無限語彙の仮名漢字変換

森 信介
日本IBM 東京基礎研究所
〒 242-8502 大和市下鶴間 1623-14
mori@fw.ipsj.or.jp

あ ら ま し
本論文では、確率的言語モデルによる仮名漢字変換において、語彙を事実上無限にする枠組を提案する。この枠組により、未知語であっても、新聞やウェブに表記が現れ、読みが各文字の可能な読みの組合せとなっている限りにおいて変換候補として挙げることが可能になる。さらに、確率的言語モデルにより前後の文脈を考慮して、未知語を含む変換候補に適切な順位を付けることができ、高い変換精度が期待できる。実験では、一般的な分野の確率的言語モデルによる仮名漢字変換システムを別の分野に適応することを行なった。実験の結果、提案手法により、一般分野の精度を損ねることなく適応分野の精度の向上を実現可能であることが示された。特に未知語であっても、正しい変換結果を得ることが可能になったことが提案手法の最大の特徴である。

キーワード　確率的言語モデル　確率的単語分割　コーパス　語彙　仮名漢字変換

Kana-Kanji Converter with an Infinite Vocabulary

Shinsuke Mori
Tokyo Research Laboratory, IBM Japan
1623-14 Shimotsuruma Yamatoshi
Kanagawaken 242-8502 Japan
mori@fw.ipsj.or.jp

Abstract
In this paper we propose a framework of a *kana-kanji* converter with an infinite vocabulary. With this framework the converter can enumerate even unknown words as conversion candidates when the string of the unknown words appear in a newspaper or a web page. In addition, the converter order the candidates of known words and unknown words properly by referring to their contexts. In the experiments, we adopted a *kana-kanji* converter based on a stochastic LM in the general domain to a special domain. The results showed that our framework is capable of improving the *kana-kanji* converter applied to a special domain without any degradation in the general domain. The main contribution is that the NLP system can propose even unknown words as the output by our framework.

Key Words Stochastic Language Model, Stochastic Segmentation, Corpus, Vocabulary, *Kana-kanji* Converter

1　はじめに

自然言語を出力とする言語処理技術の多くが確率的言語モデルを用いている。その代表例の音声認識 [1] は、音響モデルとともに確率的言語モデルを参照し、複数の候補の中から最尤の文字列を選択する。他に文字誤り訂正 [2] や仮名漢字変換 [3] やステノタイプ (速記記号) からの書き起こし [4] などが、同様の枠組で実現されている。

一般に、これらの確率的言語モデルを生成的に用いる言語処理においては、予め与えられた確率的言語モデルの語彙に含まれない単語を出力することができない。したがって、精度向上のために語彙とその文脈情報の追加が必要である。この問題は、新たな分野に言語処理システムを適応する際により顕著となる。

この問題を解決するために、本論文では、仮名漢字変換を確率的言語モデルの応用例として、新聞やウェブなどの機械可読文書に出現するすべての部分文字列を出力候補とする自然言語生成システムの枠組を提案する。具体的には、まず、新聞やウェブなどの機械可読文章を確率的単語分割コーパス [5] とみなし、確率的単語分割コーパスの全ての部分文字列を語彙 (状態) とする単語 n-gram モデル (マルコフモデル) を構築しておく。次に、仮名漢字変換の際に、入力記号列 (片仮名列) の一部に対応する確率的単語分割コーパスの部分文字列を列挙し、最も確率の高い変換候補列を単語 n-gram モデルによって算出する。この方法により、未知語であっても、新聞やウェブに表記が現れ、読みが各文字の可能な読みの組合せとなっている限りにおいて変換候補として挙げることが可能になる。現在、あらゆる分野のあらゆる話題に関して大量の機械可読文章が利用可能であるので、提案する枠組の自然言語処理システムの語彙数は事実上無限となる。

実験では、一般的な分野の確率的言語モデルによる仮名漢字変換システムを別の分野に適応することを行なった。実験の結果、提案手法により、一般分野の精度を損ねることなく適応分野の精度の向上を実現可能であることが示された。特に未知語であっても、正しい変換結果を得ることが可能になったことが提案手法の最大の特徴である。

2　確率的言語モデルとその応用

自然言語処理における確率的言語モデルの役割は、与えられた文字列がある言語の文である尤度を数値化することである。確率的言語モデルに基づく言語処理は、候補から解を選択する際にこの尤度を参照する。自動単語分割は解析系の一例であり、文字列を与えられると尤度が最大になる単語列を計算する。認識系の代表例の音声認識器では、音響信号列を入力として、尤度が最大となる文字列を算出する際に、音響モデルと併せて確率的言語モデルを参照する。

2.1　確率的言語モデル

日本語の確率的言語モデルは、日本語のアルファベット列 $\mathcal{X}^*$ が出現する確率値を記述する。これは、以下のように表される。

$$P : \mathcal{X}^* \mapsto [0,\ 1]$$

確率的モデルであるので、確率値をすべてのアルファベット列に渡って合計すると 1 以下になる必要がある。

$$\sum_{\boldsymbol{x} \in \mathcal{X}^*} P(\boldsymbol{x}) \leq 1$$

最も一般的な言語モデルは単語 n-gram モデルである。このモデルは、文を単語列 $\boldsymbol{w}_1^h = w_1 w_2 \cdots w_h$ とみなし、これらを文頭から順に予測する。

$$M_{w,n}(\boldsymbol{w}) = \prod_{i=1}^{h+1} P(w_i | \boldsymbol{w}_{i-n+1}^{i-1})$$

この式の中の w_i $(i \leq 0)$ は、文頭に対応する特別な記号であり、w_{h+1} は、文末に対応する特別な記号である。完全な語彙を定義することは不可能であるから、未知語を表わす特別な記号 UW を用意する。未知語の予測の際は、まず、単語 n-gram モデルにより UW を予測し、さらにその表記 $\boldsymbol{x}_1^{h'}$ を以下の文字 n-gram モデルにより予測する。

$$M_{x,n}(\boldsymbol{x}_1^{h'}) = \prod_{i=1}^{h'+1} P(x_i | \boldsymbol{x}_{i-n+1}^{i-1}) \tag{1}$$

この式の中の x_i $(i \leq 0)$ は、語頭に対応する特別な記号であり、$x_{h'+1}$ は、語末に対応する特別な記号である。したがって、未知語は以下のように予測される。

$$P(w_i | \boldsymbol{w}_{i-n+1}^{i-1}) = M_{x,n}(w_i) P(\mathtt{UW} | \boldsymbol{w}_{i-n+1}^{i-1})$$

2.2　自動単語分割

単語境界を明示しない言語においては、単語単位の確定が自然言語処理における最初の問題である。この問題を解決するために、単語 n-gram モデルに基づく自動単語分割器が提案されている [6] 。この方法では、以下の式で表されるように、文の生成確率が最大となる単語列を自動分割結果とする。

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}=\boldsymbol{x}}{\mathbf{argmax}}\, M_{w,n}(\boldsymbol{w})$$

永田 [6] は、10,945 文をパラメータ推定に用いて、約 97%の精度を報告している。

2.3 仮名漢字変換

確率的言語モデルによる仮名漢字変換 [3] は、キーボードから直接入力可能な記号 $\mathcal{Y}$ の正閉包 $\boldsymbol{y} \in \mathcal{Y}^+$ を入力として、変換候補 $(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots)$ を確率 $P(\boldsymbol{x}|\boldsymbol{y})$ の降順に提示する。

$$im(\boldsymbol{y}) = (\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots)$$

$$i \leq j \Leftrightarrow P(\boldsymbol{x}_i|\boldsymbol{y}) \geq P(\boldsymbol{x}_j|\boldsymbol{y})$$

この式から、仮名漢字変換器の主要な役割は、各変換候補の確率値 $P(\boldsymbol{x}|\boldsymbol{y})$ の順序関係の算出であることがわかる。逆にこの順序関係を保持している限りにおいて、実際にはこの確率値以外の値を用いてもよいと結論できる。この点を考慮に入れて、以下の式のように確率的言語モデルの分離が行なわれる。

$$\begin{aligned} & P(\boldsymbol{x}_i|\boldsymbol{y}) \geq P(\boldsymbol{x}_j|\boldsymbol{y}) \\ \Leftrightarrow\quad & \frac{P(\boldsymbol{y}|\boldsymbol{x}_i)P(\boldsymbol{x}_i)}{P(\boldsymbol{y})} \geq \frac{P(\boldsymbol{y}|\boldsymbol{x}_j)P(\boldsymbol{x}_j)}{P(\boldsymbol{y})} \\ & \qquad (\because \text{ベイズの公式}) \\ \Leftrightarrow\quad & P(\boldsymbol{y}|\boldsymbol{x}_i)P(\boldsymbol{x}_i) \geq P(\boldsymbol{y}|\boldsymbol{x}_j)P(\boldsymbol{x}_j) \qquad (2) \\ & \qquad (\because P(\boldsymbol{y}) \text{は} \boldsymbol{x}_i \text{や} \boldsymbol{x}_j \text{によらない}) \end{aligned}$$

この式において、日本語文 $\boldsymbol{x}$ の出現確率を表す $P(\boldsymbol{x})$ が確率的言語モデルであり、上述の単語 n-gram モデルを用いることができる。残りの $P(\boldsymbol{y}|\boldsymbol{x})$ は、日本語文 $\boldsymbol{x}$ が与えられたときのキーボードからの入力の記号列 (読み) の確率を表す。これは確率的仮名漢字モデルと呼ばれる。

確率的仮名漢字モデル $P(\boldsymbol{y}|\boldsymbol{x})$ は、日本語文 $\boldsymbol{x}$ が与えられたときのキーボードからの入力の記号列 $\boldsymbol{y}$ の確率を表す。あらゆる可能な日本語文に対する入力記号列の確率を推定することは不可能であるから、日本語文を単語に分割し、単語と入力記号列との対応関係がそれぞれ独立であると仮定する。このとき、単語列 $\boldsymbol{w}$ が与えられたときの入力記号列 $\boldsymbol{y}$ の確率的仮名漢字モデル M_{kk} による出現確率は以下の式で表される。

$$M_{kk}(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i) \qquad (3)$$

ここで、入力記号部分列 $\boldsymbol{y}_i$ は単語 w_i に対応する入力記号列であり、以下の条件を満たす。

$$\boldsymbol{y} = \boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_h$$

確率 $P(\boldsymbol{y}_i|w_i)$ の値は、単語ごとに読み (入力記号列) が振られたコーパスから以下の式を用いて最尤推定することで得られる。

$$P(\boldsymbol{y}_i|w_i) = \frac{f(\boldsymbol{y}_i,\ w_i)}{f(w_i)} \qquad (4)$$

この式中の $f(e)$ は、事象 e のコーパスにおける頻度を表す。

未知語に対する変換モデルは提案されておらず、仮名漢字変換器 [3] は単に入力記号列 (主に片仮名) を返す[1] 。これは、確率的言語モデルの未知語モデル $M_{x,n}(\boldsymbol{x})$ を入力記号列の未知語モデル $M_{y,n}(\boldsymbol{y})$ に置き換えることで実現される。

以上から、単語 n-gram モデルと単語単位の確率的仮名漢字モデルからなる仮名漢字変換器は、変換候補を以下の値の順に列挙する。

$$P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i)P(w_i)$$

$$\begin{aligned} & P(\boldsymbol{y}_i|w_i)P(w_i) \qquad (5) \\ = & \begin{cases} P(w_i|\boldsymbol{w}_{i-n+1}^{i-1})P(\boldsymbol{y}_i|w_i) & \text{if } w_i \in \mathcal{W} \\ P(\mathtt{UW}|\boldsymbol{w}_{i-n+1}^{i-1})M_{y,n}(\boldsymbol{y}_i) & \text{if } w_i \notin \mathcal{W} \end{cases} \end{aligned}$$

ここで $\mathcal{W}$ は確率的言語モデルの語彙を表す。

3 確率的単語分割コーパスからの言語モデルの推定

確率的言語モデルを新たな分野に適応する一般的な方法は、適応分野のコーパスを用意し、それを自動的に単語分割し、単語の頻度統計を計算することである。この方法では、単語分割誤りにより適応分野のコーパスにのみ出現する単語が適切に扱えないという問題が起こる。この解決方法として、適応分野のコーパスを確率的単語分割コーパスとして用いることが提案されている [5]。この節では、確率的単語分割コーパスからの確率的言語モデルの推定方法について概説する。

3.1 確率的単語分割コーパス

確率的単語分割コーパスは、生コーパス C_r(以下、文字列 $\boldsymbol{x}_1^{n_r}$ として参照) とその連続する各 2 文字 x_i, x_{i+1} の間に単語境界が存在する確率 P_i の組として定義される。最初の文字の前と最後の文字の後には単語境界が存在するとみなせるので、$i = 0,\ i = n_r$ の時は便宜的に $P_i = 1$ とされる。確率変数 X_i を

$$X_i = \begin{cases} 1 & x_i, x_{i+1} \text{の間に単語境界が存在する場合} \\ 0 & x_i, x_{i+1} \text{が同じ単語に属する場合} \end{cases}$$

とし $(P(X_i = 1) = P_i, P(X_i = 0) = 1 - P_i)$、各 $X_0, X_1, \ldots, X_{n_r}$ は独立であることが仮定される。

[1] 文献 [3] によると、約 33.0%の未知語が片仮名列のままで正しい変換である。

文献 [5] の実験で用いられている単語境界確率の推定方法は次の通りである。まず、単語に分割されたコーパスに対して自動単語分割システムの境界推定精度 α を計算しておく。次に、適応分野のコーパスを自動単語分割し、その出力において単語境界であると判定された点では $P_i = \alpha$ とし、単語境界でないと判定された点では $P_i = 1 - \alpha$ とする。後述する本研究の実験においてもこの方法を採用した。

3.2 単語 n-gram 頻度

確率的単語分割コーパスに対して単語 n-gram 頻度が以下のように定義される。

単語 0-gram 頻度 確率的単語分割コーパスの期待単語数として以下のように定義される。

$$f(\cdot) = 1 + \sum_{i=1}^{n_r - 1} P_i \tag{6}$$

単語 1-gram 頻度 確率的単語分割コーパスに出現する文字列 $\boldsymbol{x}_{i+1}^{k}$ が $l = k - i$ 文字からなる単語 $w = \boldsymbol{x'}_1^l$ である必要十分条件は以下の 4 つである。

1. 文字列 $\boldsymbol{x}_{i+1}^{k}$ が単語 w に等しい ($\boldsymbol{x}_{i+1}^{k} = \boldsymbol{x'}_1^l$)。
2. 文字 x_{i+1} の直前に単語境界がある ($X_i = 1$)。
3. 単語境界が文字列中にない ($X_j = 0,\ \ i+1 \leq \forall j \leq k-1$)。
4. 文字 x_k の直後に単語境界がある ($X_k = 1$)。

したがって、確率的単語分割コーパスの単語 1-gram 頻度 f_r は、単語 w の表記の全ての出現 $O_1 = \{(i,k) \mid \boldsymbol{x}_{i+1}^{k} = w\}$ に対する期待頻度の和として以下のように定義される。

$$f_r(w) = \sum_{(i,k) \in O_1} P_i \left[\prod_{j=i+1}^{k-1} (1 - P_j) \right] P_k \tag{7}$$

単語 n-gram 頻度 ($n \geq 2$) L 文字からなる単語列 $\boldsymbol{w}_1^n = \boldsymbol{x'}_1^L$ の確率的単語分割コーパス $\boldsymbol{x}_1^{n_r}$ における頻度、すなわち単語 n-gram 頻度について考える。このような単語列に相当する文字列が確率的単語分割コーパスの $(i+1)$ 文字目から始まり $k = i + L$ 文字目で終る文字列と等しく ($\boldsymbol{x}_{i+1}^{k} = \boldsymbol{x'}_1^L$)、単語列に含まれる各単語 w_m に相当する文字列が確率的単語分割コーパスの b_m 文字目から始まり e_m 文字目で終る文字列と等しい ($\boldsymbol{x}_{b_m}^{e_m} = w_m,\ 1 \leq \forall m \leq n;\ e_m + 1 = b_{m+1},\ 1 \leq \forall m \leq n-1;\ b_1 = i+1;\ e_n = k$) 状況を考える。確率的単語分割コーパスに出現する文字列 $\boldsymbol{x}_{i+1}^{k}$ が単語列 $\boldsymbol{w}_1^n = \boldsymbol{x'}_1^L$ である必要十分条件は以下の 4 つである。

1. 文字列 $\boldsymbol{x}_{i+1}^{k}$ が単語列 $\boldsymbol{w}_1^n$ に等しい ($\boldsymbol{x}_{i+1}^{k} = \boldsymbol{x'}_1^L$)。
2. 文字 x_{i+1} の直前に単語境界がある ($X_i = 1$)。
3. 単語境界が各単語に対応する文字列中にない ($X_j = 0,\ \ b_m \leq \forall j \leq e_m - 1,\ \ 1 \leq \forall m \leq n$)。
4. 単語境界が各単語に対応する文字列の後にある ($X_{e_m} = 1,\ \ 1 \leq \forall m \leq n$)。

確率的単語分割コーパスにおける単語 n-gram 頻度は以下のように定義される。

$$f_r(\boldsymbol{w}_1^n) = \sum_{(i,e_1^n) \in O_n} P_i \left[\prod_{m=1}^{n} \left\{ \prod_{j=b_m}^{e_m - 1} (1 - P_j) \right\} P_{e_m} \right]$$

ここで

$$\begin{aligned} e_1^n &= (e_1, e_2, \cdots, e_n) \\ O_n &= \{(i, e_1^n) | \boldsymbol{x}_{b_m}^{e_m} = w_m, 1 \leq m \leq n\} \end{aligned}$$

とした。

3.3 単語 n-gram 確率

決定的に単語に分割されたコーパスからの単語 n-gram 確率の最尤推定の場合と同様に、確率的単語分割コーパスにおける単語 n-gram 確率は、単語 n-gram 頻度の相対値として最尤推定される。

単語 1-gram 確率 以下のように単語 1-gram 頻度を単語 0-gram 頻度で除することで計算される。

$$P_r(w) = \frac{f_r(w)}{f_r(\cdot)} \tag{8}$$

単語 n-gram 確率 ($n \geq 2$) 以下のように単語 n-gram 頻度を単語 $(n-1)$-gram 頻度で除することで計算される。

$$P_r(w_n | \boldsymbol{w}_1^{n-1}) = \frac{f_r(\boldsymbol{w}_1^n)}{f_r(\boldsymbol{w}_1^{n-1})} \tag{9}$$

4 無限語彙の仮名漢字変換

生コーパスを確率的単語分割コーパスとみなすことで、生コーパスに出現する全ての部分文字列を語彙とする単語 n-gram モデルを推定することができる。現在、あらゆる分野のあらゆる話題に関して巨大な生コーパスが利用可能であるので、この単語 n-gram モデルを用いる仮名漢字変換システムの語彙数は事実上無限であるといってよい。この節では、このような無限語彙の仮名漢字変換システムの詳細を説明する。

4.1 単語候補の列挙

第 2.3 項で説明した仮名漢字変換の辞書は、入力記号列を受け取り、表記と式 (4) で与えられる確率値の組を返す。これが複数ある場合には全てを返す。これと同様に、無限語彙の仮名漢字変換の辞書も、入力記号列 $\boldsymbol{y}$ を受け取り、可能な表記 w とその確率 $P(\boldsymbol{y}|w)$ を単語候補として列挙する必要がある。この実現方法を以下で説明する。

1. まず、各文字 x に対して可能な入力記号列の集合 $\boldsymbol{\mathcal{Y}}_x = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k\}$ を記述した単漢字辞書を用意する。例えば $x =$「日」に対して $\boldsymbol{\mathcal{Y}}_{\text{日}} = \{$ カ, ジツ, ニチ, ニッ, ヒ, ビ $\}$ である。
2. 次に、全ての文字に対する入力記号列の集合の和集合 $\boldsymbol{\mathcal{Y}}_{\mathcal{X}} = \bigcup_{x\in\mathcal{X}} \boldsymbol{\mathcal{Y}}_x$ を既知語とする単漢字の仮名漢字変換システムを作成する。これは、入力記号列 $\boldsymbol{y}$ を受け取り、これに対応する全ての可能な文字列 w を入力記号列の生成確率 $P(\boldsymbol{y}|w)$ とともに返す。例えば、部分入力記号列 $\boldsymbol{y} =$「ニッテレ」が与えられると、可能な文字列 (変換候補) の集合として $\mathcal{W} =$ { 日テレ, 日手レ, 日照レ, ニッテレ, ニッ手レ, ニッ照レ, 荷ッテレ, $\cdots$} を生成確率とともに返す。
3. 確率 $P(\boldsymbol{y}|w)$ については、さまざまな与え方が考えられる。唯一の条件は、$w = x_1x_2\cdots x_m$ を所与として $P(\boldsymbol{y}|w)$ がアルファベット $\mathcal{Y}$ 上の確率的言語モデル (第 2.1 項参照) となっていることである。後述する実験においては、各文字の入力記号列の出現確率は一様分布であると仮定し、

$$P(\boldsymbol{y}|w) = P(\boldsymbol{y}|x_1x_2\cdots x_m) = \prod_{i=1}^{m} \frac{1}{|\boldsymbol{\mathcal{Y}}_{x_i}|} \qquad (10)$$

とした[2] 。例えば、$P($ニッテレ $|$ 日テレ$) = \frac{1}{|\boldsymbol{\mathcal{Y}}_{\text{日}}|}\frac{1}{|\boldsymbol{\mathcal{Y}}_{\text{テ}}|}\frac{1}{|\boldsymbol{\mathcal{Y}}_{\text{レ}}|} = \frac{1}{6}\times\frac{1}{1}\times\frac{1}{1} = \frac{1}{6}$ となる[3] 。

以上で述べた単語候補を列挙するモジュールは、入力記号列を受け取り、単漢字辞書にある部分入力記号列への分解を列挙する。実装においては、仮名漢字変換と同様に動的計画法 [7] を用いる。後述する実験においては、計算コストの削減のために、このモジュールが受け取る入力記号列を現実的な探索範囲 (最長 16 文字) に制限している。

4.2 単語候補の文脈のモデル化

大量の単語候補の中から適切な変換候補を選択するために、確率的単語分割コーパスから第 3 節で述べた方法で構築した単語 n-gram モデルを利用する。しかしながら、このモデルは、既知語の出現文脈を記述するという観点からは、人手により正確に単語に分割されたコーパスから推定された言語モデルほど正確ではないと考えられる。したがって、これらのモデルを以下のように補間する。

$$P(w_i|H_i) = \lambda_s P_s(w_i|H_i) + \lambda_r P_r(w_i|H_i)$$

この式中の H_i は、単語 w_i を予測する際の履歴であり、P_s と P_r はそれぞれ単語分割済みコーパス C_s から推定した確率と確率的単語分割コーパス C_r から推定した確率を表す。さらに λ_s と λ_r は両モデルの補間係数であり、削除補間 [8] によって求める。

後述する実験では、単語分割済みコーパスから推定した単語 2-gram モデルを確率的単語分割コーパスから推定した単語 2-gram モデルと補間した。したがって、実験における無限語彙の仮名漢字変換の言語モデルは以下の式を用いて次の単語の出現確率を記述する。

$$P(w_i) = \begin{cases} \lambda_s P_s(w_i|w_{i-1}) + \lambda_r P_r(w_i|w_{i-1}) \\ \quad \text{if } w_i \in \mathcal{W} \\ \lambda_s P_s(\texttt{UW}|w_{i-1}) M_{x,n}(w_i) + \lambda_r P_r(w_i|w_{i-1}) \\ \quad \text{if } w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r \\ \lambda_s P_s(\texttt{UW}|w_{i-1}) M_{x,n}(w_i), \quad \because P_r(w_i) = 0 \\ \quad \text{if } w_i \notin \mathcal{W} \wedge w_i \notin \mathcal{S}_r \end{cases} \qquad (11)$$

ここで $\mathcal{S}_r$ は生コーパスに出現する全ての部分文字列からなる集合を表す。

式 (11) と既存の確率的言語モデルとの差異は、生コーパスに出現する未知語を予測する場合、つまり $w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r$ の場合であっても、ある程度信頼できる頻度情報や文脈情報が参照できることである。また、生コーパスの自動単語分割の結果から推定した単語 n-gram モデルと補間している場合でも、自動単語分割の結果が誤っている場合には同様の差異が顕在化する。例えば、生コーパス中の文字列「日テレ」の自動分割結果において「日」と「テ」の間に単語境界があると推定される場合、$P($日テレ$) = 0$ となり、単語「日テレ」が変換候補に挙がることはない[4] 。しかしながら、提案手法では、文字列「日テレ」が生コーパスに出現していれば $P($日テレ$) > 0$ となり、これが変換候補に挙がる。さらに、この文字列の生コーパス中における頻度が高ければ、自動分割により必ず誤まって分割されるとしても $P($日テレ$) \gg 0$ となり、変換候補の上位にくる可能性が高くなる。

[2] 正確には、1 つの表記から同一の入力記号列が複数の方法で生成されることがあり、この場合には全ての生成方法による生成確率の和を計算する必要がある。

[3] $\boldsymbol{\mathcal{Y}}_{\text{テ}} = \{$ テ $\}, \boldsymbol{\mathcal{Y}}_{\text{レ}} = \{$ レ $\}$

[4] 「日」と「テレ」の間に単語境界を置く候補として出現する可能性はある。一般に、仮名漢字変換において単語という概念が必要か否か議論の余地はあるが、本論文ではこの点には触れない。

4.3 仮名漢字モデル

非常に稀にではあるが、与えられた入力記号列が既知語の入力記号列と生コーパスの部分文字列に対応する入力記号列に分解できないことがあり得る。この場合にも変換結果を出力することを保証するために、全ての入力記号にデフォルトの表記を決めておき、入力記号列に対してこのデフォルトの表記の列も候補として挙げるようにする。日本語ではこれを片仮名とするのが精度の観点から最良である [3]。各片仮名に対応する入力記号は 1 つ ($|\boldsymbol{\mathcal{Y}}_{x_i}| = 1$) であるから、

$$P(y_1 y_2 \cdots y_m | x_1 x_2 \cdots x_m) = 1 \tag{12}$$

となる。

式 (4)(10)(12) から無限語彙の仮名漢字変換における確率的仮名漢字モデルは以下のようになる。

$$P(\boldsymbol{y}_i|w_i) \tag{13}$$
$$= \begin{cases} \dfrac{f(\boldsymbol{y}_i,\ w_i)}{f(w_i)} & \text{if } \ w_i \in \mathcal{W} \\ \displaystyle\prod_{j=1}^{m} \frac{1}{|\boldsymbol{\mathcal{Y}}_{x_j}|} & \text{if } \ w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r \\ 1 & \text{if } \ w_i \notin \mathcal{W} \wedge w_i \notin \mathcal{S}_r \end{cases}$$

ここで $w_i = x_1 x_2 \cdots x_m$ である。

4.4 無限語彙の仮名漢字変換

以上から、本論文で提案する無限語彙の仮名漢字変換器は、式 (11) で与えられる確率的言語モデル $P(w_i)$ と式 (13) で与えられる確率的仮名漢字モデル $P(\boldsymbol{y}_i|w_i)$ からなる以下の評価関数の値の順に候補 $\boldsymbol{x} = w_1 w_2 \cdots w_h$ を列挙する。

$$P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i)P(w_i)$$

式 (11) と式 (13) の場合分けは全く同一であるので、合計 3 通りに場合分けされる点に注意されたい。

5 評価

無限語彙の仮名漢字変換の評価として、単語分割済みの学習コーパスがある分野と生コーパスだけがある分野において、一文に対応する入力記号列を一括変換することで得られる最尤解の精度を測定した。この節では、この結果を提示し評価を行なう。

表 1: 一般コーパス (単語分割済み)

用 途	文 数	単語数	文字数
学 習	20,808	406,021	598,264
テスト	2,311	45,180	66,874

表 2: 適応対象コーパス (単語境界情報なし)

用 途	文 数	単語数	文字数
学 習	797,345	—	17,645,920
テスト	1,000	—	20,935

5.1 実験の条件

実験に用いたコーパスは、主に新聞記事や辞書の例文からなる一般コーパスと業務日報からなる適応対象のコーパスである。一般コーパスの各文は正しく単語に分割され、各単語に入力記号列 (読み) が付与されている。これを 10 個に分割し、この内の 9 個を学習コーパスとし、残りの 1 個をテストコーパスとした (表 1 参照)。一方、適応対象のコーパスは大量にあるが、単語境界情報を持たない。この内の 1,000 文に入力記号列 (読み) を付与しテストコーパスとし (表 2 参照)、残りを確率的単語分割コーパスとして言語モデルの学習に用いた。

5.2 評価基準

実験で用いた評価基準は、各文を一括変換することで得られる最尤解と正解との最長共通部分列 (LCS; longest common subsequence)[9] の文字数に基づく再現率と適合率である。正解コーパスに含まれる文字数を N_{COR} とし、一括変換の結果に含まれる文字数を N_{SYS} とし、これらの最長共通部分列の文字数を N_{LCS} とすると、再現率は N_{LCS}/N_{COR} と定義され、適合率は N_{LCS}/N_{SYS} と定義される。

5.3 比較モデル

適応分野のコーパスの利用方法の差を調べるために、3 つの確率的言語モデルを推定し、これらに基づく仮名漢字変換器の変換精度を比較した。以下では、これらのモデルについて説明する。

モデル $\mathcal{B}$: Baseline
一般分野の単語分割済みコーパスから推定した単語

表 4: 仮名漢字変換の精度

モデル		一般分野 適合率	一般分野 再現率	適応対象分野 適合率	適応対象分野 再現率
$\mathcal{B}$	単語分割済みコーパスから推定した単語 2-gram モデル	89.80%	92.30%	68.62%	78.40%
$\mathcal{D}$	$\mathcal{B}$ と自動分割結果から推定した単語 2-gram モデルの補間	92.52%	93.17%	90.35%	93.48%
$\mathcal{S}$	$\mathcal{B}$ と確率分割結果から推定した単語 2-gram モデルの補間	92.78%	93.40%	91.10%	94.09%

表 3: テストコーパスの統計的性質

	一般分野	適応分野
カバー率	96.31%	89.02%
エントロピー	4.931	7.438
単語境界推定精度	98.52%	—

適応分野に対する値は自動単語分割の結果による。

表 5: 確率的単語分割コーパスの量と変換精度の関係

生コーパスの量		適合率	再現率
1.765×10^5 文字	(1/100 倍)	89.18%	92.32%
1.765×10^6 文字	(1/10 倍)	90.33%	93.40%
1.765×10^7 文字	(1/1 倍)	91.10%	94.09%

2-gram モデル

語彙は 9 個の部分学習コーパスの 2 つ以上に出現する 10,728 単語とした。このモデルによる一般分野と適応分野のテストコーパスに対する統計的性質を表 3 に示す。また、残りの 2 つのモデル推定で用いる自動単語分割器は、このモデルに基づき第 2 節で説明した方法で実現されている。

モデル $\mathcal{D}$: Decisive segmentation
上記のモデル $\mathcal{B}$ と適応分野の生コーパスの自動単語分割の結果から推定した単語 2-gram モデルを補間したモデル

モデル $\mathcal{S}$: Stochastic segmentation
上記のモデル $\mathcal{B}$ と適応分野の生コーパスの確率的単語分割の結果から推定した単語 2-gram モデルを補間したモデル

5.4 評価

各モデルの変換精度を表 4 に掲げる。モデル $\mathcal{B}$ とモデル $\mathcal{D}$ の適応分野に対する精度の比較から、従来の知見通りに、誤りを含む自動解析結果としてであっても、適応分野の生コーパスを利用することが言語モデルの適応に寄与することが分かる。また、一般分野に対しても、適応分野の生コーパスの自動解析結果を利用することで精度が向上している。このことから、過適応が起こっていないことが分かる。

モデル $\mathcal{D}$ とモデル $\mathcal{S}$ の精度の比較から、提案手法を利用することにより、同じ資源を用いてさらなる精度向上が両分野において実現されることが分かる。精度向上の主な要因は、未知語モデルにより既知語以外の候補も挙げることが可能になり、確率的に分割された生コーパスから推定した単語 n-gram 確率を参照することで文脈上適切な単語が選択されることである。表 4 をより詳しく見ると、文字誤り率 (100% – 再現率) の削減率は、一般分野においては 3.37%であるが、適応分野においては 9.36%となっており、一般分野における誤りの削減率よりも適応分野における誤りの削減率が大きいことが分かる。この理由は、適応対象分野に特有の単語や単語列の出現箇所において自動単語分割が誤りやすいことと、提案手法が既存手法ほどにその誤りの影響を受けないことである。

適応分野の確率的単語分割コーパスを増やすことによる変換精度向上の効果を調べるために、確率的単語分割コーパスの量を約 1/1 倍と 1/10 倍と 1/100 倍にして、適応対象の分野のテストコーパスの変換精度を測定した (表 5 参照)。この結果から、適応分野のコーパスを増やすことでさらなる精度向上が実現できることが分かる。

以上から、仮名漢字変換を用いたい分野のコーパスを可能な限り大量に収集し、これを確率的単語分割コーパスとし、提案手法を用いることで精度向上が図れることが分かった。この結果から、仮名漢字変換のみならず、音声認識や文字誤り訂正などの自然言語処理装置を新たな分野において用いる際に提案手法が有効であると結論できる。

6 関連研究

未知語の問題に対する代表的な対処法は、未知語モデルを用いる方法と、何らかの基準でコーパスなどから単語候補を抽出し辞書に追加しておく方法に大別できる。以下では、これらと提案手法との関係を順に述べる。

未知語モデルを用いる方法では、未知語の表記の生成確率を文字 n-gram モデルなどにより計算し、形態素解析などにおいて入力文のあらゆる部分文字列が未知語である可能性を考慮して解探索を行なう [6] 。品詞毎に未知語モデルを構築しておくことで未知語の品詞推定も同時に行なえる。確率的モデルによる仮名漢字変換においても同様の方法で未知語を変換候補に挙げることが可能であるが、全ての未知語が未知語を代表する一つのクラス (または品詞毎のクラス) から生成されることになるので、前後の単語という文脈情報はほとんど利用できず、正しい変換結果が期待できない。本論文で提案する手法では、この問題を確率的単語分割コーパスを用いることにより解決している。

機械可読文書 (コーパス) が利用可能な場合には、予め単語候補を抽出し辞書に追加しておく方法が提案されている。この方法の長所は、単語候補のコーパスにおけるすべての出現箇所を考慮することである。文献 [10] では Forward-Backword アルゴリズムにより単語候補の 1-gram 頻度を計算する方法を提案している。文献 [11] では、前後の文字の分布に注目することで未知語候補を抽出し、さらに各品詞として用いられる確率を推定しておき、これを辞書に追加することで形態素解析の精度が向上できることを報告している。これら予め単語候補を抽出しておく方法と比較すると、本論文で提案する枠組は、単語候補の抽出とその出現文脈情報の計算を、仮名漢字変換などの解探索の際に動的に行なっているとみることができる。本論文で提案する枠組の利点は、コーパスのすべての部分文字列を単語候補とすること (単語抽出の再現率が 100%) と、単語抽出とその結果の解析時の参照を生成確率という一貫した評価基準で行なうことによってより高い精度が期待できることである。

7 おわりに

本論文では、仮名漢字変換を例として、確率的言語モデルの応用において語彙を事実上無限にする枠組を提案した。この枠組により、未知語であっても、新聞やウェブに表記が現れ、読みが各文字の可能な読みの組合せとなっている限りにおいて変換候補として挙げることが可能になった。さらに、確率的言語モデルにより各語彙の前後の文脈を考慮して、未知語を含む変換候補に適切な順位を付けることにより高い精度が実現できることを示した。

実験では、一般的な分野の確率的言語モデルによる仮名漢字変換システムを別の分野に適応することを行なった。実験の結果、提案手法により、一般分野の精度を損ねることなく適応分野の精度の向上が実現可能であることが示された。特に未知語であっても、正しい変換結果を得ることが可能になったことが提案手法の最大の特徴である。

参考文献

[1] F. Jelinek. Self-organized language modeling for speech recognition. Technical report, IBM T. J. Watson Research Center, 1985.

[2] Masaaki Nagata. Context-based spelling correction for Japanese OCR. In *Proc. of the COLING96*, 1996.

[3] 森信介, 土屋雅稔, 山地治, 長尾真. 確率的モデルによる仮名漢字変換. 情処論, Vol. 40, No. 7, pp. 2946–2953, 1999.

[4] Anne-Marie Derouault and Bernard Merialdo. Natural language modeling for phoneme-to-text transcription. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 742–749, 1986.

[5] Shinsuke Mori and Daisuke Takuma. Word n-gram probability estimation from a Japanese raw corpus. In *ICSLP*, 2004.

[6] Masaaki Nagata. A stochastic Japanese morphological analyzer using a forward-DP backward-A* n-best search algorithm. In *Proc. of the COLING94*, pp. 201–207, 1994.

[7] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[8] Fredelick Jelinek, Robert L. Mercer, and Salim Roukos. Principles of lexical language modeling for speech recognition. In *Advances in Speech Signal Processing*, chapter 21, pp. 651–699. Dekker, 1991.

[9] Alfred V. Aho. 文字列中のパターン照合のためのアルゴリズム. コンピュータ基礎理論ハンドブック, I: 形式的モデルと意味論, pp. 263–304. Elseveir Science Publishers, 1990.

[10] Masaaki Nagata. Automatic extraction of new words from Japanese texts using generalized forward-backword search. In *EMNLP*, 1996.

[11] Shinsuke Mori and Makoto Nagao. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proc. of the COLING96*, 1996.