自然言語処理 121－3
（1997. 9. 11）

共起情報と統計的形態素解析によるＯＣＲ誤り訂正

竹内孔一　松本裕治

奈良先端科学技術大学院大学 情報科学研究科
〒630-01 奈良県生駒市高山町 8916-5
E-mail : {kouit-t,matsu}@is.aist-nara.ac.jp

あらまし
近年のインターネットの普及によりＯＣＲによるテキストの電子化はますます重要な処理となってきた．ＯＣＲテキストの誤り訂正の研究は特に英語圏で進められて来たが，1）日本語で英語のような単語間の明示的な区切りを用いないこと，2）文字種が豊富であること，から英語圏で開発された方法を日本語にそのまま適応することはできない．一方，日本語に関する先行研究ではＯＣＲの内部候補文字を用いたり，解析済みコーパスが必要なものがある．そこで，本研究ではＯＣＲの文字候補を用いずに入力文字列を訂正するシステムを提案する．本システムは，新聞記事コーパスで学習した文字 trigram，統計的形態素解析システム，単語 bigram，単語 trigram，単語共起 の各モデルを使用する．ランダムに文字置換したテキストに対して誤り訂正を行った結果，テキストの改善が観測された．
キーワード　統計的形態素解析，ＯＣＲ誤り訂正，コーパス，単語共起分布

OCR Error Correction Using Stochastic Morphological Analyzer with Probabilistic Word Model

Kouichi Takeuchi　Yuji Matsumoto

Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-01, JAPAN
E-mail : {kouit-t,matsu}@is.aist-nara.ac.jp

Abstract
In recent years, OCR error correction is becoming more and more important technique for the purpose of convert printed texts into electronic texts on computers. OCR error correction of Japanese texts is more difficult than that of English texts, because 1) Japanese texts have no white space between the words and 2) Japanese texts are written in a far richer set of characters. This paper presents OCR error correction system which uses Stochastic morphological analyzer, character trigram, word bigram, word trigram, and word co-occurrence statistcs. These stochastic models are learned using a large newspaper text corpus. We apply our system to texts which included random character substitution and observe error corrections.
key words　Stochastic Morphological Analyzer, OCR error correction, corpus

1 はじめに

近年，インターネットの普及により一旦情報を電子化してしまえば，世界規模で即座に誰でもその情報を簡単に利用できる世の中になりつつある．その中でも特に文字情報は根本的な情報伝達媒体であるため，文字情報の電子化はかなりの重要な課題である．最近の出版物は基本的にワードプロセッサを利用して作成されるため電子化には問題がないが既に紙として印刷されてきた過去の文献にも重要なものは多く，これらを電子化する事は重要な課題である．最近出現した電子化図書館では，印刷物を画像化しているが画像では文字による情報検索を行うことができず，ＯＣＲによる読み取りによる文字列の獲得が必要となる．本研究では，確率的な形態素解析器と単語や文字の統計的共起情報を用いて日本語のＯＣＲの文字読み取り誤りを訂正する方法について報告する．この際，ＯＣＲの画像情報（画像的な文字間の類似度）は利用せず文字情報のみを利用して訂正を行うためこの技術はそのまま日本語の誤字・脱字訂正の研究にも役立つ．

英語におけるスペルチェックと訂正の技術は大変進んでおり Kukich[1] にまとめられている．英語は日本語と異なり，単語ごとに区切りがあるため，単語単位で解析を行うことができる利点がある [3]．これにより単語の連続分布の利用のように単語の共起頻度を利用する方法 [4] などが提案され効果を上げている．しかし日本語では単語間の区切りが明示されないので文字の誤りにより単語区切りが変わってしまうため英語で用いられている方法を直接用いることができない．

日本語については単語単位ではなく文字の n-gram を応用した確率モデルが日本語文の文字誤りの検出・訂正に対して有効であることが明らかとなっている [8][5] [7]．また，文字種が英語に比べて多いことから，漢字やカタカナなど文字種に応じた訂正法に関する研究も行われている [6][10]．また，形態素解析を用いて訂正を行う方法も提案されている [9][2] ．永田 [2] は，ＯＣＲの文字候補付きの出力結果と品詞 trigram に基づく形態素解析システムを用いて未知語を含んだテキストデータに対して高い精度の誤り修正を行った．しかし，この方法では統計的形態素解析システムを学習するために解析対象となる文章と同じ分野のタグ付きコーパスが必要となる．

そこで我々はタグ付きコーパスを必要とせず，さらに画像的な文字間の類似度情報を用いないＯＣＲ文字読み取り誤り訂正法を提案する．つまり，ＯＣＲからは第一解の文字列だけを受取るだけとする．誤り訂正システムは３つの部分から構成されている．最初に１）文字誤り検出を行い，次に２）文字誤り候補を生成する．最後に３）候補の選択を行う．文字誤りの検出ならびに候補出力は全て文字 trigram モデルで行った．最後の候補選択については文字 trigram モデル，単語に関する統計的モデル（単語 bigram，trigram，共起頻度モデル），統計的形態素 trigram モデル を利用した．これらの学習には大量の新聞記事テキストデータを用いた．誤りは置換誤りのみを対象とし，挿入と削除誤りには対応していない．訂正実験には学習に使用しなかった新聞記事を利用してランダムに誤りを生成したデータを用いた．訂正実験の結果，単純な文字 trigram モデルや統計的形態素解析モデルよりも，統計的形態素解析 trigram モデルに単語に関する統計的モデルを重ね合わせたシステムが高い訂正能力を発揮することを示す．

2 方法

我々の提案するシステムは日本語の入力文字列のみを受取りその誤りを検出し訂正を行う．よってＯＣＲの誤り候補を用いないため誤り箇所の推定と候補出力の機構を持つ必要がある．そこで，誤り訂正システムを

1 文字誤り箇所の検出 (Detection)
2 文字候補の生成 (Genaration)
3 候補の選択 (Selection)

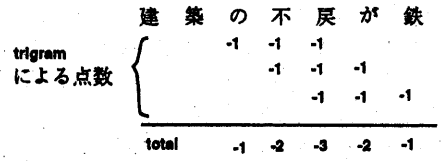
の３部分に分割し，この順に解析を進める．つまり (1) で誤りと判断された文字のみ候補を生成し選択を行うので誤りと判断されなかった場合は解析対象としない．これは誤りを見過ごす可能性もあるが，後に示すように文字 trigram モデルがかなりの精度で誤り候補を検出している．以下に各モデルについての詳細を記述する．

2.1 文字誤り箇所の検出 (Detection)

よく用いられている方法 [8][5] として，文字誤り箇所を文字 trigram で検出する場合，文字 trigram (n-gram) の確率値に対して足切り値を設けて低確率の接続箇所を誤りとしている．つまり，文字列 c において $P(c_i|c_{i-2}, c_{i-1})$ が T(足切り値) 以下なら

c_i を誤り文字と決定する方法である．しかし，真に文字 c_i のみが誤る分けでなく c_{i-1} や c_{i-2} が誤る場合もあるはずである．そこで，本システムでは文字 trigram の確率値に対する足切り値 T 以下の３文字列が出現した場合，その３文字列の最後の文字だけを誤りと見なすのではなく，その３文字全てを誤り文字の対象とする．そこで文字 trigram 確率が T 以下である３文字列に対して各々 -1 点を与え，これを文字列の文頭から順に当てはめて行き各文字においてマイナス点の多い文字を誤りとする（図１参照）．ここで $T=0$ とし，trigram 文字列の有無で判断した．

図１は文字訂正誤りシステムの具体的な計算例ならびに出力例を示している．図１下段の出力例では × 印は得点合計が -3 点の文字，△は -2 点の文字を表しており，-1 以下は○印で示している．



民間建築の不戻が鉄筋工事雨者を直撃している．
○○○○○△×△○○△××△○○○○○○○△

図 1: 文字誤り箇所の検出の出力例

2.2 文字候補の生成 (Genaration)

英語では誤り文字を修正する場合，わかち書きされているので誤った単語から単語中の文字列を利用した類似度により類推する事ができる．日本語でもカタカナ文字列などは適応できる [10] が，２字漢字が大量にある事，ならびに単語の境界が明示されない事から英語と同様の方法をとることはできない．そこで，単語の文字 trigram モデルを利用して候補文字を作成する．候補出しに対して辞書中の単語を元に文字候補を作成する方法をとらないため辞書に無い単語の文字も推定することができる．

以下では文字 trigram 確率モデルを利用して１文字，複数文字列の候補生成方法について説明する．また，これら文字列候補と誤り指摘箇所の文字列とを組み合わせる方法について述べる．

１文字の想起　目標とする１文字を m_i とする（ここで i は文字の場所を示す）と前後の文字列 $(c_{i-2},..,c_{i+2})$ を含んで，前方向の確率値

$$\begin{aligned} P_f(c_{i-2},c_{i-1},m_i,c_{i+1},c_{i+2}) \\ = \; P(m_i|c_{i-2},c_{i-1}) \times P(c_{i+1}|c_{i-1},m_i) \\ \times P(c_{i+2}|m_i,c_{i+1}) \end{aligned}$$

の上位５位を候補とする．さらに後ろ方向にも同様の計算を行い後方向の確率値

$$\begin{aligned} P_b(c_{i-2},c_{i-1},m_i,c_{i+1},c_{i+2}) \\ = \; P(m_i|c_{i+1},c_{i+2}) \times P(c_{i-1}|m_i,c_{i+1}) \\ \times P(c_{i-2}|c_{i-1},m_i) \end{aligned}$$

の上位５位を候補とする．よって前後あわせて１０候補を出力する．当然両方向から同じ文字が生成される場合があるが，それらは重ね合わせるため１０候補より候補数は減る事になる．

２文字以上の連続文字列を想起する場合　日本語文を考えると２字熟語が多くその前後を多頻度の助詞「の」や「は」で囲まれることが多いため丁度熟語が誤るとかなり候補の精度は落ちることが予測される．さらに３文字４文字となるとほぼ想起は不可能といえる．以下に説明する方法は n 文字列を生成することができるが実際には後の実験の章で連続文字列の候補生成能力を測定しその限界範囲内以上は行わない事にする．

具体的に２文字列の候補生成法の場合で説明する．目標とする文字列を m_i,m_{i+1} とし，前後の文字列を $(c_{i-2},..,c_{i+3})$ とする．１文字の場合と同様に文の左から右に向かって計算を行いその総計の確率値

$$\begin{aligned} P_f(c_{i-2},c_{i-1},m_i,m_{i+1},c_{i+2},c_{i+3}) \\ = \; P(m_i|c_{i-2},c_{i-1}) \times P(m_{i+1}|c_{i-1},m_i) \\ \times P(c_{i+2}|m_i,m_{i+1}) \times P(c_{i+3}|m_{i+1},c_{i+2}) \end{aligned} \quad (1)$$

の上位５個の $m_i, mi+1$ 文字列を候補として獲得すれば良い．さらに後方の場合も同様に行い上位５個の文字列を加えて１０候補作成する．これは３文字列以上も同様である．しかし，実際に計算の実行には多量の組合せが出るため，実際には途中で枝刈りを行っている．すなわち，$P(m_i|c_{i-2},c_{i-1})>0$ を満たす候補 m_i を取り，各候補 m_i について $P(m_{i+1}|c_{i-1},m_i)>0$ を満たす候補を作成すると

すぐに数千パスのオーダーになる．そこで各パスの合計確率値で上位300個のパスだけを残しながら計算を続けて行く．つまり中間のパス計算では300個の候補文字列が存在し最終段の総確率値で上位5個の文字列のみを採用する（図2）．

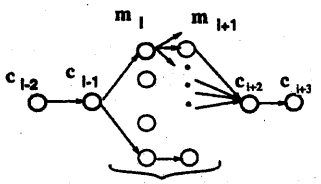


図 2: 文字誤り候補

候補文字の組合せ　図1の文字誤り検出の例を見ると1文字の誤り（「戻」→「振」，「雨」→「業」）でも周辺の文字も誤り訂正の対象となる．後の実験結果から×印と△印の文字を全て訂正の対象とするが，その中で実際に誤っている文字は1文字から全文字の可能性がある．そこで誤りを指摘している部分の文字列に対して，誤り文字を0文字から全文字列まで仮定し，それぞれについて候補文字を作成し，その組合せを候補とする．図3は「不戻が」の部分について示している．図中の「？」の部分が候補を作成する部分を表している．

後の実験で3文字列以上の誤りはほとんど候補を出しても有効でないことが明らかになる．その場合2文字列候補までの文字列を組み合わせて候補を作る．

2.3　候補の選択 (Selection)

作成された文字候補から正しい候補を選択するために確率的形態素解析システムと単語に関する統計的モデル（単語 bigram，trigram 確率と共起頻度）を利用する．これらの統計量の学習法について説明した後，使用した候補選択モデルについて記述する．

タグの無いコーパスで学習する方法　形態素解析システムや単語の統計量を持ち込む際，必ず問題となるのが単語のわかち書きの精度である．形態素解析は100%の精度では無いため解析に誤った単語列を生成することになる．しかし，ＯＣＲの文字誤り訂正において正しい単語列を生成する必要は必ずしも無い．何故なら同じ入力文字列に対しては確実に同じ単語列（と品詞列）を生成するため，その誤りを含んだ単語列の統計的分布を観測していれば元の文字列の特性を反映した結果を学習できることになる．そこで，上記の単語に関する統計的モデルを学習するときには，同じ形態素解析システムで学習コーパスを解析した出力結果を利用する．

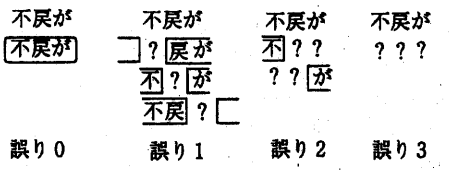


図 3: 候補文字列の組み合わせ

また，形態素解析システムは約95%程度の精度を持つ茶筌[11]があるが内部のコスト値が人手で作成されているため他の統計量と同様に確率的に扱えない．そこで，学習コーパスを 茶筌で解析させた結果出力される一意の形態素列をマルコフ学習して統計的形態素解析システムを作成する．

よって次の順序で学習を行う．まず1）人手による形態素解析システムを利用してコーパスから統計的形態素解析システムを作成する．次に2）統計的形態素解析システムで同じコーパスを解析し，その単語列を利用して単語 bigram，trigram，単語共起分布を獲得する．各々の数式を用いた表現を以下に記述する．

- 統計的形態素解析システム（形態素 trigram）

$$\begin{aligned} P(L) &= \sum_{W}\sum_{S} P(W,S) \\ &\approx P(W,S) \\ &= \prod_{i=1}^{n+1} P(w_i|s_i)P(s_i|s_{i-2},s_{i-1}) \end{aligned} \tag{2}$$

ここで L，W，S はそれぞれ入力文字列，単語列，形態素列を表している．また w は単語 s は形態素を表しており，品詞，活用型，活用形，単語表記の情報を持つ．学習の際は助詞，助動詞は単語表記まで観測している．また活用語は後方接続する語に対して活用形が影響を与えるのでそのときだけ活用形まで観測して接続確率を獲得する．s_{-1}，s_0 は文頭，

s_{n+1} は文末，w_{n+1} は空語を示している．茶筌の出力を利用して形態素接続確率，単語生成確率を学習する．

- 単語 bigram trigram モデル

$$\begin{aligned} P(W) &\approx \prod_{i=1}^{n+2} P_{tb} \\ P_{tb} &= \lambda_t P(w_i|w_{i-2}, w_{i-1}) \\ &+ (1-\lambda_t) P(w_i|w_{i-1}) \end{aligned} \quad (3)$$

ここでP(W)は上記形態素解析の出力結果である．λ_{tb} は trigram 確率と bigram 確率の混合比で実験では0.8を用いた．また，w_{-1}，w_0 は文頭，w_{n+1}，w_{n+2} は文末を表す．学習の際，形態素解析が未定義語として出力する文字列もそのまま学習するので辞書に無い未定義語の単語接続特性も獲得できる．このモデルを以降単語 bi-trigram モデルと呼ぶ．

- 単語の共起頻度の獲得

単語の共起頻度分布を獲得するために新聞記事学習コーパスの1記事内ごとに，全ての2単語間の共起頻度を数え挙げその結果を利用する．データが大量になるため他のモデルを学習するときより少ない量のコーパスでしか学習できない．そのため助詞助動詞などの機能語，数詞，形式名詞，接頭辞，接尾辞などは対象外とした．共起が10回以上のもののみを取り出し単語単独の生成回数も記録した．

候補選択モデル　候補選択モデルは上記に示した統計的形態素解析システム，単語 bi-trigram モデルの他に比較として文字 trigram モデルを用いる．さらに統計的形態素システムと単語 bi-trigram を融合した複合モデルを使用する．この複合モデルを式で表す．

$$\begin{aligned} P(L) &\approx \prod_{i=1}^{n+1} K(i) \times P(w_i|s_i) \\ K(i) &= (1-\lambda) P(s_i|s_{i-2}, s_{i-1}) \\ &+ \lambda P_{tb} \end{aligned} \quad (4)$$

つまり，形態素 trigram をもとに接続確率に関しては単語 bi-trigram でアンチスムージング化を行っている．これは名詞に対して形態素 trigram では品詞にしているため弱くこの点を補強している．

さらに単語の品詞に対する生成確率は単語単独の生成確率と見なして，単語の共起頻度の確率値と重ね合わせたモデルも用意した．式は形態素 trigram の単語の部分を以下のように重ね合わせる．

$$\begin{aligned} P(L) &\approx \prod_{i=1}^{n+1} K(i) \times S(i) \\ S(i) &= (1-\lambda) P(w_i|s_i) \\ &+ \lambda P(w_i|w_1, .., w_{i-1}, w_{i+1}, .., w_n) \end{aligned} \quad (5)$$

ただし，

$$\begin{aligned} & P(w_i|w_1, .., w_{i-1}, w_{i+1}, .., w_n) \\ &\approx \frac{C(w_i, w_1) + .. + C(w_i, w_n)}{C(w_1) + ..C(w_n)} \end{aligned} \quad (6)$$

のように近似する．ここで $C(\cdot)$ は出現回数を表している．実験では両方の λ とも0.5を用いた．

以上示した候補選択モデルは各ブロックで生成された候補の中から文全体の確率 $P(L)$ を最大にする候補を選択する．ただし，どの選択モデルでも，候補に関して以下の重み確率を掛け合わせる．ある候補を含んだ全体の文字列 L' において，その候補が上から第 k 位候補で元の文字列から h 文字入れ換えている場合 $\alpha^{k-1} \times \beta^h$ の重みを掛ける．実験の結果から $\alpha = \beta = 0.05$ を用いる．よって推測文字列 $\hat{L}$ は以下の式の確率を最大化する文字列である．

$$\hat{L} = arg\max_{L} P(L) \times \alpha^{k-1} \times \beta^h \quad (7)$$

3　実験と考察

本論文での学習とテストに用いるコーパスは日経新聞94年記事を用いた．文字 trigram，単語 bi-gram，trigram，形態素 trigram の学習には1年文の記事の4分の1の記事（約40万文）を用いた．単語の共起頻度に関してはデータが大きすぎるためさらに5分の1の記事（約8万文）のみを用いた．テスト文は学習に用いなかった記事から100文ずつ3つのテストデータを用意した．これらのテストデータに対してランダムを用いて以下のような疑似的なＯＣＲ出力結果を作成した．

- 1文中1文字置換誤り
- 1文中 n 連続文字列誤り
- 10% および 5% のランダム誤り

以上の誤りは全て置換誤りである．これらのテストデータを利用した実験の結果を以下に示す．

3.1 誤り箇所検出の実験

1文中1文字置換誤りテストデータを用いて誤り箇所検出能力の評価を行う．3種類のコーパス a, b, c を用いた実験結果を表1に示す．表中で (適合率) = (正解した数) /(誤りと指摘した数) で，(再現率) = (正解した数)/(誤り箇所の数) である．表からわかるように，×△印まで誤りと見た場合再現率が96%にも達する．ただし5倍以上の誤り箇所を指摘するがここで指摘しないと修正されないので以降×△印まで誤りと認識する．

コーパス	×まで	
	適合率	再現率
a	68(73/107)	73(73/100)
b	41(70/169)	70(70/100)
c	62(64/104)	64(64/100)
	×△まで	
a	19(96/496)	96(96/100)
b	15(96/657)	96(96/100)
c	18(98/551)	98(98/100)

表 1: 誤り箇所検出の精度

3.2 候補文字の生成

まず1文字誤りの場合の候補精度を測定する．測定する値は (再現率) = (正しく生成した文字数) /(誤り文字の総数) のみとする．解の候補を出力する範囲は前節で述べた通り×△印の部分まで測定する．適合率は正確に測定しなかったが1つの正解に対して10個以上の解を出力している．

コーパス	候補生成の精度
	再現率
a	88(88/100)
b	87(87/100)
c	84(84/100)

表 2: 1文字訂正実験の候補生成精度

結果を表2に示す．100文字の誤りに対して約86%程度の精度で正しい解を想起することができる．前節の誤り検出精度が約97%であることから誤り検出されれば91%以上の精度で正しい解を想起することができている．

次に，2，3，4文字連続誤り訂正の実験結果を表3に示す．連続文字列の想起は前章で述べたように大量のパス計算を行うため大変時間がかかる．そこでコーパスbに対してのみ実験を行った．3文字連続から急激に精度が減少する傾向が観測された．3文字連続で誤ったテストデータを見ると，もはや人間でも何を書いていたか良く分からない文章となる．実験は候補生成能力を観測するため簡単に再現率のみを示したが有効なのは2文字列連続程度である．

コーパス	候補生成の精度
	再現率
b(2文字連続)	59(118/200)
b(3文字連続)	35(105/300)
b(4文字連続)	24(95/399)

表 3: 連続文字列訂正の候補生成精度

3.3 候補の選択

ここでは以下に示す実験を行う．

- 1文字置換誤りデータを用いて候補選択モデルの能力測定
- ランダムな誤りデータに対する訂正能力の測定

最初の実験で候補選択モデルの優劣を測定する．最後の実験は10%と5%のランダム誤りエラーのテキストを訂正する．実際に論文誌をＯＣＲ処理した結果を観測して見るとほぼ91～93%ぐらいの精度であることから現実のＯＣＲ後のテキストに近いデータといえる．

測定方法として改善率，改悪率，total精度で観測する．各々を以下に示す．

$$\text{改善率} = \frac{\text{誤り文字が正しい文字に置換された数}}{\text{総誤り文字数}} \quad (8)$$

$$\text{改悪率} = \frac{\text{正しい文字が誤り文字に置換された数}}{\text{正しい文字数}} \quad (9)$$

$$\textit{total}\text{精度} = \frac{\text{訂正後の正しい文字数}}{\text{全文字数}} \quad (10)$$

測定に関しては訂正出力の第一解のみ採用する．また，以下の実験結果に使用するテストデータの特性を表4に示す．

1文字置換データ			
	a	b	c
全文字数	4751	4884	4582
総誤り文字数	100	100	100
正しい文字数	4651	4784	4482
10% 誤りデータ			
総誤り文字数	481	495	461
正しい文字数	4270	4389	4121
5% 誤りデータ			
総誤り文字数	241	248	232
正しい文字数	4510	4635	4350

表 4: テストデータの特性

候補選択モデルの能力　1文字置換誤りデータを用いて候補選択モデルの能力を測定する．表5に文字 trigram，単語 bi-trigram，形態素 trigram，単語と形態素 trigram の混合モデル，さらに単語共起分布を混合したモデルの結果を示す．単位は%である．

まず表5において1文字置換データ a，b，c のうち b の解析精度が全体的に低い．これはFMの洋楽チャートなどの特殊な文が多かったためである．b のデータに対しては文字 trigram モデルと単語 bi-trigram モデルでは total 精度で改悪してしまっている．両方に共通する点は改善率が他のモデルに対して大きく高い．しかしその分改悪率も上がってるため total 精度で敗けている．反対に形態素 trigram は改善率が低いが改悪率も低く total の精度で安定している．そこでこの2つのモデルを混合してみた．total の数字にはあまりはっきり効果が現れていないが改悪率があまり変動せず改善率は上がる傾向が見られた．このモデルにさらに単語共起分布の確率値を作用させるとさらに効果が上がり，total 精度にも少し良い影響が観測された．この結果から混合モデルがより効果的であることがわかった．

ランダム誤り対する訂正能力　表6に 10% と 5% 誤りテキストの訂正実験結果を示す．比較のために形態素 trigram で選択した場合と2種類の混合モデルの結果を示した．単語共起モデルがあまり有効に働かなかった．これは周辺の文字が誤りるため正しい共起関係を獲得できなかったことによる．

文字 trigram			
	1文字置換データ		
	a	b	c
元の精度	97.9	98.0	97.8
改善率	76.0	73.0	62.0
改悪率	0.97	1.88	1.45
total 精度	98.5	97.6	97.8
単語 bi-trigram			
改善率	77.0	75.0	66.0
改悪率	0.71	1.82	1.25
total 精度	98.8	97.7	98.0
形態素 trigram			
改善率	69.0	64.0	55.0
改悪率	0.56	1.1	0.78
total 精度	98.8	98.2	98.2
単語と形態素 trigram の混合			
改善率	72.0	70.0	59.0
改悪率	0.56	1.1	0.89
total 精度	98.9	98.2	98.2
上記モデルに単語共起分布を混合			
改善率	74.0	69.0	62.0
改悪率	0.49	1.1	0.80
total 精度	99.0	98.2	98.3

表 5: 1文字誤り訂正の実験結果

表から単語と形態素 trigram 融合モデルにおいて，10% の誤りの場合は 1.5 ～ 2.0% 程度の改善，5% の誤りの場合で 0.5 ～ 1.5% 程度の改善が見られた．

永田 [2] は形態素 trigram で選択を行い高精度を上げたが上の表では形態素 trigram より混合モデルの方が精度は勝っている．しかし，永田ほどの高精度が上げられなかったのは候補出しでかなり多くの候補を出力するため選択誤りが増えたことによる．候補の絞り込みと候補選択モデルの精度の向上が次の課題である．

3.4 まとめ

大量の新聞記事コーパスを用いて統計的な形態素解析モデル，統計的単語モデル，文字 trigram モデルを学習し，それらのモデルを用いて誤りを含むテキストの訂正を行った．誤り箇所の検出並びに候補文字の出力を文字 trigram のみで行ったがうまく働いた．候補選択モデルは従来用いられてきた文字 trigram モデルや形態素 trigram モデルより，形態素 trigram モデルに単語 bi-trigram や単語共

形態素 trigram			
	10% 誤りデータ		
	a	b	c
元の精度	89.9	89.9	89.9
改善率	42.0	37.8	40.8
改悪率	2.79	3.37	2.84
total 精度	91.6	90.7	91.5
単語と形態素 trigram の混合			
改善率	44.1	42.0	42.7
改悪率	2.72	3.01	2.94
total 精度	91.9	91.4	91.6
上記モデルに単語共起分布を混合			
改善率	43.9	42.2	42.7
改悪率	2.81	3.10	3.06
total 精度	91.8	91.4	91.5
形態素 trigram			
	5% 誤りデータ		
	a	b	c
元の精度	94.9	94.9	94.9
改善率	55.6	46.8	54.3
改悪率	1.84	2.65	2.14
total 精度	96.0	94.8	95.7
単語と形態素 trigram の混合			
改善率	59.3	53.2	57.3
改悪率	1.53	2.26	2.07
total 精度	96.4	95.4	95.8
上記モデルに単語共起分布を混合			
改善率	59.3	51.2	57.3
改悪率	1.55	2.39	2.07
total 精度	96.5	95.2	95.8

表 6: ランダム誤り訂正の実験結果

起頻度を混合させたモデルの方が良い結果が得られた．しかしながら，選択モデルの精度はまだ十分とは言えず，正しい候補があっても選択されなかったり改悪され大きな精度の獲得にまで至らなかった．

最終的には10% のランダム誤りテキストの訂正に対して平均1.7% 程度，5% 誤りに対して1% 程度の文字を修正することが確かめられた．これよりＯＣＲの解候補出力もなく，またタグ付きコーパスを全く使わなくてもＯＣＲ誤り訂正を行うことができることを示した．次回は，疑似データではなく実際のＯＣＲの処理されたテキストに対する訂正能力を評価したい．

3.5 謝辞

新聞記事を使用させていただいた日経新聞社に対して謹んで感謝の意を表します．

参考文献

[1] K. Kukich. Techniques for automatically correcting words in text. In *ACM Computing Surveys 24*, pp. 377–439, 1992.

[2] M. Nagata. Context-based spelling correction for Japanese OCR. In *Proc. COLING-96*, pp. 806–811, 1996.

[3] R.G. Webster, 中川正樹. 英語と日本語を対象にした文章誤り検出・共通点と 相違. 情報処理, *vol.37, No.9*, pp. 865–871, 1997.

[4] D. Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *ACL-94 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 88–95, 1994.

[5] 松山高明, 渥美清隆, 増山繁. n-gram による OCR 誤り検出の能力検討のための適合率と再現率の推定に関する実験と考察. 言語処理学会第 2 回年次大会発表論文集, pp. 129–132, 1996.

[6] 伊藤信泰. Bigram によるオンライン漢字認識の文脈後処理手法. 情報処理学会自然言語処理研究会, *97-6*, pp. 37–44, 1993.

[7] 森大毅, 阿曽弘具, 牧野正三. 2 重マルコフモデルを用いた日本語文書認識後処理. 情報処理学会自然言語処理研究会, *102-12*, pp. 89–96, 1994.

[8] 荒木哲郎, 池原悟, 塚原信幸. 2 重マルコフモデルによる日本語文の誤り検出並びに 訂正法. 情報処理学会自然言語処理研究会, *97-5*, pp. 29–35, 1993.

[9] 久光徹, 丸川勝美, 嶋好博, 藤澤浩道, 新田義彦. OCR 誤認識後処理の効率について. 情報処理学会自然言語処理研究会 *104*, pp. 17–24, 1994.

[10] 畑田稔, 遠藤裕英. 日本語 OCR 文における英字・カタカナのスペル誤り訂 正法. 情報処理学会論文誌, *vol.38, No.7*, pp. 1317–1327, 1997.

[11] 松本裕治, 北内啓, 山下達雄, 今一修, 今村友明. 日本語形態素解析システム『茶筌』 version 1.0 使用説明書. NAIST Technical Report NAIST-IS-TR97007, Feb 1997.