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

茨城大学 工学部 システム工学科

1 はじめに

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．

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 [1]．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は誤りの可能性があると指摘する手法である．日本語文に対しても，その効果は期待できる [2, 3]．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．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，単語 N-gram ではなく，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 を大きくとれば誤り検出，修正の精度は増す．しかし，現実的には大きな N に対しては，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 [4]，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．本論文では，平仮名 3 ～ 6-gram の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やその修正の精度を調べた．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，N の値は 4 が現実的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，ここでは 1 文字の欠落による誤りからの修正を行なうために，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 1%になる時点に最も近い頻度を閾値とする．

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 (するかどうか → 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 (するかどうか → 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 (するかどうか → 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 (するかどうか → 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．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 コーパスの大きさに応じた N

前章の設定で，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 以上の平仮名列である．長さ N より小さい平仮名列に対しては，以下のような長さ m + 2 ($0 < m < N$) 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 の値が大きいほど精度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を検出してしまう．このような不具合を避けるためには，コーパスの規模に応じた N を設定する必要がある．

3 誤り発見と修正の実験

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，及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 2000 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．

表 1: テストデータの文字列の長さ

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

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．

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．

3.2 実験結果

実験1の結果を表2に示す．一般にNが大きい方が精度は良いはずだが，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．

表2: 実験1の結果

N-gram	検出数	正解率
3-gram	313 (2000)	84.4%
4-gram	232 (2000)	88.4%
5-gram	286 (2000)	85.7%
6-gram	328 (2000)	83.6%

実験2〜5の結果を表3と表4に示す．表4中の括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3: 実験2〜5の誤り検出の結果

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	60.7%	92.1%	89.8%	92.6%	83.8%
4-gram	72.1%	95.5%	94.0%	97.0%	89.7%
5-gram	78.9%	97.4%	95.5%	98.2%	92.5%
6-gram	81.0%	97.7%	96.2%	98.2%	93.3%

実験1の正解率をp_1，実験2〜5の正解率の平均をp_2とすると，適合率(P)は$\frac{p_2}{1-p_1+p_2}$，再現率(R)は，p_2と捉えられる．P，Rを用いて，以下のF-measureにより誤り検出の評価を行なった結果を表5に示す．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．

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

5-gramが最も良いが，4-gramとの差はほとんどない．また誤り修正に限れば，4-gramの方が5-gramよりも修正できた率は高い．これは修正の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と5-gramの場合に修正の精度に大きな差はないことが予想できる．

以上より，新聞記事5年分程度から得られるN-gramを利用して誤り検出を行なう場合に，最適なNは4あるいは5であると考えられる．

表5: 誤り検出の評価

N-gram	適合率	再現率	F-measure
3-gram	84.3%	83.8%	84.1%
4-gram	88.5%	89.7%	89.1%
5-gram	86.6%	92.5%	89.4%
6-gram	83.2%	93.3%	87.9%

4 考察

4.1 最適なNについて

実験では，新聞記事5年分程度のコーパスを利用する場合には，4-gramあるいは5-gramが妥当であると結論づけた．ただしこれは誤り検出の能力だけに注目した場合である．最適なNを考える場合，検索のコストの観点も重要である．長さ4以上の平仮名列の誤り検出を行なう場合，4-gramを利用する時は4-gram表だけでよく，その種類数は約51.2万である．ところが5-gramを利用する時には，長さ4の平仮名列の頻度表と5-gram表を利用しなくてはならない．5-gram表の文字列の種類数は約105.9万である．これは4-gramの場合の約2倍に当たり，検索時間もこの比率で増える．スペルチェックとしては手軽な実装という面も必要とされるはずであり，この差は無視できない．

また閾値の問題もある．閾値を上下させることで，適合率，再現率の割合を調整できるが，5-gramの場合，実験で用いた閾値は1であり，この値の変更の自由度は少ない．一方，4-gramの場合の閾値は5であり，多少の調整が可能である．

結局，精度の差と検索時間などを総合して最適なNは求められるはずだが，新聞記事5年分程度のコーパス（これが現実的な規模のコーパスだと考える）では4-gramが現実的であると考える．6-gram以上を使う場合，精度の向上よりも検索コストの問題が大きい．

またコーパスの量による影響を調べるために，3年分の記事からN-gramを作成して同様の実験を行なった．結果は，集められた文字列の種類数は約8割に減ったが，4-gramあるいは5-gramが優れている点，誤り検出，修正に関しては4-gramと5-gramは同程度の精度である点など，5年分の記事を利用した実験と大きな差は生じなかった．最適なNとコーパスの量との関係については今後の検討が必要である．

表 4: 実験 2〜5 の誤り修正の結果

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	63.6% (775)	77.0% (1841)	79.2% (1796)	84.6% (1852)	76.1%
4-gram	78.2% (921)	87.7% (1910)	87.9% (1879)	92.4% (1939)	86.6%
5-gram	73.1% (1007)	85.4% (1947)	84.5% (1909)	88.2% (1964)	82.8%
6-gram	68.4% (1035)	81.7% (1954)	80.1% (1923)	83.5% (1963)	78.4%

4.2 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）いずれかであると仮定する →
いずれかである仮定する

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．

4.3 さらなる拡張

更に品詞 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）「太郎のゴルフクラブへの 加入が問題だ」 VS
「ゴルフクラブの太郎の 加入が問題だ」

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 [5]．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 [6]，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．

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．特に現実的な N を求めることを目的に，$N = 3,4,5,6$ 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 を利用することは効果的であること，また，新聞記事 5 年分程度のコーパスでは，$N = 4$ が実用的であると考えられること，を示した．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] Mays, Eric, Fred J. Damerau, and Robert L. Mercer : "Context based spelling collection", *Information Processing and Management*, Vol.27, No. 5, pp.517–522 (1991).

[2] 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, 7G-3, pp.181–182 (1994).

[3] 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", 情報処理学会音声言語処理研究会, SLP-19-15, pp.95–100 (1997).

[4] 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).

[5] Golding, Andrew R. and Yves Schabes : "Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction", In *34th Annual Meeting of the Association for Computational Linguistics*, pp.71–78 (1996).

[6] 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).