確率的タグ付与コーパスからの言語モデル構築

森 信介[†] **・笹田 鉄郎**[††] **・NEUBIG Graham**[††]

確率的言語モデルは、仮名漢字変換や音声認識などに広く用いられている。パラメータは、コーパスの既存のツールによる処理結果から推定される。精度の高い読み推定ツールは存在しないため、結果として、言語モデルの単位を単語 (と品詞の組) とし、仮名漢字モデルを比較的小さい読み付与済みコーパスから推定したり、単語の発音の確率を推定せずに一定値としている。これは、単語の読みの確率を文脈と独立であると仮定していることになり、この仮定に起因する精度低下がある。
このような問題を解決するために、本論文では、まず、仮名漢字変換において、単語と読みの組を単位とする言語モデルを利用することを提案する。単語と読みの組を単位とする言語モデルのパラメータは、自動単語分割および自動読み推定の結果から推定される。この処理過程で発生する誤りの問題を回避するために、本論文では、確率的タグ付与を提案する。これらの提案を採用するか否かに応じて複数の仮名漢字変換器を構築し、テストコーパスにおける変換精度を比較した結果、単語と読みの組を言語モデルの単位とし、そのパラメータを確率的に単語分割し、さらに確率的読みを付与したコーパスから推定することで最も高い変換精度となることが分かった。したがって、本論文で提案する単語と読みの組を単位とする言語モデルと、確率的タグ付与コーパスの概念は有用であると結論できる。

キーワード：確率的言語モデル, 仮名漢字変換, 確率的単語分割, 確率的タグ付与

Language Model Estimation from a Stochastically Tagged Corpus

Shinsuke MORI[†], Tetsuro SASADA[††] and Graham NEUBIG[††]

In this paper, first we propose a language model based on pairs of word and input sequence. Then we propose the notion of a stochastically tagged corpus to cope with tag estimation errors. The experimental results we conducted using *kana-kanji* converters showed that our ideas, the language model based on pairs of word and input sequence and the notion of a stochastically tagged corpus, both improved the accuracy. Therefore we conclude that the language model based on pairs and the notion of a stochastically tagged corpus are effective in language modeling for the *kana-kanji* conversion task.

Key Words: stochastic language model, Kana-kanji conversion, stochastic segmentation, stochastic tagging

[†]京都大学 学術情報メディアセンター, Kyoto University, Academic Center for Computing and Media Studies
[††]京都大学 情報学研究科, Kyoto University, School of Informatics

1　はじめに

確率的言語モデルは、統計的手法による仮名漢字変換 (森，土屋，山地，長尾 1999)(Google 2010) (村上 1991) や音声認識 (鹿野，伊藤，河原，武田，山本 2001)(Jelinek 1985) などに広く用いられている。確率的言語モデルは、ある単語列がある言語でどの程度自然であるかを出現確率としてモデル化する[1]。仮名漢字変換においては、確率的言語モデルに加えて、仮名漢字モデルが用いられる。仮名漢字モデルは、入力記号列と単語の対応を記述する。音声認識では、仮名漢字モデルの代わりに、発音と単語の対応を記述する発音辞書と音響モデルが用いられる。

確率的言語モデルの推定のためには、システムを適応する分野の大量のテキストが必要で、その文は単語に分割されている必要がある。このため、日本語を対象とする場合には、自動単語分割や形態素解析が必要であるが、ある程度汎用性のあるツールが公開されており、辞書の追加などで一般的な分野の言語モデルが構築可能となっている。

仮名漢字モデルや発音辞書における確率の推定には、実際の使用における単語の読みの頻度を計数する必要がある。しかしながら、読み推定をある程度の汎用性と精度で行うツールは存在しない[2]。したがって、仮名漢字モデルを比較的小さい読み付与済みコーパスから推定したり (森他 1999)、後処理によって、一部の高頻度語にのみ文脈に応じた発音を付与し、他の単語に関しては、各発音の確率を推定せずに一定値としている (鹿野他 2001) のが現状である。

一方で、単語 (表記) を言語モデルの単位とすることには弊害がある。例えば、「…するや、…した」という発声が、「…する夜、…した」と書き起こされることがある。この書き起こし結果の「夜」は、この文脈では必ず「よる」と発音されるので、「夜」と書き起こすのは不適切である。この問題は、単語を言語モデルの単位とする仮名漢字変換においても同様に起こる。これは、単語の読みの確率を文脈と独立であると仮定して推定 (あるいは一定値に固定) していることに起因する。

このような問題を解決するために、本論文では、まず、すべての単語を読みで細分化し、単語と読みの組を単位とする言語モデルを利用することを提案する。仮名漢字変換や音声認識において、単語と品詞の組を言語モデルの単位とすることや、一部の高頻度語を読みで細分化することが行われている (森他 1999)(鹿野他 2001)。提案手法は、品詞ではなく読みですべての単語を細分化することとみなすこともできるので、提案手法は既存手法から容易に類推可能であろう。しかしながら、提案手法を実現するためには、文脈に応じた正確な読みを様々な分野のテキストに対してある程度の精度で推定できる必要がある。このため、提案手法を実現したという報告はない。

単語を単位とする言語モデルのパラメータは、自動単語分割の結果から推定される。自動単語分割の精度は十分高いとはいえ、一定の割合の誤りは避けられない。この問題による悪影響を避ける

[1] 単語の定義に関しては様々な立場がある。本論文では、英語などの音声認識の言語モデル (Jelinek 1985) と同様に、ある言語においてなんらかの方法で認定される文字列と定義する。

[2] 音声認識では発音が必要で、仮名漢字変換では入力記号列が必要である。これらは微妙に異なる。本論文では、この違いを明確にせず両方を意味する場合に「読み」という用語を用いる。

ために、確率的単語分割 (森，宅間，倉田 2007) という考えが提案されている。この方法では、各文字の間に単語境界が存在する確率を付与し、その確率を参照して計算される単語 n-gram の期待頻度を用いて言語モデルを構築する。計算コストの削減のために、実際には、各文字間に対してその都度発生させた乱数と単語境界確率の比較結果から単語境界か否かを決定することで得られる擬似確率的単語分割コーパスから従来法と同様に言語モデルが構築される (森，小田 2009) 。

単語と読みの組を単位とする言語モデルのパラメータは、自動単語分割および自動読み推定の結果から推定される。自動単語分割と同様に、自動読み推定の精度は十分高いとしても、一定の割合の誤りは避けられず、言語モデルのパラメータ推定に悪影響がある。これを回避するために、確率的タグ付与とその近似である擬似確率的タグ付与を提案する。

実験では、タグとして入力記号列を採用し、単語と入力記号列の組を単位とする言語モデルを用いる仮名漢字変換器を構築し、単語を単位とする言語モデルを用いる場合や、決定的な単語分割や入力記号付与などの既存手法に対する提案手法の優位性を示す。

2 統計的仮名漢字変換

統計的手法による仮名漢字変換 (森他 1999) は、キーボードから直接入力可能な入力記号 $\mathcal{Y}$ の正閉包 $\boldsymbol{y} \in \mathcal{Y}^+$ を入力として、日本語の文字 $\mathcal{X}$ の正閉包である変換候補 ($\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots$) を確率値 $P(\boldsymbol{x}|\boldsymbol{y})$ の降順に提示する[3]。文献 (森他 1999) では文を単語列 $\boldsymbol{w} = w_1 w_2 \cdots w_h$ とみなし、これを単語 $w \in \mathcal{X}^+$ を単位とする言語モデルと仮名漢字モデルに分解して実現する方法を提案している。本節では、まずこれについて説明し、次に単語と読みを組とする言語モデルによる方法を提案し定式化する。

2.1 従来手法

文献 (森他 1999) では、変換候補を $P(\boldsymbol{w}|\boldsymbol{y})$ で順序付けすることを提案しており、これを次の式が示すように、単語を単位とする言語モデルと仮名漢字モデルに分解する[4]。

$$P(\boldsymbol{w}|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|\boldsymbol{w})P(\boldsymbol{w})}{P(\boldsymbol{y})} \tag{1}$$

ここで、後述するパラメータ推定のために、単語と入力記号列との対応関係は各単語において独立であるとの仮定をおく。さらに、分母 $P(\boldsymbol{y})$ は出力に依らないので、分子だけを以下のようにモデ

[3] 一般的な仮名漢字変換フロントエンドと同様に、ローマ字から (主に) 平仮名への変換が行われると仮定している。したがって、入力記号は $\mathcal{Y} =$ { A, B, ⋯, Z, 0, 1, ⋯, 9, ぁ, あ, ⋯, ん, ヴ, ヵ, ヶ, ー, =, ¥, ｀, 「, 」, ;, ’, 、, 。, !, @, #, $, %, ^, &, ∗, (,), __, +, |, ～, {, }, :, ”, <, >, ?, ・} である。

[4] 正確には、単語と品詞の組を単位とする言語モデルを提案している。

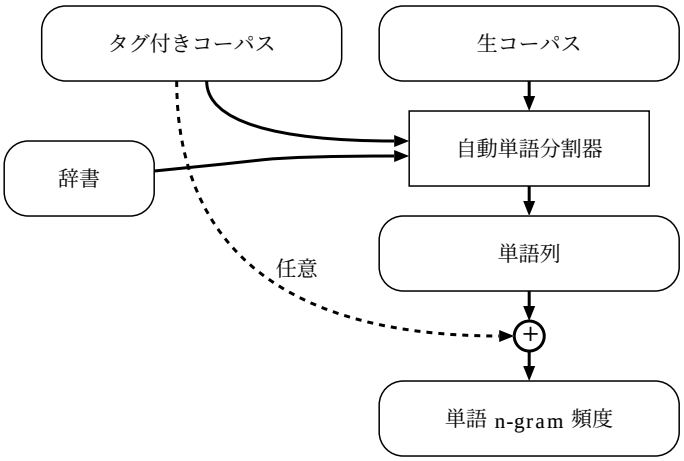

図 1　単語を単位とする言語モデルの作成の手順

ル化する。

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

$$P(\boldsymbol{y}_i|w_i)P(w_i|\boldsymbol{w}_{i-n+1}^{i-1}) \quad = \quad \begin{cases} P(w_i|\boldsymbol{w}_{i-n+1}^{i-1})P(\boldsymbol{y}_i|w_i) & \text{if} \quad w_i \in \mathcal{W} \\ P(\texttt{UW}|\boldsymbol{w}_{i-n+1}^{i-1})M_{y,n}(\boldsymbol{y}_i) & \text{if} \quad w_i \notin \mathcal{W} \end{cases} \tag{2}$$

ここで $\mathcal{W}$ は確率的言語モデルの語彙を表す。簡単のために、この式の中の w_i $(i \leq 0)$ は、文頭に対応する特別な記号 BT であり、これは文末 w_{h+1} も表す。この式の $P(w_i|\boldsymbol{w}_{i-n+1}^{i-1})$ と $P(\texttt{UW}|\boldsymbol{w}_{i-n+1}^{i-1})$ は、語彙に BT と未知語記号 UW を加えた $\mathcal{W} \cup \{\texttt{BT}, \texttt{UW}\}$ 上の n-gram モデルである。パラメータは、単語に分割されたコーパスから以下の式を用いて最尤推定する。

$$P(w_i|\boldsymbol{w}_{i-n+1}^{i-1}) = \frac{f(\boldsymbol{w}_{i-n+1}^{i})}{f(\boldsymbol{w}_{i-n+1}^{i-1})} \tag{3}$$

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。図 1 が示すように、この学習コーパスには自動単語分割の結果であることが多いが、自動単語分割器の学習に用いたタグ付きコーパスが利用可能な場合にはこれを加えることもある。

式 (2) の $P(\boldsymbol{y}_i|w_i)$ は、単語単位の仮名漢字モデルであり、パラメータは、単語に分割されかつ各単語に入力記号列が付与されたコーパスから以下の式を用いて最尤推定する。

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

式 (2) から分かるように、単語単位の仮名漢字モデルでは、単語と入力記号列との対応関係が各単

語において独立であると仮定している。この仮定により、比較的少量の入力記号列付与済みコーパスからある程度信頼できるパラメータを推定することができる。

式 (2) の $M_{y,n}(\boldsymbol{y}_i)$ は、未知語モデルであり、入力記号の集合に単語の両端を表す記号を加えた $\mathcal{Y} \cup \{\texttt{BT}\}$ 上の n-gram モデルで実現される[5]。このパラメータは低頻度の単語に対応する入力記号列から推定する。

2.2 提案手法

本論文では、言語モデルの単位を単語と入力記号列の組 $u = \langle w, \boldsymbol{y} \rangle$ とすることを提案する。その上で、以下の式のように $P(\boldsymbol{w}|\boldsymbol{y})$ をモデル化する。

$$P(\boldsymbol{w}|\boldsymbol{y}) = \frac{P(\boldsymbol{w}, \boldsymbol{y})}{P(\boldsymbol{y})} = \frac{P(\boldsymbol{u})}{P(\boldsymbol{y})}$$

分母 $P(\boldsymbol{y})$ は出力に依らないので、分子だけを以下のようにモデル化する。

$$P(\boldsymbol{u}) = \prod_{i=1}^{h} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1})$$

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \begin{cases} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) & \text{if } u_i \in \mathcal{U} \\ P(\texttt{UU}|\boldsymbol{u}_{i-n+1}^{i-1}) M_{u,n}(u_i) & \text{if } u_i \notin \mathcal{U} \end{cases} \tag{5}$$

ここで $\mathcal{U}$ は言語モデルの語彙 (単語と入力記号列の組の集合) を表す。この式の中の u_i $(i \leq 0)$ と u_{h+1} は、単語を単位とする場合と同様に、文頭と文末に対応する記号 BT である。また UU は未知の組を表す記号である。

式 (5) の $M_{u,n}(u) = M_{u,n}(\langle w, \boldsymbol{y} \rangle)$ は未知語モデルである。従来手法と同様に、大きな学習コーパスを用いれば実際の使用における未知語率は極めて低く、また未知語に対する正確な仮名漢字変換は困難であると考えて、アルファベット $\mathcal{U}$ 上の未知語モデルの代わりにアルファベット $\mathcal{Y}$ 上の未知語モデル $M_{y,n}(\boldsymbol{y})$ を用いることとする。これは、式 (2) と共通である。以上から、提案手法の仮名漢字変換は、以下の式のようになる。

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \begin{cases} P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) & \text{if } u_i \in \mathcal{U} \\ P(\texttt{UU}|\boldsymbol{u}_{i-n+1}^{i-1}) M_{y,n}(\boldsymbol{y}_i) & \text{if } u_i \notin \mathcal{U} \end{cases} \tag{6}$$

ここで $\boldsymbol{y}_i = y(u_i)$ は $u_i = \langle w_i, \boldsymbol{y}_i \rangle$ の入力記号列である。なお、$M_{u,n}(u)$ の代わりに $M_{y,n}(\boldsymbol{y})$ を用

[5] 文献 (森他 1999) によれば、あるテストコーパスにおいて未知語を構成する文字の 33.0%が片仮名であった。入力記号集合は主に平仮名からなるが、この先行研究と同様に、出力においてはこれらを片仮名とする。

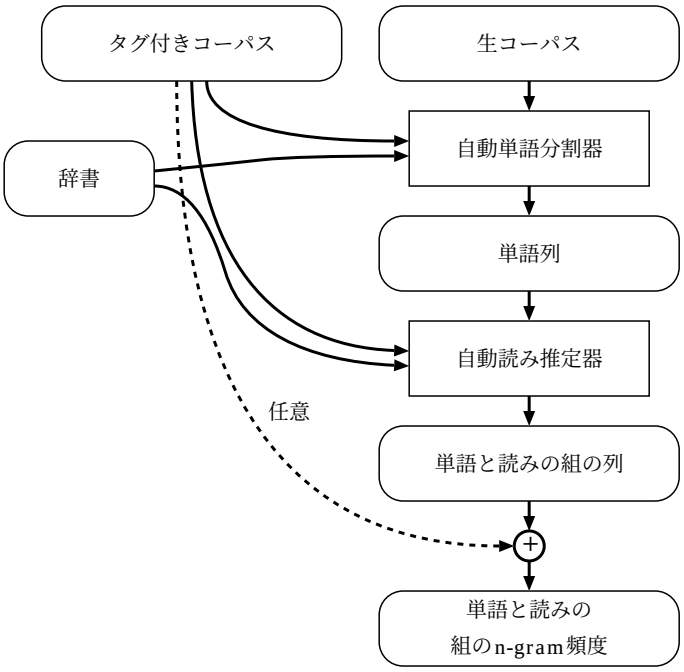

図 2　単語と読みの組を単位とする言語モデルの作成の手順

いることは以下の式で与えられる近似であり、$\mathcal{Y} \subsetneq \mathcal{X}$ であるので、入力記号列のみからなる文字列を未知語として出力することになる。

$$M_{u,n}(u) = M_{u,n}(\langle w, \boldsymbol{y} \rangle) \approx \begin{cases} M_{y,n}(\boldsymbol{y}) & \text{if } \ w \in \mathcal{Y}^+ \\ 0 & \text{if } \ w \notin \mathcal{Y}^+ \end{cases}$$

この式の $M_{y,n}(\boldsymbol{y})$ のパラメータは、学習コーパスにおける語彙 $\mathcal{U}$ に含まれない表記と入力記号列の組の入力記号列から推定する。これは、学習コーパスにおける未知の組の単語を入力記号列に置き換えた結果から $M_{u,n}(u)$ を推定しているのと同じである。

式 (6) の $P(u_i|\boldsymbol{u}_{i-n+1}^{i-1})$ と $P(\mathtt{UU}|\boldsymbol{u}_{i-n+1}^{i-1})$ は、語彙に BT と UU を加えた $\mathcal{U} \cup \{\mathtt{BT}, \mathtt{UU}\}$ 上の n-gram モデルである。パラメータは、単語に分割されかつ入力記号列が付与されたコーパスから以下の式を用いて最尤推定する。

$$P(u_i|\boldsymbol{u}_{i-n+1}^{i-1}) = \frac{f(\boldsymbol{u}_{i-n+1}^{i})}{f(\boldsymbol{u}_{i-n+1}^{i-1})} \tag{7}$$

図 2 が示すように、この学習コーパスには自動単語分割・読み付与の結果を用いることができる。さらに自動単語分割器や読み付与の学習に用いたタグ付きコーパスが利用可能な場合にはこれを加えることもできる (図 2 の点線)。

単語を単位とする従来手法と同程度の信頼性となるパラメータを推定するために、従来手法においてパラメータ推定に用いられる単語に分割されたコーパスと同程度の量の単語に分割されかつ入力記号列が付与されたコーパスが必要である。換言すれば、自動単語分割と同程度の精度で入力記号列を推定するシステムが必要である。

これまで、各単語に対する入力記号列 (読み) をその文脈に応じて十分な精度で推定する研究やフリーウェアがなかったために、提案手法は現実的ではなかったと思われる[6]。次節では、この方法を説明し、さらに入力記号列を確率的に付与することで、入力記号列の推定誤りの影響を緩和する方法を提案する。

3 仮名漢字変換のための言語資源とその処理

仮名漢字変換や音声認識のための言語モデルは、単語分割済みコーパスと生コーパスの自動単語分割結果から構築される。この節では、まずこの過程を概説する。次に、前節で提案したモデルのパラメータをより正確に推定するために、単語に入力記号列や発音などのタグを確率的に付与することを提案する。

3.1 コーパス

仮名漢字変換や音声認識のための単語を単位とする言語モデル作成においては、これらを適用する分野のコーパスが必須である。一般に、コーパスには単語境界情報がないので、自動単語分割器 (Graham，中田，森 2010) や形態素解析器 (松本，黒橋，宇津呂，妙木，長尾 1993; 丸山，荻野，渡辺 1991; 永田 1999; 森，長尾 1998; 工藤，山本，松本 2004) を用いて文を言語モデルの単位に分割し、その結果に対して単語 n-gram 頻度を計数する (図 1 参照)[7]。なお、これら自動単語分割器や形態素解析器などの自然言語処理システムは、単語分割済みあるいは品詞付与済みのコーパスから学習することが多い。その場合には、これら自然言語処理システムの学習コーパスも言語モデルの学習コーパスに加えることができる (図 1 の点線) が、実際には、これら自然言語処理システムはツールとして配布され、辞書追加程度の適応しかなされず、自然言語処理システムの学習コーパスが言語モデルの学習に利用されることは少ない。

3.2 形態素解析と自動単語分割

形態素解析は、日本語の自然言語処理の第一段階として研究され、ルールに基づく方法が一定の成果を上げた (松本他 1993)。同じ頃、統計的手法 (丸山他 1991; 永田 1999) が提案され、アルゴリ

[6]研究としては文献 (長野，森，西村 2006) があるが、公開されていない。また、読み推定ツールとして KAKASI (`http://kakasi.namazu.org/` , 2010 年 5 月) があるが、様々な分野において十分な精度とはいえない。

[7]形態素解析を利用する場合は、品詞も付与されるので、単語と品詞の組を単位とすることもあるが、仮名漢字変換や音声認識の出力に品詞は不要なので、実質的には品詞の異なる同音異義語の識別程度の効果しかない。さらに、茶筌などを利用すると「読み」も付与されるが、これらは文脈に依存しないので実質的には付与していないのと同じである。

ズムとデータの分離に成功した。統計的手法は、ルールに基づく方法と同等かそれ以上の精度を達成しており、現在では主流になっている。さらに、フリーソフトとして公開され、容易に利用可能となっている。

このような背景から、仮名漢字変換や音声認識のための言語モデル作成のために、形態素解析が用いられている。結果的に、単語 (表記) と品詞の組を言語モデルの単位とすることが多い[8]。しかしながら、仮名統計的漢字変換や音声認識等の実現には品詞情報は不要であり、形態素解析器の学習コーパス作成のコストを不必要に増大させるのみである。また、英語等の単語間に空白を置く言語の音声認識においては、言語モデルの単位として当然単語が用いられる。日本語においても単語を言語モデルの単位とする音声認識の取り組みがあり、十分な認識精度を報告している (西村，伊東，山崎 1999)。以上の考察から、本論文では、単語と品詞の組を言語モデルの単位とする手法は、単語を単位とする手法に含まれるとして、以下の議論を展開する。

言語モデルの構築においては、適応対象の分野の大量のテキストに対する統計をとることが非常に有用である。このため、形態素解析や自動単語分割等の自動処理が必須であるが、自動処理の結果は一定量の誤りを含む。この単語分割誤りによる悪影響を緩和するために、確率的に単語に分割することが提案されている (森他 2007)。この手法では、自動単語分割器によって各文字の間に単語境界がある確率を付与し、その確率を参照して計算される単語 n-gram の期待頻度を用いて言語モデルが構築される。実用上は、モンテカルロシミュレーションのように、各文字間に対して都度発生させた乱数と単語境界確率の比較結果から単語境界か否かを決定することで得られる擬似確率的単語分割コーパスから従来法と同様に言語モデルが構築される (森，小田 2009)。

3.3　自動読み推定

前節で、仮名漢字変換のための言語モデルの単位として単語と入力記号列の組を用いることを提案した。この考え自体は特に新規ではなく、以前から存在している。実際、音声認識において、数詞のあとの「本」など一部の高頻度語に文脈に応じた発音を付与する後処理が行われている (鹿野他 2001)。また、発音レベルでの書き起こしが得られる場合に、単語と発音の対応を推定し、単語と品詞と発音の組を単位をとする言語モデルを構築する研究もある (堤，加藤，小坂，好田 2002)。しかしながら、この考えを一般的な場合において実現するためには、高精度の自動読み推定システムが必要である。前述の形態素解析の研究とその成果であるフリーソフトにおいては、読みの推定は軽視されており、文脈に応じた読みを高い精度で出力する研究やフリーソフトはなかった。このため、単語と入力記号列の組や単語と発音の組を単位とする言語モデルは一般的な意味で実現されていなかった。

前節で提案した単語と入力記号列の組を単位とする言語モデルの構築においては、コーパスを単

[8] 音声認識 (鹿野他 2001) ではすべての可能な読みも付加しているが、文脈に応じた読みは付与されず、同音異義語の峻別には用いられていない。

語に分割し、文脈に応じた読みを付与することができる KyTea(京都テキスト解析ツールキット)(森, Graham 2010) を用いて適応対象の分野のテキストを自動的に単語と入力記号列の組の列に変換する (図 2 参照)。その結果から式 (7) を用いて単語と入力記号列の組の n-gram 確率を推定する。KyTea の詳細は付録 A に記述した。

3.4 確率的タグ付与

自動読み推定の結果は、形態素解析や自動単語分割等の自動処理の場合と同様に、一定量の誤りを含む。学習コーパスに含まれる読み推定誤りは、言語モデルや仮名漢字モデル、あるいは発音辞書に悪影響を及ぼす。特に、ある単語に対して至る所で同じ誤った読みを付与する場合には、非常に重大な問題となる。この問題を回避するために、確率的単語分割と同様に、単語に対する入力記号列付与や発音付与を確率的に行うことを提案する。すなわち、読み推定においては、ある単語に対する読みを決定的に推定するのではなく、可能な読みとその確率値を返すようにする。より一般的には、単語に対する読みや品詞などのタグ付与を、ある基準で最適となる唯一のタグを出力する処理ではなく、タグ t と確率値 p の組の列 $(\langle t_1, p_1\rangle, \langle t_2, p_2\rangle, \cdots)$ を出力する処理へと一般化する。この際、タグの確率値は、周辺の他の単語のタグと独立であるとの仮定をおく。この結果得られるコーパスを確率的タグ付与コーパスと呼ぶ。確率的タグ付与コーパスの文 $w_1 w_2 \cdots w_h$ は、以下のように、各単語に可能なタグと確率値の組の列が付与されている。

$$\begin{gathered}
\langle w_1, (\langle t_{1,1}, p_{1,1}\rangle, \langle t_{1,2}, p_{1,2}\rangle, \cdots, \langle t_{1,k_1}, p_{1,k_1}\rangle)\rangle \\
\langle w_2, (\langle t_{2,1}, p_{2,1}\rangle, \langle t_{2,2}, p_{2,2}\rangle, \cdots, \langle t_{2,k_2}, p_{2,k_2}\rangle)\rangle \\
\vdots \\
\langle w_h, (\langle t_{h,1}, p_{h,1}\rangle, \langle t_{h,2}, p_{h,2}\rangle, \cdots, \langle t_{h,k_h}, p_{h,k_h}\rangle)\rangle
\end{gathered}$$

ここで、$t_{i,j}$ と $p_{i,j}$ はそれぞれ、i 番目の単語の j 番目のタグとその確率を表す。このような確率的タグ付与コーパスにおける単語とタグの組の n-gram の 1 回の出現あたりの頻度 $f_1(\boldsymbol{u})$ は、以下の式で計算される期待頻度として定義される。

$$f_1(\boldsymbol{u}) = f_1(\langle w_1, t_{1,j_1}\rangle\langle w_2, t_{2,j_2}\rangle \cdots \langle w_n, t_{n,j_n}\rangle) = \prod_{i=1}^{n} p_{i,j_i} \tag{8}$$

この値をコーパスにおけるすべての出現箇所に渡って合計した結果が単語とタグの組の列 $\boldsymbol{u}$ の期待頻度である。単語とタグの組の n-gram 確率は、この期待頻度の相対値として定義される。仮名漢字変換のための言語モデル構築では、タグとして単語に対応する入力記号列を用いる。

確率的入力記号列付与のためのモデルは、単語ごとに入力記号列が付与されたコーパスからロジスティック回帰などの点推定器を推定しておくことで実現できる。

3.5　擬似確率的タグ付与

確率的単語分割の場合と同様に、確率的タグ付与コーパスに対する単語とタグの組の列の頻度の計算は、決定的タグ付与コーパスに対する頻度計算と比べてはるかに多い計算を要する。実際、対象となる組の列としての頻度が F 回とすると、式 (8) による期待頻度の計算は、各出現箇所における $(n-1)$ 回の浮動小数点の積を実行し ($F(n-1)$ 回の乗算)、その結果の総和を $(F-1)$ 回の加算により算出することになる。通常の決定的タグ付与コーパスに対する頻度の計算は、F 回のインクリメントで済むことを考えると、非常に大きな計算コストが必要である。また、非常に小さい期待頻度の単語とタグの組の列が多数生成され、これによる計算コストの増大も起こる。このような計算コストの問題は、次に述べる擬似確率的タグ付与コーパスによって近似的に解決される。

擬似確率的タグ付与コーパスは、各単語に対して都度発生させた乱数とタグの確率の比較結果から当該単語のタグを唯一に決定することで得られる単語とタグの組の列である。この手続きを複数回繰り返して得られるコーパスに対して頻度を計数することで確率的タグ付与コーパスの期待頻度の近似値が得られる。このときの繰り返し回数を倍率と呼ぶ。

擬似確率的タグ付与コーパスは、確率的単語分割コーパス (森，小田 2009) と同様に一種のモンテカルロ法となっており、近似誤差に関しては以下の議論が同様に可能である。モンテカルロ法による d 次元の単位立方体 $[0,1]^d$ 上の定積分 $I=\int_{[0,1]^d} f(x)dx$ の数値計算法では、単位立方体 $[0,1]^d$ 上の一様乱数 $x_1,\ x_2,\ \ldots,\ x_N$ を発生させて $I_N=\sum_{i=1}^{N} f(x_i)$ とする。このとき、誤差 $|I_N-I|$ は次元 d によらずに $1/\sqrt{N}$ に比例する程度の速さで減少することが知られている。擬似確率的タグ付与コーパスにおける単語とタグの組の n-gram 頻度の計算はこの特殊な場合である。すなわち、式 (8) の値は、n 次元の単位立方体中の矩形の部分領域 (i 番目の軸方向の長さが p_{i,j_i}) の体積である。したがって、誤差は n の値によらずに $1/\sqrt{FN}$ に比例する程度の速さで減少する。

4　評価

提案手法の評価のために、学習コーパスの作成の方法と言語モデルの単位が異なる仮名漢字変換を構築し、テストコーパスに対する変換精度を測定した。この節では、その結果を提示し提案手法の評価を行う。

4.1　実験条件

実験に用いたコーパスの諸元を表 1 に掲げる。学習コーパスは、L と R の 2 種類である。学習コーパス L は、現代日本語書き言葉均衡コーパス 2009 年モニター版 (Maekawa 2008) と日常会話の辞書の例文と新聞記事からなり、人手による単語分割と入力記号付与がなされている。学習コーパス R は新聞記事からなり、単語境界や入力記号などの付加情報はない。単語境界や入力記号の

表 1 コーパス

	ID	用途	文数	単語数	文字数
タグ付きコーパス	L	学習	42,170	1,162,450	1,690,818
生コーパス	R	学習	1,250,116	—	52,673,251
テストコーパス	T	テスト	1,002	29,038	43,695

正解:	私		が	高	橋	是		清	で	す	($N_{COR}=8$)
出力:	渡	し	が	高	橋	こ	れ	寄 与	で	す	($N_{SYS}=11$)
共通:			が	高	橋				で	す	($N_{LCS}=5$)

再現率 $= N_{LCS}/N_{COR} = 5/8$, 適合率 $= N_{LCS}/N_{SYS} = 5/11$

図 3 評価基準

推定は、京都テキスト解析ツールキット KyTea (Graham 他 2010)[9]によって行った。テストコーパス T は、学習コーパス R と同じ新聞の別の記事であり、変換精度の計算のために入力記号が付与されてある。

4.2 評価基準

仮名漢字変換の評価基準は、各入力文の一括変換結果と正解との最長共通部分列 (LCS; longest common subsequence)(Aho 1990) の文字数に基づく再現率と適合率である (図 3 参照)。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。図 3 の例では、これらは以下のようになる。

再現率: $N_{LCS}/N_{COR} = 5/8$

適合率: $N_{LCS}/N_{SYS} = 5/11$

これらに加えて、文正解率も計算した。これは、変換結果が文全体に渡って一致している文の割合を表す。

4.3 評価

学習コーパスの作成の方法と言語モデルの単位による仮名漢字変換精度の差を調べるために、以下の 3 通りの方法で作成された学習コーパスのそれぞれから、単語を言語モデルの単位とする仮名漢字変換 (式 (2) 参照) と単語と入力記号列の組を言語モデルの単位とする仮名漢字変換 (式 (6) 参

[9] Version 0.1.0, `http://www.phontron.com/kytea/` (2010 年 10 月)

表 2　仮名漢字変換の精度 2-gram

ID	単語分割	入力記号付与	言語モデルの単位	エントロピー [bit]	適合率 [%]	再現率 [%]	文正解率 [%]
DDw	決定的	決定的	単語 (w)	—	97.10	97.23	54.39
DSw	決定的	確率的		—	97.24	97.32	55.69
SSw	確率的	確率的		—	97.34	97.40	57.19
DDu	決定的	決定的	単語と入力記号列の組 (u)	4.842	97.34	97.40	56.59
DSu	決定的	確率的		4.816	97.48	97.47	57.78
SSu	確率的	確率的		4.800	97.51	97.48	58.58

単語分割が「確率的」であるとは、倍率 1 の擬似確率的単語分割および疑似確率的入力記号付与を意味する。エントロピー算出における予測対象は、単語と入力記号列の組であり、単語単位の言語モデル (DDw,DSw,SSw) に対しては同じ条件で算出できない。

照) を作成した。言語モデルはすべて 2-gram モデルである[10]。

DD: 決定的に単語分割し、決定的に入力記号列を付与する。

DS: 決定的に単語分割し、確率的に入力記号列を付与する。

SS: 確率的に単語分割し、確率的に入力記号列を付与する。

ここで、「確率的」は擬似確率的単語分割および疑似確率的入力記号付与を意味し、全て倍率は 1 とした。

文献 (森, 小田 2009) では、1,890,041 文字の生コーパスに対して 1〜256 の倍率による擬似確率的単語分割コーパスを評価している。その結果、倍率が 8〜32 程度で確率的単語分割コーパスと同程度の性能となっている。前後数単語の単語分割の可能性は 16〜32 通り程度 (その出現にも偏りがある) なので高頻度の単語 (候補) の高頻度の文脈はある程度大きいコーパスであれば、倍率が 1 の擬似確率的単語分割コーパスでも十分に真の分布に近い推定値が得られると考えられる。本実験での生コーパスの文字数は、この文献での実験の約 27.9 倍であり、ある程度の頻度の組の列 $\boldsymbol{u}$ の出現頻度 (第 3.5 項の F) は約 27.9 倍となっていることが期待される。したがって、倍率 (3.5 項の N) が 1 であっても、上述の文献における実験での倍率 27.9 に相当し、確率的タグ付与コーパス ($N \to \infty$) に近い性能が期待される。

3 つの学習コーパスの作成の方法と 2 つの言語モデルの単位のすべての組み合わせによる仮名漢字変換の精度を表 2 に示す。表中の ID の最初の 2 文字は学習コーパスの作成の方法を表し、次の 1 文字は言語モデルの単位を表す。文献 (森他 1999) は、単語と品詞の組を言語モデルの単位とし、

[10] 音声認識で一般的な 3-gram モデルを用いなかったのは、仮名漢字変換の先行研究 (森他 1999) とその実用化の例 (Google 2010) が 2-gram モデルを用いていること、仮名漢字変換での入力記号列は比較的短い傾向があり (第 1 著者の場合約 2.2 単語分) 長い履歴が実際にはほとんど有効ではないこと、3-gram モデルは必要となる記憶域が増大し処理速度が低下するなど実用化に向かないことである。

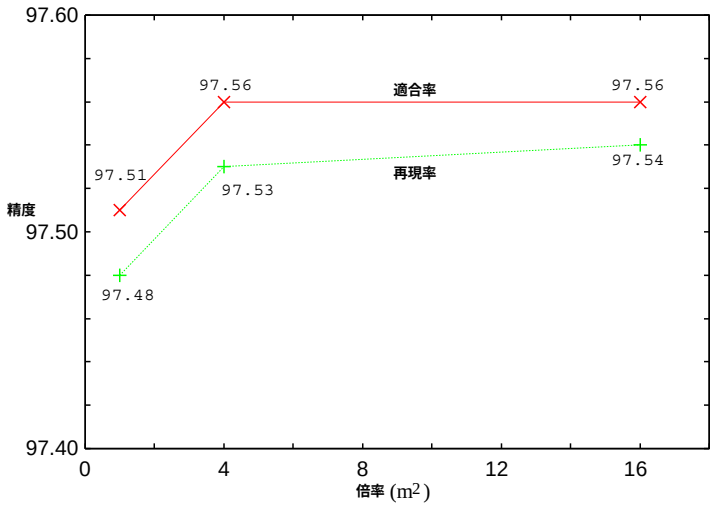

図 4　疑似確率的単語分割と疑似タグ付与の合計の倍率 (m^2) と仮名漢字変換精度の関係

生コーパスの形態素解析結果を学習コーパスに利用していないが、生コーパスの利用による精度向上は広く一般に知られているので、単語を言語モデルの単位とし、生コーパスの決定的な単語分割と入力記号列付与結果を利用する DDw が既存手法に対応するとし、これをベースラインとする。

まず、表 2 中の DDw と DSw と SSw の比較についてである。これらは、すべて単語を言語モデルの単位とする。自動分割と入力記号付与の両方を決定的に行った結果から言語モデルを推定するベースライン DDw に対して、入力記号付与を確率的に行う DSw はより高い変換精度となっている。これにより、入力記号付与を確率的に行うことが有効であることが分かる。DDw と DSw の言語モデルは共通で、違いは仮名漢字モデルのみである。このことから、入力記号付与を確率的に行うことで、仮名漢字モデルがより適切に推定できることが分かる。さらに、単語分割も確率的に行う SSw の精度は、入力記号付与のみを確率的に行う DSw よりも高くなっている。このことから、確率的入力記号付与は、確率的単語分割 (森，小田 2009) と協調して精度向上に寄与することがわかる。

次に、表 2 中の DDu と DSu と SSu の比較についてである。これらは、すべて単語と入力記号列の組を言語モデルの単位とする。この場合も、確率的に入力記号を付与することで精度が向上し、単語分割も確率的に行うことでさらに精度が向上していることが分かる。

さらに、言語モデルの単位の差異についてである。表 2 から、DDw と DDu、DSw と DSu、SSw と SSu のいずれの組の比較においても、言語モデルの単位を単語から単語と入力記号列の組に変更することで変換精度が向上していることが分かる。

最後に、提案手法 SSu における倍率と精度の関係についてである。これを調べるために、m 倍の疑似確率的単語分割の各結果に対する m 倍の疑似確率的タグ付与の結果 (合計 m^2, $m \in \{1, 2, 4\}$) を用いた場合の精度を計算した。図 4 は、倍率と精度の関係である ($m = 1$ は、表 2 の SSu と同

じ)。この結果から、倍率を上げることで、少しではあるが精度が向上することがわかる。一方で、それぞれの場合の語彙 (表記と読みの組) のサイズは順に、123,078 組、181,800 組、295,801 組であり、単語分割とタグ付与を決定的に行う DDu の 99,210 組との差は、倍率が大きくなるに従って非常に大きくなる。図 4 から精度の差は大きくないので、倍率は 1^2 か 2^2 程度が現実的であろう。

以上のことから、仮名漢字変換の言語モデルを単語から単語と入力記号列の組とし、入力記号を確率的に付与したコーパスからこれを推定することが有効であると言える。さらに、確率的単語分割と組み合わせることでさらなる精度向上が実現できると結論できる。

5　おわりに

本論文では、単語分割済みコーパスの各単語に対して、確率的にタグを付与することを提案した。具体的なタグとして単語の読みを採用し、ある単語がある読みになる確率を読みが付与されていないコーパスから推定することを実現した。さらに、単語分割済みコーパスから自動読み推定を用いて表記と読みの組を単位とする確率的言語モデルを推定し、仮名漢字変換に用いることを提案した。

実験では、単語分割や読み推定が決定的にあるいは確率的に行われているコーパスから、単語を単位とする言語モデルと、単語と読みの組を単位とする言語モデルを推定し、仮名漢字器を構築した。これら複数の仮名漢字器の変換精度を比較した結果、単語と読みの組を言語モデルの単位とし、そのパラメータを確率的に単語分割されかつ確率的に読み付与されたコーパスから推定することで最も高い変換精度となることが分かった。したがって、本論文で提案する単語と読みの組を単位とする言語モデルと、確率的タグ付与コーパスの概念は有用であると結論できる。

A　自動読み推定

本論文で用いた自動読み推定 (森，Graham 2010) は、コーパスに基づく方法であり、単語に分割された文を入力とし、単語毎に独立に以下の分類に基づいて読み推定が行われる。

$\mathbf{Q}_1$　学習コーパスに出現しているか

はい

$\mathbf{Q}_2$　読みが唯一か複数か

複数 ⇒ ロジスティック回帰 (Fan, Chang, Hsieh, Wang, and Lin 2008) を用いて読みを選択

唯一 ⇒ その読みを選択

いいえ

$\mathbf{Q}_2'$　辞書に入っているか

はい ⇒ 最初の項目の読みを選択

いいえ ⇒ 文字と読みの 2-gram モデルによる最尤の読みを選択

複数の読みが可能でその確率が必要な場合には、ロジスティック回帰の出力確率や文字と読みの 2-gram モデルによる生成確率を正規化した値を利用する。

分類器の学習に用いたコーパスは、現代日本語書き言葉均衡コーパス (Maekawa 2008) であり、辞書は UniDic (伝，小木曽，小椋，山田，峯松，内元，小磯 2007) である。

学習コーパスとして 33,147 文 (899,025 単語, 1,292,249 文字) を用い、テストコーパスとして同一分野の 3,681 文 (98,634 単語, 141,655 文字) を用いた場合の読み推定精度を測定した。評価基準は、入力記号単位の適合率と再現率である。その結果、一般的な手法である単語と読みを組とする 3-gram モデル (長野他 2006) の適合率と再現率はそれぞれ 99.07%と 99.12%であり、本論文で用いた自動読み推定の適合率と再現率はそれぞれ 99.19%と 99.26%であった。この結果から、本論文で用いた自動読み手法は、既存手法と同程度の精度となっていることがわかる。

参考文献

Aho, A. V. (1990). “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巻, pp. 263–304. Elsevier Science Publishers.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). “LIBLINEAR: A Library for Large Linear Classication.” *Journal of Machine Learning Research*, **9**, pp. 1871–1874.

Google (2010). “Google IME.”. `http://www.google.com/intl/ja/ime/` (2010 年 10 月).

Jelinek, F. (1985). “Self-Organized Language Modeling for Speech Recognition.” Tech. rep., IBM T. J. Watson Research Center.

Maekawa, K. (2008). “Balanced Corpus of Contemporary Written Japanese.” In *Proceedings of the 6th Workshop on Asian Language Resources*, pp. 101–102.

Graham Neubig，中田陽介，森信介 (2010). “点推定と能動学習を用いた自動単語分割器の分野適応.” 言語処理学会年次大会.

西村雅史，伊東伸泰，山崎一孝 (1999). “単語を認識単位とした日本語の大語彙連続音声認識.” 情報処理学会論文誌, **40** (4), pp. 1395–1403.

丸山宏，荻野紫穂，渡辺日出雄 (1991). “確率的形態素解析.” 日本ソフトウェア科学会第 8 回大会論文集, pp. 177–180.

伝康晴，小木曽智信，小椋秀樹，山田篤，峯松信明，内元清貴，小磯花絵 (2007). “コーパス日本語学のための言語資源：形態素解析用電子化辞書の開発とその応用.” 日本語科学, **22**, pp. 101–122.

永田昌明 (1999). “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.” 情報処理学会論文誌, **40** (9), pp. 3420–3431.

森信介，Graham Neubig (2010). “仮名漢字変換ログの活用による言語処理精度の自動向上.” 言語処理学会年次大会.

森信介，土屋雅稔，山地治，長尾真 (1999). “確率的モデルによる仮名漢字変換.” 情報処理学会論文誌, **40** (7), pp. 2946–2953.

森信介，宅間大介，倉田岳人 (2007). “確率的単語分割コーパスからの単語 N-gram 確率の計算.” 情報処理学会論文誌, **48** (2), pp. 892–899.

森信介，小田裕樹 (2009). “擬似確率的単語分割コーパスによる言語モデルの改良.” 自然言語処理, **16** (5), pp. 7–21.

森信介，長尾眞 (1998). “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, **5** (2), pp. 75–103.

村上仁一 (1991). “漢字かなのＴＲＩＧＲＡＭをもちいたかな漢字変換方法.” 情報処理学会第 43 回全国大会, 3 巻, pp. 287–288.

鹿野清宏，伊藤克亘，河原達也，武田一哉，山本幹雄 (2001). 音声認識システム. オーム社.

工藤拓，山本薫，松本裕治 (2004). “Conditional Random Fields を用いた日本語形態素解析.” 情報処理学会研究報告, NL161 巻.

長野徹，森信介，西村雅史 (2006). “N-gram モデルを用いた音声合成のための読み及びアクセントの同時推定.” 情報処理学会論文誌, **47** (6), pp. 1793–1801.

松本裕治，黒橋禎夫，宇津呂武仁，妙木裕，長尾眞 (1993). 日本語形態素解析システム JUMAN 使用説明書 version 1.0. 京都大学工学部長尾研究室.

堤怜介，加藤正治，小坂哲夫，好田正紀 (2002). “講演音声認識のための音響・言語モデルの検討.” 電子情報通信学会技術研究会報告, pp. 117–122.

略歴

森 信介：1998 年京都大学大学院工学研究科電子通信工学専攻博士後期課程修了. 同年日本アイ・ビー・エム (株) 入社. 2007 年より京都大学学術情報メディアセンター准教授. 現在に至る. 自然言語処理ならびに計算言語学の研究に従事. 工学博士. 1997 年情報処理学会山下記念研究賞受賞. 2010 年情報処理学会論文賞受賞. 2010 年第 58 回電気科学技術奨励賞. 情報処理学会会員.

笹田 鉄郎：2007 年京都大学工学部電気電子工学科卒業. 2009 年同大学院情報学研究科修士課程修了. 同年同大学院博士後期課程に進学. 現在に至る.

NEUBIG Graham： 2005 年米国イリノイ大学アーバナ・シャンペーン校工学部コンピュータ・サイエンス専攻卒業. 2010 年京都大学大学院情報学研究科修士課程修了. 同年同大学院博士後期課程に入学. 現在に至る. 自然言語処理に関する研究に従事.