横浜国立大学 大学院 電子情報工学専攻 専門科目

ディジタルドキュメント論(6)

Digital Document Processing

森 辰則
mori@forest.dnj.ynu.ac.jp

自然言語処理(1) --- 形態素解析

語の検索の前に

自然言語の階層的構造

□音素
- ○音声をどのように使っているかを基に考えた音の単位

□形態素
- ○意味を持つ最小の言語単位．1つ以上の音素からなる．

□語
- ○一つの意味のまとまりをなし，文法上一つの機能を持つ最小の言語単位．一つ以上の形態素からなる．

□文
- ○あるまとまった内容を持ち，形の上で完結した言語単位．一つ以上の単語からなる．

□文書・テキスト
- ○あるまとまった内容を表すための文の，順序付けられた集まり．

形態素解析(2)

文字によって記述された自然言語においては，形態素が最小の単位

□自然言語処理の第一段階

形態素解析(3)

英語の場合

□語: 空白で区切られた文字列．語が「分かち書き」されているという．

□形態素: 語を構成する単位で，「語基」(radical)と接辞(affix)からなる．
- ○play, play(play-ing)

□形態素解析
- ○(語はわかっているので)語の構成を認識し，品詞，形態素の数，性，時制，人称，格などを決定する．

□難しい点
- ○各語の品詞を同定すること(品詞の曖昧性の問題)
  - ▹例: breakfast (朝食(名詞) or 朝食を食べる/出す(動詞))

形態素解析(4)

日本語の場合

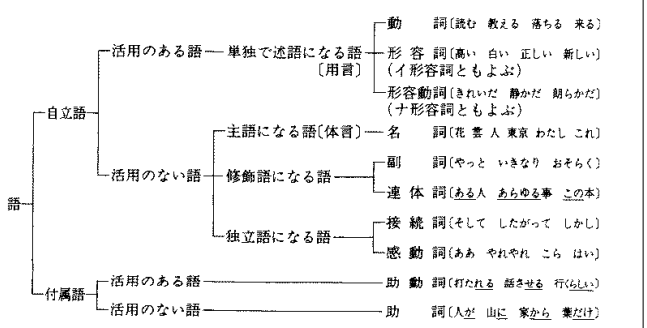
□語を区切る空白が存在しない
- ○語の厳密な定義を与えるのが困難．日本語の形態素の定義(元々，欧米語言語学に由来)も明確でない．

□語構成: 3つの結合形態
- ○活用語(屈折語): 文中での働きによって，語形が変化する語．「活用語幹」+「活用語尾」(変化する)から構成．「食べる」なら「食べ(活用語幹)+る(活用語尾)」
- ○派生語: ある語に付加的要素がついてできる語．「派生語幹」+「接辞」．「寒さ」なら「寒(派生語幹)+さ(接尾辞)」．「真冬」なら「真(接頭辞)+冬(派生語幹)」
- ○複合語:複数の語が結合して1語となった語．「自然+言語+処理」等．

□形態素解析
- ○入力文を辞書中の語の組合せに分解(単語分割)すること

□難しい点: 単語分割

日本語の品詞分類(1)



日本語の品詞分類(2)

活用型と活用形

□活用型: その語がどのような活用を起こすかというタイプ．
- ○例えば，動詞「読む」なら「マ行子音動詞(五段活用)」

□活用形: 文中のある特定の語について，その語の活用がどうなっているか．
- ○例えば，動詞「読む」が「読め-ば」となっている時には「基本条件形(仮定形)」

二単語間の連接規則に基づく日本語形態素解析

□どのような種類(品詞，活用形など)の2語が連続して現れうるかという条件は明確．
- ○2語の連接可能性を規則として与え，これを参照しながら入力文(文字列)を語のならびに変換する．

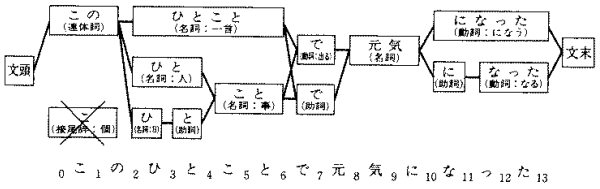
□大まかな処理
- ○1. 辞書を参照して入力文中の各位置から始まる語を取り出す．
- ○2. 連接可能性をチェックしながら取り出された語をつないでいく．

□二つの辞書を用いる．
- ○1. 単語辞書: 語の品詞，読み，活用型などを蓄える．
- ○2. 連接可能性辞書: 連接可能な2語の種類の組合せを網羅したもの．(文頭，文末も仮想的に特別な語があるとする．)

二単語間の連接規則に基づく日本語形態素解析(結果)

束構造(ラティス構造，lattice structure)で表現できる

□語を節点(ノード)，接続可能な語と語の関係を弧(リンク)で表現したグラフ構造．



単語辞書

単語辞書

見出し語	読み	品詞	活用型
日本語	にほんご	名詞：普通名詞	—
英語	えいご	名詞：普通名詞	—
読む	よむ	動詞	子音動詞マ行
書く	かく	動詞	子音動詞カ行
⋮	⋮	⋮	⋮

連接可能性辞書

連接可能性辞書

左側	右側
文頭	名詞，動詞，形容詞，形容動詞，副詞，連体詞，接続詞，感動詞
名詞 動詞 連用形	助詞：格助詞，副助詞，引用助詞，名詞接続助詞 助詞：終助詞 さ，か，かしら，ね，な，よ，なあ
動詞 基本形	助詞：終助詞 ぞ，ぜ，わ
動詞 意志形	助詞：終助詞 ぜ
動詞 基本形	助詞：述語接続助詞 し，が，とも，から，けれども，なら
動詞 連用形	助詞：述語接続助詞 つつ，ながら
動詞 未然形	接尾辞：動詞性接尾辞 れる，られる
⋮	⋮
基本形，命令形 名詞，助詞：終助詞	文末

連接可能性行列

左側＼右側	文末	助詞：格助詞	助詞：副助詞	⋮	助詞：終助詞 ぞ	助詞：終助詞 ぜ	⋮	助詞：述語接続助詞 し	助詞：述語接続助詞 つつ	⋮	動詞性接尾辞 れる	動詞性接尾辞 られる	⋮
文頭	0	0	0		0	0		0	0		0	0	
名詞	1	1	1		0	0		0	0		0	0	
動詞 基本形	1	0	0		1	1		1	0		0	0	
動詞 連用形	0	1	1	⋮	0	0	⋮	0	1	⋮	0	0	⋮
動詞 未然形	0	0	0		0	0		0	0		1	1	
動詞 意志形	0	0	0		1	1		0	0		0	0	
⋮				⋮			⋮			⋮			⋮

二単語間の連接規則に基づく日本語形態素解析(アルゴリズム)

アルゴリズムA

□0) 入力文中の位置(文字と文字の間)を指し示すポインタを用意する．初期状態として，ポインタを位置0(先頭文字の左側)におく．また「先頭」という仮想的な語が位置0で終了していると仮定し，これを「文頭ノード」と呼ぶ．

□1) ポインタの位置から始まるすべての語を単語辞書で検索し，これらを新しいノードとする．

□2) ポインタの位置で終っている語のノードとポインタの位置から始まる語のノードのすべての組について，連接可能性を連接可能性辞書により判定する．

○連接可能な組については，そのノード間に，接続可能性を表すリンクを

二単語間の連接規則に基づく日本語形態素解析(アルゴリズム)

□3) ポインタの位置を一つずつ右にずらし，そこで終了する(今までにリンクの張られた)語が存在するかどうかを調べる．これを繰り返すことにより，初めて何らかの語の末尾が現れるところまでポインタの位置をずらす．

□4) ポインタが文末よりも前の位置であれば step-1 へ．文末であれば，文末で終了している語と「文末」との連接可能性を調べ，可能な場合だけ両ノード間にリンクを張る．

□5) 最終的に「文頭ノード」から「文末ノード」に至る経路(パス，path. ノードとリンクの並び)が入力文に対する形態素解析結果である．

優先規則の導入

□束構造として得られた結果には多数の解が含まれる．

○連接可能行列だけを用いた解析結果には不適切な解があまりにも多く含まれる．

○なんらかの優先規則により尤もらしい解だけを選択することが必要となる．

形態素解析における優先規則(1)

束構造を求めなくてもよい手法

□最長一致法: 長い形態素(語)を優先して文頭から順番に調べ，始めに見つかった候補を解とする．

□2文節最長一致法: 2文節毎の長さが長い解を優先して文頭から順番に調べ，始めに見つかった候補を解とする．

※非常に高速に処理ができるが，精度が良くない．

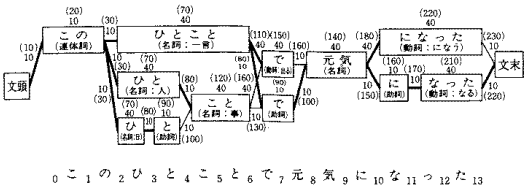
形態素解析における優先規則(2)

束構造を求める必要がある手法
- □形態素数最小法: 形態素数(語数)が一番少ない候補を解とする．
- □文節数最小法: 文節数が一番少ない候補を解とする．
- □コスト最小法: 単語や単語の連接にコスト(手間を表す評価値)を与えて，総コストの一番少ない候補を解とする．

※最初の2手法は「コスト最小法」の特殊形である．
- □形態素数最小法: 語のコストを1，連接のコストを0とする．
- □文節数最小法: 自立語のコストを1，それ以外の語と連接のコストを0とする．

コスト最小解を求める

ビタビアルゴリズム(Viterbi Algorithm)を用いる．
- □部分的最適解のみを保持していく動的計画法(ダイナミックプログラミング,Dynamic Programming, DP)の一種
- □束構造の各ノードにそのノードに至るまでのコストの最小値のみを記憶しておく．



ビタビアルゴリズムによるコスト最小解を求める形態素解析アルゴリズム

アルゴリズムB
- □0) アルゴリズムAのstep-0と同じ．ただし，文頭ノードまでの部分最小コストは0とする．
- □1) アルゴリズムAのstep-1と同じ．
- □2) ポインタの位置で終っている語をWe1,We2,...，そこから始まる語をWs1,Ws2,...とする．各Wsiについて，その左に連接可能な Wej を順次調べ，Wsiとの間にリンクを張る．このとき，以下のコストの和が最小である Wek を求め，そのコストを Wsiまでの部分最小コストとし，WekとWsiの間のリンクに特別なマークをつける．
  - ○Wekまでの部分最小コスト
  - ○Wek とWsi の間の連接コスト

ビタビアルゴリズムによるコスト最小解を求める形態素解析アルゴリズム

- □3) アルゴリズムAのstep-3と同じ．
- □4) アルゴリズムAのstep-4と同じ．
- □5) 最終的にマークのついたリンクを文末から文頭へむけて辿った経路(パス)がコスト最小の解である．

日本語形態素解析システムの実例

- □Juman (じゅまん)
  - ○京都大学で開発
  - ○最新バージョン3.61
- □ChaSen (ちゃせん，茶筅)
  - ○奈良先端科学技術大学院大学で開発
  - ○Jumanから開発の枝別をしたもの
  - ○最新バージョン2.02

ChaSen での解析結果

「昨日は君が嫌いな冷たい雨がとても静かに降っていたらしい．」

```
昨日    キノウ    昨日    名詞-副詞可能
は      ハ        は      助詞-係助詞
君      キミ      君      名詞-代名詞-一般
が      ガ        が      助詞-格助詞-一般
嫌い    キライ    嫌い    名詞-形容動詞語幹
な      ナ        だ      助動詞 特殊・ダ 体言接続
冷たい  ツメタイ  冷たい  形容詞-自立 形容詞・アウオ段 基本形
雨      アメ      雨      名詞-一般
が      ガ        が      助詞-格助詞-一般
とても  トテモ    とても  副詞-助詞類接続
静か    シズカ    静か    名詞-形容動詞語幹
に      ニ        に      助詞-副詞化
降っ    フッ      降る    動詞-自立 五段・ラ行 連用タ接続
て      テ        て      助詞-接続助詞
い      イ        いる    動詞-非自立 一段 連用形
た      タ        た      助動詞 特殊・タ 基本形
らしい  ラシイ    らしい  助動詞 形容詞・イ段 基本形
。      。        。      記号-句点
EOS
```

仮名漢字変換(1)

ワードプロセッサなど日本語入力を必要とするソフトウェアで用いられる
- □入力: 仮名で記述された文(の一部)．わかち書きがされていない．
- □出力: 漢字仮名混じり文(の一部)
- □処理: 語の区切りを見つけ出し，必要な部分を漢字に変換する．

以下の点において一般の形態素解析と同じ
- □語の連接可能性を調べる
- □優先規則を用いる

仮名漢字変換(2)

形態素解析との相違

一般の形態素解析	かな漢字変換
入力は漢字かなまじり文(曖昧性小) 入力の単位は文(曖昧性大)	入力はかなべた書き文(曖昧性大) 入力の単位は文節程度(曖昧性小)
優先規則はコスト最小法など	2 文節最長法など(実時間処理が要求される)
付属語/接辞列部分(多くの場合ひらがな列)の処理が重要であり，またそれが困難である	付属語/接辞列部分のように漢字に変換されない部分は単語分割に失敗しても問題はない
ひらがなで書かれた自立語があっても，語の区切りだけ正しく認識できればよい．例えば「いった」が「言った」であるか「行った」であるかは以降の処理に任せるとしてよい	ひらがなで書かれた自立語を漢字に変換することが重要であり，またそれが困難である

仮名漢字変換(3)

漢字への変換処理が最も重要
- □漢字の選択
  - ○本質的に意味・文脈の情報がなければ行なえない
  - ○これらの情報なしにどれだけ正しく行なえるかが重要
- □現在一般的に用いられる選択方法
  - ○w1*(単語の使用頻度) + w2*(単語が最近使われたかどうか)+ w3*(単語の属する辞書の優先度(ユーザ辞書は優先度大))
    - ▹w1,w2,w3: 重み

英語における形態素解析(1)

多品詞語の扱いが問題
- □形態素解析の段階で品詞を推定することが重要
  - ○品詞付け(part-of-speech tagging) と呼ばれる
  - ○(連接関係のような)明確な規則がない
- □入力である単語の並びに対して尤もらしい品詞列を与える
- □尤もらしさの尺度(主なもの)
  - ○各々の語について，可能性のある複数の品詞のうち，どれが一番尤もらしい品詞であるか．
    - ▹例: breakfast は多くの場合，名詞として用いられる．動詞となる場合は少ない
  - ○品詞の並びに対する優先度
    - ▹例: the の後ろに，動詞とも名詞とも解釈できる語がきた場合には名詞の解釈が優先される．

英語における形態素解析(2)

尤もらしさの尺度
- □人手で規則として与える
- □自動的に尺度を獲得する
  - ○品詞情報のついた大量のコーパスを利用し，確率的モデルを獲得する
    - ▹コーパス: 例文集のこと

品詞付けにおける確率モデル
- □入力: 単語列 W1,W2,W3,...,Wn
- □出力: 品詞列 C1,C2,C3,...,Cn
- □処理: 以下の条件付確率を最大にする品詞列を求める．

$$P(C_1, C_2, \cdots, C_n \mid W_1, W_2, \cdots, W_n) = \frac{P(C_1, C_2, \cdots, C_n) \times P(W_1, W_2, \cdots, W_n \mid C_1, C_2, \cdots, C_n)}{P(W_1, W_2, \cdots, W_n)}$$

確率モデルの近似

先の式において...
- □分母は品詞とは無関係であるから，求める品詞列は分子を最大にするもの．
- □ところが，分子の値を直接計算することは不可能
  - ○通常，W1,W2,...,Wn という単語列が十分な回数現れないため．

近似

$$P(C_1, C_2, \cdots, C_n) \sim \prod_{i=1}^{n} P(C_i|C_{i-1})$$

$$P(W_1, W_2, \cdots, W_n \mid C_1, C_2, \cdots, C_n) \sim \prod_{i=1}^{n} P(W_i|C_i)$$

例文集中の単語/品詞の出現確率(例)

各欄は左側が w の出現頻度，右側が $P(w|C)$

w \ C	N		V		DET		PREP	
time	13	0.0037	0	0.0000	0	0.0000	0	0.0000
flies	2	0.0006	2	0.0013	0	0.0000	0	0.0000
like	1	0.0003	4	0.0026	0	0.0000	7	0.0050
an	0	0.0000	0	0.0000	37	0.0336	0	0.0000
arrow	1	0.0003	0	0.0000	0	0.0000	0	0.0000
⋮	⋮		⋮		⋮		⋮	
Total	3481	1.0000	1515	1.0000	1102	1.0000	1405	1.0000

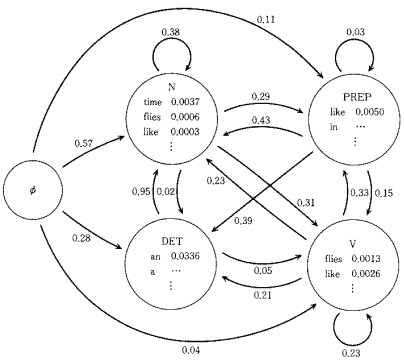
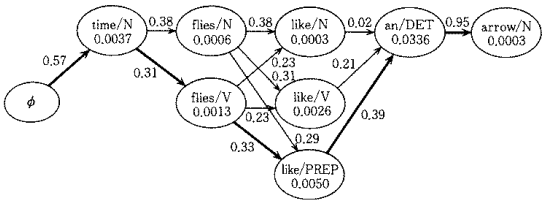
例文集中の品詞bigramの出現頻度(例)

bigram: 2語の連続

各欄は左側が品詞列 $C_{i-1}C_i$ の出現頻度，右側が $P(C_i|C_{i-1})$

C_i \ C_{i-1}	ϕ		N		V		DET		PREP	
N	392	0.57	1111	0.38	326	0.23	1050	0.95	605	0.43
V	28	0.04	918	0.31	313	0.23	52	0.05	204	0.15
DET	194	0.28	78	0.02	289	0.21	0	0.00	541	0.39
PREP	71	0.11	840	0.29	456	0.33	0	0.00	38	0.03
Total	685	1.00	2947	1.00	1384	1.00	1102	1.00	1388	1.00

単語/品詞の隠れマルコフ・モデル



ビタビアルゴリズムによる品詞付け