自然言語処理論 I

8. 形態素解析(英語の品詞のタギング)

英語の形態素解析

- 単語を形態素に分割する
  - unkindly = un + kind + ly
- 品詞を決める

- 英語は多品詞語が多い
  - 名詞のほとんどは動詞でもある
- 品詞の決定が重要な問題

品詞付け(品詞タギング)

- part-of-speech tagging
- 単語辞書をもとに各単語に品詞の候補を与える
- 優先順位付けの方法
  - 品詞の出現のしやすさに関する優先度
    - ✦ breakfast: 名詞 or 動詞
    - ✦ 名詞の方が出現しやすい
  - 品詞並びに関する優先度
    - ✦ the(冠詞) breakfast(名詞 or 動詞)
    - ✦ 冠詞の後には名詞が出現しやすい

優先度の与え方

- 人手で与える方法
- 自動的に与える方法
  - 英語の品詞付けでは確率モデルを用いた方法が昔からよく行われている

品詞タグ付けのための確率モデル

- $P(C_1,...,C_n \mid w_1,...,w_n)$
  - C_iは品詞，w_iは単語
  - この確率が最大となる$C_1,...,C_i$を求める
- ベイズの定理より

$$P(C_1,...,C_n|w_1,...,w_n) = \frac{P(C_1,...,C_n)P(w_1,...,w_n|C_1,...,C_n)}{P(w_1,...,w_n)}$$

- 分子が最大となる$C_1,...,C_n$を求めればよい

例

- 単語列: Time flies like an arrow

・品詞列: n v prep det n

$$P(n\ v\ prep\ det\ n \mid time\ flies\ like\ an\ arrow)$$
$$= \frac{P(n\ v\ prep\ det\ n)P(time\ flies\ like\ an\ arrow|n\ v\ prep\ det\ n)}{\cancel{P(time\ flies\ like\ an\ arrow)}}$$

・品詞列: n n v det n

$$P(n\ n\ v\ det\ n \mid time\ flies\ like\ an\ arrow)$$
$$= \frac{P(n\ n\ v\ det\ n)P(time\ flies\ like\ an\ arrow|n\ n\ v\ det\ n)}{\cancel{P(time\ flies\ like\ an\ arrow)}}$$

※ 確率の大きい品詞列を選ぶ

確率モデルの近似

- $P(C_1,...,C_n)P(w_1,...,w_n|C_1,...,C_n)$
- 第1項

$$\begin{aligned} P(C_1,...,C_n) &= P(C_1) \times P(C_2|C_1) \times P(C_3|C_1C_2) \\ &\quad \times ... \times P(C_i|C_1C_2...C_{i-1}) \\ &\quad \times ... \times P(C_n|C_1C_2...C_{n-1}) \\ &\simeq P(C_1)\prod_{i=2}^{n} P(C_i|C_{i-1}) \end{aligned}$$

- 第2項

$$P(w_1,...,w_n|C_1,...,C_n) \simeq \prod_{i=1}^{n} P(w_i|C_i)$$

確率モデルの近似

- まとめると

$$\begin{aligned} &P(C_1,...,C_n)P(w_1,...,w_n|C_1,...,C_n) \\ &= \prod_{i=1}^{n} P(C_i|C_{i-1}) \prod_{i=1}^{n} P(w_i|C_i) \end{aligned}$$

  - $P(C_1) = P(C_1 \mid C_0)$
    $(= P(C_1 \mid \phi))$
  - C_0(またはϕ)は文頭を表わす特別なシンボル

例(Time flies like an arrow)

・n v prep det n

$$P(n\ v\ prep\ det\ n) \simeq P(n|\phi)P(v|n)P(prep|v)P(det|prep)P(n|det)$$

$$P(time\ flies\ like\ an\ arrow|n\ v\ prep\ det\ n) \simeq P(time|n)P(flies|v)P(like|prep)P(an|det)P(arrow|n)$$

・n n v det n

$$P(n\ n\ v\ det\ n) \simeq P(n|\phi)P(n|n)P(v|n)P(det|v)P(n|det)$$

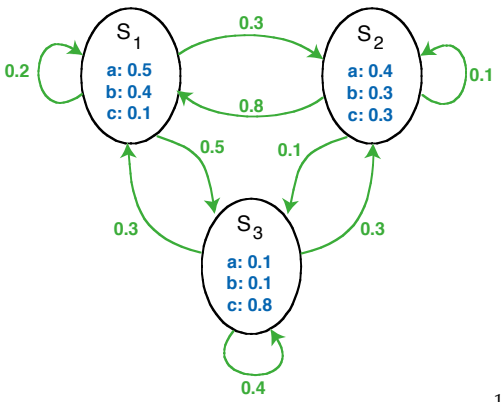
$$P(time\ flies\ like\ an\ arrow|n\ n\ v\ det\ n) \simeq P(time|n)P(flies|n)P(like|v)P(an|det)P(arrow|n)$$

隠れマルコフモデル

- 品詞付けの確率モデル $\prod_{i=1}^{n} P(C_i|C_{i-1}) \prod_{i=1}^{n} P(w_i|C_i)$ は隠れマルコフモデル
- 隠れマルコフモデルとは?
  - Hidden Markov Model (HMM)
  - 状態遷移とシンボルの出力を組み合わせた確率モデル
  - 品詞付け、音声認識で成功を収めている

隠れマルコフモデル

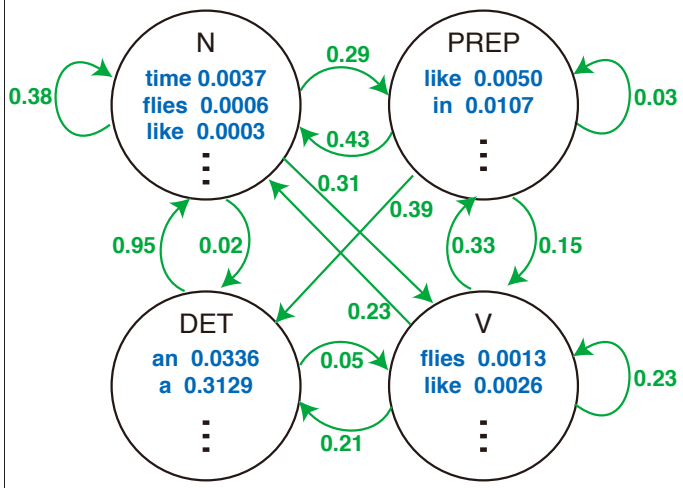
- $S_1 \rightarrow S_2 \rightarrow S_3$ と状態遷移して、シンボル列abcを出力する確率
  - $P(S_1)=0.7$
    $P(a|S_1)=0.5$
    $P(S_2|S_1)=0.3$
    $P(b|S_2)=0.3$
    $P(S_3|S_2)=0.1$
    $P(c|S_3)=0.8$
  - total: 0.00252



HMMの定義

- <S,K,Π,A,B>
  - S: 状態の集合
  - K: シンボルの集合
  - Π: 初期状態の確率
  - A: 状態遷移確率の集合 $P(S_j|S_i)$
  - B: シンボルの出力確率の集合 $P(a_i|S_i)$

品詞タグ付けのためのHMM



P(N|φ)=0.57
P(DET|φ)=0.28
P(PREP|φ)=0.11
P(V|φ)=0.04

φは文頭を表わす

品詞タグ付けのためのHMM

- $<S,K,\Pi,A,B>$
  - S: 品詞の集合
  - K: 単語の集合
  - Π: $P(C_i|\phi)$
  - A: $P(C_j|C_i)$
  - B: $P(w_i|C_i)$

HMMの学習

- 品詞付きコーパスを利用する場合
  - 品詞付きコーパス
    - ✦ 単語に品詞が付与された例文の集合
  - 品詞対の共起頻度から$P(C_j|C_i)$を推定
  - 単語の出現頻度から$P(w_i|C_i)$を推定
  - 教師あり学習
  - 訓練データを用意することが大変

$P(C_j|C_i)$

C_j \ C_i	φ	N	V	DET	PREP
N	392	1111	326	1050	605
	0.57	0.38	0.23	0.95	0.43
V	28	918	313	52	204
	0.04	0.31	0.23	0.05	0.15
DET	194	78	289	0	541
	0.28	0.03	0.21	0.00	0.39
PREP	71	840	456	0	38
	0.11	0.28	0.33	0.00	0.03
Total	685	2947	1384	1102	1388
	1.00	1.00	1.00	1.00	1.00

$P(w_i|C_i)$

w_i \ C_i	N	V	DET	PREP
time	13	0	0	0
	0.0044	0.0000	0.0000	0.0000
flies	2	2	0	0
	0.0007	0.0014	0.0000	0.0000
like	1	4	0	7
	0.0003	0.0029	0.0000	0.0050
...	...	...	...	...
Total	2947	1384	1102	1388
	1.0000	1.0000	1.0000	1.0000

HMMの学習

- プレインテキストを利用する場合
  - プレインテキスト
    - ✦ 品詞などの情報が付与されていない例文の集合
  - forward-backward アルゴリズムで学習可能
  - 教師なし学習
  - 精度が悪い

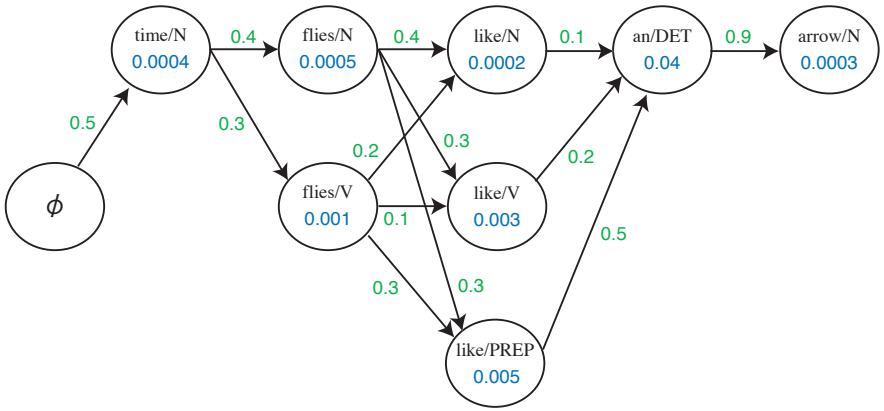
HMMによる品詞タグ付け

- 単語ラティスの作成
  - 単語辞書から各単語の品詞を求め、ノードを作る
  - ノード間にリンクを張る
  - ひとつのパス
    ＝ひとつの品詞付けの候補
- 正しい解析結果を取り出す
  - ノードに$P(w_i|C_i)$, リンクに$P(C_j|C_i)$を割り当てる
  - 確率最大のパスを求める
    - ✦ ヴィテルビ・アルゴリズム
      (ヴィタビ・アルゴリズム, Viterbi Algorithm)

単語辞書

単語	品詞
an	DET
arrow	N
flies	N,V
like	N,V,P
time	N

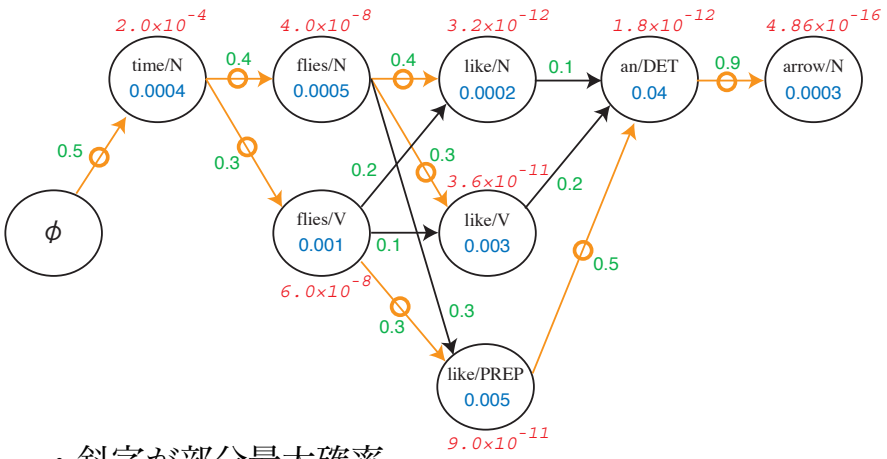
単語ラティス作成



ヴィテルビアルゴリズム

- for i=1 to n
  - 位置 i にある各ノードXについて、部分最大確率を記憶する
    - ✦ そのノードに至るパスの中で最大の確率
    - ✦ 最大確率のパスも記憶する
  - 部分最大確率の計算
    - ✦ 以下の最大値
    - ✦ (位置i−1におけるノードYの部分最大確率)×
(ノードXからノードYへの状態遷移確率)×
(ノードXにおけるシンボルの出力確率)
- 位置nのノードの部分最大確率の大きいパス＝最大確率のパス

ヴィテルビアルゴリズム



・斜字が部分最大確率

・◯が最大確率のパス

計算量

- 全てのパスの確率を計算する場合
  - 計算量はO(c^n)
    - ✦ c: 品詞の数
    - ✦ n: 入力文の単語の数
- ヴィテルビアルゴリズムの場合
  - 計算量はO(c^2n)
  - はるかに高速

まとめ

- 英語の形態素解析では品詞タギングが主な問題
- 隠れマルコフモデル(HMM)を利用した解の優先順位付け
  - 品詞並びに関する優先度
  - 単語の出現のしやすさに関する優先度
- HMMは自動的に学習できる