自 然 言 語 処 理 104－4
（1994．11．17）

複数候補を出力する非文訂正法のOCR出力の誤り訂正への応用とその候補選出について

渥美 清隆　　増山 繁
atsumi@smlab.tutkie.tut.ac.jp　masuyama@tutkie.tut.ac.jp

豊橋技術科学大学 知識情報工学系

〒441 愛知県豊橋市天伯町字雲雀ヶ丘1-1

本稿において，我々は日本語のOCR出力誤りを検出するための方法について提案する．この方法は，我々が開発した複数の誤り推定候補を出力する非文訂正法に基づいた形態素解析である。この非文訂正法は多くの誤り推定候補を出力するため，この複数の候補からの絞り込みについても述べる．

An Application of Error-Correcting Technique Generating Plural Candidates to Error-Detection for Japanese Text Read by OCR Systems and A Candidate Reduction Method

Kiyotaka ATSUMI　　Shigeru MASUYAMA

Dept. of Knowlege-based Information Engineering, Toyohashi Univ. of Tech.

1-1, Hibarigaoka, Tempaku-cho, Toyohashi-shi, Aichi-Ken, 441 Japan.

In this paper, We propose an error-detection method for Japanese text read by OCR systems. This error-detection method is a morphological analysis based on an error-correcting technique generating plural candidates proposed by us. We propose also a candidate reduction method among generated candidates because this technique generates a lot of candidates.

1 はじめに

イメージスキャナなどを用いた文字認識技術は年々向上し，その需要も増えつつあるが，文字認識後の複数候補の選択や，誤って選択した文字を検出する機構が，現在のところ十分に発達しているとは言い難い．このような誤り検出をするための研究として，隣接文字間の接続確率を用いた方法 [1] などが成果を上げているが，確率の学習を十分に行なわせるためには，かなりの量のコーパスが必要である，連接情報だけでは充分な誤り訂正ができない [2] 等の理由により実用段階には至っていない．その他，最小文節法に基づいた方法 [5] では，唯一の誤り推定候補のみを出力し，かつ，それが十分に適切な候補ではなかったため誤り検出に十分な能力を示してはいない．

一方，文脈自由文法上の非文訂正法として，文献 [3, 4] などが提案されている．これらの方法は，元の言語を文脈自由文法上の文法として規定できれば，それを誤り訂正用の文法に変換することにより，誤った文が入力されても，正しい文が入力されたときと同じように解析可能となる．しかもこれらの方法は，どこに誤りがあったのかも知ることができ，誤り訂正の方法についてのヒントも与えてくれる．特に我々が開発した方法 [4](以下，渥美らの方法とする) は，後で示す誤り訂正演算の使用回数が最小の場合だけの解析に留めず，最小から最小 $+d$ までの誤り訂正演算の使用回数までの解析結果をすべて出力する．このように解析範囲を拡大し複数個の出力を行うことにより，文法的に正しい解析のみならず意味的にも正しい解析が含まれる可能性が大幅に増加することが期待できる．このような方法が現在まであまり注目されていなかったのは，入力列の長さを n とするとき，実行時間が $O(n^3)$ もかかることや，使用する記憶容量が $O(n^2) \sim O(n^3)$ になり，適用が困難であったためだが，近年の計算機の高速化，大容量化に伴ない，十分実行可能な環境が整ったと考えている．そこで，我々は文献 [6] において渥美らの方法を形態素解析に応用した誤り検出手法を提案し，その手法について計算機実験を行ったことについて述べたが，その結果，計算機が出力した誤り推定候補が多過ぎ，そのままでは実際にうまく活用出来ないことが分かった．そこで，本稿では文献 [6] で実験した手法について述べ，さらに誤り推定候補の絞り込みの方策について述べる．

2 文脈自由文法上の非文訂正法

1節でも述べたように，文脈自由文法上の非文訂正法については [3, 4] によって議論されている．これらは，いかなる言語においても文脈自由文法の範疇において定義が可能であるとき，いかなる入力に対しても定義された文法に対して適切になるような訂正を行うことを保証するものである．

渥美らの方法では，一般文脈自由文法の解析手法として知られる Earley 法を拡張している．Earley 法は CKY 法などに比べ使用する記憶容量や実行速度の点で係数レベルで劣るが，最初に規定した文法をそのまま採用できるため，解析木の形が崩れないことや，欠落誤りに対応するための ε 導出が扱えるなどの利点を持っている．これらの利点は自然言語解析をする上では重要な要素と成り得る．ここでは，いかに Earley 法に非文訂正の能力を持たせているのかについて述べる．

2.1 非文の定義 [3]

渥美らの方法における正文とは，定義された文法によって認識できる文であると定義し，非文とは，任意の終端記号により構成される列のうち，正文ではないものであると定義する．また，いかなる非文も表 1のような 3 種類の誤りの組合せにより，正文から非文に変更した文であると見なすことができる [3].

この 3 種類の演算の逆演算を誤り訂正演算とする．最小誤り訂正演算回数による構文解析とは，これら 3 種類の誤り訂正演算を使用して，入力された非文から出発して最も近い正文にたどりつくのに必要な最小限の誤り訂正演算を利用して解析木を生成することであり，これを最小誤り訂正と呼ぶことにする．また，許容度 d の準最小誤り訂正演算回数による構文

表 1: 誤り訂正演算の種類

正しい文	誤った文	種類
$\cdots w_i\ w_{i+1}\cdots$	$\cdots w_i\ x\ w_{i+1}\cdots$	挿入
$\cdots w_{i-1}\ w_i\ w_{i+1}\cdots$	$\cdots w_{i-1}\ w_{i+1}\cdots$	欠落
$\cdots w_{i-1}\ w_i\ w_{i+1}\cdots$	$\cdots w_{i-1}\ x\ w_{i+1}\cdots$	置換

入力文を $w_0\ w_1\ w_2\cdots w_{n-1}\ w_n$,
$w_0, w_1, \cdots, w_n, x \in \Sigma$,
Σ は終端記号集合とする.

解析とは, 入力された非文からある正しい文に最小回数 $+d$ 以内の誤り訂正演算の使用回数を利用して解析木を生成することであり, これを準最小誤り訂正と呼ぶことにする.

2.2 非文訂正用文法への拡張 [3]

ある言語 L を認識するための文脈自由文法 G が存在するとき, この文法 G を変換することで, 非文訂正用の文法 G' に拡張することができる. この文法 G' はいかなる終端記号列をも認識し, その認識過程において言語 L に属するための訂正手続きを行なう. この文法 G' への拡張について簡単に述べる.

文脈自由文法 $G = (N, \Sigma, S, P)$ で, N は非終端記号の集合, Σ は終端記号の集合, S は出発記号, P は導出規則の集合である. この文法から新らしい文法 $G' = (N', \Sigma \cup \{\varepsilon\}, SP, P')$ を作るために以下の作業を行う.

1. $A \to \alpha_0\beta_1\alpha_1\beta_2\alpha_2\cdots\beta_i\alpha_i \in P$, $0 \le j \le i$, $\alpha_j \in N^*$, $1 \le k \le i$, $\beta_k \in \Sigma$ のとき, β_k を非終端記号 E_{β_k} に置き換えた規則 $A \to \alpha_0 E_{\beta_1}\alpha_1 E_{\beta_2}\alpha_2\cdots E_{\beta_i}\alpha_i$ を P' に追加する.

2. 1によって変換された非終端記号 E_a が存在するとき, 以下の規則を P' に追加する.
   (a) $E_a \to a$
   (b) $E_a \to x$, $\forall x \in \Sigma$
   (c) $E_a \to \varepsilon$

3. 以下の規則を P' に追加する.
   (a) $SP \to S$, $SP \to S\ H$
   (b) $H \to H\ I$, $H \to I$
   (c) $I \to x$, $\forall x \in \Sigma$

4. 2b, 2c, 3cで追加した導出規則はそれぞれ置換誤り, 欠落誤り, 挿入誤りを訂正するための誤り訂正演算としてマークする.

以上の処理により, 文法 G' の導出規則の集合が作成できる. 出発記号を SP とし, 非終端記号を $N' = N \cup \{E_x | \forall x \in \Sigma\}$ を定めることにより文法 G' を作成できる.

2.3 構文解析の拡張 [4]

誤り訂正用に拡張した文法 G' は文脈自由文法なので, Earley 法のままでも解析できるが, それでは準最小誤り訂正の解析木を作成することができない. そこで, 構文解析に誤り訂正演算の使用回数を数える機構を付け加え, その中から準最小誤り演算の使用回数のものだけを数え上げることにより, 準最小誤り訂正の解析木が作成できるようになる. この誤り訂正演算を数え上げる部分を, Earley 法における解析テーブルに作成される要素に対して拡張することにより実現する. Earley 法では入力列の長さを n とするとき, $n+1$ 個の解析テーブルと呼ばれる集合の枠を作成し, そこに要素を生成しながら解析を進める. そのとき生成する要素は $[A \to \alpha \bullet \beta, p]$, $A \to \alpha\beta \in P'$, $\bullet$ はそこまで解析が完了していることを示し, p はこの要素を生成した解析テーブルのポインタである. この要素を以下のように拡張する.

$$[A \to \alpha \bullet \beta, p, E]$$

$$E = \{e, b_0 b_1 \cdots b_d\}$$

ここで, e はこの要素を生成するための2.2節でマークした誤り訂正演算の最小使用回数であり, $b_0 b_1 \cdots b_d$ はビット列で, もし b_i が1ならば $e+i$ 個の誤り訂正演算を使用してもこの要素が生成できることを意味する. このように拡張した後は, Earley 法による解析とともに, 動的計画法の考え方に基づき, 入力列に対する同じ部分列を範疇とする部分木の根

に相当する要素の誤り訂正の使用回数, つまり E の部分が準最小になるように計算することで, 準最小誤り訂正の解析木を生成することが出来る. 詳しくは文献 [4] に述べられているので参照されたい.

3 非文訂正法の日本語への応用

2節で述べたように, 文脈自由文法上での非文訂正が可能であることが明らかになった. そこで, 日本語への応用を考えることにする. 日本語への応用には単語の区切りがないいわゆるべた書きや, 日本語を規定する文法の定義が問題となる. この点について以下に述べる.

3.1 べた書きへの対応

文脈自由文法上の構文解析手法はいわゆる単語単位の解析手法であり, 日本語のべた書き文では単語の区切りが曖昧なため, 文字単位の解析をすることになる. ところが文字単位の解析をすることを考えると, 辞書に登録されている見出し語のすべてを文法として記述することになり, 構文解析をする上では非常に非効率的である. そこで, 渥美らの方法において終端記号からの導出の部分を拡張することにより, 効率的な構文解析が行えるように工夫した.

渥美らの方法では i 文字目の終端記号 a を導出する場合, $[A \rightarrow \bullet a, p, E] \in I_{i-1}$ という要素があるとき, $[A \rightarrow a\bullet, p, E] \in I_i$ という要素を生成することで導出する. そこで, i 文字目から j 文字目までが辞書により品詞 b に置き換え可能であり, $[A \rightarrow \bullet b, p, E] \in I_{i-1}$ という要素が存在するならば, $[A \rightarrow b\bullet, p, E'] \in I_j$ という要素を生成する, というように拡張する. ここでのポイントは通常の構文解析では1単語あるいは1文字ごとに解析をすすめているが, 拡張した方法では複数の単語あるいは文字ごとに解析を進めることができることである. しかし, 挿入, 置換誤りの場合を考えるとき, 複数文字列が品詞という1つのラベルの元に同時に挿入, 置換されてしまうが, それを1回の誤り訂正演算として数えるのには問題がある. そこで, 置換誤りに対する終端記号の導出の場合には $E' = E + j - i - 1$ の計算により補正を加えている.

以上により, あらかじめ入力列のすべての部分列がどのような品詞を持ちえるのかを調査することにより, 効率的な構文解析が可能である.

3.2 文献 [6] の日本語文法の定義

文献 [6] で使用した文法は文節内文法を基本としたものである. 文節の定義にはいろいろとあるが, 今回は ICOT が作成した TRIE 辞書 [7] の辞書データ (約 15 万語) に記述されているインデックスを参考にして文法を作成した. この文法は動詞の活用などについては十分に記述したが, 付属語に関してはこれを精密に記述すると導出規則が複雑になるために, 一括し単純化することで文法サイズを縮小した. 導出規則の数は 54 個, 終端記号に相当する品詞の数は 152 個を定義した.

3.3 日本語文法の拡張

文献 [6] の実験結果から, プログラムの出力する誤り訂正候補が多過ぎるために, 逆に十分な誤り検出を困難にしていることが分かった. これについて主な理由としては, 3.2節で定義した文法が, 文節間の関係に対して何の制約も課していないために, 可能性のあるあらゆる文節の区切りを候補として出力してしまったからである. そこで, 今回使用した新しい文法では非常に簡単な文節間制約として, 「文末表現となっている文節は文末にあるべきだ」という制約を文法の中に埋め込んだ. これによって, 制約のある文法が制約のない文法に比べて, どのような変化をするのかについて実験で明らかにする.

4 実験

文献 [6] がどの程度の誤り検出能力を持つのか, また本稿において拡張した日本語文法によって出力がどのように変化をするのかを検証するために, 計算機実験を行った.

まず, 我々が提案する誤り検出手法をプロ

グラム化した．このプログラムに誤りの含まれている文を入力し，許容度1の準最小誤り訂正の範囲の解析を行うと，次のような結果が得られる．

２９００ｋｍの　彼方には　(地球の中で)　(一番＝的な)
２　９００ｋｍの　彼方には　(地球の中で)　(一番＝的な)
２９　００ｋｍの　彼方には　(地球の中で)　(一番＝的な)
⋮

ここで入力した文は他の入力文も含めて日経サイエンス1993年7月号pp.66～77「核-マントル境界領域」を市販のパソコン用のOCRシステムに入力した結果から得られた文である．出力結果において，スペースは文節の区切りを表し，かっこで囲まれている部分は，そこに誤りが含まれていると推定した部分である．この入力文では「一番＝的な」の「＝」が認識不能文字であり，ここに誤りが存在する．この文の誤り推定候補は，全部で25個であった．

このプログラムが出力する結果は，先に示したように複数候補を出力するので，これを単純に眺めるだけでは，どの程度うまく行くのかを見極めることができない．そこで，各文字について何回誤りが含まれていると指摘されたか，つまり，かっこの中に何回存在したかを数え上げ，それを百分率換算の頻度にしたグラフを図1に示す．

この図からも分かるように，19文字目「＝」の誤り推定が100% を示しており，この辺りに誤りが含まれていることが確信できる．これは，この文字を含むいかなる部分文字列も辞書に登録されていないためである．

それでは，図1に示した文と異なり，辞書に登録されていないような文字が存在しない例ではどうなるのかを図2に示す．この例では138個の誤り推定候補を出力し，6文字目「く」が100% になっている．これは，「く」の直前の「産」という字が本来「遠」であるのに，置換誤りを起したため文法上結び付きが

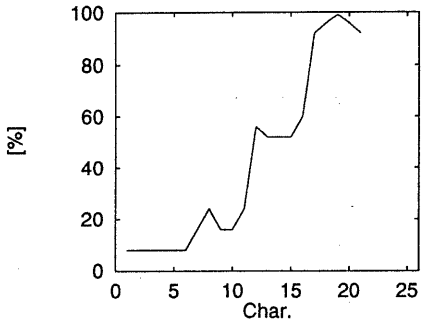

図1:「２９００ｋｍの彼方には地球の中で一番＝的な」の誤り推定

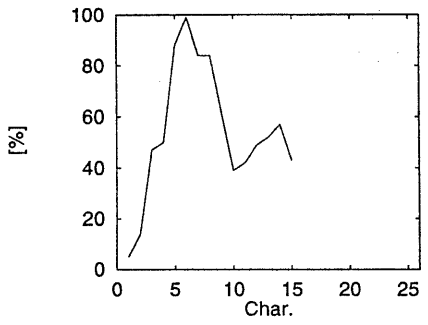

図2:「地上から産く離れたこの場所は，」の誤り推定

許されなくなり，このような結果となった．

さて，誤りを含む例のうち，比較的発見しやすいものについては例で述べた通りだが，正しい文を入力した場合の結果を図3に示す．この例では，誤りが含まれていないことを暗示するように，比較的平坦なグラフとなっている．ただ，この例では誤り推定候補が95個と多いことも特徴となっている．そこで，文節間制限を加えた文法を用いてプログラムを実行すると，誤り推定の頻度は図4に示すようにほとんど変化せずに，候補数が64個と減ることが分った．つまり文節間の制限は文節の切れ目を制限し，そのために無意味な候補の出力が制限されたと考えられる．

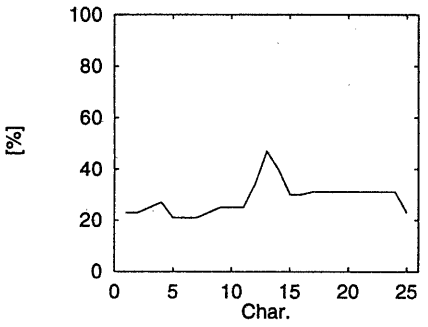

図 3: 文節間制約のない文法の「もし車で行けるとすればたった 3 日しかかからないが, 」の誤り推定

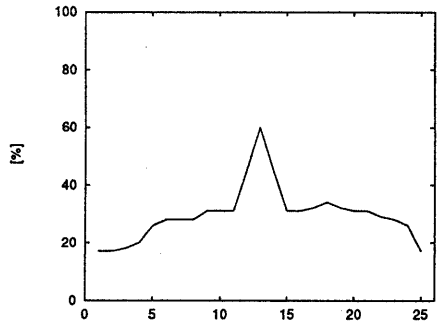

図 4: 文節間制約のある文法の「もし車で行けるとすればたった 3 日しかかからないが, 」の誤り推定

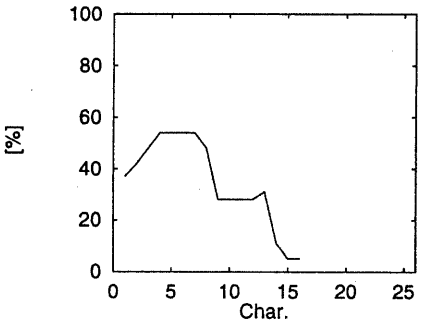

図 5: 「最下邦マントルと外核上部が接する」の誤り推定

最後に本プログラムがうまく推定できない誤りと文法の拡張による副作用について, それぞれ 1 例ずつ挙げる. うまく推定できない例としては, 図 5であり, 3 文字目「邦」が誤っているが, 誤り推定の頻度は 48% 程度と低い. 全体的に見たときにも, 前半の文字列の中に誤りが含まれているようにも見えるが, それでも十分な数字とは言えない. これは, 「邦」とい文字が文法上, 接尾語としても機能しているためであり, このような誤りは, 本稿の提案するプログラムでは十分に誤りを指摘することができない.

また, 文節間制約を加えた文法における副作用の例は図 6, 図 7である. この例の入力文には誤りは含まれていないが, 文節間の制約のある文法で解析を行うと 8 文字目付近に誤りが含まれているように見える. これは「している」という文字列が文末表現であると見做されたためであり, そのために, この部分に誤りが含まれるという推定をさせたのではないかと考えられる.

5 むすび

文献 [6] の誤り検出手法により, ある程度の誤り検出を行うことができることを確認し, 本稿で提案する文節間制約によって, 誤り推定の候補をその精度を失うことなく減らすことができることも確認した. しかし, 次のような問

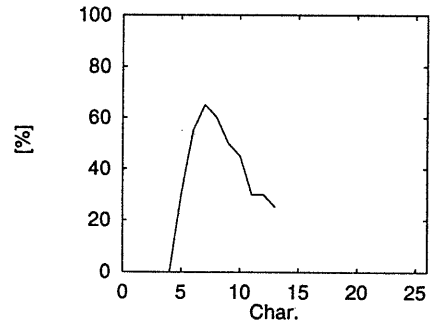

図 6: 文節間制約のない文法の「変化をしている場所がある。」の誤り推定

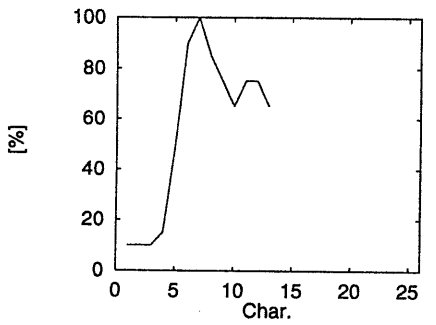

図 7: 文節間制約のある文法の「変化をしている場所がある。」の誤り推定

題点があることも分った.

- 現在定義している文法的な制約では捉えきれないような誤りが存在するときは, それを指摘することが難しい.
- 文節間制約の記述が不正確なために, 副作用として誤った誤り推定を行う.
- 複数の候補を出力した後の結果の加工方法が決まっていないので, 実際の誤り推定を行うことができない.

このうち最初の 2 点については, 今後日本語文法の精密化に努めればある程度解消されるだろうと思われるが, 隣接した文節間の制約だけでなく, その文節と隣接はしていないが同一文中の文節との間の制約も記述する必要が出ると思われるので, それについては拡張格文法などの導入を検討する. また, 3 番目の問題点については, 最初の 2 点の解決により誤り推定の候補数を減らした後, 今回のグラフ表示にしたような頻度情報を利用することを検討する予定である.

参考文献

[1] T.Araki, S.Ikehara, N.Tsukahara and Y.Komatsu: "An Evaluation to Detect and Correct Erroneous Characters Wrongly Substituted, Deleted and Inserted in Japanese and English Sentences, Using Markov Models", *PROCEEDINGS of COLING 94*, Vol.1, pp.187-193, Aug., 1994.

[2] 荒木, 池原, 塚原, 小松: "マルコフモデルを用いた OCR からの誤り文字列の訂正効果", 情処研報, 94-NL-102, Vol.94, No.63, pp.97-104, Jul., 1994.

[3] A.V.Aho and T.G.Peterson: "A Minimum Distance Error-Correcting Parser for Context-Free Languages", *SIAM J.Comput.*, pp.305-312, Dec., 1972.

[4] 渥美, 増山: "構文解析上の自由度をもった非文訂正法の一提案", 信学論, Vol.J76-D-I, pp.686-688, Dec., 1993.

[5] 渥美, 増山: “文節文法を用いたイメージスキャナの読み取り結果の誤り検出”, 情処第48回全国大会, Vol.3, pp.39-40, Mar., 1994.

[6] 渥美, 増山: “文脈自由文法上のある非文訂正技術を用いたOCR出力後の誤り検出”, 情処第49回全国大会, Vol.3, pp.221-222, Sep., 1994.

[7] (財) 新世代コンピュータ開発機構: “TRIE辞書ユーティリティ”, 1991.