日本語かな漢字変換における識別モデルの適用とその考察

徳永拓之　　岡野原大輔
株式会社 Preferred Infrastructure
{tkng, hillbig}@preferred.jp

1　はじめに

これまでのかな漢字変換システムの研究では，確率モデル，特にその中でも生成モデルを利用したものが多く，識別的な手法を用いたものはあまり見られない．しかし，近年の研究成果により，大規模データを用いた識別的なモデルの学習も容易となってきた．

生成モデルには教師なし学習が可能であるというメリットがあるが，データの生成過程をモデル化する前提上，パラメーターに対する制約が強く，教師あり学習で識別性能のみを最大化するのが目的であれば，より直接的な識別モデルを利用した方が良い結果が得られることが多い．

また，識別モデルには素性設計の自由度が高いというメリットがある．例えば品詞バイグラムと表記バイグラムの両方の素性を同時に取り込んでパラメーター推定する事は困難である．識別モデルであれば，このような場合でも識別器に与える素性を増やすだけで，それぞれの素性に与える重みは最適化アルゴリズムによって自動的に調整される．

そこで本研究では，識別的な手法を用いて構築したかな漢字変換システムと生成モデルを用いた場合とを比較し，性能差を調べた．

2　かな漢字変換とは

かな漢字変換は，仮名文字列の入力に対し，適切な漢字かな混じり文を返す問題である．入力 $\boldsymbol{x}$ についての変換候補 $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \ldots$ を，スコア $f(\boldsymbol{x}, \boldsymbol{y})$ の降順に提示する．本稿では，変換例の集合からスコア関数 $f(\boldsymbol{x}, \boldsymbol{y})$ を学習するタイプのかな漢字変換について議論する．具体的な f の構成の手順については 4 節と 5 節にそれぞれ示す．

3　関連研究

1990 年代までのかな漢字変換エンジンは，n 文節最長一致と呼ばれるヒューリスティックや，人手によって調整されたルールに基づいて変換を行うものであった．しかし，ルールベースの手法にはルールのメンテナンスが煩雑であるという問題があり，確率モデルに基づくかな漢字変換システムが提案された．

森らは確率的言語モデルを用いたかな漢字変換を提案した [6]．彼らが用いたのはクラスバイグラムと呼ばれる言語モデルを用いる手法である．

Gao らは，識別的言語モデルを利用したかな漢字変換に関する実験結果を報告している [7]．ただし，彼らの研究はドメイン適応が目的であり，生成モデルによって得られた上位 n 個の変換候補へのリランキングに識別モデルを利用している．

かな漢字変換と形態素解析には共通する部分が多い．工藤らは形態素解析に対して識別的な確率モデルである条件付確率場を適用した [8]．条件付確率場は，label bias や length bias に弱いという既存の手法の弱点がなく，隠れマルコフモデルや最大エントロピーマルコフモデルと比較し精度が良いことを示している．

4　確率的言語モデルに基づくかな漢字変換

確率的言語モデルに基づくかな漢字変換では，入力 $\boldsymbol{x}$ についての変換候補 $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \ldots$ を，確率 $P(\boldsymbol{y}|\boldsymbol{x})$ に基づいて降順に提示する．$P(\boldsymbol{y}|\boldsymbol{x})$ はベイズの定理を用いて $P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})/P(\boldsymbol{x}) \propto P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})$ と変形することができる．$P(\boldsymbol{y})$ は言語モデルそのものであり，$P(\boldsymbol{x}|\boldsymbol{y})$ はかな漢字モデルと呼ばれる．

言語モデル $P(\boldsymbol{y})$ は尤もらしい単語列に対し高い確率を与えるモデルである．かな漢字変換においては，連続する二つの単語表記に基づいた単語バイグラムモデルや，単語をクラスタリングした結果に基づいたクラスバイグラムモデルなどが多く利用される．

かな漢字モデル $P(\boldsymbol{x}|\boldsymbol{y})$ は，単語 (列) に対し尤もらしい読みがなに対し高い確率を与えるモデルである．[6] ではかな漢字モデルとして，単語列を単語に分解し，それぞれの単語の表記に対するよみがなの確率の積をかな漢字モデルとして用いている．

5 識別的手法に基づくかな漢字変換

次に，識別的手法に基づくかな漢字変換を説明する．入力 $\boldsymbol{x}$ についての変換候補 $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \ldots$ を，スコア $f(\boldsymbol{x}, \boldsymbol{y})$ に基づいて降順に提示する．

スコア関数 $f(\boldsymbol{x}, \boldsymbol{y})$ は，素性関数 Ψ の和に分解され，有効な素性の重み付き和がスコア関数の値となる．本研究では，ビタビアルゴリズムで探索ができるよう，素性関数の引数として $\boldsymbol{x}, y_{j-1}, y_j$ のみを用いるものとする．これにより，入力 $\boldsymbol{y}$ についての変換候補 $\boldsymbol{x}$ のスコアは，下記の式で計算されることになる．

$$f(\boldsymbol{x}, \boldsymbol{y}) = \sum_j \sum_k w_k \Psi_k(\boldsymbol{x}, y_{j-1}, y_j) \qquad (1)$$

ただし，素性関数は K 個存在するものとし，w_k はそれぞれの素性関数の重みパラメーターであるものとする．w_k をまとめたものを $\boldsymbol{w} \in R^K$ とする．

識別モデルは $\boldsymbol{x}$ を利用した素性関数が自由に設計できる点で生成モデルよりも強力である．

かな漢字変換のような構造学習に適用できる識別モデルとしては条件付確率場 (CRF)[3] や構造化 SVM[5, 4] などがある．素性が同じであれば構造化 SVM と CRF の性能はほぼ同じであるという報告 [2] に基づき，今回は実装が簡単であった構造化 SVM を採用した．

5.1 構造化 SVM

構造化 SVM は SVM を構造学習のために拡張したものである．本研究では 1 正則化線形構造化 SVM を用いる．

i 番目の学習用のデータの $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ で表すものとし，i 番目の学習用データに対する損失を $\mathcal{L}_i(\boldsymbol{y})$ と表記するものとする．$L_i(\boldsymbol{y})$ は，$\boldsymbol{y} = \boldsymbol{y}^{(i)}$ の時にのみ 0 となり，それ以外の場合には 0 より大きな値を取るものとする．本研究では，かな漢字変換をグラフの最短経路問題として定義した際に，正解の経路以外の頂点と辺に対して 1 を与え，$L_i(\boldsymbol{y})$ はその合計の値とした．

L1 正則化の場合，構造化 SVM における目的関数は以下の式 2 で表現される．

$$\frac{1}{n}\sum_{i=1}^{n} r_i(\boldsymbol{w}) + \lambda|\boldsymbol{w}| \qquad (2)$$

$\lambda > 0$ は損失項と正則化項とどちらを重要視するかを決めるパラメーターである．また，$r_i(\boldsymbol{w})$ は

$$\max_{\boldsymbol{y}\in\mathcal{Y}}(\boldsymbol{w}\cdot f(\boldsymbol{x}^{(i)}, \boldsymbol{y}) + \mathcal{L}(\boldsymbol{y})) - \boldsymbol{w}\cdot f(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \qquad (3)$$

となる．

この目的関数を最小化するような $\boldsymbol{w}$ を求めるため，パラメーター最適化に FOBOS[1] を用いた．

5.2 FOBOS による構造化 SVM の学習

FOBOS はパラメーター最適化手法の一種であり，オンライン型としてもバッチ型としても使うことができる．本研究ではオンライン型の FOBOS を用いた．オンライン型の場合，FOBOS は確率的勾配降下法 (以下 SGD) や劣勾配法の改良と見なすことができる．SGD ではサンプル毎に目的関数の (劣) 勾配の方向へとパラメーターを更新するが，FOBOS ではこのパラメーター更新を損失項と正則化項の二つに分割する．

FOBOS では，損失項に関しては SGD(や劣勾配法) とまったく同様に計算する．正則化項の計算方法が FOBOS の大きな特徴で，閉じた形で新しい正則化後のパラメーターの値を求める．これにより，特に L1 正則化の際にスパースな解を効率よく求めることができるようになっている．

リスト 5.1 に，FOBOS による構造化 SVM 最適化の擬似コードを示す．

```
for (x^(i), y^(i))
  y* = argmax_y f(x^(i), y) + L(x^(i), y)
  if y* ≠ y^(i)
    w^(i+1/2) = w^(i) − η∇(f(x^(i), y*) − f(x^(i), y^(i)))
    for k ∈ K
      w_k^(i+1) = sign(w_k^(i+1/2)) max{|w_k^(i+1/2)| − λη}
```

リスト 5.1 構造化 SVM による学習

$sign$ は引数の符号に合わせて 1,-1 を返す関数であり，η は学習率である．

∇ は勾配を求める関数である．ただし，SVM の損失関数は微分不可能な点を含むため，厳密には勾配を

用いることはできない．そのため，微分不可能点では勾配の代わり劣勾配を用いるものとする．

素性関数 Ψ_k を，その素性が有効になったときに k の次元にのみ 1 を返し，それ以外の次元はすべて 0 であるベクトルを返す関数と考えると，勾配は以下の式で計算できる．

$$\nabla f(\boldsymbol{x}, \boldsymbol{y}) = \sum_j \sum_k \Psi_k(\boldsymbol{x}, y_{j-1}, y_j) \qquad (4)$$

6 実験と考察

6.1 実験の設定

生成モデルと識別モデルを比較するため，京都大学テキストコーパス (ver 4.0) を用いて実験を行った．学習には 1 月 1 日分から 1 月 15 日分までの 14 日分 (計 16149 文) を用い，1 月 16 日と 17 日の 2 日分 (計 2047 文) で評価を行なった．数詞に関しては京都大学テキストコーパスのままでは明らかに学習が困難であったので，前処理として漢数字および全角数字は 1 文字で 1 形態素へと分解した．

生成モデルとしては，クラスバイグラムを言語モデルとして用いる確率的言語モデルに基づくかな漢字変換エンジンを用いた．言語モデルは，クラスバイグラム，単語表記バイグラム，単語表記ユニグラムの線形和を用いた．重み付けは特に行っていない．森ら [6] はクラスとして単語をクラスタリングした結果を用いているが，今回の実験では品詞の細分類をそのままクラスとして用いた．

識別モデルとしては，構造化 SVM に基づくかな漢字変換エンジンを用いた．素性として，クラスバイグラム，単語表記バイグラム，単語の表記と読みのペアを用いた．構造化 SVM の学習は性能が収束するまで繰り返した．実験結果として，コーパスに対して学習処理を 15 回繰り返した結果の値を記載している．学習器は C 言語で実装し，学習時間は Intel Core i7 920 (2.67GHz) において約 8 分 10 秒であった．

6.2 評価基準

評価基準として，各文を一括変換することで得られる結果と正解との最長共通部分列 (Longest Common Subsequence, LCS) の文字数に基づく精度と再現率を用いた．

正解コーパスに含まれる文字数を N_{COR}，一括変換の結果に含まれる文字数を N_{SYS}，両者の LCS の文字数を N_{LCS} としたとき，精度は N_{LCS}/N_{COR} で，再現率は N_{LCS}/N_{SYS} で定義される．

6.3 実験結果

実験結果を表 1 に示す．提案手法は精度，再現率ともに既存手法を約 3%上回った．

表 1: 変換性能の比較

変換エンジン	精度	再現率
確率的言語モデル	89.1%	88.1%
構造化 SVM	91.8%	91.4%

6.4 誤変換の例

変換結果とコーパスの比較に基づく誤変換の例を以下に挙げる．

6.4.1 同音異義語の間違い

コーパス しかし 衛星 は分離され，地球 周回 軌道に入った．

変換結果 しかし 衛生 は分離され，地球 集会 軌道に入った．

周回軌道という連語を認識できていない．“衛生” に関しては，“地球” のように少し離れた位置との共起情報を用いないと正しい変換は難しい．

6.4.2 未知語の間違い

コーパス ボー・バン・キエト 首相と首脳会談を行う．

変換結果 簿ー・晩・帰依と 首相と首脳会談を行う．

“ボー・バン・キエト” という人名が辞書に入っていなかった．

6.4.3 表記揺れの間違い

コーパス 一 歳年下の弟は中学三年になる ところ だった．

変換結果 1 歳年下の弟は中学三年になる 所 だった．

厳密には間違いではないが，コーパスと変換結果は食い違っている．

6.5 考察

森ら [6] の挙げている，同音異義語，未知語，表記揺れ，という 3 種類の誤変換の分類は，本研究にもそのまま当てはまる．

表記揺れについては個別に適応するしかないし，未知語に関しては何らかの未知語獲得装置を考えるしかないであろう．同音異義語については，“周回軌道” のように単語として辞書に取り込むことで改善できそうなもの，“衛星” のように単語の間の共起関係を用いることで改善できそうなものが存在した．

7 おわりに

本研究では，識別的な機械学習アルゴリズムを用いたかな漢字変換の手法を提案した．

提案したモデルは構造化 SVM を用いてパラメーターを最適化することによってコーパスから学習を行う．コーパスを用いた実験によって生成モデルと比較し，精度と再現率の両方で約 3%の性能の向上が得られる事を確認した．

今回の実験で用いたデータはあまり大規模ではない．大規模データに適用した場合にどこまで精度の向上が得られるかを調査することは，今後の課題である．

識別モデルの利点の一つとして，素性関数の設計の自由度が高いという点が挙げられる．しかし，今回は素性に関してはよみ，表記，クラス (品詞) の 3 種類のみしか用いていない．また，隣接している単語よりも遠い情報は素性として用いていない．入力文字列すべてを素性として組み入れた場合にどうなるかについても実験したい．

本研究で用いたオンライン学習は，変換結果のユーザーからの訂正による個人適応にも自然に応用できる．今後は今回開発したエンジンを組み込んだ日本語入力システムを開発し，実際に利用した際にどのような結果が得られるかを実験する予定である．既存のかな漢字変換エンジンでは，学習の副作用としてこれまで変換できた文の変換ができなくなってしまう点がよく問題として挙げられるが，この理由としては，そもそも理論的な裏付けの乏しいルールベースでの学習である（と考えられる）こと，ユーザーの操作誤りにより，誤った状態で確定された文を学習してしまうことの 2 点が原因として考えられる．単純にオンライン学習を適用するだけでなく，操作誤りを検知するような仕組みに関しても検討してゆきたい．

参考文献

[1] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, Vol. 10, pp. 2899–2934, 2009.

[2] Selvaraj Sathiya Keerthi and Sellamanickam Sundararajan. Crf versus svm-struct for sequence labeling. *Yahoo Research Technical Report*, 2007.

[3] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[4] Nathan Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. (online) subgradient methods for structured prediction. In *Eleventh International Conference on Artificial Intelligence and Statistics (AIStats)*, March 2007.

[5] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, Vol. 6, pp. 1453–1484, 2005.

[6] 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.

[7] Jianfeng Gao, Hisami Suzuki, and Bin Yu. Approximation lasso methods for language modeling. In *Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, July 2006.

[8] 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会自然言語処理研究会 SIGNL-161, Vol. 47, pp. 89–96, 2004.