共起性を考慮したかな漢字変換

2016/06/10 京大

山形頼之

序論

- 目標設定
- 誤変換の理由を考える
- かな漢字変換の歴史
- 現在の変換エンジンの手法

目標

- 誤変換をなくす

  碁で人間に勝つくらいなんだし、誤変換もいい加減やめて欲しい

- 個人的動機：機械学習、自然言語処理の学習

誤変換博覧会

- 大丈夫、走って変えるよ
- 深夜虹にはねます
- 女装して飛べ、あの彼方へ
- 個々にしか駐車できねえの
- あ、顔に住み着いちゃった
  - http://www.dfnt.net/t/photo/your/gohenkan.shtml, R25より改変、Google日本語入力による

考察

- 「文脈」の理解不足が大きな理由
  - 言語内の文脈と言語外の文脈がある
    - 言語内の文脈←共起の概念で捉えられないか
- 品詞しか見ていない
- 前方の文節が長くなろうとする傾向

かな漢字変換略史

- 神代(～1980年代)
  - 言語学に基づくルールによる変換
    - Canna, Wnn, ATOK, MS IME, ことえり…
- パラダイムシフト(1990年代)
  - 統計的手法の登場、アルゴリズムとデータの勝負に
  - Anthy, Google日本語入力…

かな漢字変換エンジン紹介

- Google日本語入力：品詞bigram+単語unigram
- MS IME：単語(?)trigram
- Apple：？
- ATOK：統計的手法とルールベースの併用
- SKK：単文節変換
- 関連するものとして、中文の入力エンジン、モバイル環境での誤り訂正付き入力エンジン(Swiftkey)等がある

共起性を扱う理論

- 先行研究
- 言語モデルの作り方
- 最大エントロピー法

先行研究

- Rosenfeld, R., & Roukos, S. (1993). Trigger-based language models: a maximum entropy approach

- 高岡一馬, 内田佳孝, & 松田. (2011). 非局所素性を用いたかな漢字変換

言語モデル

$$
\begin{aligned}
&P(w_1, \ldots, w_n) \\
&= P(w_1, \ldots, w_{n-1}) P(w_n \mid \{w_1, \ldots, w_{n-1}\}) \\
&= P(w_1 \mid \emptyset) \cdots P(w_n \mid \{w_1, \ldots, w_{n-1}\})
\end{aligned}
$$

どうやって $P(w_i \mid \{w_1, \ldots, w_{i-1}\})$ を推定するか

最大エントロピー法

特徴 $f_1, f_2, \ldots, f_m$ から出力 x を推定する

$P(x, f_i)$が実測値になるという条件で

$P(x)$のエントロピーを最大化する

$$P(x) = \prod_{i=1}^{m} \mu_i^{f_i(x)}$$

$\mu_1, \ldots, \mu_m$ に関する最大化問題

実装

- maxent (by Le Zhang)を使用
  - ただしscikit-learnに移行中。
- 現状は単語の表層表現をそのまま特徴として使う
- 01.words…10.wordsで学習
  - σ^2でスムージング
  - 一晩かかった(Mac Pro 2013モデル)
  - メモリ使用：学習１G, 評価140M

実行結果

走 っ て 帰 る	2.71402906882E-23
走 っ て 変え る	2.64961012244E-23
大胆 に 変え る	1.11609770297E-18
大胆 に 帰 る	1.11607272834E-18
歯 が とても 痛 い	2.46806717400E-23
葉 が とても 痛 い	2.40126293570E-23
葉 が 青々 と 茂 っ て い る	1.79972188265E-37
歯 が 青々 と 茂 っ て い る	1.85372167091E-37
家 に 住 み 着 い ちゃ っ た	7.38935880401E-40
家 に 墨 つ い ちゃ っ た	1.57282372774E-35

今後の計画

- 次回までのアクション(１ヶ月程度)
- 中期的なアクション(半年〜１年)
- リリースまでに必要なこと(２年くらい)

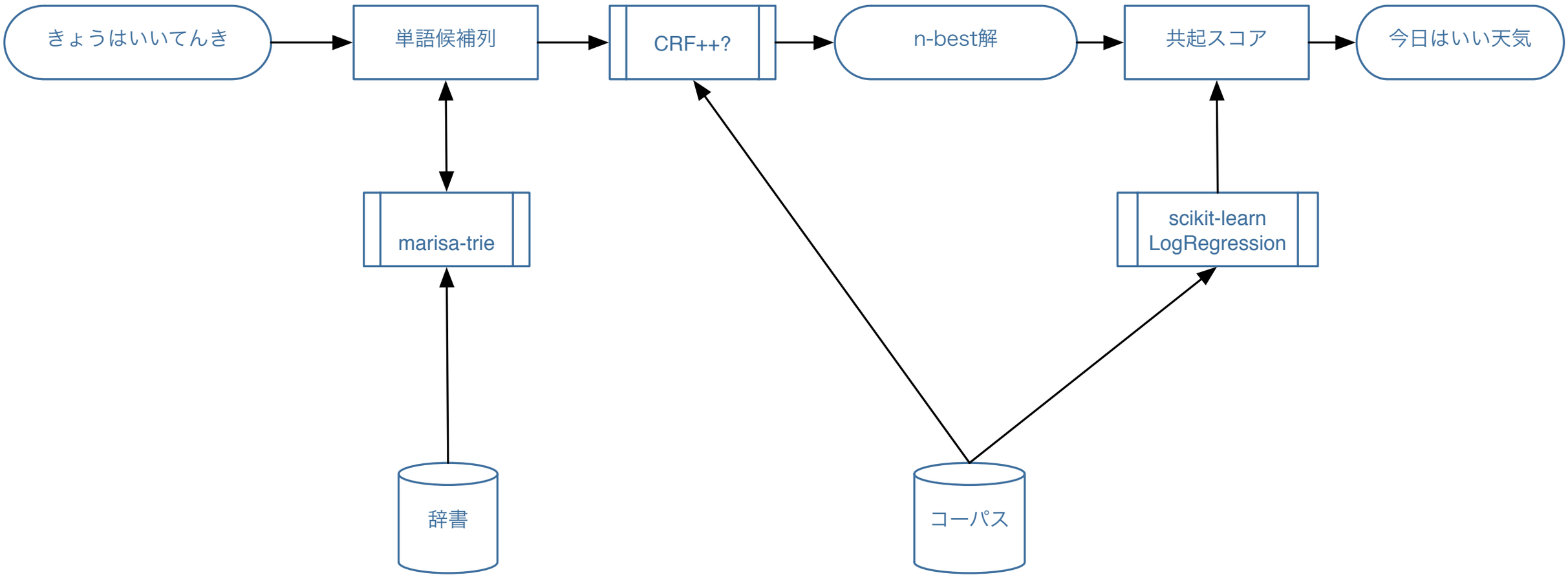
次回までのアクション

- 特徴の圧縮、出力のクラスタリング
  - 特徴ベクトル生成：hashing trick
  - 圧縮：LSA
  - 単語クラスタリング：hierarchical clustering
- 品詞bi/tri-gramモデルとの統合
- 論文読み

中期的なアクション

- 未知語(コーパスに現れない語)モデル
- 変換エンジンを作る
  - CRF
- 評価する
  - BLUE？Base-lineをどうするか？
- 論文を書く

変換エンジン(構想)



pythonで全部書く

リリースまでに必要なこと

- 予測変換(？)
- 学習(cache model?)
- 大規模コーパス
  - KyTea等による自動単語分割、品詞タグ付けの利用
- クライアント(特にMac)
  - オープンソースの変換エンジンはMozc以外壊滅している