

# メディア情報処理

## — テキスト・自然言語処理 —

学術情報メディアセンター  
森 信介

# 目次

1. 概論, 文字列検索, 言語統計, 文書検索

KWIC, TF · IDF

2. 単語分割, 品詞付与, 形態素解析

規則に基づく方法, 機械学習

3. 構文解析, 意味解析, 機械翻訳

述語項構造, 意味表現

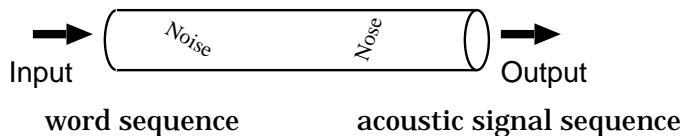
4. 言語モデル, 仮名漢字変換

# 読書案内

1. A Mathematical Theory of Communication  
C. E. Shannon, Bell System Technical Journal,  
Vol. 27. pp. 379–423 and 623–656, 1948.
2. Prediction and Entropy of Printed English  
C. E. Shannon, Bell System Technical Journal,  
Vol. 30, pp. 50–64, 1951.
3. 確率的言語モデル  
北 研二, 東京大学出版会, 1999.

# 生成的な確率モデルによるアプローチ

- 雑音のある通信路モデル (Noisy Channel Model)



- モデル構築

- 複数のモデルに分割 (分析・分割)
- 各モデルを独立に作る (枚挙)
- 解探索 (統合)

$$\hat{I} = \operatorname{argmax} P(I|O) = \operatorname{argmax} P(I)P(O|I)$$

# 音声認識

- 雑音のある通信路モデルの典型

$$\hat{w} = \operatorname{argmax}_w P(w|s) = \operatorname{argmax}_w P(w)P(s|w)$$

- 言語モデル:

$$\begin{aligned} P(w) &= P(w_1)P(w_2|w_1) \cdots P(w_k|w_1w_2 \cdots w_{k-1}) \\ &= \prod P(w_i|H_i) \end{aligned}$$

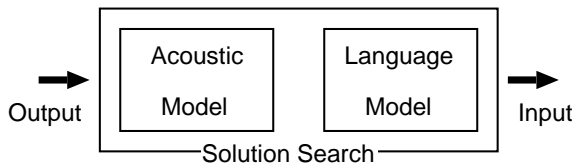
- 音響モデル:

$$\begin{aligned} P(s|w) &= P(s_1|w_1)P(s_2|w_2) \cdots P(s_k|w_k) \\ &= \prod P(s_i|w_i) \end{aligned}$$

# 音声認識

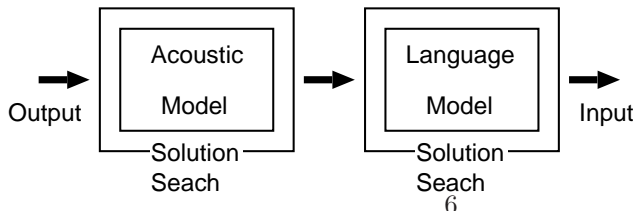
- 雑音のある通信路モデルの典型

$$\hat{w} = \operatorname{argmax} P(w|s) = \operatorname{argmax} P(w)P(s|w)$$



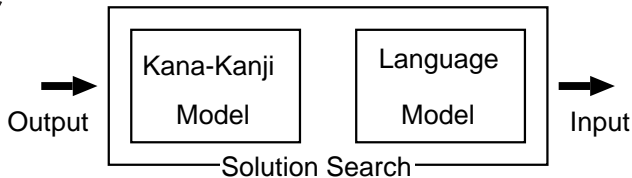
- もし解探索も分割すると

$$\hat{w} = \operatorname{argmax} P(w|\hat{h}), \quad \hat{h} = \operatorname{argmax} P(h|s)$$

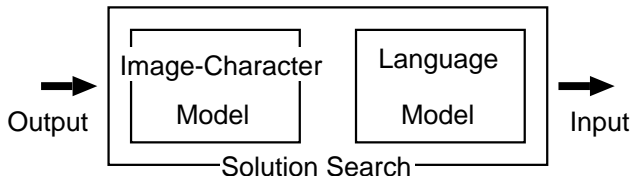


# その他の応用

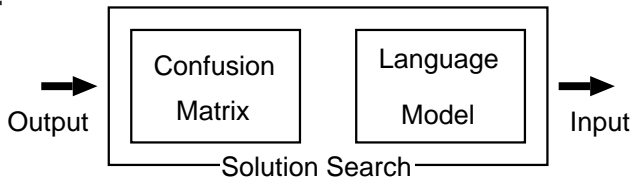
- 仮名漢字変換



- 文字認識



- 文字誤り訂正



# 確率的言語モデル

- 確率的言語モデル  $P(x)$

1. 日本語のアルファベット列が出現する確率値

$$P(x) \in [0, 1]$$

2. すべてのアルファベット列の合計は1以下

$$\sum_{x \in \mathcal{X}^*} P(x) \leq 1$$

## 記号の定義

- 日本語の文字集合を  $\mathcal{X}$  とする ( $|\mathcal{X}| = 6878$ )
- 文は文字の列 ( $\vec{x} = x_1x_2 \cdots x_h$ )
- 文末に特別な文区切り記号  $x_{h+1} = \text{BT} \notin \mathcal{X}$  を付加

例文) 我輩は猫であるBT

# 文字0-gramモデル

例文)  $\vec{x} =$  我輩は猫であるBT

- 文の先頭から順に文字単位で予測 (記述)

$$P(\text{我}) = \frac{1}{|\mathcal{X}|} \quad 12.75[\text{bit}]$$

$$P(\text{輩}) = \frac{1}{|\mathcal{X}|} \quad 12.75[\text{bit}]$$

⋮

$$P(\text{る}) = \frac{1}{|\mathcal{X}|} \quad 12.75[\text{bit}]$$

$$P(\text{BT}) = \frac{1}{|\mathcal{X}|} \quad 12.75[\text{bit}]$$

---

$$P(\vec{x}) = \frac{1}{|\mathcal{X}|^8} \quad 8 \times 12.75[\text{bit}]$$

# 文字 1-gram モデル

- 個々の文字の出現確率  $P(x)$  を与え、文の生成確率を以下で計算

$$P(\vec{x}) = P(x_1 x_2 \cdots x_h \text{BT}) = \prod_{i=1}^{h+1} P(x_i)$$

- 頻出文字に高い確率 (短い符合) を!!
- すべての文字の出現確率の合計は 1 以下

$$\sum_{x \in \mathcal{X}} P(x) \leq 1$$

- 最も無駄がないのは等号が成り立つとき

# 文字 1-gram モデル (つづき)

- 文字の出現確率をコーパスから最尤推定

$$P(x) = \frac{f(x)}{\sum f(x)} \quad \text{ex.) } P(\text{の}) = 0.05$$

–  $f(\text{BT})$  は文の数

- コーパスに出てこない文字の確率が 0 になる

$$f(x) = 0 \Rightarrow P(x) = 0 \quad \text{ex.) } P(\text{蟲}) = 0$$

– そのような文字を含む文の生成確率が 0 に

ex.)  $P(\text{王蟲の怒りを鎮める冒頭のシーンです}) = 0$

# 未知文字記号 UX の導入

1. 文字集合を既知文字  $\mathcal{X}_k$  と未知文字  $\mathcal{X}_u$  に分割
2. 既知文字と UX の確率はコーパスから最尤推定  
ex.) 王蟲は蟲の王様だ BT  $\Rightarrow$  王 UX は UX の王様だ BT

$$f(\text{UX}) = 2, \quad P(\text{UX}) = \frac{f(\text{UX})}{\sum f(x)}$$

3. 未知文字は未知文字記号を経由して一様分布

$$M_{\mathcal{X},1}(x_1^h) = \prod_{i=1}^{h+1} P(x_i)$$
$$P(x_i) = \begin{cases} P(x_i) & \text{if } x_i \in \mathcal{X}_k \\ P(\text{UX}) \frac{1}{|\mathcal{X}_u|} & \text{if } x_i \notin \mathcal{X}_k \end{cases}$$

# 単語 1-gram モデル

- 文は単語の列 ( $\vec{w} = w_1 w_2 \cdots w_h = w_1^h$ )
- 文末に特別な文区切り記号  $w_{h+1} = \text{BT}$  を付加

例文)  $\vec{w} = \text{我輩は猫であるBT}$

- 文の先頭から順に単語単位で予測 (記述)

$$\begin{aligned} M_{\mathcal{W},1}(\vec{w}) &= M_{\mathcal{W},1}(w_1 w_2 \cdots w_h \text{BT}) \\ &= \prod_{i=1}^{h+1} P(w_i) \end{aligned}$$

# 単語 1-gram モデル (つづき)

- 文の先頭から順に単語単位で予測 (記述)

$$M_{\mathcal{W},1}(\vec{w}) = M_{\mathcal{W},1}(w_1 w_2 \cdots w_h \text{BT}) = \prod_{i=1}^{h+1} P(w_i)$$

- 未知語は特別な記号 (UW) を経由して未知語モデル  $M_{uw}$  で予測

$$P(w_i) = \begin{cases} P(w_i) & \text{if } w_i \in \mathcal{W}_k \\ P(\text{UW})M_{uw}(w_i) & \text{if } w_i \notin \mathcal{W}_k \end{cases}$$

$\mathcal{W}_k$ : 既知語の集合 (語彙)

- 未知語は無限にあるので  $M_{uw}(w) = \frac{1}{|\mathcal{W}_u|}$  とはできない

# 未知語モデル

- 単語は文字の列 ( $w = x_1x_2 \cdots x_{h'}$ )
- 語末に特別な単語区切り記号  $x_{h'+1} = \text{BT} \notin \mathcal{X}$  を付加

例文) 王蟲BT

- 単語は文字列  $\Rightarrow$  文字 1-gram モデルを利用

$$M_{\mathcal{X},1}(w) = M_{\mathcal{X},1}(x_1^{h'}) = \prod_{i=1}^{h'+1} P(x_i)$$

- 文のモデルと同じ!!

# パラメータ推定

- 単語分割済みコーパスが必要

ex.) 王蟲は蟲の王様だ  $\Rightarrow f(\text{蟲}) = 1 (\neq 2)$

- 自動分割

低コスト, 不正確 (特に想定外の分野で)

- 手動分割

高コスト, (比較的) 正確

# パープレキシティ

単語単位の

- 文字エントロピーと平均単語長から計算

1. テストコーパス  $C_t$  に対して文字単位のエントロピー  $H$  を計算 ( $|C_t|$  は  $C_t$  の文字数)

$$H = -\frac{1}{|C_t|} \log_2 \prod_{\vec{w} \in C_t} M_{w,n}(\vec{w})$$

2. 単語単位のパープレキシティを計算

$$PP = 2^{H \times \overline{|w|}}$$

$\overline{|w|}$  は平均単語長 (文字数)

- 次の単語が等確率とみた場合の平均の単語数

# 単語 2-gram モデル

- 単語単位で直前の単語を条件として予測

$$M_{\mathcal{W},2}(\vec{w}) = M_{\mathcal{W},2}(w_1 w_2 \cdots w_h \text{BT}) = \prod_{i=1}^{h+1} P(w_i | w_{i-1})$$

ex.)  $P(\text{大学} | \text{京都})$ : 「京都」の直後の「大学」の確率

- 未知語は特別な記号 (UW) を経由して未知語モデルで予測

$$P(w_i | w_{i-1}) = \begin{cases} P_E(w_i | w_{i-1}) & \text{if } w_i \in \mathcal{W}_k \\ P_E(\text{UW} | w_{i-1}) M_{\mathcal{X}}(w_i) & \text{if } w_i \notin \mathcal{W}_k \end{cases}$$

- 文頭に特別な記号 BT を付加  $\Rightarrow$  記述の簡略化

例文)  $\vec{w} = \text{BT 我輩は猫であるBT}$

# パラメータ推定 (単語 2-gram モデル その2)

- 単語 2-gram 確率はコーパスから最尤推定

$$P_{MLE}(w_x|w_y) = \frac{f(w_y w_x)}{f(w_y)} \quad \text{ex.) } \frac{f(\text{京都 大学})}{f(\text{京都})}$$

ちなみに  $\sum_{w_x} f(w_y w_x) = f(w_y)$  のはず

単語 1-gram 頻度が 0 の単語が語彙に含まれていると  $f(w_y) = 0 \Rightarrow$  上式の分母が 0!!

1. 学習コーパスに出現する単語のみから語彙を構成
2. 前件の頻度  $f(w_y)$  によって場合分け

## 補間 (単語 2-gram モデル その 3)

- 学習コーパスに出現しない単語 2-gram はないと言えるか

ex.) もし  $f(\text{京都 大学}) = 0$  なら  $P(\text{大学}|\text{京都}) = 0$   
テストコーパスにその単語 2-gram があると  
生成確率は 0!!

- 補間: 単語 1-gram 確率を少し混ぜる

$$P(w_i|w_{i-1}) = \lambda_1 P_E(w_i) + \lambda_2 P_E(w_i|w_{i-1})$$

ここで  $0 \leq \lambda_i \leq 1$  ( $i = 1, 2$ ),  $\lambda_1 + \lambda_2 = 1$  である

ex.)  $f(\text{京都 大学}) = 0 \Rightarrow P(\text{大学}|\text{京都}) = \lambda_1 P_E(\text{大学}) > 0$

## 補間係数の推定 (単語 2-gram モデル その4)

1. 学習コーパスの一部をヘルドアウトコーパス  $C_H$  とする

$C_H$  でテストコーパスを模擬する

2. 残りの学習コーパス  $C_L$  から単語 2-gram 確率  $P_L$  を推定

3.  $C_H$  の生成確率が最大になるように  $(\hat{\lambda}_1, \hat{\lambda}_2)$  を推定 (EM アルゴリズム)

$$(\hat{\lambda}_1, \hat{\lambda}_2) = \operatorname{argmax}_{(\lambda_1, \lambda_2)} \prod_{C_H} P_L(w_i | w_{i-1})$$

# 単語 $n$ -gram モデル

- 直前の  $(n - 1)$  単語を条件として予測

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

- 未知語は特別な記号 (UW) を経由して未知語モデルで予測

単語 2-gram モデルと同様

- 補間

$$P(w_i | w_{i-n+1}^{i-1}) = \sum_{k=1}^n \lambda_k P_E(w_i | w_{i-k+1}^{i-1})$$

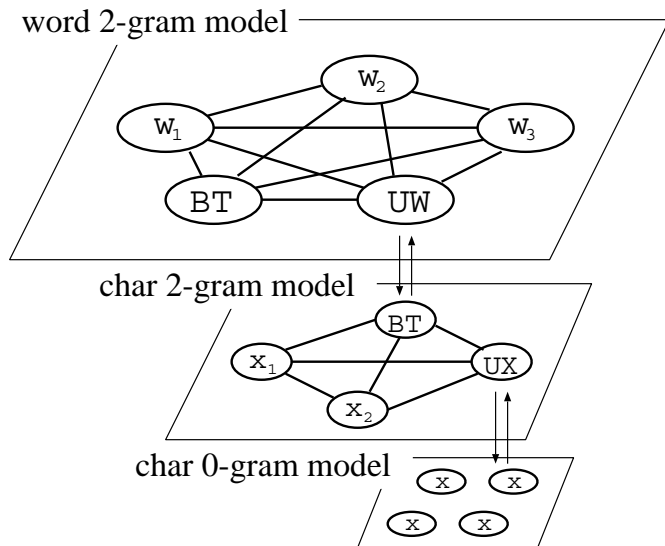
# 階層的な確率的言語モデル

- 未知語モデルに文字  $n'$ -gram モデルを用いる

ことも可 
$$M_{\mathcal{X}, n'}(x_1^{h'}) = \prod_{i=1}^{h'+1} P(x_i | x_{i-n'+1}^{i-1})$$

- 言語モデルの代入

1. あるアルファベット上の言語モデルの記号  $S$  (ex. UW) を...
2. 別のアルファベット上の言語モデルが生成する記号列で置き換える



# 応用

- 自動単語分割 [Nagata 94]

生成確率が最大となる単語列

$$\hat{w} = \operatorname{argmax}_{w=x} M_{w,n}(w)$$

- 音声認識
- 仮名漢字変換 [森 ほか 1998]  
 $y$ : 入力記号列,  $x$ : 文字列

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

$$i \leq j \Leftrightarrow P(x_i|y) \geq P(x_j|y)$$

$$\Leftrightarrow P(y|x_i)P(x_i) \geq P(y|x_j)P(x_j)$$

# 仮名漢字変換

- 最初に実用化された大規模自然言語処理
  - 規則に基づく方法 (人手によるスコア調整)
- 統計的手法 [森 ほか, 1998]
  - MS-IME 2007 (単語 3-gram モデル?)
  - 次世代ことえり?
  - ATOKも部分的に統計を使っているらしい

# 仮名漢字変換

- 言語モデル: 単語 2-gram モデル
- 仮名漢字モデル: 各単語での独立性を仮定

候補列挙のための尤度

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

$$P(y_i|w_i)P(w_i) = \begin{cases} P(w_i|w_{i-n+1}^{i-1})P(y_i|w_i) & \text{if } w_i \in \mathcal{W} \\ P(\text{UW}|w_{i-n+1}^{i-1})M_{y,n}(y_i) & \text{if } w_i \notin \mathcal{W} \end{cases}$$

$\mathcal{W}$ : 確率的言語モデルの語彙

# 変換処理過程 その1

入力 イジョウノコウタイガアルト  
イ/胃/Dict  
イ/医/Dict  
イ/意/Dict  
イ/言/Dict  
イ/い/Dict  
イジ/意地/Dict  
イジ/維持/Dict  
ジ/字/Dict  
ジ/辞/Dict  
ジ/地/Dict  
ジ/寺/Dict  
ジョ/女/Dict  
ジョ/助/Dict

## 変換処理過程 その2

入力 イジョウノコウタイガアルト  
イジョウ/異常/Dict  
イジョウ/委譲/Dict  
イジョウ/以上/Dict  
ジョウ/状/Dict  
ジョウ/条/Dict  
ジョウ/嬢/Dict  
ジョウ/上/Dict  
ウ/産/Dict  
ウ/う/Dict  
ウ/生/Dict  
ウ/雨/Dict  
ウノ/宇野/Dict  
ノ/乗/Dict

# 変換処理過程 その3

入力 イジョウノコウタイガアルト

ノ/廬/Dict

ノ/ノ/Dict

ノ/載/Dict

ノ/の/Dict

ノコ/残/Dict

ノコ/のこ/Dict

コ/古/Dict

コ/故/Dict

コ/粉/Dict

コ/超/Dict

コ/こ/Dict

コ/木/Dict

コウ/光/Dict

# 変換処理過程 その4

入力 イジョウノコウタイガアルト  
コウ/工/Dict  
コウ/凍/Dict  
コウ/鋤/Dict  
コウ/校/Dict  
コウ/こう/Dict  
ウ/産/Dict  
ウ/う/Dict  
ウ/売/Dict  
ウ/雨/Dict  
ウタ/うた/Dict  
ウタ/歌/Dict  
タ/建/Dict  
タ/立/Dict

# 変換処理過程 その5

入力 イジョウノコウタイガアルト

タ/他/Dict

タ/た/Dict

コウタイ/交代/Dict

コウタイ/後退/Dict

タイ/帯/Dict

タイ/タイ/Dict

タイ/体/Dict

タイ/隊/Dict

イ/胃/Dict

イ/医/Dict

イ/意/Dict

イ/言/Dict

イ/い/Dict

# 変換処理過程 その6

入力 イジョウノコウタイガアルト

ガ/画/Dict

ガ/が/Dict

ア/亜/Dict

ア/空/Dict

ア/会/Dict

ア/あ/Dict

アル/歩/Dict

アル/ある/Dict

ル/ル/Dict

ル/る/Dict

ト/取/Dict

ト/と/Dict

ト/解/Dict

# 実装

## 入力

イジョウノコウタイガアルト

## 出力

以上/Dict の/Dict 交代/Dict が/Dict  
あ/Dict る/Dict と/Dict

- 実装

- 動的計画法 ( $O(h)$ )  $\approx$  形態素解析
- AC法による辞書検索

# 精度

評価基準: LCS(最長共通部分系列)による

$$\text{適合率} = |LCS|/|SYS| \quad (94\% \sim 98\%)$$

$$\text{再現率} = |LCS|/|COR| \quad (94\% \sim 98\%)$$

## 言語資源

1. 単語境界&読み付与コーパス 56,924文
2. テキスト 8,341,771文
3. UniDic 223,377語

# 読み推定

- 音声合成 (TTS; Text-To-Speech) の一部
  - 単語と読みの組を単位とした  $n$ -gram モデル
    - 〈メニュー, めにゅー〉 〈的, てき〉 〈に, に〉
    - 〈は, は〉 〈十分, じゅうぶん〉 〈だ, だ〉
- 形態素解析では読みは重視されていない
- 精度は 99% 程度
  - アクセント や 休止時間の推定も言語処理

# 翻字 (Transliteration)

- 機械翻訳や読み推定の一部
  - 固有名詞と思われる未知語

ex.) Descartes ⇒ デカルト

- 文字列と読みの組の  $n$ -gram モデルなど

$$\begin{cases} \text{gou/グ r/ル met/メ} \\ \text{gou/グ r/ル me/メ t/無音} \end{cases}$$

英語: 4,419 語, 仏や伊も必要?

cf. Google Map

# 言語処理システムの公開

あるいは製品化

- 分野適応が課題（医療、法律、...）
- 精度がすべてではない
  - － 体感の精度
  - － インターフェース
  - － サポート体制
- 法的な問題
  - － 言語資源の著作権
  - － 個人情報