2004－NL－162　(26)
2004／7／16

生コーパスからの単語 N-gram 確率の推定

森 信介　宅間 大介
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, ta9ma@jp.ibm.com

あ ら ま し
確率的言語モデルは、音声認識やスペルチェッカーなどの言語処理において重要な役割を担っている。最も一般的な確率的言語モデルは単語 n-gram モデルであるが、実用的な予測力を実現するには、正しく単語に分割された対象分野のコーパスが必要である。日本語では単語を明示しないので、自動単語分割の結果を人手で修正する。これには、対象分野の語彙の知識がある作業者があたる必要があり、多大な時間とコストがかかる。この問題を解決するために、本論文では生コーパスから単語 n-gram 確率を推定する方法を提案する。この方法を用いれば、対象分野の例文を収集するだけで確率的言語モデルの能力が向上する。実験では、確率的言語モデルの応用の一つとしての仮名漢字変換器の精度が、生コーパスを利用することにより向上することを示す。

キーワード　確率的言語モデル　生コーパス　音声認識　仮名漢字変換　スペルチェッカー

Word N-gram Probability Estimation from a Raw Corpus

Shinsuke Mori　Daisuke Takuma
Tokyo Research Laboratory, IBM Japan
1623-14 Shimotsuruma Yamatoshi
Kanagawaken 242-8502 Japan
mori@fw.ipsj.or.jp, ta9ma@jp.ibm.com

Abstract
Statistical language modeling plays an important role in a state-of-the-art language processing system, such as speech recognizer, spelling checker, etc. The most used language model (LM) is word n-gram model, which needs sentences annotated with word boundary information. In this paper, we present a method to build a language model directly from a raw corpus. In the experiments, we estimated an LM from newspaper articles, interpolated it with an LM built from a relatively small segmented corpus, and tested it in an input method in Japanese. An input method based on our method eliminated about 15% of the errors in a baseline model.

Key Words　Stochastic Language Model, Raw Corpus, Speech Recognition, *Kana-kanji* Converter, Spelling Checker

1 はじめに

実用化に至っている言語処理技術の多くは、確率的言語モデルに基づいている。音声認識システム [1] の多くが、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。文字誤り訂正 [2] では、確率的言語モデルの尤度に基づいて不自然な文字列とその訂正候補を列挙する。単語分割や形態素解析 [3] に代表される言語解析においても、ユーザーがその出力を直接必要としないので、本質的に必要かという疑問はあるが、少なくとも研究者や開発者の道具として確率的言語モデルを利用する方法が用いられている。

確率的言語モデルは、単語や文字の頻度に基づいており、応用分野の大量の例文 (コーパス) が必要不可欠である。実用的なモデルは単語を単位としているので、コーパスには単語境界の情報が付与されている必要がある。したがって、日本語などの単語境界が明示されない言語のモデル構築には、コーパス中の大量の文に単語境界を正しく付与する作業が不可欠である。例えば、音声認識を医療分野に応用する場合には、カルテや医療所見の機械可読の例文を収集し、それら例文を、新聞などの既存のコーパスから構築された自動単語分割システムにより単語に分割し、その結果を一文ずつ人手で修正する。このようにして得られた単語分割済みコーパスから認識語彙を選択し、読みを付与し、コーパスにおける頻度を計数することで医療分野向けの音声認識システムの言語モデルが出来上がる。実用に耐える認識精度を確保するためには、一般的に最低数万文からなる単語分割済みコーパスが必要であり、この準備にかかる時間とコストは開発の大半を占める。医療などのように、専門用語を多く含む場合には、既存の自動単語分割システムの精度も低く、さらに多くの人にとって単語を正しく認定し読みを振ることが困難であり、このことが開発コストをさらに押し上げ、短時間での開発を阻害している。

前述の問題を解決するために、本論文では、単語境界情報を含まないコーパスから確率的言語モデルを構築する方法を提案する。提案手法では、まず、各文字間に単語境界が存在する確率を推定する。次に、これを用いて生コーパスを確率的に単語に分割されたコーパスとみなし、単語 n-gram 確率を推定する。実験では、比較的小さい単語分割済みコーパスと巨大な生コーパスが利用可能である状況を前提として、これらから複数の言語モデルを構築し、仮名漢字変換を題材としてその効果を検証する。未知語の変換候補を挙げるために必要な表記と読みを記述するモデルとして、文字単位の未知語モデルを提案する。この未知語モデルと生コーパスから推定した単語 n-gram 確率により、文脈情報を含む語彙が、比較的小さい単語分割済みコーパスに出現する単語から巨大な生コーパスに出現する単語に大きく拡大するする。実験の結果、提案手法により変換誤りの約 15%が削減された。また、同じ未知語モデルが読み推定システムにも応用可能であり、精度が向上することを示す。

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。形態素解析器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品詞と表記の組の列を計算する。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。

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

単語 n-gram モデルの一般化の一つとして、クラス n-gram モデルが提案されている [4]。このモデルでは、単語はクラスと呼ばれる単語のグループに属しており、まずクラスが予測され、ついで単語が予測される。

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

$$P_c(w_i|\boldsymbol{c}_{i-n+1}^{i-1}) = \begin{cases} P(c_i|\boldsymbol{c}_{i-n+1}^{i-1})P(w_i|c_i) & \text{if } w_i \in \mathcal{W} \\ P(\mathtt{UW}|\boldsymbol{c}_{i-n+1}^{i-1})M_{x,n}(w_i) & \text{if } w_i \notin \mathcal{W} \end{cases}$$

この式中の c_i は i 番目の単語が属するクラスである。この式では、各単語は唯一のクラスに属すると仮定されている。単語とクラスの関係の推定には、単語クラスタリングが用いられる [4, 5]。

2.2 自動単語分割

日本語においては、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 [7] 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 $(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_k)$ を確率 $P(\boldsymbol{x}|\boldsymbol{y})$ の降順に提示する。

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

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

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他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。

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

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、上述のクラス 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り、日本語文を単語に分割し、それらの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 [7] 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、クラス 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による候補は、以下の値の順に列挙されることになる。

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

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

[1] 文献 [7] 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。

3　未知語の仮名漢字モデル

未知語の問題は、自然言語処理において不可避である。第 2 節で説明した確率的言語モデルによる仮名漢字変換では、片仮名のまま出力されるが、変換候補が挙げられることが望ましい。テキスト音声合成 [8] においても未知語の読みが推定されることが望ましい。テキスト音声合成の言語処理部分 (フロントエンド) は、アクセント推定 [9] のために、各単語の品詞情報が読みとあわせて必要である。これらを同時に推定するために、各単語に読みと品詞が付与されたコーパスから推定された確率的言語モデルによる形態素解析が提案されている [3]。しかしながら、この手法においても、未知語の読み推定は未解決となっている。この節では、未知語の仮名漢字変換や読み推定を可能とする未知語の仮名漢字モデルを提案する。

3.1　文字単位の未知語モデル

漢字をはじめとする日本語の文字の多くは複数の読みを持っているが、文の中で使われている場合には、読みの曖昧性はほとんどない。ここでも問題は、実際の利用における文字と読みの対応関係をモデル化することである。この関係の記述に、読みの文字列と文字の組 $z = \langle \boldsymbol{y}, x \rangle$ をアルファベットとする n-gram モデルを利用することを提案する。このモデルによれば、k 文字からなる単語 $w = \boldsymbol{x}_1^k$ とその読み $\boldsymbol{y}$ の同時生成確率は以下のように計算される。

$$
\begin{aligned}
P(\boldsymbol{y}, w) &= \sum_{\boldsymbol{y}=\boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_k} M_{z,n}(\boldsymbol{y}, w) \qquad (6)\\
M_{z,n}(\boldsymbol{y}, w) &= \prod_{i=1}^{k} P(z_i | \boldsymbol{z}_{i-n+1}^{i-1})
\end{aligned}
$$

この式中で $z_i = \langle \boldsymbol{y}_i, x_i \rangle$ であり、z_i $(i \leq 0)$ は語頭を表す特別な記号であり、z_{k+1} は、語末を表す特別な記号である。

3.1.1　自動読み振りへの応用

未知語の読み推定システムは、未知語の表記 w を与えられると、以下の式を用いて読みを推定する。

$$
\begin{aligned}
\hat{\boldsymbol{y}} &= \operatorname*{\mathbf{argmax}}_{\boldsymbol{y}} P(\boldsymbol{y}|w)\\
&= \operatorname*{\mathbf{argmax}}_{\boldsymbol{y}} \frac{P(\boldsymbol{y}, w)}{P(w)}\\
&= \operatorname*{\mathbf{argmax}}_{\boldsymbol{y}} P(\boldsymbol{y}, w)\\
&\qquad (\because P(w) \text{ は } \boldsymbol{y} \text{ と独立})
\end{aligned}
$$

この式中の $P(\boldsymbol{y}, w)$ は本来式 (6) によって計算されるべきであるが、計算を簡単にするために以下の近似を行なう。

$$
P(\boldsymbol{y}, w) \approx \max_{\boldsymbol{y}=\boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_k} M_{z,n}(\boldsymbol{y}, w) \qquad (7)
$$

3.1.2　仮名漢字変換への応用

確率的言語モデルによる仮名漢字変換において未知語の変換候補を列挙するためには、未知語に対しても式 (4) の表記を条件とする入力記号列の確率が定義される必要がある。このために、以下のように式 (6) の同時確率 $P(\boldsymbol{y}, w)$ を式 (1) の未知語の表記の確率モデル $M_{x,n}(\boldsymbol{x})$ で除する。

$$
P(\boldsymbol{y}_i | w_i) = \frac{P(\boldsymbol{y}_i, w_i)}{M_{x,n}(w_i)} \qquad (8)
$$

式 (2) の言語モデルの確率と式 (4) の既知語に対する仮名漢字モデルと式 (8) の未知語に対する仮名漢字モデルを組み合わせることで、仮名漢字変換の未知語の可能性も含む候補群を順序づける評価関数は以下のように定められる。

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

自動読み振りと同様に、$P(\boldsymbol{y}_i, w_i)$ の計算には、式 (6) を用いる代わりに式 (7) の近似を採用した。

4　生コーパスからの言語モデルの推定

この節では、生コーパスから単語 n-gram 確率を推定するための数学的に適正な方法を提案する。この方法は、生コーパスの文の各文字間に単語境界が存在する確率に基づいており、単語に分割されているコーパスを利用する場合は、これが 1 または 0 に決定されている特殊な場合として包含される。

4.1　単語境界確率

比較的小さい正確に単語に分割されたコーパス C_s が利用可能であることを前提として、まず文字 x_i と x_{i+1} の間に単語境界が存在する確率 P_i を推定する。正確に単語に分割するコストを考慮すると、コーパス C_s のサイズは、全ての 2 文字の組み合わせに対する単語境界確率をある程度の精度で推定できるほど大きいことは期待できない。よって、全ての文字を文字種で分類し、以下のように文字種ごとに単語境界確率をコーパス C_s から最尤推定する。

$$
P_i = \frac{f_s(c(x_i), \mathtt{BT}, c(x_{i+1}))}{f_s(c(x_i), \mathtt{BT}, c(x_{i+1})) + f_s(c(x_i), c(x_{i+1}))}
$$

この式中の $c(x)$ は文字 x の属する文字種を表し、BT は単語分割済みコーパスの単語境界記号を表し、$f_s(\boldsymbol{x})$ はコーパス C_s における文字列 $\boldsymbol{x}$ の頻度を表す。

文字種としては、漢字・記号・平仮名・片仮名・アラビア数字・西洋文字の 6 分類を採用した。西洋文字には、ラテン文字とギリシャ文字とキリル文字が含まる。文末記号と文頭記号にも一つのクラスを割り当てた。

上述のように、単語境界確率を単に前後の文字からのみ推定するのではなく、決定木 [10] や PPM [11] などを用いてより広範囲を参照することも可能である。

4.2 確率的単語分割済みコーパス

単語境界確率を用いると、生コーパス C_r(以下、文字列 $\boldsymbol{x}_1^{n_r}$ として参照) は、各文字境界 (x_i と x_{i+1} の間) が P_i の確率で分割されているコーパスとみなすことができる。この前提のもと、全単語の頻度の和 (単語 0-gram 頻度) は以下のように計算される。

定理 1 生コーパス C_r の中の期待単語数は以下のようになる。

$$1+\sum_{i=1}^{n_r-1} P_i$$

証明 1 X_i を以下で定義される確率変数とする。

$$X_i=\begin{cases}1 & x_i \text{ と } x_{i+1} \text{ が異なる単語に属する場合}\\ 0 & x_i \text{ と } x_{i+1} \text{ が同一の単語に属する場合}\end{cases}$$

すると、X_i の期待値は $E(X_i)$ は P_i に等しい。ゆえに、期待値の加法性[2] から、期待単語数は以下のようになる。

$$1+E(\sum_{i=1}^{n_r-1} X_i)=1+\sum_{i=1}^{n_r-1} E(X_i)=1+\sum_{i=1}^{n_r-1} P_i \quad \blacksquare$$

4.3 単語 1-gram 確率

生コーパス中の文字列 $\boldsymbol{x}_{i+1}^{i+k}$ が単語 $w=\boldsymbol{x}_{i+1}^{i+k}$ である必要十分条件は以下の 3 つである。

1. 文字 x_{i+1} の直前に単語境界がある ($X_i=1$)。
2. 単語境界が文字列中にない ($X_{i+j}=0,\ \ i+1\le\forall j\le i+k-1$)。
3. 文字 x_{i+k} の直後に単語境界がある ($X_{i+k}=1$)。

したがって、単語 w の生コーパス中の確率的頻度 f_r は、単語 w の表記の全ての出現 $O_1=\{i\,|\,\boldsymbol{x}_{i+1}^{i+k}=w\}$ に対する確率的頻度の和として、以下のように定義される。

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

証明 1 と同様に、f_r が生コーパス C_r における w の期待頻度であることが示せる。したがって、単語 1-gram 確率は以下のようになる。

$$P_r(w)=f_r(w)/f_r(\cdot)$$

ここで

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

である。以下の定理により P_r が確率として正しく定義されていることが示される[3] 。

定理 2

$$\sum_{w\in\mathcal{S}_r} f_r(w)=f_r(\cdot)$$

ここで $\mathcal{S}_r$ は C_r の全ての部分文字列の集合である。

証明 2

$$\begin{aligned}\sum_{w\in\mathcal{S}_r} f_r(w) &= \sum_{w\in\mathcal{S}_r}\sum_{\boldsymbol{x}_i^k=w} P_i\left[\prod_{j=i+1}^{k-1}(1-P_j)\right]P_k\\ &= \sum_{0\le i<k\le n_r} P_i\left[\prod_{j=i+1}^{k-1}(1-P_j)\right]P_k\\ &= \sum_{i=0}^{n_r-1} P_i\sum_{k=i+1}^{n_r}\left[\prod_{j=i+1}^{k-1}(1-P_j)\right]P_k\\ &= \sum_{i=0}^{n_r-1} P_i \qquad (\text{図 1 参照})\\ &= 1+\sum_{i=1}^{n_r-1} P_i=f_r(\cdot)\quad(\because P_0=1)\ \blacksquare\end{aligned}$$

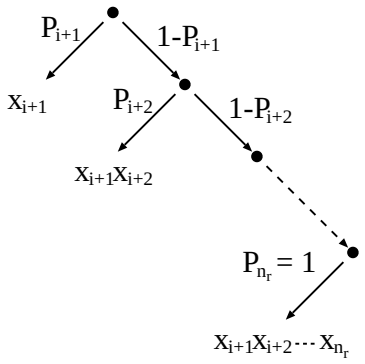
4.4 単語 n-gram 確率

単語 1-gram と同様に、生コーパスにおける単語 n-gram 確率を定義することができる。式が表面的には複雑になるため、以下では 2-gram の場合のみを示す。

$$P_r(w_2|w_1)=\frac{f_r(w_1w_2)}{f_r(w_1)}$$

[2] 確率変数 X と Y に対して $E(X)+E(Y)=E(X+Y)$

[3] 確率論 [12] において S_r を標本空間とし、その全ての部分集合を事象とする。



文字 x_{i+1} の直前に単語境界があるとの仮定のもと、葉に対応する文字列が単語である確率は根から葉までのエッジの確率の積となる。

図 1: 証明 2 の概念図

ここで

$$f_r(w_1w_2) = \sum_{i\in O_2}\left(P_i\left[\prod_{j=1}^{k-1}(1-P_{i+j})\right]P_{i+k} \times \left[\prod_{j=1}^{l-1}(1-P_{i+k+j})\right]P_{i+k+l}\right)$$

$$O_2 = \{i \mid \boldsymbol{x}_{i+1}^{i+k} = w_1 \wedge \boldsymbol{x}_{i+k+1}^{i+k+l} = w_2\}$$

である。P_r が確率として正しく定義されていることは、単語 1-gram 確率の場合と同様の論法により示される。

4.5 仮名漢字変換への応用

生コーパスから推定された単語 n-gram 確率は、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生コーパス C_r から提案手法で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 [13] によって求める。

実験では、単語分割済みコーパスから推定したクラス 2-gram モデルを生コーパスから推定した単語 1-gram モデルと補間した。したがって、生コーパスを利用する仮名漢字変換器は、以下の評価関数の値の順に候補を列挙する。

$$P(\boldsymbol{y}_i|w_i)P(w_i) \tag{9}$$

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

この式から分かるように、提案する仮名漢字変換器は、単語分割済みコーパスに出現しない単語であっても、頻度情報を参照して候補を列挙することができる。さらに、生コーパスの単語 n-gram 確率を利用した場合、未知語の文脈情報も利用することが可能となる。

4.6 既存手法

生コーパスと単語の一覧 $\mathcal{D}$ が所与の場合に、単語の 1-gram 情報を得る方法としては、以下の方法が提案されている。

文字列頻度 [14]

この手法では、単語 1-gram 頻度は文字列としての頻度 $f'(w) = |O_1|$ として計算され、単語 0-gram 頻度はその和 $f'(\cdot) = \sum_{w\in\mathcal{D}} f'(w)$ とする。

最長一致文字列頻度

文字列頻度では、計数が重複することにより頻度が実際よりも高く推定されるという問題がある。永田 [15] はこの問題を軽減するために、頻度推定の対象の単語 w を部分文字列としてもつ単語の推定頻度を以下のように割り引くことを提案している。

$$f''(w) = f'(w) - \sum_{w'\in\mathcal{D}\wedge w\in w'} f'(w')$$

一般的に、これらの推定頻度は、Viterbi 再推定の初期値として利用される [16] 。しかしながら、この再推定の計算量は大きいので、膨大な生コーパスを利用する場合や、$n \geq 2$ の単語 n-gram 確率の再推定に利用するのは現実的ではない。

単語 n-gram 頻度は、生コーパスを単語に自動分割した結果から計算することも可能である。辞書が所与の場合には最長一致法を利用することができる。単語分割済みコーパスが利用できる場合には、それを用いて構築された自動単語分割システムによって、生コーパスを単語に分割することで単語 n-gram 頻度が計数可能となり、その結果から確率的言語モデルを推定できる。

本論文で提案する手法の優位性は以下の通りである。

- 生コーパスの全体を読むのは、単語 0-gram 頻度の計

算と文字列頻度の効率的計算のための suffix array[17] の構築の 1 回のみである。

- 任意の $n \geq 1$ に対する単語 n-gram 頻度の推定に必要なのは、生コーパスにおけるその単語 n-gram の文字列の各出現位置における直前の文字と直後の文字を参照することのみで、生コーパスの追加やモデルの次数の変更に際して膨大な再計算を要しない。

上記の優位性は、提案手法の単語 n-gram 確率の定義が数学的に適切であることによる点に注意されたい。

5 評価

未知語の仮名漢字モデルと生コーパスから推定した言語モデルの評価として、読み推定と仮名漢字変換の実験を行なった。

5.1 実験の条件

実験には EDR コーパス [18] を用いた。このコーパスの各文は単語に分割されており、各単語には読みと品詞が付与されている。コーパスは 10 個に分割され、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。それぞれのコーパスに含まれる文数と形態素数と文字数は表 1 の通りである。実験に用いたい生コーパスは、毎日新聞の 1997 年の記事 (54,415,092 文字) である。

5.2 読み推定

我々が用いた評価基準は、片仮名表記された読みの推定結果と正解との最長共通部分列 (LCS; longest common subsequence)[19] の文字数に基づく再現率と適合率である。EDR コーパスの読みに含まれる文字数を N_{EDR} とし、読み推定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_{EDR} と定義され、適合率は N_{LCS}/N_{SYS} と定義される。

既存のモデルは表記と品詞と読みの直積を単位とする 2-gram モデルである。このモデルと、このモデルに文字単位の未知語モデルを付与した場合とを比較した。表 2 はこの結果である。この結果から、読み推定のための未知語モデルにより、誤りが約半数に減少していることが分かる。学習コーパスには 126,937 の異なり語 (正確には、表記と品詞と読みの直積) が含まれており、テストコーパスに対するカバー率は 98.58%であった。結果を精査したところ、既知語に関しては、読みが文脈依存する少数の単語に誤りが散見されるものの、ほとんど正しく読まれていた。約 80%の未知語は正しく読みが推定されており、誤りの大半は固有名詞などの特殊な読みを持つ低頻度語であった。これらを事前に登録することで、未知語モデルを含む読み推定システムは、テキスト音声合成の言語処理部分として十分実用に耐えると考えられる。

表 1: 単語分割済みコーパス

用途	文数	形態素数	文字数
学習	187,022	4,595,786	7,252,558
評価	20,780	509,261	802,576

表 2: 読み推定の精度

言語モデル	適合率	再現率
3 項組の 2-gram モデル *	98.32%	97.48%
+ 未知語モデル	99.16%	99.17%

* 表記と品詞と読みからなる。

5.3 仮名漢字変換

仮名漢字変換の評価基準は、各文を一括変換することで得られる最尤解と正解との比較により得られる再現率と適合率である。これらの定義は、上記の読み推定の場合と同様である。

以下のモデルに基づく仮名漢字変換器を構築し、変換精度を比較した。

1. 単語分割済みコーパスから構築したクラス 2-gram モデル
2. 単語分割済みコーパスから構築したクラス 2-gram モデルと生コーパスの自動単語分割の結果から構築した単語 1-gram モデルを補間したモデル
3. 単語分割済みコーパスから構築したクラス 2-gram モデルと本論文で提案する方法により生コーパスから推定した単語 1-gram モデルを補間したモデル

各モデルの変換精度を表 3 に掲げる。(1) と (2) の精度の比較から、従来の知見通りに、誤りを含む自動解析結果として生コーパスを利用することが言語モデルの改善に寄与することが分かる。(2) と (3) 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適切な単語が選択されるこ

表 3: 仮名漢字変換の精度

model	precision	recall
(1) クラス 2-gram モデル (baseline)	94.27%	95.22%
(2) (1) と自動分割結果に対する単語 1-gram モデル	95.19%	95.62%
(3) (1) と確率分割に対する単語 1-gram モデルと文字単位の未知語モデル	96.24%	96.04%

とである。例えば、入力記号列「タカハシコレキヨ」は、(3) においては、「高橋是清/タカハシコレキヨ」が未知語であるにもかかわらず正しく変換された。これは、生コーパスに文字列「高橋是清」が出現していることと、入力記号列「タカハシコレキヨ」が一定の確率で「高橋是清」に対応することが未知語モデルにより推定されるからである。(1) と (2) では「高橋 これ 寄与」と高頻度の既知語の列として誤変換された。学習コーパスにおける「是/コレ」や「清/キヨ」の頻度は低く、「是/コレ」と「清/キヨ」が連続する例 (2-gram) が出現しないことによる。

これらの結果は、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の精度向上や新たな分野への適用における提案手法の有効性を示す。

6 おわりに

本論文では、表記と読みの確率的な関係を記述する文字単位の未知語モデルについて述べた。これを読み推定システムに応用することにより、約半数の誤りが解決されることを示した。さらに、生コーパスから単語単位の言語モデルを構築する方法を提案した。実験では、仮名漢字変換を応用例として、変換精度が向上することを確認した。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] 永田昌明. 統計的言語モデルと n-best 探索を用いた日本語形態素解析法. 情処論, Vol. 40, No. 9, pp. 3420–3431, 1999.

[4] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. Class-based *n*-gram models of natural language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, 1992.

[5] 森信介, 長尾眞. 形態素クラスタリングによる形態素解析精度の向上. 自然言語処理, Vol. 5, No. 2, pp. 75–103, 1998.

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

[7] 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.

[8] R. E. Donovan. Topics in decision tree based speech synthesis. *Computer Speech and Language*, Vol. 17, pp. 43–67, 2003.

[9] 匂坂芳典, 佐藤, 大和. 日本語単語連鎖のアクセント規則. 信学論, Vol. J66-D, No. 7, pp. 849–856, 1983.

[10] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall, Inc., 1984.

[11] W. J. Teahan and John G. Cleary. The entropy of English using PPM-based models. In *DCC*, 1996.

[12] David Williams. *Probability With Martingales*. Cambridge Mathematical Textbooks, 1991.

[13] 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.

[14] Richard Sproat and Chilin Shih William Gale Nancy Chang. A stochastic finite-state word-segmentation algorithm for chinese. *Computational Linguistics*, Vol. 22, No. 3, pp. 377–404, 1996.

[15] Masaaki Nagata. A self-organizing Japanese word segmenter using heuristic word identification and reestimation. In *Proc. of the WVLC97*, 1997.

[16] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov process. *Inequalities*, Vol. 3, pp. 1–8, 1972.

[17] Udi Manber and Gene Myers. Suffix arrays: A new method for on-line string searches. *SIAM J. Comput*, Vol. 22, No. 5, pp. 935–948, 1993.

[18] 日本電子化辞書研究所. EDR 電子化辞書仕様説明書, 1993.

[19] 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.