自然言語処理論I

10. 言語の統計モデル

言語の統計的な分析

- 統計的な分析とは?　(講義でやること)
  - 文字や単語の頻度を数える　} n-gram統計
  - 文字や単語の共起頻度を数える　} ジップの法則
  - 文の生成確率を求める ← マルコフモデル
- 目的は?
  - 言語の性質を明らかにする
  - 自然言語処理への応用 ← 応用

頻度

- 文字の頻度
  - ある文字が出現する頻度
- Wall Street Journal での例
  - 教科書p.14, 表2.1
    - ✦ 相対頻度(%)
  - 16万文字
  - 頻度が大きい文字: 空白, e, t, a, i, o, ...

n-gram統計

- n文字の共起頻度(相対頻度)
  - 2-gram, 3-gram, 4-gram, ...
  - n=1のとき→文字の頻度
- 2-gram統計の例
  - 教科書pp.16-17, 表2.2
  - 27×27の配列
  - 1回も出現しない文字の並びもある
- n-gram統計の例
  - 教科書p.18, 表2.3
  - 英語の文章によく現われる文字列がわかる

n-gram統計

- 日本語の場合
  - 教科書p.19, 表2.4
  - 英語と比べて文字種が多い
    - ひらがな、カタカナ、漢字
    - 4000〜5000字
    - 統計データを作成することが困難
  - 典型的な言い回しや単語がわかる
  - 元のテキストに大きく依存する
    - 文書のジャンル
    - 年代

単語の統計

- 今までは文字単位の統計
- 単語単位でも同様の統計を調べることができる
- 単語の頻度
  - Wall Street Journalの例
    - 相対頻度

単語の相対頻度(%)

the	5.066	on	0.6236	be	0.4486
of	2.796	said	0.5471	its	0.4420
to	2.698	by	0.5442	are	0.4089
a	2.317	at	0.5338	an	0.3778
and	1.939	from	0.5304	has	0.3668
in	1.861	as	0.5196	will	0.3625
for	0.9719	with	0.5095	have	0.3561
that	0.9514	Mr.	0.4914	million	0.3334
is	0.7772	it	0.4825	or	0.3047
The	0.7715	was	0.4667	he	0.3035

単語の統計

- 単語のn-gramも同様に求められる
  - p(the,dog)
  - p(the,dog,runs)
- nをあまり大きくすることは現実的ではない
  - 表のサイズが大きくなる
  - 統計的に信頼できるほど十分な量のテキストが得られない

ジップの法則

- 英語の単語の頻度分布に関する法則
- 単語の数え方
  - 異なり数
    - ✦ 単語の種類を数える
  - のべ数
    - ✦ 単語の出現を数える
  - 例: do in Rome as the Romans do
    - ✦ 単語ののべ数は7
    - ✦ 単語の異なり数は6

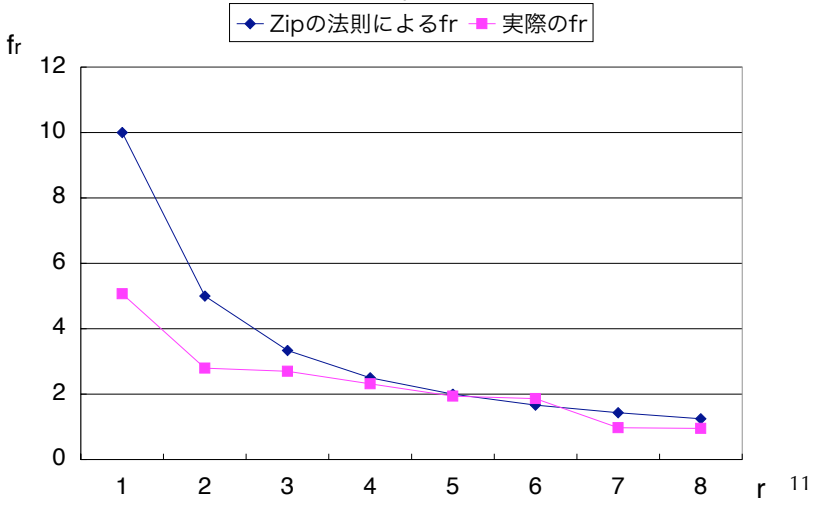
ジップの法則

- $f_r \simeq \frac{0.1}{r}$
  - r: ランク(単語の出現頻度の順位)
  - f_r: ランクが r である単語の相対頻度
- $f_r \times r =$ 定数 (=0.1)
- f_rはf_1の1/rである

- 英単語ののべ数に関する法則

ジップの法則

- 実際のデータとの比較



ジップの法則

- $v_n(x) \simeq \frac{0.1n}{x^2}$
  - $v_n(x)$：出現頻度がx回である単語の異なり数
  - n: 文章の単語の総数
- n=100,000のとき
  - 頻度1の単語の異なり数は10,000
  - 頻度2の単語の異なり数は2,500
  - 頻度100以上であるような単語の異なり数は1個または0個
- 英単語の異なり数に関する法則

ジップの法則の改良

- $p_r = \frac{c}{r}, \quad \sum_{r=1}^{v} p_r = 1$
  - cは定数、vは単語の異なり数
- $p_r = \frac{c}{r^b}, \quad \sum_{r=1}^{v} p_r = 1$
  - b>0, c>0
- $p_r = \frac{c}{(r+a)^b}, \quad \sum_{r=1}^{v} p_r = 1$
  - 0 < a < 1, b > 0, c > 0

マルコフモデル

- 文の生成確率を求める
  - 文字列の生成確率P(c_1, ..., c_m)を求める
- マルコフモデルによる近似
  - Markov model
  - 文字c_iの生成確率は直前のn−1個の文字のみに依存すると仮定
  - n-gramモデルとも呼ばれる

マルコフモデル(n-gramモデル)

- $$P(c_1 ... c_m) = P(c_1) \times P(c_2|c_1) \times P(c_3|c_1c_2) \times \ ... = \prod_{i=1}^{m} P(c_i|c_1c_2 \ ... \ c_{i-1}) \simeq \prod_{i=1}^{m} P(c_i|c_{i-n+1} \ ... \ c_{i-1})$$
- n=2のとき、$\prod_{i=1}^{m} P(c_i|c_{i-1})$
  - bi-gramモデル
- n=3のとき、$\prod_{i=1}^{m} P(c_i|c_{i-2}c_{i-1})$
  - tri-gramモデル

n-gramモデルの求め方

- n-gram統計を使う
- n=2のとき
  - 2-gram統計はp(c_i,c_j)
  - $P(c_j|c_i) = \frac{p(c_i, c_j)}{\sum_j p(c_i, c_j)}$
- n>=3のときでも同様
  - $P(c_k|c_i, c_j) = \frac{p(c_i, c_j, c_k)}{\sum_k p(c_i, c_j, c_k)}$

$P(C_j \mid C_i)$

C_i \ C_j	a	b	c	...	z	(合計)
a	0 0	0.00154 0.0191	0.00332 0.411	...	0.00009 0.00111	0.08072 1
b	0.00112 0.0760	0.00006 0.00407	0 0	...	0 0	0.01474 1
l	l	l	l		l	l
z	0.00010 0.179	0 0	0 0	...	0.00005 0.0893	0.00056 1

上段: n-gram統計 $p(c_i,c_j)$　下段: n-gramモデル $p(c_j \mid c_i)$

$P(C_k \mid C_i,C_j)$

C_i,C_j \ C_k	a	b	c	...	z	(合計)
aa	0 –	0 –	0 –	...	0 –	0 –
ab	0.00020 0.1299	0.00006 0.0390	0 0	...	0.00013 0.0844	0.00154 1
l	l	l	l		l	l
zz	0.00001 0.3333	0 0	0 0	...	0.00001 0.3333	0.00003 1

上段: n-gram統計 $p(c_i,c_j,c_k)$　下段: n-gramモデル $p(c_k|c_i,c_j)$

n-gramモデル

- 文字のn-gramモデル $\prod_{i=1}^{m} P(c_i|c_{i-n+1} \ldots c_{i-1})$
  - ある文字列の後、どの文字がどれだけの確率で続くか?
  - P(b | a) = 0.0191
- 単語のn-gramモデル $\prod_{i=1}^{m} P(w_i|w_{i-n+1} \ldots w_{i-1})$
  - ある単語列の後、どの単語がどれだけの確率で続くか?
  - P(百万石 | 加賀) = 0.2163
- いずれも文の生成確率を与える

応用

- 英語の品詞タグ付け
- 文字認識, OCR
  - 文字のbi-gramモデルを使う
  - 2つの認識結果の候補があるとする
    - ahl
    - all
  - 文字列の生成確率をbi-gramで計算
    - P(ahl) = P(a|ϕ) P(h|a) P(l|h)
    - P(all) = P(a|ϕ) P(l|a) P(l|l)
  - P(all)>P(ahl)なら、allが正しいとみなす

応用

- 音声認識
  - 音声認識結果の優先順位付けに単語列や音韻列のn-gramモデルを使う
- (辞書を使わない)単語分割
  - n-gramモデルの確率分布 $P(c_j|c_i)$ を調べる
  - 次に生成される文字の確率分布が一様分布に近い → 単語境界である可能性が高い
- テキスト間の類似性判定
  - テキストの統計情報を手がかりとする

応用

- テキスト間の類似性検定
  - 2つのテキストが同一著者によるものか?
- 利用できる統計情報
  - 数値パラメタ
    - ✦ 単語の頻度、単語のランク、文の長さ、埋め込み文の数
  - 比
    - ✦ 異なり語数/のべ語数、名詞数/動詞数、能動文/受動文、単文/複文、具体名詞/抽象名詞

テキスト間の類似性検定

- χ^2検定
  - 文書の特徴を特徴ベクトル(f_{1i}, ..., f_{ni})で表現

	テキスト1	テキスト2	
f_1	f_{11}	f_{12}	N_1
f_2	f_{21}	f_{22}	N_2
l	l	l	l
f_n	f_{n1}	f_{n2}	N_n
	n_1	n_2	N

ただし、

$N_i = f_{i1} + f_{i2}$

$n_1 = \Sigma f_{i1}$

$n_2 = \Sigma f_{i2}$

$N = n_1 + n_2$

テキスト間の類似性検定

- テキスト1とテキスト2が同じなら、
  $\frac{f_{i1}}{n_1} = \frac{f_{i2}}{n_2} = \frac{N_i}{N}$ が成立
- f'_{i1}, f'_{i2}をf_{i1}, f_{i2}の期待値とすると...

$$f'_{i1} = \frac{n_1 N_i}{N}, \quad f'_{i2} = \frac{n_2 N_i}{N}$$

- 現実の値f_{i1}, f_{i2}と期待値f'_{i1}, f'_{i2}のずれの大きさをχ^2ではかる

$$\begin{aligned} \chi^2 &= \sum_i \frac{(f_{i1} - f'_{i1})^2}{f'_{i1}} + \sum_i \frac{(f_{i2} - f'_{i2})^2}{f'_{i2}} \\ &= \sum_i \left(\frac{{f_{i1}}^2}{n_1} + \frac{{f_{i2}}^2}{n_2} \right) \frac{N}{N_i} - N \end{aligned}$$

テキスト間の類似性検定

- χ^2の値が小さければ小さいほど、2つのテキストは似ている
  - 一定の閾値以下なら、2つのテキストは同一であると判定
- χ^2の閾値は次の要素から決まる
  - 自由度
    - ✦ 特徴ベクトルの長さn
    - ✦ 文書の数m
  - 有意水準

まとめ

- 言語の統計的な分析
  - n-gram統計
  - ジップの法則
  - マルコフモデル
- 応用
  - 英語の品詞タグ付け、文字認識、音声認識、単語分割、テキスト間の類似性判定