非局所素性を用いたかな漢字変換

高岡一馬　内田佳孝　松田寛
株式会社ジャストシステム
{kazuma.takaoka, yoshitaka.uchida, hiroshi.matsuda}@justsystems.com

1. はじめに

かな漢字変換に統計的手法を導入するにあたってはいくつかの困難があるが、本稿では特に以下の3点の解決に注目し議論する。

(1)入力に区切りの手がかりがすくなく、同音語がおおいため形態素解析などにくらべ解空間が広い、(2)変換が逐次おこなわれるため入力が文頭から文末まで完全なかたちであたえられない、(3)同音語による曖昧性が顕著で非局所的な文脈(文内共起)を考慮する必要がある。

これらの点について、条件付き確率場(CRF)法をもちいた枠組みでの解決法を提案する。

2. かな漢字変換のモデル化

かな漢字変換はおおまかに入力となるひらがな列から対応する漢字かな交じり文字列を推定するタスクであるが、本稿では以下のようなモデルを前提に議論をすすめる。

入力をひらがな列 x とし、出力を単語列 $\mathbf{w}=w_1,w_2, ..., w_n$ とする。単語 w はそれぞれ読み Rw、表記 Hw、品詞 Pw をもつ。

推定には CRF 法を採用し、unigram 素性として Rw, Hw, Pw を、bigram 素性として Pw_iPw_{i+1} を使用する。

3. 区切りと表記の推定

前節のように本稿では形態素解析に類似したモデルで推定をおこなうが、ひらがな列を入力とするため形態素解析における文字種のように区切りの手がかりとなるものがすくない。そのため解空間が広くなり、推定がむずかしく計算量もおおくなってしまう。一方で、ユーザインタフェイスでは誤解析の訂正操作が区切り推定と表記推定のそれぞれについて別におこなわれることが一般的である。このとき、区切りの訂正結果が表記の再推定の制約として利用される。入力ひらがな列から直接漢字かな交じり文字列を推定するのではなく、まず区切り推定をおこない、その結果を制約として表記推定をおこなえばこのような操作体系にも合致し、解空間もせばめることが可能である。

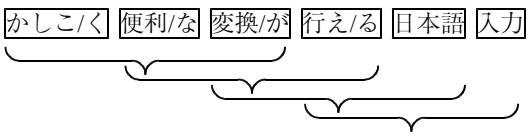

図1　3文節訓練事例の作成

区切り、表記推定を逐次おこなうモデルを 2 段階モデルとよび、以下のように定義する。

区切り推定はひらがな列 x を入力、単語ごとに区切られたひらがな列 **Rw** を出力とする。この推定では表記素性 Hw はもちいない。おなじ読み、品詞をもつ単語は同一のものとしてあつかう。

表記推定は **Rw** を入力とし、単語列 **w** を出力とする。あたえられた単語区切りはくつがえさない。区切り推定の結果として副次的に品詞情報 **Pw** がえられるが、このモデルでは表記推定の入力としていないので両者の品詞推定結果はくいちがうことがある。

4. 訓練事例の単位

かな漢字変換では対話的に処理をすすめるため、入力が文末まであたえられない状態で推定をおこなう必要がある。訓練事例に完全なかたちの文をあたえると実行時の推定と齟齬が起こる可能性がたかい。なるだけユーザがあたえる入力にちかいかたちで訓練事例を作成する必要がある。

ユーザの入力は一般的にいわゆる文節に相当するものを単位としているとかんがえられるが、1文節では文節間の関係が学習できないことから n 文節に分割する。n の値は実験により決定する。

すべての文節間の関係をとりこむために、1 文

1) 共起パスの複製

1-1) 中間ノード、後件単語ノードをコピー

1-2) 前件単語ノードから中間ノード、後件単語ノードの間にあるすべてのエッジをコピー

1-3) 後件単語ノードから後方にでるエッジをコピー

1-4) 後件単語ノードに共起素性を付与

2) 重複パスの除去

2-1) 前件単語ノードに隣接するコピー元中間ノードから前件単語ノードに接続するエッジを削除

2-2) 中間ノードをコピー

2-2) 中間ノードのもつエッジのうちコピー元後件単語ノードに接続しないエッジをコピー

図2 共起区間展開アルゴリズム

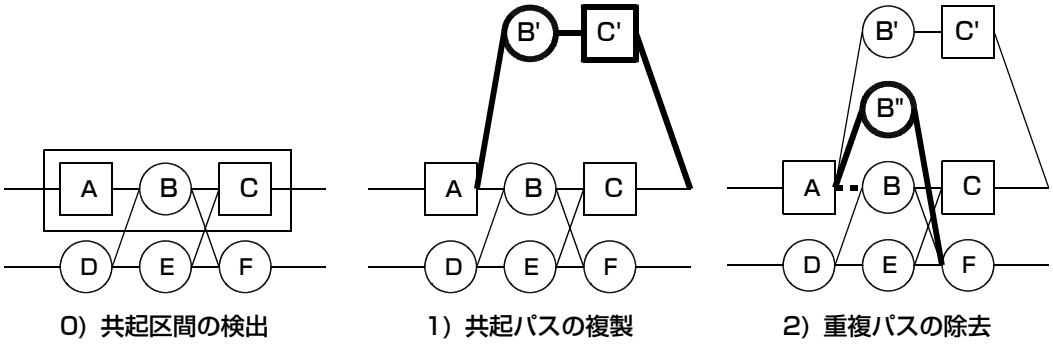

図3 共起区間展開アルゴリズムの例

単位の訓練事例にたいして n 文節幅のウィンドウを 1 文節ずつずらしながら適用することで n 文節単位の訓練事例を作成する(図 1)。

5. 非局所素性の導入

かな漢字変換における困難の 1 つに同音語がおおいことがあげられ、この曖昧性の解消には語の共起関係の利用が有効である。

共起関係にはさまざまなものがあるが、本モデルで利用する語の共起をつぎのものに限定する。

- 文内に共起する自立語 2 語のペア
- 自立語どうしの距離は n 文節以内

以下では共起ペアのうち前方にある語を前件単語、後方にある語を後件単語とする。

共起素性は n-gram でとらえられない非局所的な素性であり、モデルへの導入は計算の工夫が必要である。訓練時は解空間内の正解以外の解についても共起素性を付与する必要がある一方、解析時にはなるべくちいさな時間・空間計算量で処理する必要がある。そこで訓練時、解析時それぞれでことなる方法で共起素性を導入した。

5.1. 共起区間展開法

CRF 法では事例の学習に解空間すべての素性の期待値を数え上げる必要があり、この計算の効率的な方法として forward-backward アルゴリズムが利用される。計算効率をそこなわずに共起素性を導入するため、ラティス構造の拡張によって共起関係を表現する手法を提案する。

この手法では共起関係が成立した区間を部分的に共起が成立したパスとそれ以外のパスに分離し、共起成立パスにのみ共起素性をあたえることで非局所素性を表現する。共起パスは 1 つの前件単語ノードと 1 つの後件単語ノード、0 個以上の中間ノードからなる。

共起パスの分離は以下の図 2 の手順でおこなう。

この手法では共起が成立する区間の数や構造の複雑さによって計算量がおおきくなるが、前節でのべたように訓練事例を n 文節としているため影響は限定される。

5.2. 拡張Viterbi法

前接する全ノードの共起遷移リストの和をとる。ただし、同一内容のパスがあればコスト最小のパスのみをのこす。
if 現ノード n が前件単語ノード該当する
　　共起遷移リストにパス<n>を追加
else if 共起遷移リストが空でない
　　if　ノード n が共起遷移リスト内のパスの後件単語ノード
　　　　成立した共起のうち最小のパスを最適共起パスに記録
　　else
　　　　共起遷移リストの各パスにワイルドカードを追加

図 4 拡張 Viterbi 前向き探索操作

解析時は最尤解をもとめるために Viterbi アルゴリズムをもちいるが、共起素性をあつかうため以下のような拡張をおこなう。

ラティスの各ノードに最適共起パス情報と共起遷移リストを追加する。最適共起パスにはそのノードで成立した最小コストの共起パスを記録する。共起遷移リストにはそのノード以降に成立する可能性のある共起パスを列挙する。共起パスのうち中間ノードはワイルドカードとして記録する。

前向き探索は通常の Viterbi アルゴリズムに加えて図 4 の操作をおこなう。

最適経路選択は通常の Viterbi アルゴリズムとおなじく文末からノードに記録された最適前件をたどるが、最適共起パスが記録されていればその経路にそって遷移する。最適共起パス内にワイルドカードがあればノードに記録された最適前件を選択する。

以上の手順で比較的効率よく最尤解をもとめることができるが、入力が長くなるにつれて共起遷移リストがおおきくなる。現実的には共起の有効距離を限定し、共起遷移リスト内のパスがある程度の長さになれば破棄するなどの操作が必要である。

6. 評価実験と考察

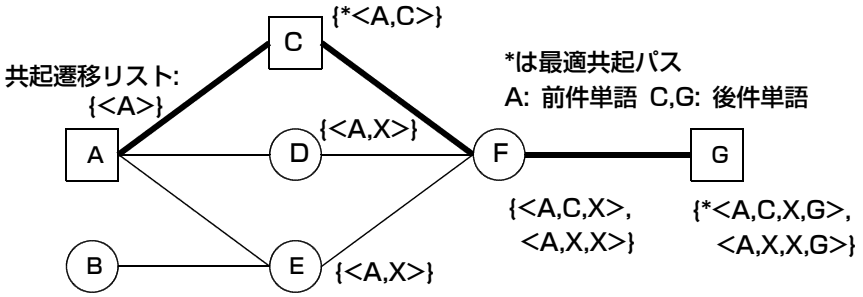

図5 拡張Viterbiアルゴリズムの例

精度評価は単語単位でおこない、とくにことわらないかぎり表記と品詞が正しく推定できれば正解とみなした。

6.1. 2段階モデル

区切り表記を一括で推定するモデル(同時モデル)と 2 段階推定モデルの比較をおこなった。訓練には機械的に形態素情報を付与した新聞コーパス数万文、評価には約 1,000 文のコーパスをもちいた。結果を表 1 に示す。2 段階モデルの方がよい精度がえられた。

また学習時間は同時モデルにくらべ 2 段階モデルは 1 割程度となった。

2段階モデルの区切り誤り

2 段階モデルでは区切り推定に失敗すると後段の表記推定では正解がえられない。また区切り推定時には表記情報がえられないことから、同時モデルとくらべ特徴的な区切り誤りが発生することが予測された。

実験の結果、2 段階モデルの区切り誤りのうち同時モデルで正解となっている事例は 7,985 件中 91 件であり、2 段階モデル固有の区切り誤りは誤解析の 1%程度であった。

以上のことから、2 段階モデル固有の区切り誤りは存在するものの精度にはおおきく影響しないといえる。

6.2. n文節ウィンドウ

この実験では 1,000 文のコーパスを 5 分割交差検定して使用した。訓練コーパスは 1 文単位のもの、1 文節ずつに分割したもの、ウィンドウ幅を 2 文節として分割したものの 3 種類を使用した。評価は 1 文単位と 1 文節単位でおこなった。学習には同時モデルをもちいた。結果を表 2 にしめす。

1 文と 1 文節で訓練単位と評価単位が合致しているときの精度がもっともよいが、合致しないときは極端にわるくなっている。2 文節ウィンドウではどちらの評価単位にたいしても比較的よい精度をたもっており、入力長にたいして頑健な手法であるといえる。

文節頭、文節末のあつかい

n 文節ウィンドウをもちいた場合、文頭と文節頭、文末と文節末がおなじものとしてあつかわれる。品詞 n-gram からみると文頭と文節頭にそれほど違いはないが、文末と文節末は性質の違いがあると予想された。

表 1　2 段階モデル精度(F 値)

	区切り	表記	品詞
同時モデル	0.977	0.938	0.932
2 段階区切り	0.979	-	(0.961)
2 段階表記	-	0.940	0.934

表 2　2 文節ウィンドウ精度(平均 F 値)

	1 文節評価	1 文評価
1 文学習	0.883	0.927
1 文節学習	0.929	0.777
2 文節学習	0.925	0.925

しかし、じっさいに文頭と文節頭、文末と文節末をそれぞれ区別して学習をおこなっても結果に有意差はなかった。

文頭と文節頭だけでなく文末と文節末の区別も考慮する必要はないということがわかった。

6.3. 共起素性の導入

実験では共起の有効距離を 2 文節内とし、3 文節ウィンドウのコーパスを訓練にもちいた。学習は同時モデルでおこなった。

共起素性がない場合とくらべ共起区間展開法を使用したときの学習時間は 1.08 倍であった。拡張 Viterbi 法をもちいた解析時間は共起素性を使用しないときとくらべ 2.67 倍となった。

また解析結果は共起素性導入前にくらべ誤り率が約 5%減少した。

誤り率の減少がすくないが、これは正解中の共起成立数がすくないことと共起成立箇所でも共起情報なしで正解がえられていることが原因である。

7. まとめ

本稿ではかな漢字変換の効率的な統計的モデルについて提案をおこなった。区切り、表記推定の分割、訓練事例の n 文節化、学習、解析への共起素性の導入によって、精度向上と処理の効率化が可能であることが確認できた。

本稿で提案した手法の一部は日本語入力システム「ATOK」の辞書学習に応用され、変換精度向上に寄与している。