あいまいな日本語のかな漢字変換

小町　守[†1]　森　信介[†2]　徳永拓之[†3]

近年 Web を中心としてユーザが入力する文書が爆発的に増大している。Web テキストでは次から次に新しい用語が生まれるため、新語を人手で辞書に登録する方法は現実的ではない。また、高い精度で解析するための辞書作成には高度な言語学的知識が必要であり、多大な労力がかかる。そこで、本研究では統計的手法に基づき、大規模な Web データを用いたかな漢字変換システムを提案する。本システムのポイントは、大規模テキストから推定した言語モデルを用いてかな漢字変換を行うことである。提案手法は言語学的知識を必要とせず、メンテナンスも用意である。

Japanese, the ambiguous, and Input Methods

Mamoru Komachi , Shinsuke Mori and Hiroyuki Tokunaga

Japanese inputh method is one of the most difficult problems for Japanese PC users. The rapid growth of WWW in recent years creates a vast amount of newly-coined terms and poses a yet another challenge to Japanese input method. It is expensive for PC users to add new terms to its dictionary since it requires linguistic knowledge to annotate. In this paper, we propose a statistical approach to Japanese Input Method Editor. This approach uses a bigram-based language model and Kana-Kanji conversion model. We use a large-scale corpus for the language model to include all the words that appear in the corpus. The proposed approach does not require linguistic knowledge and is easy to maintain.

1. はじめに

近年 Web を中心としてユーザが入力する文書が爆発的に増大している。Web テキストでは次から次に新しい用語が生まれるため、新語を辞書に登録する方法は現実的ではない。また、高い精度で解析するための辞書作成には高度な言語学的知識が必要であり、多大な労力がかかる。

そこで、本研究では統計的かな漢字変換手法[1)] に基づき、大規模な Web データを用いたかな漢字変換システムを提案する。本システムのポイントは、単語の品詞情報の代わりに大規模テキストから推定した文字の連接情報を用いて単語分割と変換を行うことである。大量のデータを用いれば品詞情報を補うことができると考えられ、品詞情報に頼らないことで辞書のメンテナンスの問題を克服する。また、Web データを処理した大規模コーパス中に出現する単語を用いることで、特別な未知語処理を組み込まない場合でも適切な確率推定が行えるようになる。

†1 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
†2 京都大学
Kyoto University
†3 プリファードインフラストラクチャー
Preferred Infrastructure

2. 関連研究

これまでのかな漢字変換システムとしては、ヒューリスティックに基づく変換が広く用いられてきた。

2.1 ルールによる変換

Canna[⋆1] に代表されるかな漢字変換システムは、複雑なルールを用いて変換を行う。これらのルールは言語学的な直観に基づいて作成されたものであるが、時としてアドホックであり、数学的な裏付けはない。また、Canna の辞書 cannadic[⋆2] には人手による単語コスト（ある 1 つの単語の出現しやすさ）と単語同士の連接コスト（2 つの単語のつながりやすさ）が付与されており、総合コストが低い候補が選択される。しかしながら、これらのルールやコストのメンテナンスには言語学（国語学）的知識が必要であり、必ずしもソフトウェアの開発者・ユーザが言語学に詳しいとは限らず、メンテナンスのハードルとなっている。かな漢字変換が人手で書き尽くせる程度の単語・ルールで表現できる問題であれば高精度に変換できることが期待

⋆1 http://canna.sourceforge.jp/
⋆2 http://cannadic.oucrc.org/

されるが、特に近年の Web の拡大により、これらの多様な言語表現を網羅することは不可能に近い。

2.2 N 文節最長一致法（後ろ向き N 文節評価最大法）

かな漢字変換では文節を用いた変換を行うと効率的に探索空間を減らして高精度に変換できることが知られている。文節分割のヒューリスティックとして、N 個の文節の分割結果がいちばん長くなるようような分割を行う、というヒューリスティックがある。以前の ATOK や VJE、Wnn[⋆1] がこのカテゴリに入る。しかしながら、たとえば Wnn では変換に関する数十個に渡るパラメタを最適化しなければならず、統一的に最適化する方法も用意されていないため、一般ユーザには調整が難しい。そして Wnn で用いられている pubdic+[⋆2] といった辞書は、品詞の数が数百あるなど、日本語とシステムの双方に詳しい開発者でないと適切な品詞を付与することができないといった問題点がある。また、これらのソフトウェアで用いられている変換アルゴリズムは、ヒューリスティックに基づくものであり、ルールによる変換と同様、理論的な背景に乏しい。

2.3 最小コスト法

文全体の変換結果を得るとき、連接コストと単語コストが最小になるような候補を出力する手法を最小コスト法と言う。WXG や MS-IME 2007 以前の MS-IME がこのカテゴリに属する。このコストは人手により付与されていたが、試行錯誤により最適値を決定しなければならないため非常に労力がかかり、客観的な評価が難しい、という問題がある。

2.4 機械学習（識別モデル）による変換

ここ数年のうちに急速に主要 Linux ディストリビューションにて採用が広がったかな漢字変換ソフトとして Anthy[⋆3] がある。Anthy は辞書としては前述の cannadic を用いているため、基本的に cannadic の問題点（コストのメンテナンスが大変）を受け継ぐほか、付属語を用いた変換を行うため、変換には付属語辞書のカバー率が重要となる。変換モデルとして現在は最大エントロピー法に基づいた機械学習による識別モデルを用いた変換を行っているが、学習に用いたデータが少量であるため十分なモデルを学習することができず[⋆4]、人手によるパラメータの調整が依然必要である[⋆5] 。

3. 統計的かな漢字変換

統計的かな漢字変換では、かな漢字変換を数学的な枠組みで定式化し、変換規則や辞書に相当する確率モデルをコーパスから自動的に学習し、パラメータを最適化できる、という利点がある。短所としてはコーパスの質と量に性能が依存する点と、変化の微調整が難しい点であるが、本研究では Web から獲得した大規模なコーパスを使うため、前者の問題を解決することができる。

統計的かな漢字変換では、与えられたかな文字列の入力 y に対して変換候補 $(x_1, x_, \cdots)$ を確率 $P(x|y)$ の降順に提示する。

$$i \leq j \Longleftrightarrow P(x_i|y) \geq P(x_j|y) \tag{1}$$

この確率値が最大のものがもっとも尤もらしい変換候補となり、尤もらしさ順に確率値が並んでいることが統計的かな漢字変換の基本原理である。$P(x|y)$ を直接推定する方法があればそのままかな漢字変換に用いることができるが、一般にこの確率値を直接推定することは難しいため、この確率値は統計的機械翻訳や音声認識と同様、ベイズの定理を用いることによって以下のように推定することができる。

$$P(x_i|y) \geq P(x_j|y) \tag{2}$$

$$\Longleftrightarrow \frac{P(y|x_i)P(x_i)}{P(y)} \geq \frac{P(y|x_j)P(x_j)}{P(y)} \tag{3}$$

$$\Longleftrightarrow P(y|x_i)P(x_i) \geq P(y|x_j)P(x_j) \tag{4}$$

この式において、かな漢字交じり文 x の生起確率 $P(x)$ は確率的言語モデルと呼ばれ、$P(y|x)$ は確率的かな漢字変換モデルと呼ばれる。以下でそれぞれについて述べる。

まず確率的言語モデルについて述べる。確率的言語モデルは、与えられた文字列がある言語の文である尤度を数値化したものであり、統計的機械翻訳や音声認識でも用いられている。最も一般的な確率的言語モデルは単語 n-gram モデルであり、このモデルは文を単語列 $w_1^h = w_1 w_2 \cdots w_h$ からなるものと見なし、文頭から順に単語を予測する。

$$M_{w,n}(w) = \Pi_{i=1}^{h+1} P(w_i|w_{i-n+1}^{i-1}) \tag{5}$$

ここで $w_i(i < 0), w_{h+1}$ はそれぞれ文頭と文末を表す記号である。一般的にかな漢字変換では単語分割されないままかな文字列が入力されるため、単語分割が最初に大きな問題となる。この問題を解決するために、

⋆1 http://freewnn.sourceforge.jp/
⋆2 http://www.remus.dti.ne.jp/~endo-h/wnn/
⋆3 http://sourceforge.jp/projects/anthy/
⋆4 コーパスを削除し学習の影響を取り除くという修正まで行われている。http://www.geocities.jp/ep3797/anthy_dict_01.html
⋆5 http://www.fenix.ne.jp/~ G-HAL/soft/nosettle/♯anthy

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。[2)]

$$\hat{w} = arg\max_{w=x} M_{w,n}(w) \tag{6}$$

本研究でも同様に文字列 x として与えられる文の確率が最大となるような単語分割を自動分割結果として用いる。

先行研究では、辞書のカバー率を補うために未知語を予測し、さらにその未知語の表記を文字 n-gram で予測する手法が提案されている[3)] が、本研究では大規模なコーパスを用いることで、未知語モデルを使用せず表記のみに基づいて変換する。大規模コーパスには可能な単語や連接が網羅されているため、素朴な未知語モデルで十分であると考えられる。

次に確率的かな漢字変換モデルについて述べる。確率的かな漢字変換モデルとは、かな漢字交じり文 x が与えられたときのキーボードからの入力記号列 y の確率を表す。単語列 w が与えられたときの確率的かな漢字変換モデルによる確率は以下の式で表す。

$$M_{kk}(y|w) = \Pi_{i=1}^{h} P(y_i|w_i) \tag{7}$$

ここで入力記号部分列 y_i は単語 w_i に対応する入力記号列であり、

$$y = y_1 y_2 \cdots y_h \tag{8}$$

を満たす。確率 $P(y|w)$ の値は単語分割および読みの付与がなされたコーパスから最尤推定によって計算する。

$$P(y_i|w_i) = \frac{f(y_i, w_i)}{f(w_i)} \tag{9}$$

ただし $f(e)$ はコーパス中における事象 e の頻度である。

以上で述べた単語候補を枚挙するシステムは、入力記号列 y を受け取りあらゆる部分文字列から変換可能な単語列 w を出力するモデルであり、以下のようになる。

$$P(y|x)P(x) = \Pi_{i=1}^{h} P(y_i|w_i)P(w_i) \tag{10}$$

$$P(y_i|w_i)P(w_i) = P(w_i|w_{i-n+1}^{i-1} P(y_i|w_i) \tag{11}$$

しかしながら、単純に単語候補を列挙すると、計算量が $O(N^h)$ （ただし N は状態数）かかり、長い文字列を変換する際非常に非効率である。そこで、前向き探索として動的計画法（Viterbi 探索）を用いることで、効率的に最尤候補（1-best 解）を出力することが可能である。すなわち、文頭から i 番目までの単語列の同時確率 $P(w_1 \cdots w_i)$ の最大値を $\Phi(w_i)$ とすると、以下の関係が成立する。

$$\Phi(w_i) = \max_{w_{i-1}} \Phi(w_{i-1})P(w_i|w_{i-1}) \tag{12}$$

この関係を用いて文頭から順次 $\Phi(w_i)$ を求めれば、文頭から文末までの同時確率の最大値 $\Phi(w_n)$ を計算量 $O(hN)$ で効率的に求めることができる。

また、N-best 解の出力は後ろ向き A*探索を用いることで、計算量と記憶量を抑えつつ効率的に列挙することができる。

4. 統計的かな漢字変換の実装

本研究ではプログラミング言語 Python を用いて上記の統計的かな漢字変換システムを実装した。ソースコードやドキュメントは Google Code ⋆1 上で公開している。確率的言語モデルには Google 日本語 N グラム[4)] を用いた。Google 日本語 N グラムは 200 億文の Web コーパスから単語 N グラム（$1 \leq N \leq 7$）を抽出したもので、今回は単語 1-gram および 2-gram を用いた。単語 1 グラムは 250 万タイプ、単語 2 グラムは 8,000 万タイプからなり、フリーで入手可能な最大規模の辞書 NAIST-jdic⋆2 は約 40 万語からなる辞書であるため、約 6 倍程度とかなり大きな規模の辞書と見ることができる。大量のキーから値（頻度）を引くためのデータベースマネージャとしては、一般的に用いられている GDBM や BDB といった実装では 100 万を超える巨大なキー集合を扱うのは非効率で現実的ではないため、巨大なキー集合を扱うのに適した TokyoCabinet⋆3 を用いた。⋆4確率的かな漢字変換モデルの学習には、形態素解析器 MeCab⋆5 を用いて解析した毎日新聞 13 年分のデータを使用した。

かな漢字変換システムはオンラインデモ⋆6でサービスしており、Ajax を用いてブラウザ経由でアクセスし、サーバ=クライアント方式で変換している（図 1）。こうした理由の一つとしては、Google 日本語 N グラムは商用利用不可・研究目的限定で使用許諾されたデータなので、サーバ側にのみデータを保持する形でないと公開ができないことがある。また、このようにする利点としては、かな漢字変換のログをサーバ側に蓄積

⋆1 http://code.google.com/p/chaime/
⋆2 http://sourceforge.jp/projects/naist-jdic/
⋆3 http://tokyocabinet.sourceforge.net/
⋆4 データ容量の圧縮という観点からは Succinct Data Structure を用いた Tx (http://www-tsujii.is.s.u-tokyo.ac.jp/ hillbig/tx.htm) のほうが検索速度と使用記憶容量の観点から望ましいが、ここでは次に述べるように巨大なディスクを設置できるサーバ側にリソースを置けばよいため、検索速度を優先した。
⋆5 http://mecab.sourceforge.net/
⋆6 http://cl.naist.jp/m̃amoru-k/chaime/

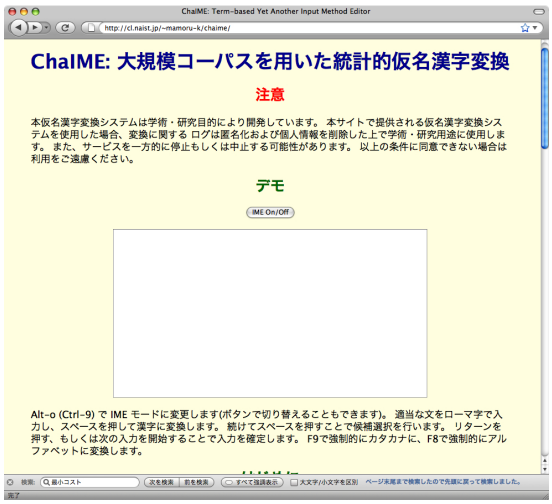

図 1　ブラウザによる入力インタフェース

し、将来変換ログを用いた予測入力や変換候補のリランキングを導入することができる点のほか、海外のインターネットカフェでかな漢字変換環境がインストールされていなくても、フォントさえあれば利用できるという点がある。逆にブラウザ経由にする欠点は、オフラインで使えない、ブラウザ以外の入力に使うためには変換結果をコピー and ペーストせねばならない、外部プログラムとの連携が困難である、といった点である。

ブラウザ経由でかな漢字変換を提供する他のシステムとしては Sumibi[⋆1] および AjaxIME[⋆2] がある。

前者は生コーパスを用いて単語 1-gram/2-gram/スキップ 2-gram を計算するもので、ユーザが任意の生コーパスを追加することができる点で提案手法と共通しているが、入力時ユーザは単語分かち書きをしないといけないという制約がある。また、Sumibi は辞書に載っていない単語の頻度を計算することができないため、生コーパスの処理に用いた形態素解析器と同じ単位で分かち書きをしなければならず、辞書にない複合名詞の場合は単漢字変換を繰り返す必要があり、入力の手間がかかる。

後者の AjaxIME は形態素解析器 MeCab と IPADic[⋆3] を用い、言語モデルを IPA コーパスから条件付き確率場（CRF）によって推定したものであり、変換エンジンは mecab-skkserv[⋆4] と同一である。これは確率的な推定を行っているという理論的な裏付けがあるが、かな漢字変換モデルは統計的モデルを使用していない。また用いたコーパスが 4 万文と小さく、Google 日本語 N グラムの 1/500,000 であり、データの過疎性の問題がある。

⋆1 http://www.sumibi.org/
⋆2 http://ajaxime.chasen.org/
⋆3 http://sourceforge.jp/projects/ipadic/
⋆4 http://chasen.org/~taku/software/mecab-skkserv/

5. 実　　験

変換のサンプルを以下に示す。変換にはジャストシステムのサイト[⋆5] に掲載されている例文を使用し、最尤候補（1-best 解）を掲げる。

表 1 から分かるように、提案手法は新聞記事のようなテキストデータや「サイト」などの新語に対して適切な変換ができるという特徴を持つ。これは確率的言語モデルに Google 日本語 N グラムを用いていることと、確率的かな漢字変換モデルの学習に新聞記事を用いているため、例文のような文の変換に適しているためだと考えられる。

一方、提案手法が変換に失敗する文として以下のものがある（単語境界に/を付した）。

- 貴社/の/記者/が/記者/で/帰社/し/た/。
- ここ/で/は/着物/を/切る
- 民主/ワイオミング/州/は/小浜/市/大正（正しくは「オバマ氏大勝」）

最初 2 つの事例では、2-gram モデルは隣り合う単語の連接しか考慮しないため、「記者/が/記者/で」や「着物/を/切る」という日本語としてはありそうにない単語の並びを棄却することができなかった、という問題であるが、3-gram や 4-gram などを言語モデルとして使うことによって改善されると思われる。最後の事例は「オバマ/氏」という未知語が含まれるため解析に失敗している例であるが、最新のニュース記事に対して自動解析をかけたコーパスを追加することで、「オバマ/氏」も正しく変換できるようになると考えられる。

6. まとめと今後の予定

確率モデルに基づく統計的かな漢字変換システムを実装した。提案手法は既存のヒューリスティックなかな漢字変換システムと異なり、理論的根拠があり、辞書や変換ルールを人手で調整する必要がなく、自動でコーパスから学習できる、という利点がある。また、Web から獲得した大規模なコーパスを用いることで、書き言葉でも話し言葉でも高精度な変換を行うことができることを示した。

本手法では言語モデルの学習に複数のコーパスを用いることで、入力している文章の分野に適応したかな

⋆5 http://www.justsystems.com/jp/products/atok/

表 1 変換サンプル

アルゴリズム	請求書の支払い日時	近く市場調査を行う。	初めっから持ってけばいいのに。	熱々の肉まんにぱくついた。
ChaIME	請求書の支払日時	近く市場調査を行う。	初めっからもってけばいいのに。	熱々の肉まんにぱくついた。
ATOK 2007	請求書の市は来日時	知覚し冗長さを行う。	恥メッカら持って毛羽いいのに。	熱々の肉まん二泊着いた。
Anthy 9100d	請求書の支払い日時	近く市場調査を行う。	恥メッカらもって毛羽いいのに。	あつあつの肉まん２泊付いた。
AjaxIME	請求書の支払いに知事	近く市場調査を行う。	始っから持ってけば良いのに。	熱熱の肉まんにぱくついた。
アルゴリズム	その後サイト内で	去年に比べ高い水準だ。	昼一までに書類作っといて。	そんな話信じっこないよね。
ChaIME	その後サイト内で	去年に比べ高い水準だ。	昼イチまでに書類作っといて。	そんな話信じっこないよね。
ATOK 2007	その五歳都内で	去年に比べた海水順だ。	昼一までに書類津くっといて。	そんな話心十個内よね。
Anthy 9100d	その後サイト内で	去年に比べたかい水準だ。	昼一までに書類作っといて。	そんなはな視診時っこないよね。
AjaxIME	その後再都内で	去年に比べ高い水準だ。	肥留市までに書類作っといて。	そんな話神事っ子ないよね。

漢字変換を行うことができる枠組みとなっている。たとえばブログを書いているときには Web をクロールしたブログデータから作成した言語モデルを使用し、論文を書いているときには手元にある日本語論文誌・研究会報告データから作成した言語モデルを用いることで、高精度な変換が可能になる。過去に入力した単語の履歴からこれらのトピックを判定し、言語モデルの切り替えを行いつつ変換を行う手法の実装は今後の課題である。

また、サーバ=クライアント型のかな漢字変換システムから取得した変換履歴を用いた変換システムの作成も検討している。たとえば間違った入力で確定し、バックスペースで削除してから直す、という操作をする人が多い、という調査結果が知られている。こういった変換のセッションから間違った操作を特定することで、精度の高い訓練データをマイニングすることが考えられる。

謝　　辞

適宜アドバイスをくださった奈良先端科学技術大学院大学浅原正幸氏、ジャストシステム高岡一馬氏、Google 工藤拓氏、NTT 永田昌明氏に感謝する。本研究の一部は奈良先端科学技術大学院大学情報科学研究科大学院教育改革支援プログラムの支援を受けた。

参 考 文 献

1) 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, 情報処理学会論文誌, Vol.40, No.7, pp. 2946–2953 (1999).
2) 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, 情報処理学会論文誌, Vol.40, No.9, pp. 3420–3431 (1999).
3) 森信介, 小田裕樹：自動未知語獲得による仮名漢字変換システムの精度向上, 言語処理学会第 13 回年次大会論文集 (2007).
4) 工藤拓, 賀沢秀人：Web 日本語 N グラム第 1 版 (2007).

付録　質疑応答

Q（疋田@トヨタ IT センター） 変換精度が低くても ATOK はストレスが少ないと言われているが、精度とストレスの関係はあるのか？

A これが出来るということではなく，出来ないとストレスを感じられるというのがかな変換の難しいところである。また、ストレスは変換精度だけでなくインタフェースの影響も強く受けるため、精度が悪かったからといってすぐストレスになるわけではない。

Q（長@一橋大学） 半角全角を自動で認識してくれるというものはないのか？自分の入力の癖を学習して自然に入力できるようになってほしい。

A ATOK も MS-IME も半角らしきものは半角に変換する。これもインタフェースの問題で、精度を上げるには自動でやらず、手動でモードを切り替えるしかない。SKK がうまくいっているのは同音異義語が特定のものに偏っているから。また、学習したからといって統計的な枠組みでうまくいくとは限らない。

Q（山口@東京理科大学） 学習すればするほど悪くなるというのは、統計的な法則に反しているのでは？

A 正しい入力が来ればいいが、間違った入力が来ても計算機は間違っているということが分からないため、適切なコストを推定できないと言う問題がある。これを解決するには変換のログを見て変換操作のセッションの情報を使えば対処できると考えている。