスティック情報としては，入力文字列のある位置から始まる単語を認定する段階で複数の候補が存在する場合，入力文字列における単語の長さが長いものから優先する最長一致法が用いられている．この手法を用いた解析手順は，形態論的制約を正規文法としてとらえた場合，適用可能な書き換え規則に優先順位を付けて，その順位の高い書き換え規則から優先して適用するという手順になる．すなわち，X, Y_iを非終端記号，ω_iを終端記号としてω_iに対応する入力文字列の部分列の長さを$|\omega_i|$で表わすとき，ある時点でm個の書き換え規則，

P_1 : $X \rightarrow \omega_1 Y_1$

P_2 : $X \rightarrow \omega_2 Y_2$

⋮

P_m : $X \rightarrow \omega_m Y_m$

が適用可能ならば，$|\omega_i|$の値が最大となる書き換え規則P_iから優先して適用する手順となる．これは書き換え規則に優先順位を付けた文脈自由言語のバックトラック方式の構文解析アルゴリズムと同じ手順である(1)．このアルゴリズムの能率については文献(1)で議論されているが，その議論において文脈自由文法を正規文法で置き換えても正規文法の特長は活かすことができず，正規文法に対しても同様な結論を得る．従って，最長一致法を用いた場合の能率は入力文字列の長さnに対して，最悪の場合に必要とするステップ数は$O(c^n)$，メモリ数は$O(n)$となる．ここでCはnに無関係な定数である．

これまで最長一致法は多くのシステムに用いられているが，二つの問題点が考えられる．第一の問題点は，長く一致した単語を優先する根拠が明らかでないことである．最長一致法の歴史を調べると，海外の文献では英単語の語幹抽出アルゴリズム(stemming algorithm)において用いられている(2)．この場合は，接辞辞書の検索回数を減らすことが目的となっている．国内の文献では，文節単位に分かち書きされた入力文を対象とする仮名漢字変換において自立語の認定に用いられている(3)．ただし，この場合の最長一致法は尤度の評価としてではなく解析の順序付けのために用いられている．このように短い単位に対して各々の根拠を持って用いられてきた手法が，べた書き日本語文に対して流用されていると考えられる．

第二の問題点は，単語の長さという局所的な対象を評価しているため，文全体に対する尤度の評価になっていないと考えられることである．

以上のような問題点はあるが，必要とするメモリ数が少ない，プログラミングが容易などの利点も持っている．

3. 文節数最小法

解析結果の尤度を解析結果の文節数を用いて評価する手法が文節数最小法である．すなわち，解析結果に複数の候補がある場合には文節数が少ないものから優先する手法である．このヒューリスティック情報を用いる場合には，表方式のアルゴリズムを使用することができ，その能率は最悪の場合に必要とするステップ数，メモリ数ともに入力文の長さnに対して$O(n^3)$となる(4)．また，特別の場合として，文節数が最

小となる解析のみを求める場合の能率は，最悪の場合に必要とするステップ数，メモリ数ともに$O(n)$となる．

日本語文の統語規則は，文節を構成する単語の並びを規定する形態論的規則（文節構造規則と呼ぶ）と文を構成する文節間の係り受け関係を規定する構文論的規則（係り受け構造規則と呼ぶ）としてとらえることができる[5]．従って，文節構造規則を満たす解析が正しい日本語文であるためには，少なくとも係り受け構造規則を満たさなければならない．係り受け構造規則は文節間の意味的呼応関係を規定するものであるから文節数が少ない解析結果の方が係り受け構造規則を満たす確率は高いと考えられる．これが文節数で尤度を評価する根拠である．このように考えると，最長一致法は解析結果の文節数を局所的な単語の長さで近似的に評価しているとみなすこともできる．

文節数最小法は文節数で尤度を評価しているため，文全体の尤度を評価できると考えられる．

4. 実験

2.，3.で最長一致法と文節数最小法について簡単に考察したが，このようなヒューリスティック情報に基づいた手法の有効性を理論的な側面だけでとらえることは困難である．本章では，これら二つの手法を比較するために行った実験の概要を示し，5.でその結果を報告する．

4.1 述語の定義

次節で実験に用いた最長一致法のアルゴリズムを示すが，本節では，その記述を簡明にするために幾かの述語を定義する．なお以下の議論ではn長さの入力文字列，

$$\Delta = \delta(1)\,\delta(2)\cdots\delta(n)$$

を固定する．

［定義１］　単語構造

単語wの綴りW（活用語の場合は終止形の綴り），品詞H，活用情報Kからなる３項系列(W,H,K)を単語wの単語構造と呼ぶ．□

［定義２］　述語 WS,J,E,C

i)　入力記号列Δの部分列$\delta(i+1)\delta(i+2)\cdots\delta(j)$に対して，$w=\delta(i+1)\delta(i+2)\cdots\delta(j)$なる単語$w$が存在し，その単語構造が$\alpha$であることを$WS(i,j,\alpha)$で表わす．

ii)　単語構造αが自立語の単語構造であることを$J(\alpha)$で表わす．

iii)　単語構造αの単語が文節末の語になりうることを$E(\alpha)$で表わす．

iv)　単語構造α_1の単語w_1と単語構造α_2の単語w_2の連接w_1w_2が文節の部分列になり得ることを$C(\alpha_1,\alpha_2)$で表わす．□

これらの述語を用いて，一般に用いられる形態論的制約は次の文節構造規則として定義される．

［定義３］　文節構造規則

n長さの入力記号列Δに対して，

$i_0(=0)<i_1<\cdots<i_m(=n)$

なる整数$i_0,i_1,\cdots,i_m$と単語構造$\alpha_1,\alpha_2,\cdots,\alpha_m$が存在して，次のi,ii,iii,を満たすとき$\Delta$は左文節を成すといい，さらにivも満たすとき文節を成すという．

i.　$WS(i_{l-1},i_l,\alpha_l)$　$(l=1,2,\cdots,m)$

ii.　$J(\alpha_1)$

iii.　$C(\alpha_{l-1},\alpha_l)$　$(l=2,3,\cdots,m)$

iv.　$E(\alpha_m)$　□

ただし，i,ii,iii,ivの条件は文節を成すための必要十分条件ではなく，必要条件にしかすぎない．

4.2 最長一致法のアルゴリズム

最長一致法には単語最長一致法，文節最長一致法，二文節最長一致法などのいろいろな方式がある．そこで議論を明確にするため，実験に用いた最長一致法のアルゴリズムを示す．文節数

最小法を用いた表方式のアルゴリズムについては既に報告しているので本稿では割愛する[4].

アルゴリズムの記述において，プッシュ・ダウン・スタックをSTACKとし，ANSLISTは整数 i,j と単語構造 α からなる3項系列 (i,j,α) のリストとする．また，$\Gamma(i),\Gamma_F(i),\Gamma_J(i)$ は同様な3項系列の集合であり，$\Gamma_F(i),\Gamma_J(i)$ は次のように定義される．

［定義4］ 集合 $\Gamma_F(i),\Gamma_J(i)$

$\Gamma_F(i)=\{(i,j,\alpha) \mid i<j\leq n, WS(i,j,\alpha), \neg J(\alpha)\}$

$\Gamma_J(i)=\{(i,j,\alpha) \mid i<j\leq n, WS(i,j,\alpha), J(\alpha)\}$ □

集合 $\Gamma_F(i),\Gamma_J(i)$ を求めることは，各々付属語辞書検索ルーチン，自立語辞書検索ルーチンに入力記号列 $\mathcal{O}$ の部分列 $\mathcal{O}(i+1)\mathcal{O}(i+2)\cdots\mathcal{O}(n)$ を与えて，その最左部分列としての付属語，自立語の位置及び単語構造を全て求めることに相当する．

［アルゴリズム］

初期状態　ANSLIST＝NIL，STACK＝ϕ

入力　記号列 $\mathcal{O}=\mathcal{O}(1)\mathcal{O}(2)\cdots\mathcal{O}(n)$

出力　ANSLIST

(1) $\Gamma_J(0)$ を求める．

(2) $\Gamma_J(0)$ の各要素の優先順位を求める．優先順位が最高の要素 $(0,j,\alpha)$ について，

ANSLIST＝ANSLIST・$(0,j,\alpha)$

とし，残りの要素を優先順位の低いものから順にSTACKにプッシュする．

（以下ANSLISTの末尾を (i,j,α) とする）

(3) $j=n$ ならば，

$E(\alpha)$ ならばANSLISTを出力して停止する．

$\neg E(\alpha)$ ならば(9)へ．

(4) $\Gamma(j)=\Phi$ とする．

(5) $\Gamma_F(j)$ を求め，$\Gamma_F(j)$ の各要素 (j,k,β) について $C(\alpha,\beta)$ ならば

$\Gamma(j)=\Gamma(j)\cup\{(j,k,\beta)\}$

とする．

(6) $E(\alpha)$ ならば $\Gamma_J(j)$ を求め，

$\Gamma(j)=\Gamma(j)\cup\Gamma_J(j)$

とする．

(7) $\Gamma(j)=\Phi$ ならば(9)へ．

(8) $\Gamma(j)$ の各要素の優先順位を求める．優先順位が最高の要素 (j,k,β) について，

ANSLIST＝ANSLIST・(j,k,β)

とし，その他の全ての要素を優先順位の低いものから順にSTACKにプッシュしたのち，(3)へもどる．

（バックトラックの手続き）

(9) STACK＝Φ ならば，ANSLIST＝NILにしてANSLISTを出力したのち停止する．

(10) STACKの先頭をポップする．これを (l,m,β) とする．

(11) ANSLISTの末尾の項目の第2項目が l に等しくなるまでANSLISTの末尾の項目を取り去る．

(12) ANSLIST＝ANSLIST・(l,m,β) にして(3)へもどる． □

アルゴリズムにおいて記号'・'は連接の操作を示す．例えば，ANSLIST＝(A B C)のとき，

ANSLIST＝ANSLIST・D

を実行した結果はANSLIST＝(A B C D)となる．

アルゴリズムを実行した結果，ANSLIST＝NILならば入力記号列は解析不能であり，その他の場合はANSLISTに入力記号列の形態素解析の結果が得られる．ここで示したアルゴリズムは一つの解析だけを求めるアルゴリズムであるが，複数個の解析が存在する場合には(3)でANSLISTを出力したのちに(9)から実行を再開することにより全ての解析を求めることができる．

ステップ(2)，(8)における優先順位の評価は，

i) 自立語よりも付属語を優先する．

ii) 入力文における単語の長さが長いものを優先する．

iii) 活用する語を優先する．

とした．ここで評価の順序はi, ii, iiiの順である．

このアルゴリズムは一般的なバックトラック方式のアルゴリズムであるから詳しい説明は省略し，ANSLISTの意味について述べる．

［定義5］ 述語C

単語構造α, βに対してC(α, β)であることは，C(α, β)であるか，又はE(α)かつJ(β)であることと等価である．□

［定義6］ 述語LB

4変数述語LBを次のように再帰的に定義する．

i) WS(0, j, α)かつJ(α)であることとLB(0, j, α)であることは等価である．

ii) LB(i₁, i₂, α)かつWS(i₂, i₃, β)かつC(α, β)ならばLB(i₂, i₃, β)である．

iii) 述語LBはi, iiで定義されるものだけである．□

整数iと単語構造αが存在しLB(i, j, α)のとき入力記号列Oの部分列σ = O(1)O(2)…O(j)は左文節列をなすといい，さらにE(α)であるようなαが存在するときσは文節列をなすという．

［定理1］

ANSLISTの末尾の項目が(i, j, α)ならば，LB(i, j, α)である．□

証明はiに関する帰納法で容易にできるので省略する．

4.3 単語辞書

実験に用いた単語辞書は自立語辞書と付属語辞書からなる．自立語辞書には約83,000語の自立語が収納されており，付属語辞書には約300語の付属語，形式名詞，補助動詞が収納されている．本稿では単語辞書のデータ構造の詳細については述べないが，一回の検索ルーチンの実行で集合F(i), F₂(i)を能率良く求めることができるデータ構造を用いている．

4.4 入力文

実験に用いた入力文としては，武者小路実篤の「人生論」から1,000文を用意した．入力文は仮名表記のべた書きであり，一文の平均長さは44文字で，平均20文字毎に読点'、'で分かち書きされている．

5. 実験結果

本章では4.で述べた最長一致法と文献(4)で述べた文節数最小法の比較を行った解析実験の結果について報告する．

5.1 文節数最小法の有効性

1,000文の入力文について実験した結果，960文は文節数が最小となる解析の中に正解が存在した．ここで正解とは原文と一致する解析である．ただし，原文との比較は品詞，活用情報の

原文	解析	件数
その|人	其の日と	13
良く|して	治して	5
子が|ない	漕がない	2
気が|する	起臥する	2
～の|ない|人	～の|内及と	2
この|蚊の	子のかの	1
その|産む|能力	国生|無能力	1
死|その|もの	始祖の|もの	1
見て|嫌に	未更やに	1
もう|あとは	盲唖とは	1
具合|良く|育って	具合よ|養だって	1
目|座着	名車	1
本屋が|良い|本を	本屋がよ|要本を	1
招き|易い|点だ	招きや|水天だ	1
そう|たいした	早退した	1
しかし|金と	庵レかぬと	1
どうか|すると	同化すると	1
ーフーフ|書く|気は|ない	ーフーフか|基はない	1
どう|言う|人を	同意|雨飛とを	1
甲の|見た|美が	甲のみ|旅が	1
男が|何か	男仮名|か	1

表-1

みを対象とし，綴りの区別はしないものとする．例えば，'きかい'に対しては'機械'，'機会'などの名詞が対応するが，これらの区別はしていない．文節数が最小となる解析に正解が得られなかった解析を表-1に示す．ここで'|'は文節の切れ目を示す記号である．表-1において原文'その人'の正解が得られなかった原因は，自立語辞書に一単語として'其の日'が登録されていたことにある．このような複合語を自立語辞書に登録する場合には，文節数に相当する荷重の情報を付加する必要があると思われる．表-1からも明らかなように実験した1,000文に対しては，正解は文節数が最小となる解析か，その次に文節数が少ない解析の中にすべて含まれている．このことは文節数最小法の有効性を十分に示している．

文節数が最小となる解析には，入力文字列の長さ10文字に対して平均2通りのあいまいな解析が存在する．

5.2 能率の比較

図-3に入力文字列の長さに対して，解析に必要としたメモリ数のグラフを示す．文節数最小法に対しては，文節数が最小となる解析のみを求めるのに要するメモリ数が示してある．

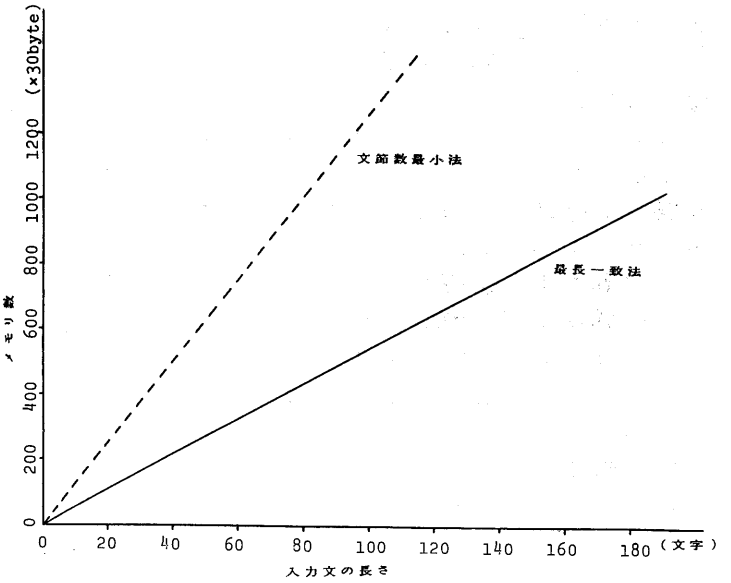

図-3 メモリ数の比較

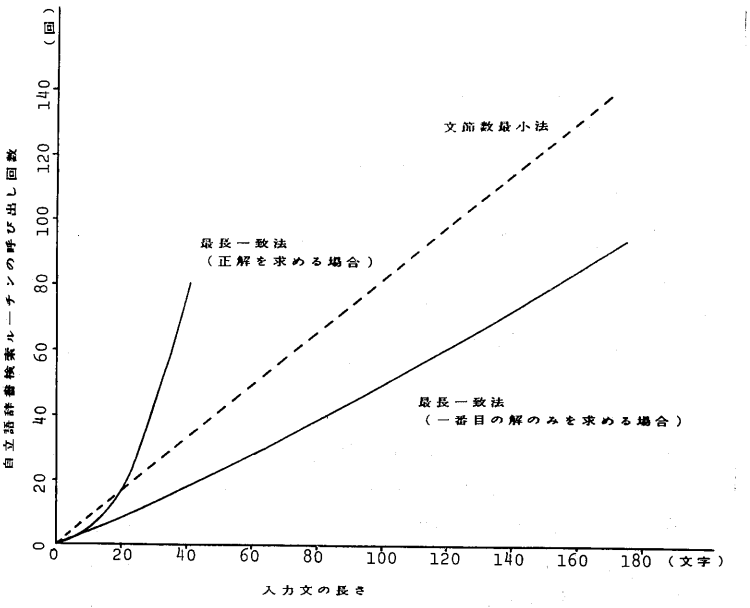

図-4 自立語辞書の検索回数の比較

現在の計算機技術においては，本稿で述べた形態素解析の処理時間はほとんど二次記憶の検索時間で費される．そこでステップ数の比較は自立語辞書の検索回数の比較で行った．入力文字列の長さに対して，解析中に行った自立語辞書の検索回数のグラフを図-4に示す．最長一致法については一番目の解を求めるのに要した検索回数と正解を求めるのに要した検索回数を示す．

5.3 解析の品質の比較

解析の品質の比較として，ここでは一番目に出力される解析の誤った部分

の入力文字列における長さを比較する．誤った部分の入力文字列における長さの入力文字列長に対する割合の平均は，最長一致法で12.4％，文節数最小法では7.0％である．ただし，文節数最小法では第一番目の解析を選択するために次のような評価をした．

i) 文節数が少ないものを優先する．
ii) 自立語の次は付属語を，付属語の次は自立語を優先する．
iii) 活用のある語を優先する．

ここで評価の順序はi, ii, iiiの順である．

5.4 尤度の評価

2.で最長一致法は文全体に対する尤度の評価ができないことを述べたが，300個の入力文について，正解が一番目と二番目に出力された文の数を比較したものを表-2に示す．この場合の文節数最小法による尤度の評価はほぼ同様である．

	最長一致法	文節数最小法
第一番目	73	124
第二番目	3	28

表-2 第一番目，第二番目に出力される解析に正解が得られた文数

6. あとがき

5.で示した実験結果より，第一番目の解を求めるのに要する処理時間，必要とするメモリ数の点では最長一致法が優れている．しかし，正解を求めるまでに要する処理時間や解析の品質，つまり尤度の評価の点では文節数最小法が優れている．これまで形態素解析の解析結果に対する尤度の評価には，ほとんどの場合最長一致法が用いられてきたが，文節数最小法も同等又はそれ以上の能力を持っている．今後その利用法などに関する研究が必要であろう．

最後に，本実験に使用した自立語辞書の作成に初期の頃から従事し，以来十数年間これに専念して来られた九州芸工大稲永紘之講師，小西彬充助手，最長一致法の実験に関して多大な御協力をいただいた日本ユニバックの吉田正行氏をはじめ実験システムの実現のために多くの方々の協力をいただいた．深く感謝の意を表する．

なお，本研究は昭和56～57年度21世紀文化学術財団学術奨励金および昭和56～57年度文部省科学研究費一般研究(B)によった．

7. 参考文献

(1) Aho, A.V., Ullman, J.D.: "The Theory of Parsing, Translation and Compiling, Volume 1 : Parsing", Prentice Hall, 1972.

(2) J.B. Lovins: "Development of Stemming Algorithm", MT, Vol.11, 1968, 22-31.

(3) 栗原，黒崎："仮名文の漢字混り文への変換について"，九大工集，Vol.39, 4, 1967, 659-664.

(4) 吉村，日高，吉田："表方式を用いた文節構造分析アルゴリズムとその能率について"，情報処理，計算言語学研資 25-6, 1980.

(5) Hitaka, T., Yoshida, S.: "A syntax parser based on the case dependency grammar and its efficiency", Proceedings of the COLING 80, 1980, 295-302.