確率的LSAを用いた日本語同音異義語誤りの検出・訂正

三品拓也[†,☆] 貞光九月[††] 山本幹雄[†††]

本論文ではかな漢字変換誤り，特に同音異義語の選択誤りを対象とした日本語スペルチェックの方法を報告する．同音異義語誤りの判定には局所的な情報と大域的な情報の両者が必要であるが，本論文では大域的な情報をモデル化するために確率的 LSA を用いることを提案・検討する．評価実験として，人為的に誤りを混入させたテストデータを用いた誤り検出・訂正実験を行った．局所的な情報のモデル化に従来からよく使われている $n$gram モデルのみを利用した手法をベースラインとして比較した．ベースラインシステムでは再現率 93.8%，適合率 79.0%（F 値 85.8%）であった性能が，確率的 LSA と組み合わせることにより再現率 95.5%，適合率 83.6%（F 値 89.2%）と改善された．

Detection and Correction of Japanese Homophone Errors Using Probabilistic LSA

Takuya Mishina,[†,☆] Kugatsu Sadamitsu[††] and Mikio Yamamoto[†††]

We report a method of a Japanese spell checker for homophone errors which often occur in Japanese input process using a kana-kanji conversion system. Error detection methods need both of local and global information around a target word. In this paper, we propose and investigate use of a probabilistic LSA for modeling global information. We will show experimental results of performance to detect and correct homophone errors which are generated randomly. We use a simple method based on $n$gram models as a baseline system. Ngram models are common for Japanese spell checkers to model local information. In the results, although detection rates of the baseline system are 93.8% in recall, 79.0% in precision (85.8% in F-measure), those of a combination system of an $n$gram model and a probabilistic LSA increase to 95.5% in recall, 83.6% in precision (89.2% in F-measure).

1. はじめに

本論文では，かな漢字変換に起因する同音異義語の誤りを，$n$gram と確率的 LSA [1] を組み合わせることによって検出・訂正するシステムを提案・評価する．

日本語文を人手で入力する機会は，ワードプロセッサや PC の普及により増大しており，入力支援システム，特に入力誤りの検出・訂正システムの必要性は増している．入力誤りは 2 種類に大きく分けることができる．1 つは文字の脱落（必要な字が欠ける）・挿入（不必要な字が入る）・置換（あるべき字とは異なる字が入る）といった，文字単位での入力誤りである．もう 1 つはかな漢字変換における同音異義語誤りである．

文字単位の誤りを含む文では多くの場合形態素解析にも失敗するため，誤り検出・訂正においても文字単位での処理が要求される．既存の手法としては，文字 $n$gram や品詞 $n$gram を用いる手法[2),3)]，マルコフ連鎖モデルを用いた手法[4)]，複合語における隣接単語を用いる手法[5)] があり，OCR における文字認識誤りの訂正などでよく用いられている．しかし，本論文が対象とする人手での入力はかな漢字変換を用いる場合がほとんどであり，文字単位での誤りが発生するとそもそも漢字変換に失敗してしまう場合が多い．このため，かな漢字変換によって入力を行う場合，文字単位での誤りはその時点で気付いて修正することが多くなると考えられる[6)]．

翻って同音異義語誤りについて考えると，かな漢字変換を利用した入力では非常に多い誤りであることが容易に察せられる．同じ読みを持つ単語から文脈にふ

† 筑波大学大学院理工学研究科
Master's Program in Science and Engineering, University of Tsukuba
†† 筑波大学情報学類
College of Information Sciences, University of Tsukuba
††† 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba
☆ 現在，日本アイ・ビー・エム株式会社東京基礎研究所
Presently with Tokyo Research Laboratory, IBM Japan Ltd.

さわしい漢字を選択する作業は今のところユーザに委ねられており，単純なミスや知識不足によって容易に誤りを含む文書が作成されてしまう．たとえば以下のような誤りは，ワードプロセッサによる文書や web 上で非常によく観察される．

例 1 新しいプロセッサを 内臓［内蔵］したマシンが発表された．

例 2 ピアノを両手でうまく 引く［弾く］には練習が必要だ．

例 1 は，「内蔵」はサ変名詞であるが「内臓」はそうでないため，単語連鎖確率などの局所的な情報で比較的判定しやすい☆．例 2 は品詞が等しく，格構造も似ているため局所的には判定できない．この場合，距離は離れているが「ピアノ」などのキーワード情報から「弾く」の方が妥当であると判断されるべきである．実際の判定にはどちらの情報も重要であり，2 つの情報を併用する必要がある．

同音異義語誤りを検出・訂正する既存の手法としては，前後の単語（特に訂正対象語と複合語をなす語）との連接関係を利用する方法が多い[7)~9)]．同音異義語の選択を語義選択としてとらえた手法も多く，決定リストを用いる方法[10),11)]，trigram による局所情報と Naive Bayes による大域情報を組み合わせて誤りを検出・訂正する手法[12)] がある．

しかし，これまでの方法は，大域的とはいってもチェック対象単語の前後 3〜10 単語を見ている程度である．文献 13) によれば，大域的モデルとして Naive Bayes 法を用いた場合，前後 4 単語以上の文脈を考慮すると性能が劣化することが報告されている．しかし，明らかに 4 単語より広い範囲の情報も有効である場合も多い．たとえば，例 2 では記事全体が楽器や音楽について書かれていれば，記事全体に現れる多くのキーワードが「弾く」を支持する証拠として用いられるべきである．

本論文では，大域的な情報としてより広い範囲の情報（記事すべて）を用いるために確率的 LSA を利用し，trigram モデルを併用することにより同音異義語誤りを高い精度で検出・訂正できる方法を提案する．以下，2 章で確率的 LSA を簡単に紹介した後に，3 章で今回の提案手法で用いた trigram モデルと併用した誤り判定方法を述べ，4 章でいくつかの準備を行ったあと，5 章で新聞記事に人為的に誤りを混入させたテキストを用いて評価を行う．評価結果として，764 の同音異義語の集合（1,741 単語）を用いたとき，5%の誤りを含むテストデータに対して，再現率 95.5%・適合率 83.6%で誤りを検出できることを示す．

☆「ないぞうした」という読みに「内臓した」と変換するかな漢字変換システムはまずありえない．しかし，実際に Google で検索すると「内蔵した」が約 10 万件，「内臓した」が約 3,000 件ヒットし（2004 年 3 月現在），「ないぞうした」を含む web ページの約 3%は誤っている．

2. 確率的 LSA

確率的 LSA（Probabilistic Latent Semantic Analysis，以下 PLSA）は，複数の unigram モデルの混合モデルと考えることができる．PLSA において，文脈 h を条件とする単語 w の確率 $p(w|h)$ は，次式で与えられる．

$$p(w|h) = \sum_{t=1}^{m} p(t|h)p(w|t) \tag{1}$$

ここで t は unigram モデルの番号，m は混合数，$p(t|h)$ は文脈 h における t 番目の unigram モデルの重み，$p(w|t)$ は t 番目の unigram モデルにおける w の確率である[1)]．スペルチェックに応用する場合，文脈 h とはスペルチェック対象文書全体であり，単語 w には文書中のあいまい語（「内蔵」「引く」など）が入る．すなわち，式 (1) は対象文書全体の文脈に従って，同じ読みの単語グループ中のどの単語が出やすいかの確率を与える式であると解釈できる．各 unigram モデルを求めるためには EM アルゴリズムが使われ，以下のような訓練データ D の尤度を最大化する（学習）．

$$\mathcal{L}(D;\theta) = \sum_{w}\sum_{d\in D} n(w,d)\log\sum_{t} p(w|t)p(t|d) \tag{2}$$

$n(w,d)$ は記事 d 中の単語 w の出現頻度，$p(t|d)$ は記事 d における t 番目のモデルの重みである．実際の計算では局所最適解に落ちるのを避けるために deterministic annealing（または Tempered EM）法を用いる[1)]．

学習を行った後である未知の文脈が与えられた場合の単語の出現確率を求めるには，文脈に応じて混合比 $p(t|h)$ を決定する必要がある（適応）．Hofmann は EM アルゴリズムによって $p(t|h)$ を求めているが[1)]，EM アルゴリズムは記事に出現した単語に過適応して性能が低下する場合がある．このため，過適応を避けるために今回は変分ベイズ学習による適応を行った[14)]．

3. PLSA を利用したスペルチェック

本論文で作成するスペルチェッカは，統計的言語モデルによってスペルチェック対象文書（以下単に対象文書という）に出現した単語の確率とその同音異義語

の確率を計算し，より確率の高い方をその文脈にふさわしい単語であると判断することでスペルチェックを行う．

3.1 単語の尤度 $L(w)$ の定義

ある文脈中の単語の出現確率を求めるには様々な方法があるが，本論文ではベイズ適応による PLSA 確率と back-off スムージングされた $n$gram 確率を unigram rescaling 法[15)]☆で混合し，次のように単語 w の尤度 $L(w)$ を定義する☆☆．今回の提案手法は $n$gram+PLSA であり，残りの 2 つは比較のためのものである．

$n$gram+PLSA（提案手法）

$$L(w) = \frac{p_{PLSA}(w|h_G)}{p_{uni}(w)} p_{ngram}(w|h_L) \tag{3}$$

$n$gram（ベースライン）

$$L(w) = p_{ngram}(w|h_L) \tag{4}$$

PLSA

$$L(w) = p_{PLSA}(w|h_G) \tag{5}$$

ここで $p_{PLSA}(w|h_G)$ は PLSA によってモデル化される大域的出現確率であり，$p_{ngram}(w|h_L)$ は $n$gram でモデル化される局所的出現確率である．PLSA においては単語が出現した記事全体で適応を行い，$n$gram 確率は前向き $n$gram 確率 $p_f(w_i|w_{i-n+1}^{i-1})$ と後向き $n$gram 確率 $p_b(w_i|w_{i+1}^{i+n-1})$ との幾何平均とする．

$$p_{ngram}(w_i|h_L) = \sqrt{p_f(w_i|w_{i-n+1}^{i-1}) p_b(w_i|w_{i+1}^{i+n-1})} \tag{6}$$

ただし，文の先頭や末尾で，前向き・後向き確率の計算をしようとしても履歴部分が足りない場合は，残り一方の確率のみを用いて，幾何平均はとらないことにする．

3.2 誤り検出

対象文書中に現れた単語が同音異義語を持つ場合は，その単語は誤りである可能性がある．誤りであるかどうかを検出するため，w の尤度 $L(w)$ とその同音異義語（置換候補）w' の尤度 $L(w')$ の比の対数 $d(w, w')$ を以下のように定義する．

$$d(w, w') = \log \frac{L(w)}{L(w')} \tag{7}$$

同音異義語が複数ある場合にはその文脈において最も尤度の高い単語が置換候補であると見なして，その単語との間で d を計算する．ここでもし統計的言語モデルが文脈を完全に把握しているのであれば，$d < 0$ となった場合に出現した単語が誤りであると判断できることになる．しかし一般にそこまで完全にモデル化できるとは限らないので，本論文におけるスペルチェッカはこの d の値が一定の閾値以下になった場合には「出現した単語は誤りである」と判断し，置換候補として w' を提示することにする．

4. 実験の準備

4.1 同音異義語リストとその内部表現

誤り検出の対象となる単語および置換候補は，同音異義語リストとして用意する．日本語の同音異義語の集合を収集する方法としてはいくつかの方法が考えられるが（たとえば文献 16)），本論文では以下のような同音異義語の集合を「同音異義語リスト」として用いた．

- 読みが同一である．
- 文字数が 2 文字である．
- 少なくとも 1 文字は漢字を含む．
- 固有名詞ではない．
- アラビア数字を含まない．
- コーパス中に一定回数以上出現する．

文字数を 2 文字に制限したのは，これ以上の文字数からなる単語は，「取付け」「取付」のような，正書法に起因する同音異義語を多数含むからである．固有名詞は人名・地名などの人間でも判別不能な同音異義語を含むため除外した．また，アラビア数字を含む単語には，「1 月」「一月」など意味的に同一でスペルチェックの意味がないものが多いためこれも除外した．それらの単語を除外した後で，コーパスに一定回数以上出現する単語を選択して同音異義語リストとする．

具体的な例を**表 1** に示す．この表は毎日新聞 2000 年版[17)] に 100 回以上出現した単語の中から上記の条

表 1 同音異義語リストの例
Table 1 An example of homophone word lists.

読み	単語		
オリ	降り	下り	折り
フリ	降り	振り	
イライ	以来	依頼	
イガイ	意外	以外	
ヒク	弾く	引く	

☆ unigram rescaling 法は PLSA と $n$gram モデルを混合する 1 つの方法であり，線形補間よりも高い性能であることが報告されている．

☆☆ 提案手法では正規化を行っておらず確率の定義から外れるため，ここでは「確率」ではなく「尤度」という言葉を使うことにする．なお，式 (3) を尤度ではなく確率にしたい場合は単語全体で正規化すればよい．しかし，次章で述べる判定方法では，同一文脈における単語の確率の比を用いるため，正規化項がキャンセルされる．そのため，確率にする必要はない．

表 2 置換候補リストの例
Table 2 An example of words for replacement.

現れ	候補		
降り	下り	折り	振り
下り	降り	折り	
折り	降り	下り	
振り	降り		
以来	依頼		
依頼	以来		
意外	以外		
以外	意外		
弾く	引く		
引く	弾く		

件に合致する単語を抽出したものの一部である．全体では 764 の同音異義語の集合（1,741 単語）を得られた．

実際にシステムで使っているのは，同音異義語リストをさらに置換候補リストに変換したものである．これは**表 2** のようなリストで，ある現れ単語とその置換候補との組を記録したものである．このリストでは，読みが違っていても同じ表記となるような単語は 1 つにまとめてある．異なる読みでも表記が同じであれば，両者の候補を考慮する必要があるためである（例：表 2 の 1 行目「振り」）．

4.2 テストデータの作成

本論文がスペルチェックの対象とする文書は人間が一般的なかな漢字変換ソフトウェアを用いて作成したデータであるから，テストデータとしても人間が作成したデータを用いるのが望ましい．しかし，人間が書いた文章は個人の傾向や校正の進み具合によって誤りの混入比率が実に様々であって，典型的なテストデータを得るのはそれほど容易ではない．そこで今回はコーパスに対して人為的に一定の割合（具体的には 1%，5%，10%の 3 種類）で誤りを混入させてテストデータを作成する．その際，ある単語に対して同音異義語が複数ある場合は，置換候補のうちで最も unigram 確率の高い単語に置換する．かな漢字変換を利用する場合，ありふれた単語を稀な単語に誤るよりは，稀な単語をありふれた単語に誤る確率の方が高いことが考えられるからである☆．

4.3 閾値の決定

今回のスペルチェッカでは，閾値は訓練データからの学習で獲得することにし，現れ単語ごとに個別に決定するものとする．「現れ単語ごと」とは，表 2 の各行ごとに閾値を設定するという意味である☆☆．具体的には訓練データの一部をテストデータと見なして人為的に 5%の誤りを混入させ，一度スペルチェックを行う．ここでの誤り混入作業はテストデータ作成と同様に行う（unigram 確率最大の単語にランダムに置換する）．そこで現れ単語ごとに F 値（4.4 節参照）最大となるような閾値を設定する．なお，実験では比較としてすべての閾値を同一とする場合についても検討を行った．

4.4 評価指標

本論文では，再現率（R）・適合率（P）・F 値（F）を評価指標とする．

$$R = \frac{\text{スペルチェッカの正解数}}{\text{テストデータ中の誤り単語数}} \tag{8}$$

$$P = \frac{\text{スペルチェッカの正解数}}{\text{スペルチェッカの検出単語数}} \tag{9}$$

$$F = \frac{2 \times R \times P}{R + P} \tag{10}$$

再現率と適合率の間には，再現率を上げようとすれば適合率が下がり，適合率を上げようとすれば再現率が下がるというトレードオフの関係がある．どちらに重きを置くかはスペルチェッカの使われ方によって異なるが，今回はそれらの調和平均である F 値を用いて性能を評価する．本論文におけるスペルチェッカは指摘をどの程度厳しく行うかを閾値によって調整可能であるので，実際の応用の際は適宜閾値を変えればよい．また，ここでいう「スペルチェッカの正解数」は，検出のみが目的であれば「スペルチェッカが誤りであると判断した単語のうち，実際にテストデータ中で誤りであった単語の数」であり，訂正まで行うのであれば「スペルチェッカによって誤りを正しく訂正できた単語の数」となる．一般に検出だけするよりは訂正まで行った方が難易度は高いが，これについても実験で比較・検討を行うことにする．

4.5 実験条件

主な実験条件を**表 3** に示す．なお，PLSA モデルは文献 14) における「中頻度語彙」モデルと同一である．具体的には混合数（m）200，繰返し演算の初期値は乱数で与え，文献 14) の Tempered EM スケジュールに基づいて計 42 回の繰返しを行った．コーパスは茶筌[18] によって形態素解析を行い，1 形態素を 1 単語として扱った．また，$n$gram モデルは CMU-Cambridge SLM Toolkit [19] を用いて構築した back-off trigram

☆ もしこの仮定が誤っていたとしても，ランダムに置換するよりはスペルチェックの難度は上がるので，性能を過剰に高く見積もる危険性は少ない．

☆☆ たとえば「以外」を「意外」に誤る閾値と，逆に「意外」を「以外」に誤る閾値とは別に設定されるし，読みが違っても現れが同じ場合（「下り」「折り」「振り」を「降り」に誤る場合など）は同一の閾値が設定される．

表 3 実験条件
Table 3 Experimental conditions.

パラメータ	使用コーパス	備考
PLSA モデル	毎日新聞 99 年版	語彙は頻出 103 語を除く出現頻度上位 19,000 語
trigram モデル	毎日新聞 94〜99 年版	語彙は出現頻度上位 20,000 語
あいまい語リスト	毎日新聞 2000 年版	100 回以上出現する同音異義語 764 組 1,741 語
テストデータ	毎日新聞 2000 年版	1%・5%・10%の割合で同音異義語を置換
閾値学習データ	毎日新聞 99 年版	5%誤りを混入させた場合の F 値を最大とする閾値

（Good-Turing discounting）を用いた．

5. 実験結果

5.1 閾値の設定方法と尤度計算方法の関連

図 1 は，横軸に再現率，縦軸に適合率をとり，テストデータの誤り混入率（1%，5%，10%），尤度計算方法，閾値の設定方法の 3 つのパラメータを変えながら実験した結果である．曲線で表されているのが閾値を全単語同一として徐々に変えていった場合のもので，破線が *n*gram のみ（ラベル “ngram”），実線が *n*gram と PLSA の併用（同 “*n*gram+PLSA”）のものであり，1%，5%，10%の誤りを含む各テストデータに対応して計 6 本である．四角・丸・三角の点で表されているのが現れ単語ごとに最適な閾値を学習した場合で，白抜き点が *n*gram のみ，黒塗り点が *n*gram+PLSA を表しており，曲線と同様に 1%（四角），5%（丸），10%（三角）の誤りを含む各テストデータに対応して計 6 点である．グラフは右上端点が再現率・適合率・F 値 100%を表しており，曲線や点が右上に近いほど高性能であることを示している．

まず図 1 の中で，すべての現れ単語について同一の閾値とし，その閾値を徐々に変えていった場合の結果（曲線部分）について考察する．この場合，*n*gram のみの場合と *n*gram+PLSA では，わずかに *n*gram+PLSA の方が高性能であるが，大きな違いはない．

一方，あらかじめ現れ単語ごとに個別の閾値を学習しておき，テストセットに対して性能を評価した結果（丸・三角・四角の各点）について見てみる．この実験では，閾値は単語ごとに最良の点に設定されるので，全単語の閾値を同一とした場合よりもきめ細かく閾値を決めたことになる．この場合，*n*gram のみの場合（白抜き）よりも *n*gram+PLSA の場合（黒塗り）の方が明らかに高い性能を発揮している．*n*gram のみを用いる場合は，現れ単語ごとに個別に閾値を設定した場合と，全単語同一の閾値にした場合とでほとんど性能が同一になっていることが分かる（曲線と白抜き点がほぼ重なっている）．PLSA は単語ごとに判別能力が大きく異なるために，個別に閾値を設定する場合と全単語同一とした場合とで差が生まれるものと考えられる（5.3 節参照）．

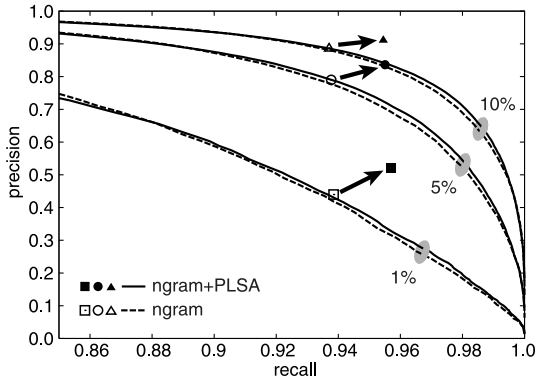

図 1 *n*gram と *n*gram+PLSA の性能比較
Fig. 1 Precision-recall curves for the 3 test sets with or without the PLSA.

なお，どのような場合にでもいえることだが，誤りを多く含む文書に対しては効果が大きい．これは現れた表記が実際に誤りである確率が高ければ高いほど，誤りの指摘が簡単だからである（誤指摘が少なくなるので適合率が上昇する）．

5.2 尤度計算方法の違いによる結果の考察

表 4 は，テストデータに混入させた単語の誤り比率と尤度計算に使う統計的言語モデルを変えた場合の再現率・適合率・F 値の結果についてまとめたものである．「混入させた誤り」の括弧内の数値は実際の誤り単語数，「検出」はスペルチェッカが誤りであると判断した単語数，「正解」は検出単語のうち実際に誤りであった単語の数である．閾値は現れ単語ごとに学習したものを用いた．

PLSA 単独と *n*gram 単独とを比較すると，*n*gram 単独の方が明らかに性能が高い．これは，同音異義語判定には大域的情報よりも局所的情報がより効果的であることを示唆する．一方 *n*gram+PLSA は *n*gram を上回る性能を発揮しており，PLSA は *n*gram がモデル化する情報とは別の大域的情報をうまくモデル化していることが分かる．

5.3 個別の単語についての考察

表 5 は，一部の同音異義語の組について個別に性能を比較したものである．表は「候補」カラムの単語

表 4 誤り混入率と尤度計算方法を変えた場合の性能比較
Table 4 Comparison of the performance for the 3 test sets and 3 types of methods.

混入させた誤り	尤度計算	検出	正解	再現率（%）	適合率（%）	F 値（%）
1%（17704）	$n$gram+PLSA	32477	16941	95.7	52.2	67.5
	$n$gram	37787	16615	93.8	44.0	59.9
	PLSA	65637	13214	74.6	20.1	31.7
5%（87874）	$n$gram+PLSA	100378	83926	95.5	83.6	89.2
	$n$gram	104269	82399	93.8	79.0	85.8
	PLSA	120246	65606	74.7	54.6	63.0
10%（177121）	$n$gram+PLSA	185197	169062	95.5	91.3	93.3
	$n$gram	186994	165968	93.7	88.8	91.2
	PLSA	185844	131934	74.5	71.0	72.7

表 5 5%の誤りを含むテストデータに対する単語ごとの性能例
Table 5 Examples of the performance for some words using the test set containing 5% errors.

現れ	候補	混入させた誤り	$n$gram	$n$gram+PLSA
			F 値（%）	F 値（%）
内臓	内蔵	10	72.0	87.0
内蔵	内臓	9	70.6	94.1
引く	弾く	8	56.0	76.2
弾く	引く	20	51.4	71.4
景観	警官	43	64.4	92.0
警官	景観	14	33.9	77.8
以来	依頼	70	94.8	95.8
依頼	以来	292	95.6	95.6
議院	議員	451	98.2	98.5
議員	議院	13	92.9	89.7
自身	自信	114	84.3	85.8
自信	自身	363	89.9	92.1
以外	意外	36	80.0	52.2
意外	以外	202	94.8	95.3
作る	創る	18	33.3	19.0
創る	作る	126	92.5	93.4
写す	移す	16	81.3	55.3
移す	写す	262	98.5	97.6

であるべき箇所が「現れ」カラムの単語として出現した際の性能を示している．たとえば第 1 行目は，本来「内蔵」であるべきところに「内臓」という誤り単語が出現した場合の誤り箇所（単語）数と性能である．

表を見ると，「引く」「弾く」のような前後広い範囲に依存すると思われる単語については，PLSA を加えることにより性能が改善されていることが分かる．逆に「作る」「創る」のような，人間でも判断するのが困難な同音異義語の場合や，「意外」「以外」のようにあまり文脈に依存しない単語の場合は，PLSA を加えることによってかえって性能が悪化している．また，名詞とサ変名詞の組の場合，「依頼」「以来」に対してはそれほどの性能改善がみられない半面，「内蔵」「内臓」に対しては PLSA との併用で性能が向上しており，PLSA の併用が有効に作用するものとそうでないものがあることが分かる．

効果の違いがどこからくるのか，**表 6** から考察してみる．この表は，「内蔵・内臓」「以来・依頼」「意外・以外」という 3 つの同音異義語の組について，それぞれの単語の直後に来る出現頻度上位 10 単語と，それぞれの単語が出現した記事の出現頻度上位 10 名詞を毎日新聞 2000 年版を用いて数え上げ，頻度順に並べたものである．直後の単語による情報はすなわち局所的依存関係を示すもので $n$gram が，記事中に出現する単語（主に名詞）による情報はすなわち大域的依存関係を示すもので PLSA がモデル化する．

「内臓」「内蔵」の組では，直後に来る単語に違いがあり，$n$gram だけを用いた場合でも一定程度の性能は出ている．しかし記事全体で単語の出現傾向を見てみると，「内臓」には特段の特徴はないが，「内蔵」は極端に情報家電系の単語が並んでおり☆，局所的情報にもまして特徴的である．このような，使われる話題が明らかに異なる同音異義語の場合，PLSA がうまく作用して，$n$gram よりも $n$gram+PLSA の方が高い性能を発揮できているものと考えられる．

一方，「以来」「依頼」のペアの場合，直後に来る単語だけを見ると，両者でかなりの違いがある．両者のリストには句読点を除いて同じものが 1 つもなく，局所的情報だけでも単語を判断しやすい．たとえば「以来」ではほとんどの場合に直後が「、」もしくは数字であり（数字は「〜以来 2 年ぶり」といった文脈で頻出する），「依頼」のそれと比較すれば判断は容易であるといえる．記事全体の傾向としては多少の違いがある程度であるが（「依頼」の「容疑」など），$n$gram だけでかなり判別がつくため，PLSA を加えても性能はそれほど変わらないという結果になったものと考えられる．

☆ テストセットが 2000 年版の新聞記事であることから，2000 年 12 月に開始された BS デジタル放送関連の単語ばかり並んでいる印象があるが，KWIC によりほかにも家電関連の文脈中に多数出現していることを確認している．

表 6 局所および大域的文脈における共起単語の例
Table 6 Examples of co-occurrence words in local and global contexts.

	内臓		内蔵		以来		依頼		意外		以外	
直後に来る単語出現頻度	23	を	49	し	2627	、	517	し	280	な	1634	の
	13	に	19	の	1021	の	176	を	279	に	863	に
	12	の	19	さ	229	。	145	さ	37	性	438	は
	11	疾患	14	テレビ	197	2	112	が	23	だっ	291	で
	10	が	7	、	152	1	94	する	10	でし	88	、
	9	破裂	6	型	139	初めて	80	。	9	や	75	でも
	4	など	5	時計	126	3	50	で	8	だ	40	から
	3	脂肪	5	。	98	と	30	、	6	。	28	が
	3	検査	4	する	91	4	29	者	5	で	23	を
	3	や	3	で	67	5	27	は	5	かも	23	も
記事全体の名詞出現頻度	342	こと	405	放送	13027	年	3243	こと	2453	こと	11235	こと
	275	人	273	デジタル	10734	日	3057	者	1894	人	8462	人
	189	年	250	こと	10633	こと	2802	人	1387	年	7813	年
	153	日	220	テレビ	9280	人	2256	日	985	的	6520	日
	113	者	212	BS	5571	的	2243	年	923	者	5991	者
	104	的	202	日	5108	日本	1225	容疑	864	日	5288	的
	100	中	170	万	5075	者	1225	的	828	日本	4035	日本
	86	もの	167	データ	3499	回	1125	同	693	中	2961	万
	79	体	154	者	3481	円	1122	万	664	それ	2917	中
	79	時	148	年	3336	万	1102	円	582	氏	2867	円

表 7 *n*gram に PLSA を加えた場合に F 値が改良・改悪された単語数
Table 7 The numbers of improved/disimproved words in F-measure when the PLSA is added.

F 値	単語種類数（現れ）
1%以上改善	901（61.0%）
ほぼ同等	186（12.6%）
1%以上改悪	391（26.4%）

表 8 5%の誤りを含むテストデータに対する誤り検出能力と誤り訂正能力の比較
Table 8 Comparison of the performance between error detection and correction using the test set containing 5% errors.

正解基準	再現率（%）	適合率（%）	F 値（%）
検出	95.5	83.6	89.2
訂正	94.9	83.1	88.6

「意外」「以外」の組では，直後に来る単語のリストに違いがあり，*n*gram による性能はかなり高い．ところが記事全体の傾向を見てみると，リストに並ぶ単語は内容だけでなく順位まで類似しており，2 つの単語はどのような文脈にもまんべんなく出現しているという傾向が読み取れる．このような同音異義語の組に対しては，PLSA を加えることによりかえって性能が悪化していることが分かる．

PLSA によって F 値が改善・改悪された単語数を**表 7** にまとめておく．全体的に見れば，多くの単語において性能が改善されていることが分かるが，改悪してしまった単語も少なくない．これらは，局所的な情報で十分に訂正できる単語に対して，PLSA による情報がノイズとして働いているためであると考えられる．つまり，PLSA の効果は改善から改悪まで単語によって大きく異なり，F 値を最大とする閾値も PLSA の付加によって単語ごとに大きく変化するものと考えられる．単語ごとに閾値を調整することにより提案手法（*n*gram+PLSA）の性能が大きく向上した（5.1 節）一因はこの点にあるものと推察される．とはいえ，改悪してしまった単語に関しては PLSA を使わないでおけばいいので，同音異義語の組ごとに PLSA の寄与度を変える（PLSA で悪化するような単語では *n*gram のみを使うことにする）ことで全体性能をさらに向上させることができると考えられる[12]．

5.4 誤り検出と誤り訂正の難易度についての考察

検出能力と訂正能力の比較を**表 8** に示す．一般的に考えて，検出よりは訂正の方が困難であるが，提案手法においては検出能力と訂正能力はほとんど同等であることが分かる．

5.5 実験結果の全体的考察

最後に全体的な結果について考察してみる．実験の結果，以下のようなことが分かった．

- *n*gram だけを用いた場合であっても，5%の誤りを含む文書に対して，再現率 93.8%，適合率 79.0%と比較的高い性能であり，PLSA を加えれば再現率 95.5%，適合率 83.6%，F 値 89.2%となる．
- 閾値をすべての現れ単語について同一とした場合は PLSA を付加しても性能はそれほど変化しな

いが，閾値をあらかじめ現れ単語ごとに学習しておく場合，PLSA を加えたほうが高性能である．
- PLSA の付加によって *n*gram ではモデル化できない大域的情報を取り込むことができるため，一部の単語を除きほとんどの現れ単語については性能が向上する．
- 誤り訂正と誤り検出の難易度はほとんど同じである．

6. おわりに

本論文では同音異義語誤りを訂正・検出するため，PLSA と *n*gram を併用したモデルによるスペルチェッカについての検討を行った．その結果，*n*gram のみを用いて尤度を計算した場合に比べて性能が高く，現れ単語ごとに閾値を調整した場合には特に高い性能を発揮することが分かった．

これ以上の性能向上を図る場合，改良を加えるべき箇所は，言語モデル部分とスペルチェック部分の 2 つに分けられる．言語モデル部分では，PLSA 自体の改良・適応範囲の最適化があげられる．今回は文脈適応を記事全体で行ったが，同一の記事中であっても距離が非常に遠い単語についてはノイズとなっている可能性がある．適応を行う際の前後の文脈範囲については適切な範囲を検討する必要がある．またスペルチェック部分については，*n*gram が得意とする同音異義語，PLSA が得意とする同音異義語があることを考えると，両者の重みを単語ごとに変えたりといった工夫を施すことでさらに性能を向上させられる可能性がある．今後はこれらについてさらに検討し，改良を図って実用に堪えるスペルチェッカを作りたい．

参考文献

1) Hofmann, T.: Probabilistic Latent Semantic Indexing, *Proc. 22nd Annual ACM Conference on Research and Development in Information Retrieval*, Berkeley, California, pp.50–57 (1999).
2) 新納浩幸：平仮名 N-gram による平仮名列の誤り検出とその修正，情報処理学会論文誌，Vol.40, No.6, pp.2690–2698 (1999).
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，情報処理学会研究報告 SLP-19-15 (1997).
4) 荒木哲郎，池原 悟，佐藤政伸，榮代正男：マルコフ連鎖モデルを用いた日本語文の置換型，挿入型及び脱落型誤りの検出・訂正法の改善，電子情報通信学会論文誌，Vol.J85-D-II, No.1, pp.66–78 (2002).
5) 奥 雅博：日本語文推敲支援システム REVISE における複合語同音異義語誤りの検出および訂正支援手法，電子情報通信学会論文誌，Vol.J79-D-II, No.11 (1996).
6) 田中穂積（監修）：自然言語処理—基礎と応用，（社）電子情報通信学会 (1999).
7) 伊吹 潤，徐 国偉，斉藤孝広，松井くにお：校正支援システム Joyner における表記誤りの訂正方式，情報処理学会研究報告 NL-117-21 (1997).
8) 脇田早紀子，金子 宏：変換ミスチェッカーのための辞書生成，情報処理学会研究報告 NL-111-5 (1996).
9) 奥 雅博，松岡浩司：文字連鎖を用いた複合語同音異義語誤りの検出とその評価，自然言語処理，Vol.4, No.3, pp.83–99 (1997).
10) 新納浩幸：複合語からの証拠に重みをつけた決定リストによる同音異義語判別，情報処理学会論文誌，Vol.39, No.12, pp.3200–3206 (1998).
11) 新納浩幸：表記情報をデフォルトの証拠として用いた決定リストによる同音異義語の誤り検出，情報処理学会論文誌，Vol.41, No.4, pp.1046–1053 (2000).
12) Golding, A. and Schabes, Y.: Combining trigram-based and feature-based methods for context-sensitive spelling correction, *Proc. 34th ACL*, pp.71–78 (1996).
13) Golding, A.: A Bayesian hybrid method for context-sensitive spelling correction, *Proc. 3rd WVLC*, pp.39–53 (1995).
14) 三品拓也，山本幹雄：確率的 LSA に基づく ngram モデルの変分ベイズ学習を利用した文脈適応化，信学技報 NLC2002-73, pp.13–18 (2002).
15) Gildea, D. and Hofmann, T.: Topic-based language models using em, *Proc. 6th European Conference on Speech Communication and Technology* (*EUROSPEECH*) (1999).
16) 新納浩幸：誤りやすい同音異義語の収集，情報処理学会研究報告 NL-126-1 (1998).
17) 毎日新聞社：CD-毎日新聞 1994 年版–2000 年版，日外アソシエーツ.
18) 松本裕治，北内 啓，山下達雄，平野善隆，松田寛，高岡一馬，浅原正幸：形態素解析システム『茶筌』version 2.2.3 使用説明書 (2001).
19) Clarkson, P. and Rosenfeld, R.: Statistical Language Modeling Using the CMU-Cambridge Toolkit, *Proc. ESCA Eurospeech* (1997).

(平成 15 年 10 月 22 日受付)
(平成 16 年 7 月 1 日採録)

三品　拓也（学生会員）

昭和 55 年生．平成 16 年筑波大学大学院理工学研究科修了．同年日本アイ・ビー・エム（株）入社，現在同社東京基礎研究所研究員．

貞光　九月

昭和 56 年生．平成 16 年筑波大学情報学類卒業．現在筑波大学大学院システム情報工学研究科在学中．自然言語処理の研究に従事．

山本　幹雄（正会員）

昭和 61 年豊橋技術科学大学大学院修士課程修了．同年（株）沖テクノシステムズラボラトリ研究開発員．昭和 63 年豊橋技術科学大学情報工学系教務職員．平成 4 年同助手．平成 7 年筑波大学電子・情報工学系講師．平成 10 年同助教授．博士（工学）．自然言語処理，音声言語情報処理の研究に従事．電子情報通信学会，言語処理学会，人工知能学会，音響学会，ACL 各会員．