Vol. 40 No. 6 情報処理学会論文誌 June 1999

平仮名 N-gram による平仮名列の誤り検出とその修正

新 納 浩 幸†

本論文では，日本語の平仮名列で生じる書き誤りを検出，修正する手法として平仮名 N-gram を提案する．また妥当な N の値についても考察する．

単語 N-gram により文書中の誤り検出，修正が可能であるが，大規模な N-gram は N が 3 の場合でさえ，構築するのが困難である．また日本語の場合，形態素解析が必要である点，N-gram 表の検索コストが高い点などから，手軽に利用できる手法ではない．ただし，平仮名列中に生じる書き誤りに限定すれば，平仮名文字に対する N-gram を構築することで，上記の問題を回避し，平仮名列中の誤り検出，修正が可能となる．ここで，N を大きくとれば誤り検出の再現率は高くなるが，コーパスのスパース性から適合率が低くなる．つまり最適な N の設定にはコーパスの量と再現率への重みが影響する．

本論文では，現実規模のコーパスとして新聞記事 5 年分を利用した．そこから平仮名 3〜6-gram を作成し，各々を利用した場合の平仮名文字の挿入，削除，置換，転置による誤りの検出とその修正の効果を調べた．結果，平仮名列中の誤り検出，修正に対して平仮名 N-gram が有効であることおよび，新聞記事 5 年分では 4-gram の利用が実用的であることを示した．

Detection and Correction for Errors in Hiragana Sequences by a Hiragana Character N-gram

Hiroyuki Shinnou†

In this paper, we propose the hiragana character N-gram method to detect and correct errors in Japanese hiragana sequences. Further, we investigate about the proper N.

It is known that the word N-gram method is effective to detect and correct errors in texts. However, it is difficult to construct word N-gram, even the case of N = 3. Moreover, in Japanese, this method requires the morphological analysis and high cost for searching an N word sequence from the word N-gram table. Thus, at the moment the word N-gram method for the text revision is not reasonable. However, if the target of the revision is limited to simple errors in Japanese hiragana sequences, by using the hiragana character N-gram we can detect and correct their errors without above problems. In this method, N-gram with the high N has the high recall, but the low precision because of the corpus sparseness problem. So, we must consider the corpus size and the weight of the recall to set the proper N.

In experiments, we constructed 3,4,5 and 6-gram respectively from newspaper five years articles. By using their N-gram tables respectively, we examined the effectiveness of the revision for simple errors in hiragana sequences, which are caused by a single hiragana character insertion, deletion, substitution and reversal. We conclude that the hiragana character N-gram is effective to detect and correct errors in hiragana sequences, and N = 4 is proper realistically.

1. はじめに

本論文では，日本語の平仮名列で生じる書き誤りを検出，修正する手法として平仮名 N-gram を提案する．また妥当な N の値についても考察する．

英文のスペルチェックの最も簡易な実装は，辞書にない単語をタイプミスであると指摘することである．このレベルの書き誤りに対応する日本語文のスペルチェックシステムは，有益であることは明らかだが，広く利用されているものはない．なぜなら，日本語文章の場合，単語切りを行なうためには形態素解析が必要であるし，誤り箇所から誤った単語切りも生じ，辞書に単語が存在する，しなしで単純にスペルミスを検出することは困難だからである [1]．

一方，単語 N-gram を利用して文章中の誤った単語の検出，修正が可能であることが知られている [2]．これは N 個の単語列とその頻度などの統計的データを表の形で予め用意しておき，存在しない単語列や出現回数の少ない単語列を誤りの可能性があると指摘する手

† 茨城大学工学部システム工学科
Faculty of Engineering, Ibaraki University Dept. of Systems Engineering

法である．日本語文に対しても，その効果は期待できる [3),4)]．ただしスペルチェックに利用できる大規模な N-gram は N が 3 の場合でさえ，構築するのが困難である．また日本語の場合，単語 N-gram を利用するためには，スペルチェックの際に形態素解析が必要である点，単語の N-gram の種類数は膨大であり，その検索コストが高い点などから，スペルチェックとして手軽に利用できる手法とは考えられない．文字 N-gram の利用も考えられる．OCR の認識修正に応用された研究もあるが，文書校正に対しての評価は行われていない．また文字 N-gram では同音異義語の誤りが検出困難という問題もある．ただし，対象を日本語の平仮名列中に生じる書き誤りに限定すれば，平仮名文字の N-gram を構築することで，上記の問題を回避しつつ，誤り検出やその修正が可能である．

平仮名 N-gram を利用する場合，一般に，N が大きい方が誤りかどうかを判別する判別能力が高い．しかし，現実的には N が大きいと，コーパスのスパース性から過度に誤りを検出してしまう結果となる．また，平仮名列すべてを用意することも考えられるが [5)]，平仮名列であってもその種類数は膨大であり，この場合も過度に誤りを検出してしまう．つまり N を大きくとれば誤り検出の再現率は高いが，コーパスのスパース性から適合率が低くなる．このため妥当な N を設定するためには，コーパスの量と再現率への重みを考慮しなくてはならない．

本論文では，現実規模のコーパスとして新聞記事 5 年分を利用した．このコーパスから平仮名 3 〜 6-gram を作成し，各 N-gram を利用した場合の平仮名文字の挿入，削除，置換，転置による誤りの検出やその修正の効果を調べた．平仮名列の誤り検出や修正には平仮名 N-gram が有効であること，新聞記事 5 年分では N の値は 4 が妥当と考えられることを結論とする．

2. 平仮名 N-gram による誤り検出と修正

2.1 平仮名 N-gram の構築

平仮名 N-gram の構築は容易である．コーパスを 1 本の長い文字列と考え，平仮名以外の文字を K という文字に変換しておく．i 番目の位置の文字から $i+N-1$ 番目の位置の文字までからなる長さ N の文字列を考え，その文字列が以下のいずれかのパターンになっている場合に，その文字列を取り出す．

- H H …. H
  N 文字すべてが平仮名
- K H H … H
  先頭文字が平仮名以外で残りの N - 1 文字が平仮名
- H H … H K
  先頭から N - 1 文字が平仮名で末尾文字が平仮名以外

ここで，最初のケースのすべての文字が平仮名である文字列だけを取り出してもよいが，精度向上のために残りの2 つのケースの文字列も抽出している．上記の操作を $i = 0$ から順にコーパスの最後の位置に至るまで繰り返し，取り出した文字列の頻度表を作成することで平仮名 N-gram が構築できる．

2.2 平仮名 N-gram による誤り検出

先ほど構築した平仮名 N-gram を頻度の昇順に並べ，同時に総頻度を測る．頻度の少ないものから順に頻度の累計をとってゆき，累計が総頻度の x%になる時点に最も近い頻度を閾値とする．またこの x をここでは**閾値割合** と呼ぶことにする．

ある平仮名列 α に書き誤りが存在するかどうかの判定は以下に従う．まず文字列K α K から N-gram を取り出し，それぞれの文字列の頻度を平仮名 N-gram から調べる．それら頻度の最小値（ この値を平仮名列 α に対する **N-gram 最小頻度**と呼ぶことにする）が先の閾値以下である場合に，平仮名列 α に書き誤りが存在すると判定する．

2.3 平仮名 N-gram による誤り修正

平仮名列の誤りは以下の 4 つのパターンのいずれかであると仮定する．

削除 平仮名列中のある位置の平仮名 1 文字が欠落した誤り
例）(正) するかどうか → (誤) するどうか

挿入 平仮名列中のある位置に，ある平仮名 1 文字が挿入された誤り
例）(正) するかどうか → (誤) するかがどうか

置換 平仮名列中のある位置の平仮名 1 文字が，ある平仮名と入れ替わった誤り
例）(正) するかどうか → (誤) するかとうか

転置 平仮名列中のあるとなりあう2 つの平仮名文字が交換された誤り
例）(正) するかどうか → (誤) するかうどか

平仮名列 α に書き誤りが存在すると判定した場合，上記の 4 つのパターンの誤りから α が生じるようなすべての平仮名列を列挙する．それらすべての平仮名列に対する N-gram 最小頻度を求め，それらの値のうち最大の値を持つ平仮名列を α に対する修正とする [6)]．また N-gram 最小頻度の大きい値から5 つとった平仮名列を修正候補とし，修正候補の中に本来の平仮名列が含まれていた場合を修正の正解とする．

2.4 最適な N に影響を及ぼす要素

前節の設定で，平仮名列の誤り検出，修正が可能ではあるが， N をいくつにするかという問題が残る．

N-gram を作成した場合，誤り検出，修正の対象となるのは，長さ $N-1$ 以上☆の平仮名列である．長さ $N-1$ より小さい平仮名列に対しては，以下のような長さ $m+2$ ($0 < m < N-1$) の文字列をコーパスから収集しておき，各 m ごとの文字列の頻度表を使って，前章と同様の手法を適用すればよい．

- K H H ... K
  先頭と末尾の文字が平仮名以外で間の平仮名文字列の長さが m

この枠組では，N の値が大きいほど誤りかどうかを判別する判別能力が高いことは明らかである．しかし現実的には，N が大きいとき，コーパスのスパース性から登録されていない文字列が多くなり，結果的に過剰に誤りだと判定してしまう．誤り検出の観点で見れば，N が大きいほうが再現率は高いが，適合率が低いことになる．当然，コーパスを大きくしてゆけば，スパース性の影響が小さくなってゆくので，徐々に N の大きな N-gram においても適合率が上がってゆく．つまり最適な N を設定する場合には，以下の2 つの要素を考慮する必要がある．

- コーパスの量
- 再現率への重み

本論文では現実規模と考えられる新聞記事 5 年分のコーパスから妥当な N の値も考察する．

3. 誤り検出と修正の実験

3.1 実験の設定

日本経済新聞 CD-ROM の '90 年度版から '94 年度版，つまり 5 年分の新聞記事から平仮名 3 〜 6-gram を作成し，それらを各々利用した場合の，誤り検出，およびその修正の効果を調べた．

まず，テストデータとして，先の新聞記事とは別の新聞記事を用意し，そこから長さ 6 以上の平仮名列を 2,000 種類取り出した．具体的には毎日新聞 '94 年度版の CD-ROM に存在する文の始めから順に長さ 6 以上の平仮名列を 2000 種類になるまで取り出した．テストデータの平仮名列の長さ別の総数を表 1 に示す．

このテストデータに対して，平仮名 3 〜 6-gram を各々利用して，以下の実験を行なった．

実験1 各平仮名列に誤りがあるかどうかを判定する．

☆ N でないのは，ここでの N-gram の作成方法が "K H ... H" や "H ... H K" を含めているからである．

表 1 テストデータの文字列の長さ
Table 1 Length of sequences in test data

文字列長	テスト数
6	723
7	489
8	290
9	202
10	115
11	74
12	41
13	25
14	18
15 以上	23
合計	2000

実験2 各平仮名列の適当な位置の1 文字を取り除くことで作成した平仮名列に誤りがあるかどうかを判定する．また誤りがあると判定した場合，その修正候補を求める．

実験3 各平仮名列の適当な位置に適当な平仮名1 文字を挿入することで作成した平仮名列に誤りがあるかどうかを判定する．また誤りがあると判定した場合，その修正候補を求める．

実験4 各平仮名列の適当な位置の1 文字を適当な平仮名1 文字に変更することで作成した平仮名列に誤りがあるかどうかを判定する．また誤りがあると判定した場合，その修正候補を求める．

実験5 各平仮名列の適当な隣合う2 文字を入れ換えることで作成した平仮名列に誤りがあるかどうかを判定する．また誤りがあると判定した場合，その修正候補を求める．

実験1 では，どの程度過度に誤りを検出するか調べる．残りの実験はそれぞれ，削除，挿入，置換，転置によって生じた誤りを含む平仮名列に対して，どの程度検出，修正できるかを調べる．また長さ 6 の平仮名列は実験2 では長さ 5 になってしまい，6-gram がやや不利である．公平な比較のために，実験2 については長さ 7 以上の平仮名列 1277 種類に対して実験を行った．

3.2 閾値と誤り率を固定した実験結果

本手法により誤り検出を行なう場合，検出の正解率や再現率は閾値に依存する．本節では閾値割合を 1.0% に設定して実験を行なう．閾値割合を 1.0% にしたときの各 N-gram の閾値となる頻度を表 2 に示す．

実験 1 の結果を表 3 に示す．一般に N が大きい方が正解率は良いはずだが，4-gram の方が 5-gram や 6-gram よりも良い結果となった．これは 5-gram, 6-gram のスパース性が原因である．

実験 2〜5 の結果を表 4 と表 5 に示す．表 5 中の括

表 2 閾値割合 1% による閾値
Table 2 Threshold corresponding to threshold ratio 1%

N-gram	閾値
3-gram	75
4-gram	5
5-gram	1
6-gram	0

表 3 実験 1 の結果
Table 3 Result of experience 1

N-gram	検出数	正解率
3-gram	313 (2000)	0.844
4-gram	232 (2000)	0.884
5-gram	286 (2000)	0.857
6-gram	328 (2000)	0.836

弧内の数値は誤り検出で検出できた文字列の種類数を表し，修正はこれらの文字列に対して行なった．

表 4 実験 2〜5 の誤り検出の結果
Table 4 Detection results of experience 2-5

N-gram	実験 2 (削除)	実験 3 (挿入)	実験 4 (置換)	実験 5 (転置)	平均
3-gram	0.607	0.921	0.898	0.926	0.838
4-gram	0.721	0.955	0.940	0.970	0.897
5-gram	0.789	0.974	0.955	0.982	0.925
6-gram	0.810	0.977	0.962	0.982	0.933

表 5 実験 2〜5 の誤り修正の結果
Table 5 Correction results of experience 2-5

N-gram	実験 2 (削除)	実験 3 (挿入)	実験 4 (置換)	実験 5 (転置)	平均
3-gram	0.636 (775)	0.770 (1841)	0.792 (1796)	0.846 (1852)	0.761
4-gram	0.782 (921)	0.877 (1910)	0.879 (1879)	0.924 (1939)	0.866
5-gram	0.731 (1007)	0.854 (1947)	0.845 (1909)	0.882 (1964)	0.828
6-gram	0.684 (1035)	0.817 (1954)	0.801 (1923)	0.835 (1963)	0.784

実験 1 の正解率を p_1，実験 2〜5 の正解率の平均を p_2 とする．また実際の文書で平仮名列に誤りが生じる確率を r とする．r をここでは**誤り率**と呼ぶことにする．いま文書中の全ての平仮名列の数を T とすると，正しい平仮名列の数は $T(1-r)$ であり，誤った平仮名列の数は Tr である．正しい平仮名列を誤りだと誤って検出する数は，$T(1-r)(1-p_1)$ であり，誤りの平仮名列を正しく誤りだと判別できる数は，Trp_2 である．結局，誤りとして検出する数は，$T((1-r)(1-p_1)+rp_2)$ であり，そのうち Trp_2 が正しい検出であるので，適合率 (P) は，

$$P = \frac{rp_2}{(1-r)(1-p_1)+rp_2}$$

となる．また再現率 (R) は，

$$R = p_2$$

となる．P，R を用いて，以下のF 値により誤り検出の評価を行なった結果を表 6 に示す．

$$F = \frac{(\beta^2+1.0)*P*R}{\beta^2*P+R}$$

ただし，ここでは再現率と適合率の重みを等しくして，$\beta = 1.0$ とした．また，誤り率 r も 0.01 に固定した．

表 6 誤り検出の評価
Table 6 Evaluation of error detection

N-gram	適合率	再現率	F 値
3-gram	0.051	0.838	0.097
4-gram	0.072	0.897	0.134
5-gram	0.061	0.925	0.115
6-gram	0.054	0.933	0.103

4-gram が最も良い．また誤り修正に関しても 4-gram が最も優れている．修正に関しては対象となった平仮名列やその数の違いもあるので，単純に比較はできないが，4-gram は他の N-gram と比べて劣ることはないだろう．4-gram の結果を見ると，適合率は 1 割弱 (0.072) と低いが，再現率が約 9 割 (0.897) と高い．文書校正を目的とした場合，再現率の方が重要であり，この程度の値であれば，有用なツールとして利用できるであろう．

3.3 誤り率とF 値

誤り率 r を 0.001 から 0.001 刻みで 0.05 まで変化させた場合のF 値の変化を調べた．ただし閾値割合は 1.0% に固定した．結果を図 1 に示す．

各 r の点で 4-gram のF 値が最も高かった．また，各 N-gram の比較を行なう場合に，誤り率は影響を与えないと考えられる．

3.4 閾値とF 値

閾値割合 x を，0.0% から 0.1% 刻みで 5.0% まで変化させ，先の実験の適合率，再現率，そしてF 値の変化を調べた．ただし閾値割合が 0.0% に対する閾値の頻度は 0 とした．また $r = 0.01$ として固定した．それぞれの結果を図 2，図 3，図 4 に示す．

図 4 から，どの N-gram に対しても閾値割合が低い方がF 値が高いことが確認できる．閾値割合が最も小さな点 0.0% は，各 N-gram において最も高いF 値を示す．これは閾値割合が低い方が適合率が高くなるか

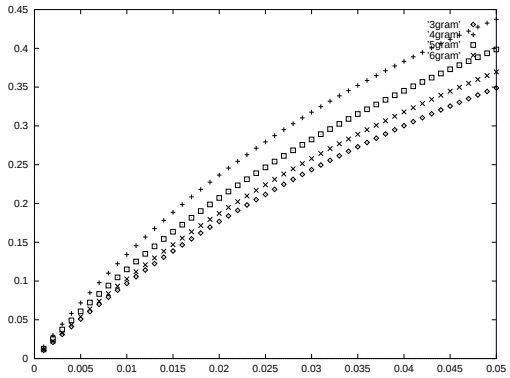


図 1 誤り率とF 値
Fig. 1 Relation between error ratio and F-measure

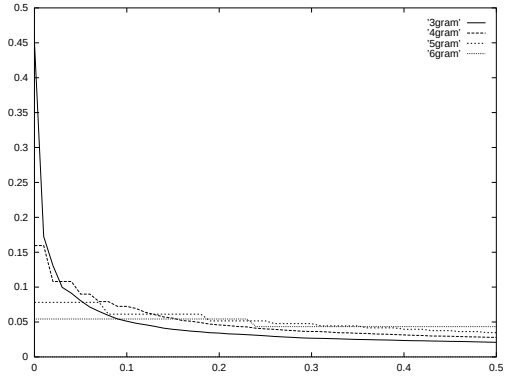


図 2 閾値割合と適合率
Fig. 2 Relation between threshold ratio and precision

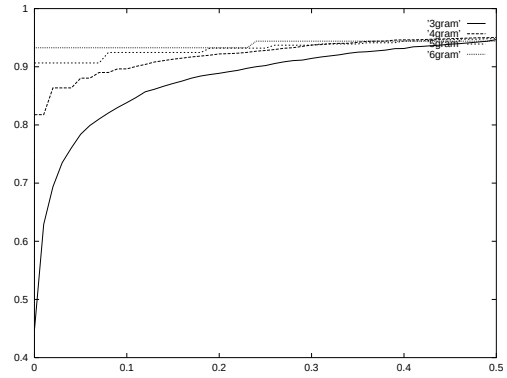


図 3 閾値割合と再現率
Fig. 3 Relation between threshold ratio and recall

らである．適合率と再現率の値には大きな開きがあり，F 値の評価において適合率と再現率が同じ重みの場合には，結果的に適合率の優劣がF 値の優劣を決める．また図 2 からも分かるように，閾値割合が 0.0% の点では 3-gram が最も適合率が高く，結果的に 3-gram から最も高いF 値 0.449 が得られる．

また頻度が 0 に対応する閾値割合が最小の閾値割合

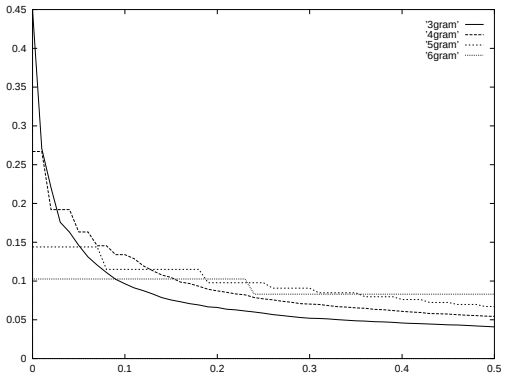


図 4 閾値割合とF 値
Fig. 4 Relation between threshold ratio and F-measure

であり，この値よりも小さな閾値割合は意味がない．この最小の閾値割合を**最小閾値割合**と呼ぶことにする．今回の実験で得られた各 N-gram の最小閾値割合とそのときのF 値を表 7 に示す．

表 7 最小閾値割合とF 値
Table 7 F-measure corresponding to minimum threshold ratio

N-gram	最小閾値割合	F 値
3-gram	0.01%	0.449
4-gram	0.17%	0.267
5-gram	0.72%	0.144
6-gram	2.08%	0.102

3.5 再現率に重みをおいた評価

上記までの実験は適合率と再現率を同じ重みでF 値の評価を行なった場合の結果である．F 値の評価では，適合率と再現率の値の大きさの差から適合率の影響が大きい．また，実際の文章にはほとんど誤りが含まれないために，できるだけ誤りを検出しない手法の方が，結果的に適合率が上がる．そのため適合率と再現率を同じ重みで評価する場合には，できるだけ誤りを検出しない手法の方がF 値も高くなる．3-gram を利用した判別はできるだけ誤りを検出しない手法に対応するので，上記の実験でも，その特徴があらわれた．

適合率と再現率への重みは利用目的に依存する．簡易に単純な誤りだけを検出したい場合には，再現率よりも適合率が大事であろうし，文書の最終的な校正に利用する場合には，再現率が重要になるだろう．適合率と再現率の重みを等しくした場合，上記の実験からは 3-gram が優れていた．ただしその場合，適合率は 0.451 と高いが，再現率は 0.447 であり，高い数値ではない．

再現率に重みを置いて，F 値の定義式の β を 2.0,

3.0 とした場合の，F 値の変化を図 5 と図 6 に示す．$\beta = 1.0$ の場合のF 値の変化が図 4 である．

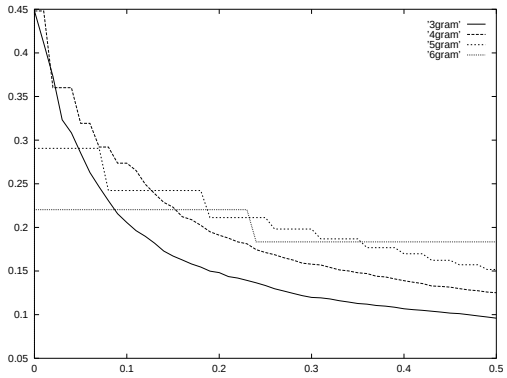


図 5 $\beta = 2$ の閾値割合とF 値
Fig. 5 Relation between threshold ratio and precision in $\beta = 2$

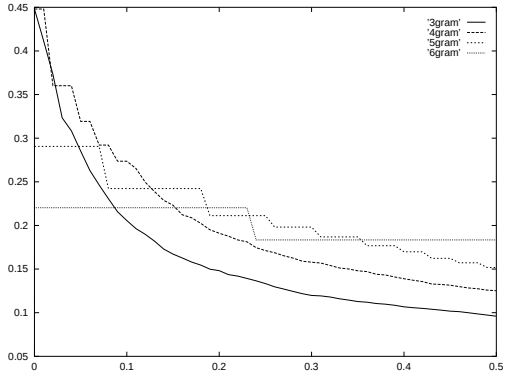


図 6 $\beta = 3$ の閾値割合とF 値
Fig. 6 Relation between threshold ratio and precision in $\beta = 3$

$\beta = 2$ の場合，最も高いF 値は 4-gram の最小閾値割合の 0.17% 時の 0.448 であった．この時の適合率は 0.160，再現率は 0.818 であった．$\beta = 3$ の場合も，最も高いF 値が得られるのは 4-gram からである．

つまり再現率に重みをつけた場合には，3-gram よりも 4-gram の方がF 値が高くなる．5-gram や 6-gram のF 値が 4-gram のF 値よりも大きな値になるのは，閾値割合を大きくとる必要がある．

4. 考　　察

4.1 最適な N について

実用的な校正システムとして認められるための適合率，再現率の下限がどの程度であるかを決定するのは困難であろう．ツールとして使うのであるから，それらの値は利用目的や利用者に依存する．ただし筆者自身は，適合率 0.1（ 10 個の検出で 1 個が実際に誤り）再現率 0.9（ 誤りの 9 割は検出できる）あたりが達成できれば，一般の文書校正に利用可能だと 考える．このように目標とする適合率と再現率を固定すれば，利用するコーパスから最適な閾値割合と 最適な N が決定できる．

新聞記事 5 年分からの N-gram では適合率 0.1，再現率 0.9 を満すような閾値割合や N は存在しなかった．ただし閾値割合 1% と $N = 4$ の場合に，適合率 0.072，再現率 0.897 であり，非常に近い値が実現できている．この点で提案した平仮名 N-gram の手法は利用可能だと 考える．

本手法は明らかにコーパスを大きくしてゆくことで，適合率や再現率を 高く することができる．ただし 上限は存在するかもしれない．またあまりにも巨大なコーパスを想定することは現実的ではなく，本実験で利用した新聞記事 5 年分程度が現実的な規模だと思われる．これらの点から 適合率 0.1 と再現率 0.9 程度の値が本手法の達成できる 現実的な値であろうと 予想する．

また目標におく 適合率と再現率の値によって，最適な N は変化する．概略，N-gram の N が小さい方が適合率は高く，N-gram の N が大きい方が再現率は高い．つまり 誤り 検出の評価方法によって，最適な N は変化する．実験では 5 年分の新聞記事を 利用したが，適合率と再現率を同じ 重みで評価すれば，3-gram が優れており，再現率に重みをおくと 4-gram 方が優れていた．文書校正に利用する場合には，再現率の方が重要なので，4-gram が実用的だと 結論できる．5-gram が 4-gram よりも良い結果を 出すには，5 年分の新聞記事では不足であり，結果的に $N = 5$ 以上の N-gram は現実的ではない．

また今回の実験では頻度 0 に対応する閾値割合を最小閾値割合と 定義したが，頻度 0 に対応する閾値割合は実際に求めることできない．ここでは頻度 1 に対応する閾値割合の 1/2 をその値としている．頻度 1 に対応する閾値割合を 最小閾値割合と 定義した場合，各 N-gram の最小閾値割合とそのときのF 値を 表 8 に示す．

このとき興味深い結果として，各 N-gram の最小閾値割合で各 N-gram を 評価する場合には，その N-gram のF 値が最も高い．つまり，3-gram の最小閾値割合 0.02% で評価する場合には，3-gram が最も優れ，4-gram の最小閾値割合 0.33% で評価する場合には，4-gram が最も優れ，5-gram の最小閾値割合 1.44% で評価する 場合には，5-gram が最も優

表 8 頻度1 に対する閾値割合とF 値
Table 8 F-measure corresponding to 1 of threshold

N-gram	閾値割合	F 値
3-gram	0.02%	0.360
4-gram	0.33%	0.192
5-gram	1.44%	0.115
6-gram	4.17%	0.083

れ，6-gram の最小閾値割合 4.17% で評価する場合には，6-gram が最も優れている．最初に行なった実験では閾値割合を 1.0% に設定して評価を行なっている．このために，1.0% よりも小さな最小閾値割合をもつ 3-gram，4-gram が評価の対象になり，N の大きさから 4-gram の方がよい結果を得ることが予想できる．また実際に実験結果もそうであった．この特徴は閾値と N-gram の N を設定する際のヒントになると思われる．閾値と N-gram の N の関係が分っていれば，目標とする適合率から閾値割合が決定され，次に閾値割合から N-gram の N が決定される．この点の実験は今後の課題である．

最後に，本手法は各 N-gram からの判断を線形結合により統合させて判断するように拡張できることも注記しておく．これはコーパスのスパース性に対処するためのスムージングの手法 [7] と捉えられ，コーパスを大きくし高次の N-gram で判別することに相当する．そのため，この形に拡張すれば，再現率の高い N の大きな N-gram からの判別を行え，しかも適合率の向上も期待できる．この方向での拡張は今後の課題である．

4.2 現実に生じた誤りの検出

前章の実験では誤りを人工的に作成して，その検出，修正に対する本手法の有効性を調べた．ここでは，実際に文書中に生じた誤りに対する本手法の有効性を考察する．対象とした文書は（社）日本電子工業振興協会の自然言語処理技術委員会の平成 8 年度の活動報告書「自然言語処理システムの動向に関する調査報告書」[8] の第 4 章 (約 200K バイトのテキストファイル) である．この文書を詳細に読むことにより，長さ 4 文字以上の平仮名列の誤りとして以下の 3 箇所を発見することができた．

(1) …キーワードによる検索が可能 なもとのと、…(148 ページ 28 行目)
(2) …分類先 でなければなない。(161 ページ 5 行目)
(3) …省 くことにでより 小さな…(175 ページ 6 行目)

上記の文書中，長さ 4 文字以上の平仮名列は 2,447 種類存在した．誤り率としては 0.0012 である．それらに対して，実験で作成した平仮名 4-gram から誤り検出と修正を試みた．また閾値割合は 1.0% とした．誤りとして検出された文字列は 37 種類であり，前述した 3 つの誤りはすべて含まれていた．この点で再現率は 100% である．また，この 3 つに対する修正を行なうと，最小 N-gram が最大のもの，つまり第 1 候補が正解であった．

過度に誤りとして検出した文字列は 34 種類あったが，その原因の内訳は以下のとおりである．

- 単語が「漢字列+ 平仮名列」から構成され，末尾の平仮名列とその単語に接続する平仮名列とで不自然な平仮名列になる (8 種類)
  - …急 ぎのものに 返事を書く…(118 ページ 26 行目)
  - …本人への成 りすましや 契約内容の改竄…(119 ページ 27 行目)
- 通常は漢字で表記するものを平仮名で表記している (7 種類)
  - …機能 をかんたんに 呼び出せる…(134 ページ 21 行目)
  - …電子化という観点 からのべる…(164 ページ 34 行目)
- 頻度が低い平仮名からなる単語を含む (3 種類)
  - …スピードと情報量 のいたちごっごとなる可能性…(118 ページ最下行)
  - …常識 がくつがえってしまったのである…(171 ページ 12 行目)
- 誤りではないが不自然な表現である (2 種類)
  - …容易 となってきたとともに、…(158 ページ最下行)
  - …各ソフトウェア それぞれにさまざまな 工夫…(134 ページ 13 行目)
- その他，利用したコーパスでは余り生じなかった平仮名列である (14 種類)
  - …管理者 によるなんらかの チェック…(147 ページ 7 行目)
  - …閾値を越 えたらより 大きな…(176 ページ 7 行目)

自然な平仮名列も誤りとして検出されていたのが目だったが（上記の「その他」の分類），これは N-gram の作成の元になったコーパスが新聞記事だが，検出対象の文書が論文調のものであったことも考えられる．

ここで用いた文書では，未検出となった誤りの平仮名列はなかった．しかし，当然，これは再現率に関して完全であることを意味するわけではない．例えば，

白木は現実に生じた誤り平仮名列を 9 種類示しているが [5]，その中で長さ 4 以上の以下の 4 つの平仮名列に対して，誤り検出をを行なってみると，2 番目の「…例えば，図ようなのネットワークから，…」が検出できなかった．これは 4-gram の閾値割合 1.0% における再現率の低さから生じている．閾値割合を 5.2% まであげれば検出できた．また平仮名 5-gram を利用した場合も誤りだと検出できた．

(1) …自然なつながりがもつようにする必要がある．(検出 ○)
(2) …例えば，図ようなのネットワークから，…(検出 ×)
(3) …可能な連体節がである場合は，…(検出 ○)
(4) …出力する方が適切がであると考え…(検出 ○)

注意として，この例の「ようなの」は長さが 4 の平仮名列ではあるが 4-gram と 5-gram で差が生じている．それは，**2.1** で述べたように，本論文では N-gram を作成するときに完全に平仮名文字列だけを対象にしているのではなく，K H H … H や H H … H K の形も含めているからである．実際に「ようなの」から 5-gram を取ると，

Kようなの
ようなのK

となるので 4-gram との差が生じる．

本節の実験の適合率は 0.081 であった．前章の実験の適合率は 0.072 (閾値割合 1.0%での値) であったので，前章の実験よりも良い値である．これは先の実験では長さ 6 以上の平仮名列を対象としたためである．実際は長さが短い平仮名列の方が割合としては多く，長さの短い平仮名列の方が誤りかどうかの判断の正解率が高いからだと考えられる．

4.3 品詞 N-gram との統合

実験では削除による誤りに対する正解率が悪い．これは 1 文字削除によって，生成される平仮名列が妥当な平仮名列となる場合が多いからである．例えば，以下の例1 では「のいずれかである」という平仮名列は誤りであるが，例2 では「のいずれかである」という平仮名列は正しい．

例1) (正) 0 か1 のいずれかであると 仮定する
(誤) 0 か1 のいずれかである 仮定する
例2) (正) 0 か1 のいずれかである 場合は…

この種の誤りを対象の平仮名列だけから検出することは難しく，他の情報を利用する必要がある．有効なアプローチとして，品詞 N-gram との統合が考えられる．対象となる平仮名列を含む前後の単語を含めた単語列の品詞列のパターンからその平仮名列の品詞パターンに誤りがある可能性が検出でき，そこから誤りの検出ができる．修正も本手法の方法を併用することで可能である．

ただしこの場合，形態素解析が必要になり，本手法の長所に矛盾してしまう．この場合は，利用目的が最終的な校正という位置付けになるであろう．

4.4 さらなる拡張

さらに品詞 N-gram との統合では検出できない誤りのタイプとして，文脈依存の平仮名列もありうる．例えば，以下の例3 では「されることを」と「されることに」という平仮名列はその例文中では正しいが，入れ換えると誤りになる．さらにそれらの平仮名列の直前，直後の単語品詞は同一であるので，品詞列からも誤りを検出することはできず，文脈依存の平仮名列となっている．

例3) 反対 されることを 考慮する
反対 されることに 対処する

文脈依存の平仮名列に対しては，語義選択手法が利用できる．N-gram 手法と統合して利用することが有効であろう [9]．ただし平仮名列は付属語的な表現である場合が多く，文脈が平仮名列を決定するケースは少ないと予想する．

平仮名列以外の文字列への拡張としては，独立した漢字1 文字(その漢字文字の前後が漢字文字でないもの) を平仮名と同列に扱い [10]，N-gram を求めることが考えられる．上記のような漢字1 文字は助詞あるいは助動詞的な句の一部であることが多く，これによって本手法の適用範囲が広がるはずである．あるいは，本論文では句読点も漢字と同列に扱ったが，平仮名列の先頭と末尾に位置する文字が句読点の場合は，それらを漢字と区別することも有効であろう．

5. おわりに

本論文では，日本語の平仮名列で生じる書き誤りを対象に，平仮名 N-gram を利用してその誤りを検出，修正することについて述べた．特に妥当な N を求めることを目的に，新聞記事 5 年分を利用して，$N = 3, 4, 5, 6$ の場合についてそれぞれ試した．その結果，平仮名列の誤り検出，修正に対しては，平仮名 N-gram が有用

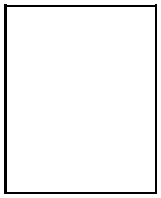
であること，また，新聞記事 5 年分のコーパスでは，$N = 4$ が妥当と考えられること，を示した．N-gram の大きさと最適な閾値の関係の調査，スムージング手法の導入，品詞 N-gram との統合，平仮名以外の文字種への拡張を今後の課題とする．

謝辞 本実験で利用したコーパスおよび評価文は，日本経済新聞 CD-ROM '90〜'94 版と毎日新聞 CD-ROM '94 版から得ています. 利用を許可していただいた日本経済新聞社と毎日新聞社に深く感謝します.

参 考 文 献

1) 田中靖大: 日本語のスペルチェッカーを目指して, *bit*, Vol. 30, No. 10, pp. 19–22 (1998).
2) Mays, E., Damerau, F. and Mercer, R.: Context based spelling collection, *Information Processing and Management*, Vol. 27, No. 5, pp. 517–522 (1991).
3) 丸山宏: N グラムモデルによる日本語単語の並べ替え実験, 情報処理学会第 49 回全国大会論文集, pp. 181–182 (1994).
4) 石場正大, 竹山哲夫, 青木恒夫, 兵藤安昭, 池田尚志: 品詞 N-gram 統計情報を用いた日本語文書における誤り検出法について, 音声言語処理研究会 SLP-19-15, 情報処理学会 (1997).
5) 白木伸征, 黒橋禎夫, 長尾眞: 大量の平仮名列登録による日本語スペルチェッカの作成, 言語処理学会第 3 回年次大会論文集, pp. 445–448 (1997).
6) Kernighan, M., Church, K. and Gale, W.: A Spelling Correction Program Based on a Noisy Channel Model, *COLING-90, Vol.2*, pp. 205–210 (1990).
7) 北研二, 中村哲, 永田昌明: 音声情報処理, 森北出版株式会社 (1996).
8) (社) 日本電子工業振興協会:『自然言語処理システムの動向に関する調査報告書』, 97-情-2 , 自然言語処理調査委員会 (1997).
9) Golding, A. and Schabes, Y.: Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction, *34th Annual Meeting of the Association for Computational Linguistics*, pp. 71–78 (1996).
10) 新納浩幸, 井佐原均: 疑似 N グラムを用いた助詞的定型表現の自動抽出, 情報処理学会論文誌, Vol. 36, No. 1, pp. 32–40 (1995).

(平成 10 年 4 月 3 日受付)
(平成 11 年 3 月 5 日採録)



新納 浩幸(正会員)

昭和 36 年生．昭和 60 年東京工業大学理学部情報科学科卒業. 昭和 62 年同大学大学院理工学研究科情報科学専攻修士課程修了. 同年富士ゼロックス，翌年松下電器を経て, 平成 5 年 4 月茨城大学工学部システム工学科助手にて着任．平成 9 年 10 月同学科講師，現在に至る．博士 (工学).