利用過程で得られる言語情報を活用する音声言語処理システム

森 信介 (京都大学), 前田 浩邦 (京都大学)

2009年10月1日

音声言語処理の学習データの収集

- 音声言語処理の統計的アプローチ
  アルゴリズム と データ の分離に成功

- 研究パターン

  1. コストをかけてを正解データを準備
  2. 機械学習の適用
  3. 結果の (比較) 評価

- データをだれがどのように用意するのか?

  1. 使われる音声言語処理システムを作成・配布
  2. ユーザーとのインタラクションを記録

使われる音声言語処理

- とにかく**普通のユーザーに**使われる物を作るべし
  品詞や構文木を知りたいユーザーなどいない

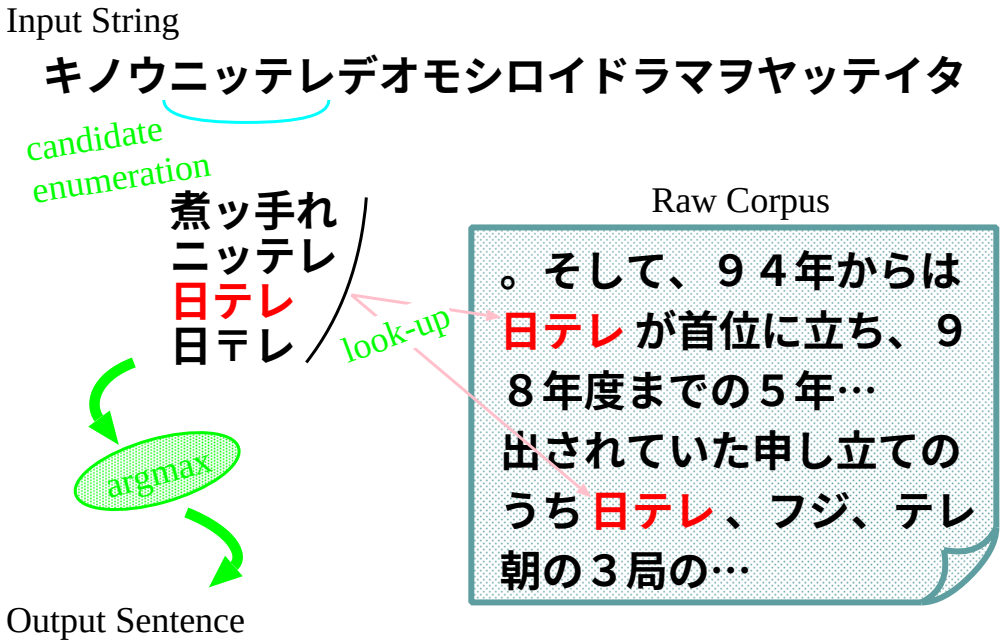
IBM 時代の上司曰く
構文解析の精度が 5%上がって
どんなビジネスインパクトが…

- 使ってもらえればデータも集まる

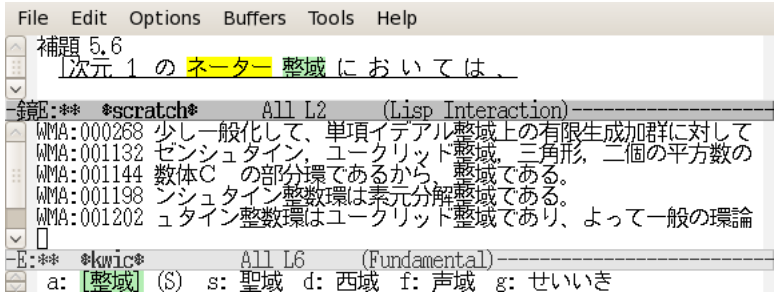
新機能付き仮名漢字変換の配布 (第１弾)

1. 商品と比較しても遜色ない完成度
   - 十分な変換性能
   - 快適なインターフェイス
2. 研究を活かした便利な新機能

テキストの部分文字列も候補に挙げる仮名漢字変換



実稼働中 (私は使ってます)



⇐ 執筆中の文章

⇐ 部分文字列候補の KWIC (任意)

⇐ 変換候補

⇖ テキストの部分文字列 (未知語)

1. ベース言語モデル
   - 98,450 語 (UniDic ∪ 読み付きコーパス)
   - 単語 2-gram モデル (単語分割済みコーパス + 新聞記事)
2. 適用対象分野の生テキスト
   - Wikipedia の数学関連のページ
   - 擬似確率的単語分割 (倍率 8)

ログ

1. 19130 52033 ⇐ **時間 (epoch time)**
2. １７/１７/IN ⇐ **最尤解**
3. １７/１７/IN ⇐ **ユーザーの選択**

---

1. 19130 52043
2. 正規/せいき/IN の/の/IN 後半/こうはん/IN に/に/IN
3. 世紀/せいき/IN の/の/IN 後半/こうはん/IN に/に/IN

---

1. 19130 60848
2. 開/ひら/IN い/い/IN た/た/IN 類体/るいたい/IV 論/ろん/IN
3. 開/ひら/IN い/い/IN た/た/IN 類体/るいたい/IV 論/ろん/IN

⇖ **Wikipediaの部分文字列**

---

1. 19130 52925
2. 非/ひ/IN 可換/かかん/IV 類/るい/IN 体/たい/IN 論/ろん/IN
3. 非/ひ/IN 可換/かかん/IV 類/るい/IN 体/たい/IN 論/ろん/IN

⇖ **単語の定義に合わず**

さまざまな適応分野

ID	出所	文字数
WMA	Wikipedia 数学	6,777,849
MPT	1996 年〜2008 年に森信介が書いた 国内研究会の論文のテキスト (LaTeX)	109,782
WWW	World Wide Web の一部	29,194,122
USR	あるユーザーの Web browser & Mailer	??
NLP	自然言語処理 & 年次大会 & YANS	??
BMA	岩波「現代数学の基礎」全 23 冊の OCR 結果	??
TRL	ある企業の部署でのメールのログや会議の資料	??

- **音声言語処理でのログの活用** (やりたい人はどうぞ)
  仮名漢字変換, 音声認識, 単語分割, 読み推定, 機械翻訳, ...
- **OCR の結果でも動作するのか**

確率的言語モデルによる仮名漢字変換 [森 ほか 1998]

- 入力記号列 y に対応する文字列 x を列挙

$$im(y) = (x_1,\ x_2,\ \cdots)$$

確率の降順に

$$\begin{aligned} i \leq j \ &\Leftrightarrow\ P(x_i|y) \geq P(x_j|y) \\ &\Leftrightarrow\ P(y|x_i)P(x_i) \geq P(y|x_j)P(x_j) \end{aligned}$$

1. $P(x)$: **言語モデル** (日本語文字列の傾向を記述)

   P(我輩は猫である) $>$ P(我が背は猫である)

2. $P(y|x)$: **仮名漢字モデル** (読みの曖昧性を記述)

   P(ハイ | 背), P(セ | 背)

単語 n-gram モデル (言語モデル)

- 文を単語列 $w_1^h = w_1 w_2 \cdots w_h$ と見なし文頭から順に予測

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

- 語彙 $\mathcal{W}$ に含まれない単語 (未知語) の予測
  1. 単語 n-gram モデルにより未知語記号 UW を予測
  2. 表記 (文字列)$x_1^{h'}$ を文字 n-gram モデルにより予測

$$\begin{cases} P(w_i | w_{i-n+1}^{i-1}) & = & M_{x,n}(w_i) P(\mathtt{UW} | w_{i-n+1}^{i-1}) \\ M_{x,n}(x_1^{h'}) & = & \prod_{i=1}^{h'+1} P(x_i | x_{i-n+1}^{i-1}) \end{cases}$$

- パラメータは単語分割済みコーパスから推定

仮名漢字モデル

- 入力記号列と日本語文との確率的対応関係を記述
- 単語と入力記号列との対応関係が単語ごとに独立であると仮定
- 単語列 w が与えられたときの入力記号列 y の出現確率

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

※ 単語 w_i に対応する入力記号部分列 y_i は以下の条件を満たす

$$y = y_1 y_2 \cdots y_h$$

- $P(y_i|w_i)$ は単語ごとに入力記号列が付与されたコーパスから推定

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

テキストの部分文字列も候補に挙げる仮名漢字変換

[森 ほか 2007]

1. 仮名漢字モデル

- 各文字に対応する入力記号列が一様に出現すると仮定

$$\mathcal{Y}_{抗} = \{\text{コウ}, \text{アラガ}\}$$
$$\mathcal{Y}_{体} = \{\text{タイ}, \text{テイ}, \text{カラダ}\}$$

単漢字辞書

$$P(\text{コウタイ} \mid \text{抗体}) = \frac{1}{|\mathcal{Y}_{抗}|}\frac{1}{|\mathcal{Y}_{体}|} = \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$$

2. 生テキストからの言語モデル構築

$\Rightarrow$ $\Rightarrow$ $\Rightarrow$

言語モデル構築

- テキストの確率的単語分割

  … 単$_{0.2}$ 項$_{0.9}$ イ$_{0.1}$ デ$_{0.1}$ ア$_{0.1}$ ル$_{0.9}$ 整$_{0.3}$ 域$_{0.3}$ 上$_{0.9}$ の …

  ※分割確率は MaxEnt で推定 (要 単語分割済みコーパス)

- 期待頻度

$$
\begin{aligned}
& f(\text{イデアル}, \text{整域}) \\
&= 0.9 \times (1 - 0.1)^3 \times 0.9 \times (1 - 0.3) \times 0.3 \\
&= 0.1240029
\end{aligned}
$$

- 単語 n-gram 確率を計算

$$
P(\text{整域} \mid \text{イデアル}) = \frac{f(\text{イデアル}, \text{整域})}{f(\text{イデアル})}
$$

疑似確率的単語分割コーパス

確率的単語分割コーパス + 乱数
= 揺れのある単語分割済みコーパス ($\times$ M)
$\approx$ 確率的単語分割コーパス

例) 単$_{0.2}$ 項$_{0.9}$ イ$_{0.1}$ デ$_{0.1}$ ア$_{0.1}$ ル$_{0.9}$ 整$_{0.3}$ 域$_{0.3}$ 上$_{0.9}$ の

試行	結果
1	単 項 / イ デ ア ル / 整 / 域 上 / の
2	単 項 / イ デ / ア ル / 整 域 / 上 / の
3	単 項 / イ デ ア ル / 整 域 上 / の
4	単 / 項 / イ デ ア ル / 整 域 / 上 / の

- あとは通常の単語分割済みコーパスと同じ

課題1. 様々な情報を統合的に扱う自動単語分割器の学習

- **不完全な単語分割済みコーパスからの学習** [COLING08]

  再-発|す　る|と|細　気　管　支|が|傷-害|さ-れ|ま-す|。

- **単語列・複合語リスト** [NL-193, PACLING09]

  |動-脈|管|開-存|症|，|クレアチンキナーゼＭＢ|

- **能動学習** (まったく手つかず)

課題2. 変換ログの活用 (やりたい人はどうぞ)

※変換ログはノイズありの文断片 (単語分割済み, 読み付き)

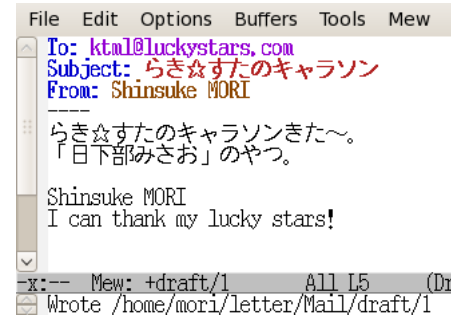
- 仮名漢字変換, 音声認識 (ex. 講義の音声認識)
  とりあえず変換ログからのモデル構築
- 読み推定　for 音声合成
  精度向上の実現は簡単!?
- 単語分割
  たぶん何とかなる (ノイズをどう扱うか)
- 機械翻訳, 誤り訂正
  どうすればいいのか (差分は大きくないか)
- 他に何かある?

他の音声言語処理システムの配布&ログの回収

課題 3. Firefox Plugin 等の開発

– Hélas! Et j'ai lu toutes les pages –

1. Browserがテキストを自動的に集収
2. 辞書と言語モデルを随時更新



ATOK: らキス多
MS-IME: らキス他
Anthy: 良きすた
kagami(仮): らき☆すた