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

白木 伸征　黒橋 禎夫　長尾 眞

京都大学大学院工学研究科 電子通信工学専攻
{nshiraki, kuro, nagao}@kuee.kyoto-u.ac.jp

1 はじめに

現在，ワープロやパソコンなどが急速に普及し，それらを用いて日常的に大量の文書が作成されている．これらの文書の入力ミスを自動的にチェックし，訂正するスペルチェッカは実用上非常に有用である．英語のように単語の区切りに空白がある言語では，かなり性能の良いスペルチェッカが開発されている [1]．これに対して，単語の区切りが明確でない日本語文のスペルチェックは，形態素解析と同じ，あるいはそれ以上の難しさがある．これまでに行われてきた日本語スペルチェッカに関する研究でも形態素解析の手法を元にするものが中心であった [2, 3, 4]．

ところが，ワープロ，パソコンなどでのかな漢字変換に基づく入力文の特徴を考慮すれば，形態素解析とは全く異なる方法でスペルチェッカを実現することが可能となる．かな漢字変換に基づく入力では，まず平仮名列を入力し，そのうち必要な部分を漢字に変換する．このとき，漢字についてはかな漢字変換が一種の確認の作業となり，誤りは比較的少ない．また，誤りがあったとしても同音異義語などの微妙な間違いであり，これを自動検出することは非常に難しい．これに対して，平仮名列部分はかな漢字変換で何も処理されないため，誤字脱字などの単純な入力ミスが多く残ってしまう傾向にある．したがって，かな漢字変換入力文を対象とするスペルチェックでは平仮名列の扱いを中心に考えればよいといえる．

平仮名列に問題を限定すれば，それ以外の部分，すなわち漢字，カタカナの部分は平仮名列の区切りでしかないと考えられる．このように考えると，空白を単語区切りとする英語文の最も単純なスペルチェック，

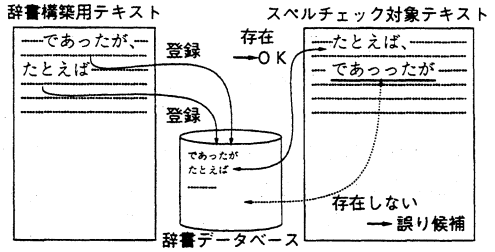

図 1: スペルチェックの方法

すなわち，空白で区切られた文字列が辞書にあるかどうかをチェックするという方法をそのまま日本語に適用することができる．すなわち，漢字，カタカナで区切られた平仮名列が辞書 (日本語としての正しい平仮名列を登録したデータベース) を調べることでスペルチェックの機能を実現することができる．

以下にその具体的な方法と評価実験について述べる．

2 方法

本研究で提案するスペルチェックの方法は，次の通りである (図 1)．

- 大量の辞書構築用テキストから平仮名列を抜き出し，ハッシュなどを用いて辞書データベースを作成する．
- スペルチェック対象テキスト中から平仮名列のみを抜き出し，辞書データベース中にその平仮名列があるかどうかを調べる．なければ誤り平仮名列候補とし，その部分を反転させるなどしてユーザに示す．ただし，1 文字の平仮名が誤っているかどうかはこの方法では調べることができないため，平仮名列の長さは 2 文字以上とする．

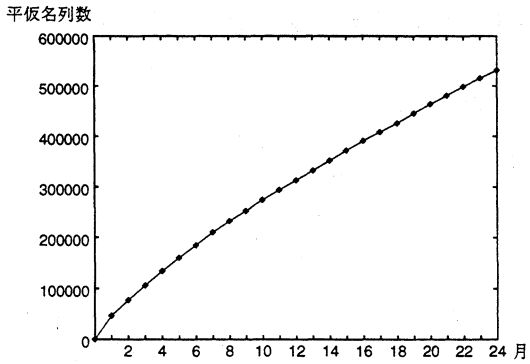

図 2: 月ごとの平仮名列の異なりの数

この方法では平仮名列が辞書データベース中に存在するかどうかを調べるだけであるため，非常に高速にスペルチェックを行うことができる．また，辞書データベースには大量の平仮名列が網羅的に必要となるが，辞書構築用テキストから平仮名列を抜き出して登録するだけなので，実現することは困難ではない．

このような方法でスペルチェッカの機能が実現可能であるかどうかを調べるために，まず予備実験を行った．予備実験に用いた辞書構築用テキストは，毎日新聞 CD-ROM の 94 年，95 年の二年分のテキストである．このテキスト中には，2 文字以上の平仮名列が 531,755 種類あった．図 2は新聞のテキスト量(月単位) と，平仮名列の異なりの数の関係を示したものである．このグラフから分かるように，24ヶ月分の新聞記事では平仮名列の種類が網羅されるわけではない．しかし，本研究ではこの規模の辞書データベースによるスペルチェックの有効性を報告する．

このような辞書データベースを用いて実際に種々のテキストに対してスペルチェックを行ったところ，以下の二つのことが分かった．

1. 平仮名列が長い場合，正しいものでも辞書データベースに存在しないことが多い．それらの平仮名列を調べたところ，平仮名列の中に「**もの**」あるいは「**こと**」を含むものが非常に多いことが分かった．そこで，「もの」，「こと」を含む平仮名列が辞書データベースに存在しない場合，その前後で平仮名列を区切り，それぞれについて辞書データベースに登録されているかどうかを調べることにした．例えば，テキストに「がないことによる」という平仮名列があり，それが辞書データベース中にない場合，「がない」と「による」を辞書データベースで調べる．これらが両方とも辞書データベース中にあれば「がないことによる」を正しい平仮名列，そうでなければ誤り平仮名列候補と判断する．

表 1: 毎日新聞中の 2 文字，3 文字平仮名列の出現頻度

	2 文字平仮名列	3 文字平仮名列
平仮名列延べ	2934971	1763013
平仮名列異なり	2746	19955
平均頻度	1068.82	88.35

2. 辞書構築用テキストに誤りの含まれている可能性がある．実験を行った結果，スペルチェック対象テキストに誤りがあっても，それが誤り候補として抜き出されないことがあった．これは次のような場合である．

「方法 **でを**(を) 用いることができる．」
「主題と述語が与 **えらた**(えられた)．」

この原因は，毎日新聞の CD-ROM テキスト中に，これらの平仮名列が次のような形で誤植として存在したためである．

「大会新の５５分５９秒 **でを** マーク，」
「命令はひそかに頴娃にも伝 **えらた**．」

大量の辞書構築用テキストから辞書データベースを構築しようとすれば，このような誤植が含まれてしまうことは避けられない問題である．そこで，辞書構築用テキスト中の平仮名列の出現頻度を調べ (表 1)，出現頻度が極端に低いものを辞書データベースから削除することにした．削除する平仮名列は，その出現頻度が同じ長さの平仮名列の出現平均頻度の 1% 以下であるものとした．具体的には，2 文字の平仮名列では頻度 10 以下のもの，3 文字の平仮名列では頻度 1 のものを削除した．4 文字の平仮名列の出現平均頻度は 26.15 であるので，4 文字以上の平仮名列では削除は行わなかった．上記の例では，「でを」は頻度が 3,

表 2: 人為的誤りに対する誤りの発見数

	誤りの発見数
ランダム位置の一文字を消去	3909 (78.18%)
ランダム位置の一文字を変更	4860 (97.20%)
ランダム位置に一文字を挿入	4930 (98.60%)

「えらた」は頻度が1であったので，辞書データベースから削除した．

次節の実験で評価するスペルチェッカは，上記の2点の改善後のシステムである．

3 実験

はじめに，提案するスペルチェックの方法の基本的な性能を調べるために，人為的に誤り平仮名列を作成し，それが本手法で発見できるかどうか，すなわち辞書データベース中に存在しない平仮名列となっているかどうかを調べた．具体的には毎日新聞中からランダムに5000種類の平仮名列を取り出し，以下のような操作を施した．

- ランダムな位置の一文字を消去 (ただし，3文字以上の平仮名列のみを対象)
- ランダムな位置の一文字を変更 (変更後の文字もランダムに選択)
- ランダムな位置に一文字を挿入 (挿入する文字もランダムに選択)

このようにして作成した誤り平仮名列が辞書データベース中に存在するかどうかを調べた結果が表2である．ランダムな位置の一文字を変更した場合と，ランダムな位置に一文字挿入した場合では，それが辞書データベース中に存在することは少なく，すなわち高い確率でその誤りを発見できることが分かった．これに対し，ランダムな位置の一文字を消去すると，それが別の正しい平仮名列となることもあり，誤りとして発見できない場合が20%程度であるということが分かった．

次に，ある論文の原稿 (全体で23,000字程度) に対してスペルチェックを行い，本手法の実際の有効性を

表 3: 論文テキスト中の誤り

1	…自然 **なつながりがもつようにする** 必要がある．
2	…「〜する」「〜した」などの表現が用 **いらる**．
3	例えば，図 **ようなの** ネットワークから，…
4	…「咲く」を含む文 **でのは** 次ような対応関係を…
5	…可能な連体節 **がである** 場合は，この連体節を…
6	…説明した方法 **でを** 用いることができる．
7	…文として出力する方が適切 **がであると** 考え，
8	主題と述語が与 **えらた** 場合に意味ネットワーク…
9	可能な場合にはすべて並列節 **といて** 出力した…

表 4: 論文テキストに対する実験結果

		辞書データベース中の有無			
		有		無	
対象テキストの	正	(a)	2208	(c)	77
平仮名列の正誤	誤	(d)	2	(b)	7

調べた．まずその論文テキストを人手で注意深く調べた結果，表3に示す9つの誤りが存在した．

この論文テキストに対してスペルチェックを行った結果をまとめると，表4のようになる．

このうち (a) は平仮名列が正しく，辞書データベース中に存在する場合で，スペルチェッカとして正しく機能している場合である．

(b) は平仮名列が誤り，辞書データベース中に存在しない場合である．これは誤り平仮名列候補として示されるものが実際に誤りである場合で，スペルチェッカとして正しく機能している．表3の 1,2,3,4,6,7,8 はこうして取り出される．

(c) は平仮名列が正しく，辞書データベース中に存在しない場合である．この場合，正しい平仮名列を誤り候補として抜き出すため，一応問題となる．しかし，実際にスペルチェックを計算機のエディタなどで用いる場合，誤り候補として示される平仮名列の数がこの程度であれば，すべてを順に調べることも十分可能である (そのうち約1割以上は (b) のように本当の誤りが発見される)．正しい平仮名列が誤り候補となる場合は，以下のように分類できる (前節で説明した「こと」，「もの」の特別処理は，この問題に対する一つ

の対策である).

- 丁寧語：「それはお読**みいただかなくとも**結構です．」
- 漢字を平仮名で表現：「出力**をならべるだけでは**不十分である．」
- 長い平仮名列：「利子が半分**ぐらいしかつかなくなってしまうことがあります**．」
- 日本語として不自然な表現：「他の文節に係**れるかどうかをひとつずつ**チェックすることである．」
- 送り仮名の違い：「多様な文章生成を行**なうためには**，」(標準的な送り仮名は「行う」)
- 口語的：「あの人の曲は伴奏が同**じっぽいし**，」

これらの平仮名列は新聞中で用いられることが少なく，辞書データベースにも存在しないため，誤り候補と判断される．

(d) は平仮名列が誤り，辞書データベース中に存在する場合である．これは誤りを見落とすという場合であり，スペルチェックとしてはもっとも深刻な問題である．実際に論文テキスト中で抜き出されなかった誤りは，表3の5の「がである」と9の「といて」である．これらの平仮名列は，それぞれ毎日新聞テキスト中の以下の部分に存在した．

> 「わが国を代表する超一流企業**がである**．」
> 「『偏見や誤解**といて**』　札幌市のイラストレーター…」
> 「病院にも特に治療の必要はない高齢者が五年，六年**といて**，…」

これらは，スペルチェック対象テキストでは誤りであるが，辞書構築用テキストでは誤りではない，すなわち平仮名列の正誤が文脈依存であるということによる．しかしこのようなものの多く，例えば「といて」などは非常に限定された文脈でしか使われないため，辞書構築用テキスト全体の中での頻度もそれほど多くはならない．したがって，辞書構築用テキスト中で出現頻度の低い平仮名列は，限定された文脈でしか用いることができないものであり，スペルチェック対象テキストの文脈でも正しいかどうかは疑わしい．そこで，このような平仮名列を**準誤り候補**としてユーザに示すことが考えられる．この頻度としては，1%~5%程度が適当であると考えられるが，この問題の具体的検証は今後の課題とする．

4　おわりに

本研究で提案した日本語スペルチェックの方法は，大量の平仮名列を用意するだけで良く，非常に単純である．それにもかかわらず，実際の試用実験でもかなり有効に働くことが確かめられた．

スペルチェック対象テキストが辞書構築用テキストと同じような文体である場合は良いが，それらが大きくずれる場合には正しい平仮名列が誤り候補となる割合が非常に高くなる．例えば，新聞記事テキストから作成した辞書データベースを用いて口語的な電子メイルのテキストのスペルチェックを行うことは難しい．そのため，スペルチェック対象テキストの文体に合わせた辞書構築用テキストを用意することが必要であるが，現在のように電子化テキストが溢れている社会ではそれも困難ではないと考えられる．

本研究で作成したスペルチェッカプログラムは，インターネットWWWの長尾研究室のホームページ `http://www-nagao.kuee.kyoto-u.ac.jp/` で公開している．

参考文献

[1] Karen Kukich. 1992. Techniques for Automatically Correcting Words in Text. ACM Computing Surveys, Vol.24, No. 4, pages 205-210.

[2] 高尾哲康，西野文人：　日本語文書リーダ後処理の実現と評価 情報処理学会論文誌 Nov. 1989. Vol.30, No.11

[3] 伊藤伸泰，丸山宏：　OCR入力された日本語文の誤り検出と自動訂正 情報処理学会論文誌 May. 1992, Vol.30, No.11

[4] Masaaki NAGATA. 1996. Context-Based Spelling Correction for Japanese OCR, In Proceedings of COLING-96, pages 806-811.