単語と入力記号列の組の1-gram(ゆにぐらむ)モデルによる統計的仮名漢字変換

森 信介
http://plata.ar.media.kyoto-u.ac.jp/mori/

1 除論

統計的仮名漢字変換の本質を理解するためのフリーウェアー SIMPLE (Statistical Input Method for Personal Learning and Education; 仮) について説明する。統計的仮名漢字変換を最初に提案している文献 [1] では、単語 (表記) を単位とする言語モデルを提案している[1]。本論文では、実装がより簡単な、単語と入力記号列の組を単位とする 1-gram モデルによる方法について説明する。

2 統計的仮名漢字変換

仮名漢字変換は、キーボードから直接入力可能な入力記号 (読み) $\mathcal{Y}$ の列 $\boldsymbol{y} \in \mathcal{Y}^+$ を入力とし、日本語の文字 $\mathcal{X}$ の列 $\boldsymbol{x} \in \mathcal{X}^+$ を変換結果として提示する。統計的手法では、変換結果の選択にコーパスなどに対する統計量を用いる。本論文では、これを雑音のある通信路モデルで定式化する。モデル化においては、ユーザーの直感や入力記号列付与の観点から、文を単語列 $\boldsymbol{w} = w_1 w_2 \cdots w_h$ とみなす。

2.1 動作原理

雑音のある通信路モデルでは、雑音を含む観測 $\boldsymbol{y}$ から通信路の入力 $\hat{\boldsymbol{w}}$ を以下のように推定する[2]。

$$\hat{\boldsymbol{w}} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{y}) \tag{1}$$

本論文では、統計的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。その上で、以下の式のように $P(\boldsymbol{w}|\boldsymbol{y})$ をモデル化する。

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

この式を式 (1) に代入し、分母 $P(\boldsymbol{y})$ が出力によらないことに留意して式変形し、以下の式を得る。

$$\hat{\boldsymbol{w}} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}} \frac{P(\boldsymbol{u})}{P(\boldsymbol{y})} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}} P(\boldsymbol{u})$$

この式の $P(\boldsymbol{u})$ を以下のように単語と入力記号列の組の 1-gram モデルでモデル化する。

$$P(\boldsymbol{u}) = \prod_{i=1}^{h+1} P_1(u_i)$$

[1] 文字ベースのはある。ユーザーの直感とある程度合致する単語を単位とするのはこれが初めて。
[2] $\mathbf{argmax}_x f(x)$ は $f(x)$ が最大となる x を返す。

この式の中の u_{h+1} は、文末に対応する記号 BT である。一般的に $P(u)$ の値は、コーパスから推定される。コーパスに出現する組のみの列に変換できない入力に対しても妥当な変換候補を出力するために、$P(u)$ を以下のように u が既知組 ($u \in \mathcal{U}$) か否かに応じて場合けする。

$$P_1(u_i) = \begin{cases} P(u_i) & \text{if } u_i \in \mathcal{U} \\ P(\mathtt{UU})M_u(u_i) & \text{if } u_i \notin \mathcal{U} \end{cases} \tag{2}$$

この式の UU は未知語記号であり $M_u(u) = M_u(\langle w, \boldsymbol{y} \rangle)$ は未知語モデルである。大きな学習コーパスを用いれば既知組 $\mathcal{U}$ は十分大きくすることができ、未知語率は極めて低なる。したがって、未知語モデルとして、入力記号 $\mathcal{Y}$ の 0-gram モデル $M_{y,0}(\boldsymbol{y})$ を用いる[3]。

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

入力記号 0-gram モデルは、入力記号列 $\boldsymbol{y} = y_1 y_2 \cdots y_{h'}$ と単語末記号 BT を 1 文字ずつ一様分布を用いて生成する。

$$M_{y,0}(\boldsymbol{y}) = \prod_1^{h'+1} \frac{1}{|\mathcal{Y} \cup \{\mathtt{BT}\}|} = \prod_1^{h'+1} \frac{1}{|\mathcal{Y}|+1} = (|\mathcal{Y}|+1)^{-(h'+1)}$$

以上から、単語と入力記号列の組の 1-gram モデルによる統計的仮名漢字変換は、以下の式のようになる。

$$\begin{cases} \hat{\boldsymbol{w}} = \mathbf{argmax}_{\boldsymbol{w}} P(\boldsymbol{u}) \\ P(\boldsymbol{u}) = \prod_{i=1}^{h+1} P_1(u_i) \\ P_1(u_i) = \begin{cases} P(u_i) & \text{if } u_i \in \mathcal{U} \\ P(\mathtt{UU})(|\mathcal{Y}|+1)^{-(h'_i+1)} & \text{if } u_i \notin \mathcal{U} \end{cases} \end{cases} \tag{3}$$

ここで $h = |\boldsymbol{w}|$, $u_{i+1} = \mathtt{BT}$, $h'_i = |\boldsymbol{y}_i|$ である。

2.2 パラメーター推定

各 $u = \langle w, \boldsymbol{y} \rangle$ に対する式 (2) の $P(u)$ の実際の値は、タグ付きコーパスから推定する。タグ付きコーパスの各文は、単語に分割され、各単語に入力記号列が付与されている必要がある。以下に例を示す。

> テキスト/てきすと 解析/かいせき 器/き ＫｙＴｅａ/ＫＹＴＥＡ で/で 解析/かいせき
> す/す る/る

まず、コーパス全体を走査し、一部の組を未知組記号に置き換える。また、文末に BT を加える。この結果、上述の例は以下のようになる。

> テキスト/てきすと 解析/かいせき 器/き UU で/で 解析/かいせき す/す る/る BT

次に各組の頻度と総頻度を計数する。上述の例では以下のようになる。

> f(テキスト/てきすと) = f(器/き) = f(UU) = f(で/で) = f(す/す) = f(る/る) = f(BT) = 1
> f(解析/かいせき) = 2

最後に、確率 $P(u)$ を以下の式を用いて最尤推定する。

$$P(u) = \frac{f(u)}{\sum f(u)} \tag{4}$$

[3] 0-gram モデルは、すべての記号が一様な確率で出現するモデルで、この呼び名は、n-gram モデル ($n \geq 1$) との整合性を考慮した便宜的なものである。

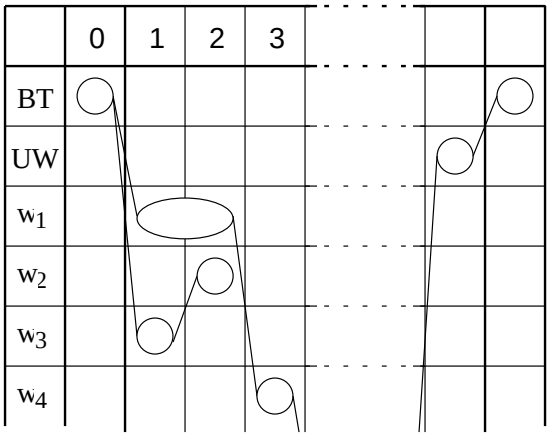

図 1: 動的計画法による解探索

上述の例では以下のようになる。

$$P(\text{テキスト}/\text{てきすと}) = P(\text{器}/\text{き}) = P(\texttt{UU}) = P(\text{で}/\text{で}) = P(\text{す}/\text{す}) = P(\text{る}/\text{る}) = P(\texttt{BT}) = 1/9$$
$$P(\text{解析}/\text{かいせき}) = 2/9$$

上述の処理から分かるように、信頼性のあるパラメータを推定するために、単語に分割されかつ入力記号列が付与された文が大量に必要である。機械可読のテキストは容易に入手可能であるので、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とも言える。この問題は、KyTea (`http://www.phontron.com/kytea/`) により解決される。

3 解探索

最尤の組の列は、動的計画法 [2] を用いることで入力の長さに比例する時間で探索することができる。本節では、まず辞書引きについて説明し、次に解探索の詳細について述べる。

3.1 辞書引き

辞書は、以下のように、入力記号列を受け取り、対応する組の列を返す。

$$\boldsymbol{y} \mapsto (u_1, u_2, \cdots) = (\langle w_1, \boldsymbol{y}\rangle, \langle w_2, \boldsymbol{y}\rangle, \cdots)$$

列になるのは、同音異義語などのように、同じ入力記号列に対して複数の変換候補があるからである。また、組の入力記号列の部分は、辞書引きの際の入力と同じなので本質的には不要であるが、言語モデルの単位が組であるので、入力記号列を含む組を返すこととする。

3.2 解探索

式 (3) から分かるように、組の列の生成確率には加法性がある[4]。したがって、解探索には動的計画法を用いる。これには、以下の 2 通りの方法がある。

1. ある入力記号位置から始まる組を列挙する
2. ある入力記号位置で終わる組を列挙する

今後、辞書引きを AC 法や DFA 法で実現することを念頭に、後者を採用する。
あとは図 1 を参照して推して知るべし (すみません、書きます)。

[4] 熱帯半環になっている。

表 1: コーパス

	文数	単語数	文字数
学習	239	?,???	?,???
テスト	26	???	???

表 2: 各モデルの変換精度

言語モデル	学習コーパス	適合率	再現率	文正解率
組の 1-gram モデル	MPT	91.4%	93.0%	30.8%
単語 2-gram モデル	BCCWJ etc.	97.3%	97.4%	57.2%
組の 2-gram モデル	BCCWJ etc.	97.5%	97.5%	58.6%

下の 2 つはテストコーパスが異なり、あくまでも参考値である。

4 評価

この節では、実際のデータを用いた実験の結果を提示し、本論文で提案する単語と入力記号列の組の 1-gram モデルによる統計的仮名漢字変換の評価を行なう。

4.1 言語資源

実験に用いたコーパスは、MPT(Mori's Paper Corpus) である[5]。各文は、単語に分割され、各単語には入力記号列が付与されている。コーパスは 10 個に分割され、このうちの 9 個からパラメータを推定し、残りの 1 個に対してテストした。学習コーパスとテストコーパスの大きさは表 1 の通りである。

4.2 評価基準

我々が用いた評価基準は、各文を一括変換することで得られる最尤解と正解の最長共通部分列 (longest common subsequence)[4] の文字数に基づく再現率と適合率である。正解コーパスに含まれる文字数を N_{COR} とし、仮名漢字変換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。例として、コーパスの内容と変換結果が以下のような場合を考える。

コーパス

私 が 長 尾 真 で す 。

変換結果

渡 し が 長 尾 マ コ ト で す 。

この場合、最長共通部分列は「が長尾です。」の 6 文字であるので、$N_{LCS} = 6$ となる。コーパスに含まれる文字数は 8 であり、変換結果に含まれる文字数は 11 であるので、$N_{COR} = 8$, $N_{SYS} = 11$ である。よって、再現率は $N_{LCS}/N_{COR} = 6/8$ となり、適合率は $N_{LCS}/N_{SYS} = 6/11$ となる。

4.3 評価

頻度が 2 以上の組 (432 個) を既知の組とた。この結果、学習コーパスのカバー率は 96.8%であった。このようにして得られた仮名漢字変換器によりテストコーパスの入力記号列を文ごとに一括変換した。表 2 は、変換結果である。

[5]MPT(えんぷてぃー) は、`http://plata.ar.media.kyoto-u.ac.jp/mori/research/` からダウンロードできる。

簡便さを追求した手法の割にはまあまあである。

5 さらなる改善のためのアイデア

提案手法には非常に多くの改良の余地がある。以下では、これらを精度向上と実装上の工夫に分けて概説する。

5.1 精度向上

- 大規模コーパス
  - 正確な入力記号推定 `http://www.phontron.com/kytea/`
  - 確率的単語分割・入力記号列付与 [5]
- 長い履歴の参照
  - 高次の n-gram モデルの利用 [5]
  - 連語獲得と連語 2-gram モデル [7]
- 動的適応
  キャッシュ&トリガーモデル [8]

5.2 実装上の工夫

実装上の工夫は、主に速度の向上と記憶域の削減である。

- 組のクラスタリングとクラス 2-gram モデル (精度向上というよりは記憶域の削減) [1][6]
- 辞書引き
  日本語のように単語境界のない言語の文の辞書引きは、ある長い文字列に対する複数文字列の検索と本質的に同じで、AC 法に基づいた辞書引き [9] を用いるのが効率が良い。さらに速度が必要な場合は、AC 法の失敗関数を展開することで得られる決定性オートマトン (DFA)[3] を用いるのがよい。
- 整数演算による確率計算の実現

5.3 機能拡張

- なんといっても予測の定式化かな

6 欠論

本論文では、単語と入力記号列の組の 1-gram モデルによる統計的仮名漢字変換について説明したが、特に論じることはなかった。

A プログラムとの対応

本論文で説明した仮名漢字変換エンジンのプログラムは、`http://plata.ar.media.kyoto-u.ac.jp/mori/research/topics/KKC/` からダウンロード可能である。本節では、プログラムと本文中の数式などとの対応について説明する。

A.1 プログラムを通じて共通の変数

`$word` : w (例: "京大")
`$kkci` : $\boldsymbol{y}$ (例: "きょうだい")
`$pair` : u (例: "京大/きょうだい")

A.2 モデル作成により確定する変数

`@KKCInput` : $\mathcal{Y}$
`%PairFreq` : $u \mapsto f(u)$ (`keys(%PairFreq)` $= \mathcal{U}$)
`$Freq` : $\sum f(u)$
`%Dict` : $\boldsymbol{y} \mapsto (u_1, u_2, \cdots)$

A.3 解探索時の変数

`$posi` : 解析文字位置 (辞書引きの右端)
`$from` : 辞書引きの左端 (当該ノードの左端)
`$POSI` : 入力の長さ (tt $posi の上限)
`@VTable` : 動的計画法の表
`@best` : 表のある箇所の最良のノード (一時変数)
`$logP` : ノードの生成確率の負対数値
`$node` : 後ろ向き探索のときのノード (`@Vtable` の要素)

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] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. アルゴリズムイントロダクション 第 2 巻 アルゴリズムの設計と解析手法. 近代科学社, 1995.

[3] 森信介. Dfa による形態素解析の高速辞書検索. EDR 電子化辞書利用シンポジウム, 1997.

[4] Alfred V. Aho. 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, pp. 263–304. Elsevier Science Publishers, 1990.

[5] 森信介, 笹田鉄郎, NEUBIG Graham. 確率的タグ付与コーパスからの言語モデル構築. 情報処理学会研究報告, 第 NL196/SLP81 巻, 2010.

[6] 森信介, 西村雅史, 伊東伸泰. クラスに基づく言語モデルのための単語クラスタリング. 情報処理学会論文誌, Vol. 38, No. 11, pp. 2200–2208, 1997.

[7] 森信介, 山地治, 長尾真. 予測単位の変更による n-gram モデルの改善. 情報処理学会研究報告, 第 SLP19 巻, pp. 87–94, 1997.

[8] Roland Kuhn and Renato de Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, pp. 570–583, 1990.

[9] Hiroshi Maruyama. Backtracking-free dictionary access method for japanese morphological analysis. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 208–213, 1994.