N-gramとN-posのアンサンブルによるクラウド仮名漢字変換

李 陽　　陳 暁キン　　何 径舟　　黄 イン　　呉 闖
Baidu Inc.
{liyang10, chenxiaoxin, hejingzhou, huangjun, wuchuang}@baidu.com

1　はじめに

かな漢字変換は，形態素解析や機械翻訳などと同じく，シーケンス変換モデルに属す．とりあえずかな漢字変換の数学上の定義をする．

読みシーケンス y に対して、単語シーケンス x は以下の式て求められる．

$$x = \arg\max_{x} P(x|y)$$

ベイジアン変換によって

$$P(x|y) \propto P(y|x)P(x)$$

が得られる．

ここの $P(y|x)$ は発音モデルで，$P(x)$ は言語モデルである．

かな漢字変換は日本語や中国語などに独特の自然言語処理の応用であり、東アジア以外ではよく知られていない．よって，それについての研究が少ない．

また，今までの研究に最も多く使われる言語モデルは $N-gram$ と $N-POS$ の2つである．$N-POS$ に対しては mozc[2] や [3] などがあり，$N-gram$ に対しては Social IME[1] などがある．

$N-gram$ と $N-POS$ にそれぞれ長所と短所がある．本文は $N-gram$ モデルと $N-POS$ モデルを比較し，この2つのモデルのアンサンブル手法を提案し、評価した．

2　クラウド入力

サーバが発音文字列を受信し，サーバ側で計算し，かな漢字変換の結果をユーザに表示するシステムを，クラウド入力と言う．

PC で使われる今までの普通のかな漢字変換システムに対しては，かな漢字変換ができるだけ少ない資源を使用することが要求されるので，使用できるメモリと CPU などの資源は制限される．なので，辞書のサイズも，$N-gram$ モデル共起データのサイズも制限される．しかも，$N-gram$ モデル共起データを圧縮するほど精度が低くなる．

ただし，専用の高性能サーバの計算力を使って，変換結果をクライアント側に送信すれば，サーバのすべての計算資源は使用できるので，こういった問題がなくなる．

サーバ側の計算力を使うかな漢字変換システムの試みは，Social IME[1] や Baidu IME がある．

3　$N-gram$ と $N-POS$ の比較

$N-gram$ モデルは以下の式によって定義されている．

$$p(w_i = w|c) = p(w_i = w|w_{i-N+1}^{i-1})$$

ここの c は文脈で，w は単語を指す．$N-gram$ モデルは前の $(N-1)$ 個の単語の情報を全部持っていて，N 個目の単語が強く制約されるので，精度は高いが，大規模のコーパスが要求される．なお，数字や名前などはすべてコーパスに含まれないので，それらの単語について，$N-gram$ モデルの精度が低い．$N-POS$ モデルは単語を文法や機能構造でクラスに分類し，これらのクラスで，次の単語の出現確率を決める．すなわち，

$$\begin{aligned} &p(w_i = w|c) \\ &= p(g(w_i)|g(w_{i-N+1}^{i} - 1)) \cdot p(w_i = w|g(w_i)) \end{aligned}$$

ここに，$G = g_1, g_2, ..., g_t$ を単語クラスの集合とする．なお，単語が所属するクラスの出現確率を前 $N-1$ 個の単語のクラスに制約され，単語自身の出現確率は所属するクラスに制約されると仮定すれば，上の式が下

記のようになる．

$$p(w_i = w|c) \\ = p(g(w_i)|g(w_{i-N+1}^{i-1})) \cdot p(w_i = w|g(w_i) = g_j)$$

ただし，ひとつの単語が複製のクラスに所属することもあって，それを考えると，下記の式が得られる．

$$p(w_i = w|c) = \\ \sum_{g_j \subset G} p(g(w_i) = g_j|g(w_{i-N+1}^{i-1})) \cdot p(w_i = w|g(w_i) = g_j)$$

$N-POS$ モデルは単語ではなく，クラスで確率を計算しているので，コーパスの規模が大きくない場合も精度は出る．

つまり $N-gram$ はコーパスにある事例の多い単語に対しでは精確が，数字や名前など過疎な単語に対しては $N-POS$ モデルの方が精確である．

4 アンサンブル手法

本研究最初は $N-gram$ モデルと $N-POS$ モデルのアンサンブル手法を 2 つ提案し，評価した．そのうち，手法 1 は単に重み付けで結果をアンサンブルしている．手法 2 はより複雑で，$N-gram$ の $k-best$ の結果を $N-POS$ とアンサンブルし，並び順を変える．

以下はそれぞれの手法について説明をする．

4.1 手法 1

- グラフを作る．
- グラフのすべて可能のルートに対して $N-gram$ モデルと $N-POS$ を使って，確率を計算する．
- それぞれのルートに対して，$N-gram$ モデルで確率を計算する．
- それぞれのルートに対して，$N-POS$ モデルで確率を計算する．
- それぞれのルートに対して，$N-gram$ モデルの確率と $N-POS$ モデルの確率に重み付けて，最終の確率を計算する．
- 前のステップで計算し最終のた確率で $k-best$ の結果を決める．

4.2 手法 2

- グラフを作る．
- グラフのすべての可能のルートに対して $N-gram$ モデルを使って，確率を計算する．
- $k-best$ の結果を $N-gram$ モデルの確率で決める．
- $k-best$ のルートに対して，$N-POS$ モデルの確率を計算する．
- $k-best$ のルートに対して，$N-gram$ モデルの確率と $N-POS$ モデルの確率に重みを付けて，最終の確率を求める．
- この最終の確率で，$k-best$ の並び順を変える．

5 訓練コーパスと実験データ

訓練コーパスはクローリングしたウェブページの一部にした．

実験データは選出した 2112 句の文章．

6 実験結果

評価基準は完全一致率を用いている．実験結果は表 1 に示したように，アンサンブルにより，手法 1 と手法 2 の精度が $N-gram$ と $N-POS$ より高いから，手法の有効性を検証できた．この 2 つの手法のうち，手法 2 の精度が手法 1 より高い．

現在の Baidu IME のクラウド入力は手法 2 を用いている．

手法	正解数	正解率
N-gram	1402	66.38%
N-POS	1312	62.12%
手法 1	1431	67.75%
手法 2	1463	69.27%

表 1: 実験結果

7 おわりに

本論文では，$N-gram$ と $N-POS$ のアンサンブルアルゴリズムについて紹介した．

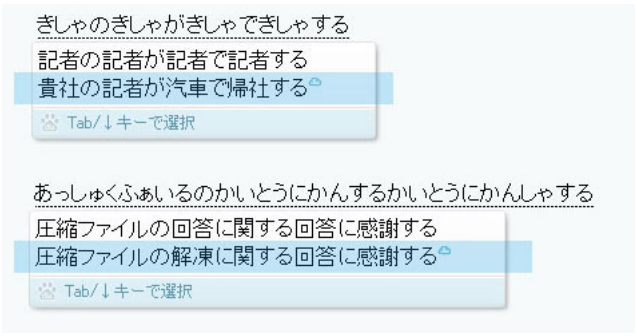

図 1: クラウド入力の例

図 2: オンライン IME の例

図 1 で示した Baidu IME の中で実装されるクラウド入力機能と図 2 で示した Baidu IME が新たに提供するオンライン IME サービスなどがこのアルゴリズムを用いている.

これからも精度を向上し，より精度の高い日本語入力システムを提供することを目指す.

参考文献

[1] 奥野陽, 萩原将文. インターネットを用いた日本語入力システム.

[2] 工藤拓, 小松弘幸, 花岡俊行, 向井淳, 田畑悠介. 統計的かな漢字変換システム mozc. 言語処理学会 第 17 回年次大会 発表論文集, pp. 948–951, 2011.

[3] 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, jul 1999.