自 然 言 語 処 理 146－2
（２００１．１１．２０）

PPM法を用いたかな漢字変換の学習モデル

丸 山 卓 久[†]　田中 (石井) 久美子[††]　武 市 正 人[†††]

かな漢字変換では、ユーザが過去に入力した文章中の情報を学習する。学習手法には、1. 前に入力した単語を次回先頭候補にする、2. 単語の頻度情報を登録する、3. 単語の共起情報を登録する、4. 単語の n-gram 情報を登録する、のような手法が考えられる。1〜3 までの手法を用いた学習モデルは、既に存在するが、4 の単語 n-gram 情報を用いたモデルはあまり例が無いと思われる。そこで、本研究では、単語 n-gram 情報を用い、圧縮の分野で用いられている学習手法である PPM(Prediction by Partial Matching) 法を用いたかな漢字変換の学習手モデルを提案し、その性能評価を行った。

A Learning Model for Kana-Kanji Conversion Systems based on PPM Method

Takahisa Maruyama,[†] Kumiko Tanaka-Ishii[††]
and Masato Takeichi[†††]

The efficiency of kana-kanji conversion systems can be improved dynamically by the user's personal corpus accumulated while the user processes text. The learning of such corpus is performed in real time by updating either of the following informations of the candidate preferences: frequency, co-occurrence, or n-gram. In this paper, we first propose a learning model based on PPM (Prediction by Partial Match), that is one of the learning methods based on n-grams. Then, we report its efficiency by comparing the result with those of other models.

1. はじめに

近年、多くの人が個人で、PC や PDA や携帯電話などのコンピュータを所有しており、それらを用いて日本語を入力する機会が多い。個人でこのような機械を所有し日本語入力を行う機会が多ければ、そのかな漢字変換システムを個人ごとに個別化できれば、効率の良い日本語入力が実現できる。そこで、近年のかな漢字変換システムは、ユーザーが過去に入力した文章中の情報を学習することにより、個人ごとにシステムを個別化し、より効率の良い変換を試みている。学習法には様々な手法が考えられ、

(1)　一度入力した単語を、次回先頭候補とする
(2)　単語の出現頻度を用いる
(3)　単語の共起情報を用いる
(4)　単語 n-gram の情報を用いる

のような手法がある。これらの学習手法のうち、1 から 3 は従来のかな漢字変換システムで用いられている。例えば、ATOK[6] では 2、3 が、MS-IME[5] では、1、2、3 が、Wnn[3] では 2 が、SKK[4] では 2 が用いられている。しかし、4 の手法を用いた学習モデルは、あまり例が無いと思われる。そこで、本研究では、4 の単語 n-gram 情報を用いる学習手法を提案する。単語 n-gram を用いる学習手法に圧縮の分野で用いられている PPM 法がある[1)2)]。今回は、PPM 法を用いたかな漢字変換の学習モデルを構築し、その性能評価を行った。

また、1 から 4 の学習手法は、登録している情

† 東京大学大学院工学系研究科情報工学専攻
Department of Information Engineering, School of Engineering University of Tokyo
†† 東京大学大学院情報学環
Interfaculty Initiative in Information Studies, Graduate School of The University of Tokyo
††† 東京大学大学院情報理工学系研究科数理情報学専攻
Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo

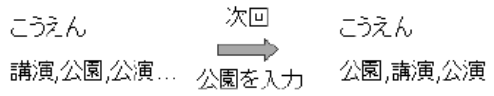

図 1 次回先頭候補にする手法

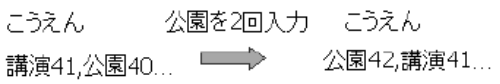

図 2 頻度情報を蓄える手法

報が動的に変化していくが、これらの学習手法の性能評価は我々の知る限り行われていない。これらの学習手法の優劣を評価することは、今後かな漢字変換システムを構築する際に非常に重要である。そこで、1 から 3 の手法について学習モデルを構築し、4 の PPM 法を用いる学習モデルとの性能評価を行った。

2. 学習手法

本章では、前章で述べた 4 つのかな漢字変換の学習手法について、詳しく述べる。

2.1 入力単語を次回先頭候補にする手法

この手法は、最近入力した単語は直後に出現しやすいという考え方に基づいた手法である。この手法では、前に入力した単語を、次に同じひらがな列から変換するときがあれば、その単語を第一候補にする。この手法は、実際のかな漢字変換では、MS-IME などで行われている。以下では、この手法を MTF(Move To Front) の手法と呼ぶ。

候補の順位の決定は、まず前に入力した単語を先頭に持ってくる。そして、その前まで先頭にあった単語は一つ順位を落として、第二候補となる。このようにして、一番最近に出現した単語を先頭にくるように積み重ねていく。例えば、ある時点で「こうえん」という入力に対して候補が「講演、公園...」という順番で出現し、「公園」で決定したとする。そうすると次に「こうえん」を入力するときは「公園、講演...」という順番で候補が出てくる (図 1)。以上のような手順で、候補の順位付けを行っている。

2.2 頻度情報を蓄える手法

単語の出現頻度により、候補の順位付けを行う手法である。入力した単語の頻度を 1 増やすことにより、学習を行っている。実際にかな漢字変換では、MS-IME、ATOK、Wnn、SKK などでこの手法は用いられている。以下では、この手法を頻度の手法と呼ぶ。

候補の順位の決定は、その候補の単独での出現頻度の大きさで、順位付けを行う手法である。例えばある時点で、「こうえん」という読みに対して、「講演」の出現頻度が 41 回で 1 番目、「公園」が 40 回で 2 番目だったとする。ここで、「こうえん」の候補を頻度順に並べるので、「講演、公園...」という順に現れる。これ以降に「公園」を 2 回入力すると「公園」の頻度は 42 回となり「講演」の 41 回よりも大きくなる。そのため、次に「こうえん」を入力するときは、「公園、講演」という順に出現することになる (図 2)。

2.3 共起情報を蓄えていく手法

入力した文章の文脈情報を、ある単語とある単語が同時に出現するという共起関係を蓄えることにより学習を行う手法である。この手法は、MS-IME、ATOK などで用いられている。以下では、この手法を共起の手法と呼ぶ。まず初めに、共起情報を説明する。

過去の入力で、「先生 が 講演 する」という文章が現れたとする。スペースは文節区切りである。このときの共起情報は、「先生 が」、「先生 講演」、「先生 する」、「が 講演」、「が する」... のような二つの単語対である。ここで、「講堂 で 先生 が こうえん」という文章の「こうえん」を変換したいとする。「こうえん」の候補として「講演、公園...」が考えられる。このとき、「講堂 講演」、「先生 講演」や「講堂 公園」、「先生 公園」などの共起関係を考えることにより候補の順位を決定していく (図 3)。過去に共起情報を利用した文書中に含まれる同音異義語の誤りの訂正の研究があり、良い結果を得ている[7]。そこで、今回は、そこで用いられていた手法をかな漢字変換に応用することを考えた。以下では、順位決定手法を詳しく説明する。

ある単語列 $C = w_1, w_2, \ldots, w_{n-1}$ が存在し、この次に来る単語 w_n を予測することを考える。予測したい単語 w_n は、その単語の d 文字前ま

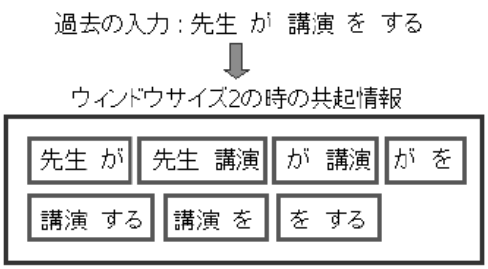

図 3 共起情報

での単語列に依存する。d をウィンドウサイズと呼ぶ。この単語列の後にくる最適な w_n を求める問題は、以下のように表せる。

$$\hat{w_n} = argmax_{w_n} P(w_n|w_{n-d},\ldots,w_{n-1}) \quad (2.1)$$

これをベイズの定理を用いて変形する。

$$\hat{w_n} = argmax_{w_n} \frac{P(w_{n-d},\ldots,w_{n-1}|w_n)P(w_n)}{P(w_{n-d},\ldots,w_{n-1})} \quad (2.2)$$

ここで、分母は w_n に依存しないので、無視することができる。

$$\hat{w_n} = argmax_{w_n} P(w_{n-d},\ldots,w_{n-1}|w_n)P(w_n) \quad (2.3)$$

ここで、$w_{n-d},\ldots,w_{n-1}$ は互いに独立に出現すると仮定すると、

$$\begin{aligned} & P(w_{n-d},\ldots,w_{n-1}|w_n)P(w_n) \\ & \cong P(w_n)\prod_{i=n-d}^{n-1} P(w_i|w_n) \\ & = P(w_n)\prod_{i=n-d}^{n-1} \frac{P(w_i,w_n)}{P(w_n)} \end{aligned} \quad (2.4)$$

となる。$P(w_n)$ は、w_n の出現回数、$P(w_i,w_n)$ は、$(w_i \quad w_n)$ という共起関係の出現回数を表している。

それぞれの候補について上式を計算し、その値の大きい順に候補の順位付けを行う。

2.4 n-gram 情報を蓄える手法

入力した文章の文脈情報を、単語 n-gram 情報を蓄えることにより学習を行う手法である。このような学習手法として PPM(Prediction by Partial Matching) 法がある。PPM 法は、圧縮の分野で用いられている手法であり、前の文脈を学習することにより効率の良い圧縮を実現する手法である。今回は、この学習法を、かな漢字変換の学習に適用することを考えた。以下では、PPM 法について述べる。

PPM 法では、$c = w_1 w_2 \ldots w_{n-1}$ という単語列が出現しているとき、その次に w_n という単語がくる確率 $P(w_n)$ を、単語 n-gram を用いて推定する。この確率は、以下のようにして求める。

$$P(w_n) = \sum_{d=-1}^{dmax} u_d P_d(w_n) \quad (2.5)$$

d は、オーダーで、w_n の d 文字前まで文脈であるとみなす、つまり w_n の決定に d 文字前までの情報を用いる。例えば、$d = 2$ のときは、w_n、w_{n-1} を文脈であるとみなし、この二つの単語との n-gram 情報を用いて $P_2(w_n)$ を計算する。この確率に重み u_d をかけて、加算することにより $P(w_n)$ を計算する。重みは、正規化されており、文脈が長い、つまりオーダーが大きいほど大きな重みをかけるようになっている。

$d \geq 0$ のとき $P_d(w_n)$ は以下のようにして求める。

$$P_d(w_n) = \frac{c_d(w_n)}{C_d} \quad (2.6)$$

C_d はオーダー d の文脈の出現頻度、$c_d w_n$ はオーダー d の文脈の次に w_n が現れた回数を表している。$d = -1$ のとき、$P_{-1}(w_n)$ は全ての単語が等確率で出現するとしたときの確率である。

重み u_d は以下のようにして求める。

$$w_d = (1 - e_d)\prod_{i=d+1}^{m} e_i \quad -1 \leq d < m \quad (2.7)$$

ここで、m は、$P_d(w_n)$ を求めることができる、最大の d である。PPM 法には上式で現れた e_d の求め方により、いくつかの種類がある。今回は PPMA 法、PPMC 法の二つを用いた。PPMA 法、PPMC 法における e_d の求め方は以下の通りである。

PPMA:

$$e_d = \frac{1}{C_d + 1} \quad (2.8)$$

PPMC:

$$e_d = \frac{q_d}{C_d + q_d} \quad (2.9)$$

q_d は、ユーザーが過去に入力した文章中で、オーダー d の文脈の後に出現した異なり単語数である。

また、すべてのオーダーについて、$P_d(w_n)$ を求めると、計算に非常に時間がかかってしまう。そこで、オーダーが大きいほうから順に $P_d(w_n)$ を求めていき、あるオーダーにおいて $P_d(w_n) \neq 0$ になったら、それより下のオーダーについては計算を行わないことにより計算時間を短縮する手法がある。この手法を exclusion と呼ぶ。exclusion は PPMA、PPMC の両方に適用が可能である。以上のようにして $P(w_n)$ を求め、この値が大きい順に候補の順位付けを行う。

3. 実験データ

前章で述べた 4 つの手法は、種々の確率や、頻度情報を利用することにより、候補の順位付けを行っている。そのため、これらの確率や、頻度を算出する必要があり、辞書、コーパスを用いて算出する。これには、基本辞書、学習コーパスの 2 つがある。また、性能評価を行うためには、実際にひらがなから漢字などに変換するテストデータも必要である。以下では、これら 3 つについて説明する。

3.1 基本辞書

この辞書は、ひらがなから候補を列挙するために用いられる。また、それぞれの単語の出現頻度を保持している。そのため、この辞書の一項目は、「ひらがな列 ひらがなに対応する単語 単語の出現頻度」のような構成になっている。また、この辞書は 94 年度毎日新聞の記事 1 年分を用いて作成した 186898 単語により構成されている。

3.2 学習コーパス

これは、ユーザごとに学習を行うためのコーパスである。つまり、このコーパスから単語に対する読み、単語の出現頻度情報、単語間の共起情報、n-gram 情報などを抽出し、個別化を行う。また、このコーパスは個人のメールから作成したものと、94 年度毎日新聞の社会面から作成したものと、プログラミング言語のテキストから作成したもの、の 3 種類とした。それぞれについて、学習量の差による評価を行うため、0 から 5 万単語の学習コーパスを作成した。表 1 に個人のメール、表 2 に社会面、表 3 にテキストの学習コーパスの詳細を示す。

表 1 学習コーパス (メール)

総単語数	異なり単語数	基本辞書と重なる単語数
10000	1935	1746
20000	2808	2575
50000	4514	4049

表 2 学習コーパス (社会面)

総単語数	異なり単語数	基本辞書と重なる単語数
10000	3128	2962
20000	5036	4829
50000	8985	8630

表 3 学習コーパス (テキスト)

総単語数	異なり単語数	基本辞書と重なる単語数
10000	2006	1450
20000	2868	2004
50000	4286	2884

3.3 テストデータ

実際にかな漢字変換テストを行うためのデータである。個人のメール、新聞の社会面及びプログラミング言語のテキストのうち学習コーパスとして使用しなかった残りの部分を抽出し、それをすべて形態素に区切り、ひらがなに直したものである。このひらがなを漢字などに変換し候補を列挙する。表 4 に個人のメール、表 5 に社会面、表 6 にプログラミング言語のテキストのテストコーパスの詳細を示す。

4. 実　　験

前章で述べた、基本辞書、学習コーパスを用いて、頻度や確率などのユーザ情報を算出し、それによりテストデータのひらがなを漢字などに変換した。評価法は、列挙した候補の中で正解が何番目に現れているかを、3 万単語のテストデータについて平均を取った値で評価する。つまり、この値が小さいほど性能が良いということである。

4.1 各手法の比較および学習量による比較

上に述べた 4 つの手法を用いて、学習量の差による性能差を見るために学習量を「なし」から「5 万単語」まで変化させ、変換実験を行った。PPM 法および、共起の方法における、ウィンドウサイズは 4 とした。実験結果は、メールが図 4、社会面が図 5、テキストが図 6 のようなグラフとなった。縦軸が正解が出現する順位の平均値、横軸が学習量 (× 10000 words) であ

表 4　テストコーパス (メール)

総単語数	異なり単語数	基本辞書と重なる単語数
30000	3381	3054

学習コーパスと重なる単語数			両方と重なる単語数		
1 万	2 万	5 万	1 万	2 万	5 万
1336	1750	2289	3111	3138	3191

表 5　テストコーパス (社会面)

総単語数	異なり単語数	基本辞書と重なる単語数
30000	6345	6067

学習コーパスと重なる単語数			両方と重なる単語数		
1 万	2 万	5 万	1 万	2 万	5 万
1666	2401	3478	6201	6234	6297

表 6　テストコーパス (テキスト)

総単語数	異なり単語数	基本辞書と重なる単語数
30000	2632	1782

学習コーパスと重なる単語数			両方と重なる単語数		
1 万	2 万	5 万	1 万	2 万	5 万
1442	1754	2086	2140	2218	2338

る。メール、社会面、テキストともにグラフは上から順に、頻度の手法、MTF の手法、共起の手法、PPMC(exclusion)、PPMC(exclusion なし)、PPMA(exclusion)、PPMA(exclusion なし) となっている。メールの結果の PPMA(exclusion なし) を見てみると、5000 単語学習したときは、平均で、約 1.15 番目に正解が現れていることになる。正解が 1 番目に出てくる単語を除いて、平均値を求めたところ平均して約 3 番目に正解が現れている。また、平均値が 1.15 のとき、115 回入力すると 100 単語入力できる。この二つのことより 100 単語中約 92 単語が 1 回で入力できるということが分かる。同様にして、学習なしの時の平均値約 1.3 のときは、100 単語中 85 単語が 1 回で入力できることが分かる。また、メールの実験結果において PPMA 法 (exclusion なし) で 10 万単語学習させたとき、1.06 になるが、このときは 100 単語中 97 単語が 1 回で入力できることになる。

この結果を見ると PPMA(exclusion なし) の手法が一番良く変換できているが、PPM 法の中での性能差はごくわずかなものである。また、exclusion をする場合としない場合での差はわずかなものであり、計算時間を考慮すると exclusion をする場合のほうがより実用的ではないかと思われる。

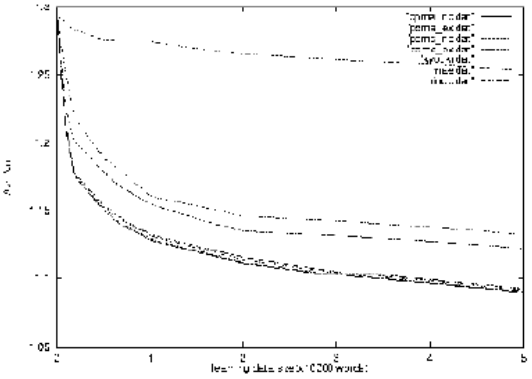

図 4　種々の手法の結果 (メール)

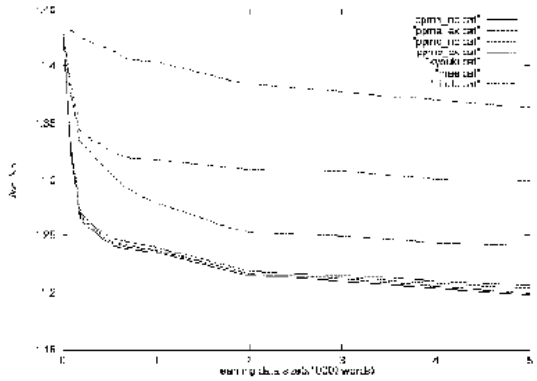

図 5　種々の手法の結果 (社会面)

また、学習量による差は、MTF、頻度、共起、PPM 法のいずれの手法においても似たような傾向を示しており、学習量が少ない間は性能が非常に良く上がっているが、学習量が多くなってくると上がり方が減ってくる。特に、学習の効果が顕著に現れているのは、学習量が 1 万単語から 2 万単語の間までであり、それ以上学習してもあまり効果はあがっていない。

5. まとめと今後の展望

以上の結果より、単語 n-gram を用いる学習手法である PPM 法が一番良く変換が行えていることが分かった。exclusion をする場合としない場合では、しない場合のほうがわずかに性能は

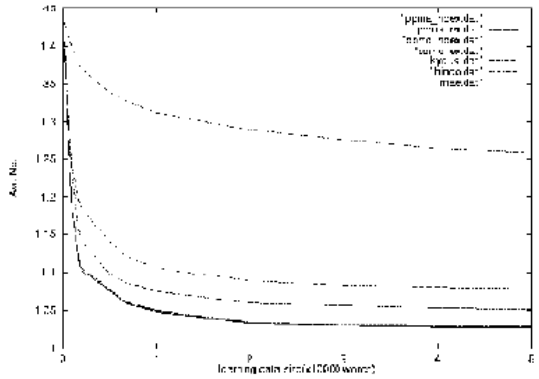

図 6 種々の手法の結果 (テキスト)

良いが、計算時間は exclusion をするほうが短縮できているため、PPMA 法の exclusion が一番適当であると思われる。

最近使用した単語を先頭候補にする手法では、最近使用した単語は直後に出現しやすいという考え方に基づいている。そこで、この考え方を応用して PPM 法において蓄積する n-gram を最近覚えた n-gram により大きい重みを加えることにより、さらに PPM 法の性能を上げることができるのではないかと考えられる。そこで、今後このような手法も考えていく予定である。

参 考 文 献

1) Timothy C.Bell, John G.Cleary, Ian H.Witten, "TEXT COMPRESSION", PRENTICE HALL
2) W.J Teahan, "Probability estimation for PPM", In Proceedings NZCSRSC'95. Available from http://www.cs.waikato.ac.nz/wjt/papers/NZCSRSC.ps.gz
3) http://www.tomo.gr.jp
4) http://openlab.ring.gr.jp/skk/index-j.html
5) 日経バイト AUGUST 1999
6) http://www.justsystem.co.jp/tech/atok/text_00.html
7) Hideya Iwasaki, Kumiko Tanaka-Ishii, Tateno Kei, Masato Takeichi, "Context Sensitive Detection and Correction of Homonym Errors for Japanese Text", IRAL, September, 2000