自然言語処理入門
「ここではきものをぬぐこと」

東京大学　情報基盤センター
（情報理工学系研究科　数理情報学専攻、
学際情報学府　兼担）
中川裕志
nakagawa@dl.itc.u-tokyo.ac.jp
http://www.r.dl.itc.u-tokyo.ac.jp/~nakagawa/

参考文献

- 岩波講座　言語の科学　全11巻
  - 形態素解析は第3巻、構文解析は第8巻、談話、対話は第7巻、文法と意味は第4巻、音声は第2巻、情報処理は第9巻
- 東大出版会　言語と計算
  - 談話、対話は第3巻、統計的言語処理は第4巻、情報検索は第5巻
- 学会誌、国際会議論文集など
  - 言語処理学会誌「自然言語処理」（中川は編集長をしています。）
  - 人工知能学会誌
  - 情報処理学会　論文誌
  - Computational Linguistics (ACL の　journal)
  - Proceedings of ACL
  - Proceedings of COLING
  - ACM SIGIR

ここではきものをぬぐこと

- 「ここで　はきもの　を　脱ぐこと」か
  「ここでは、着物　を脱ぐこと」か
- 「にわにはにわがある」
- 「庭には庭がある」か
  「庭に埴輪がある」か
- 単語の切れ目を見つける→形態素解析
- 形態素とは、文字より大きく、しかしそれ以上分割できない言語単位

なぜ形態素解析か

- 単語が認識できないと、文の意味を組み立てられない。
- わざわざ仮名で書くから難しいのでは、
- 最初から「ここで履物を脱ぐこと」なら簡単？
- ワープロの日本語入力は仮名漢字変換
- 音声認識結果は音の列→仮名の列
  → 漢字やカタカナ交じりの文字列ではない。

日本語と英語

- 英語の場合、最初から単語は空白で区切られている。（ただし、音声認識の場合は日本語と同じ問題）
- 屈折型言語　　apples → apple
  studied → study
- 屈折した表現から原形そして品詞を求める。
- 日本語は単語の切れ目が表面的には分からない。
- 膠着型言語　　単語に分割することが必要。そして品詞も求める。

日本語の形態素解析

- 日本語形態素解析で用いられるヒューリスティックな方法
- 最長一致法

  先頭から辞書において一致する最長の単語を当てはめる。

  全国都道府県議会議長席→

  全国　都道府県議　会議　長　席
- 分割数最小法

  辞書を調べて、すべての可能分割を求め、その中で最小分割数のものを選ぶ。

  全国都道府県議会議長席→　全国　都道府県　議会　議長　席
- 字種切り法

  字種の変化点を単語の境界とみなす。

  カラフルな電子メール → カラフル　な　電子　メール

文法情報に基づく形態素解析の枠組み

にわにはにわがある

辞書：

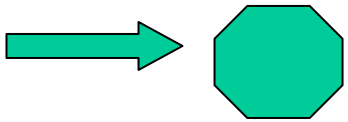
にわ＝名詞：庭、二羽

はにわ＝名詞：埴輪

に＝助詞

は＝助詞

が＝助詞



(1)辞書引きの早さ

(2)辞書と入力文をつき合わせるが曖昧さ解消

にわ（名詞：庭　or 二羽）、

に（助詞）、はにわ（名詞：埴輪）、

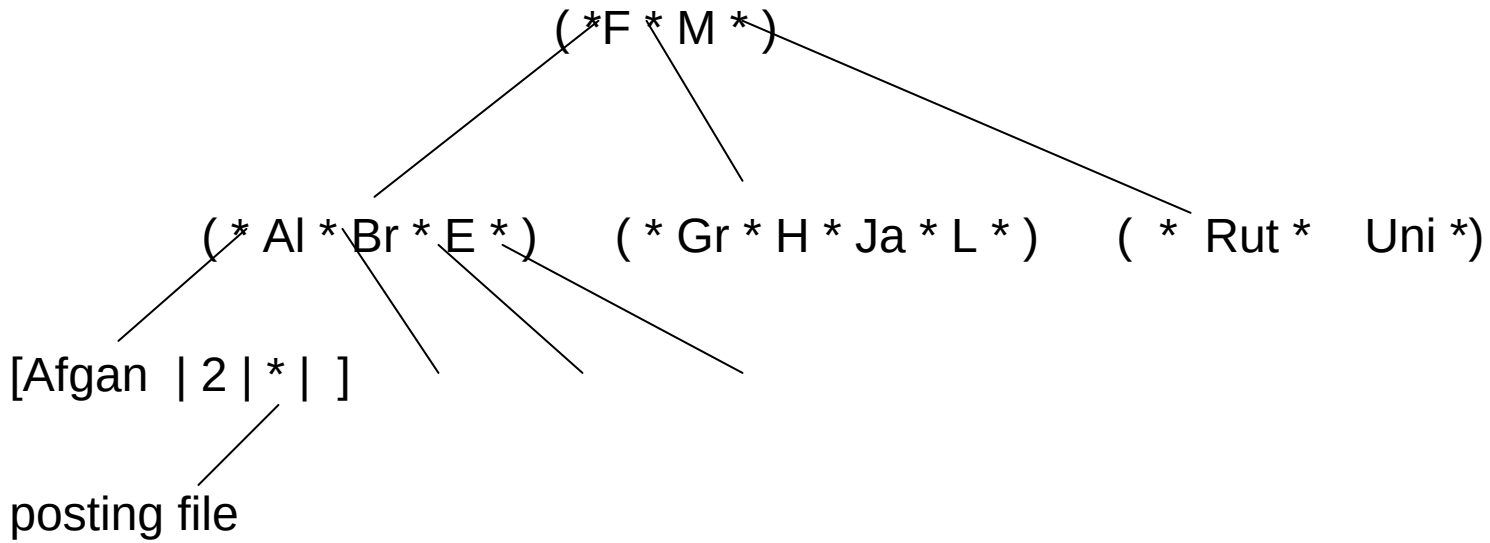
が（助詞）

辞書引きの早さ

- 膨大な数の単語の中から入力文の部分に一致する単語を高速に検索することが重要
- いろいろなデータ構造の工夫有り
- B木
- トライ
- PATRICIA木

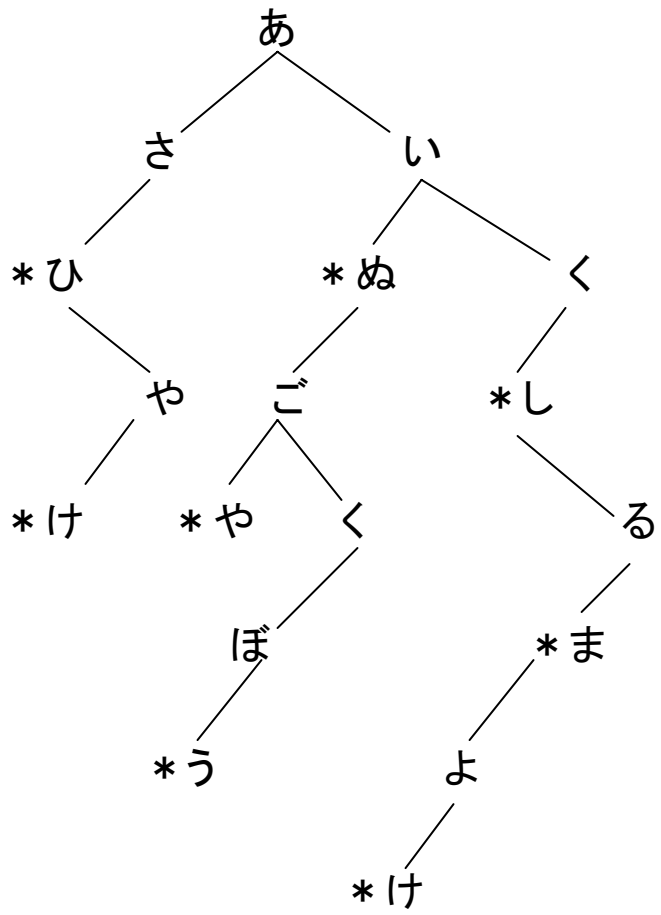
B木

- 膨大な数のキーワードの中から質問として与えられたキーワードを高速に検索することが重要



Trie（トライ）

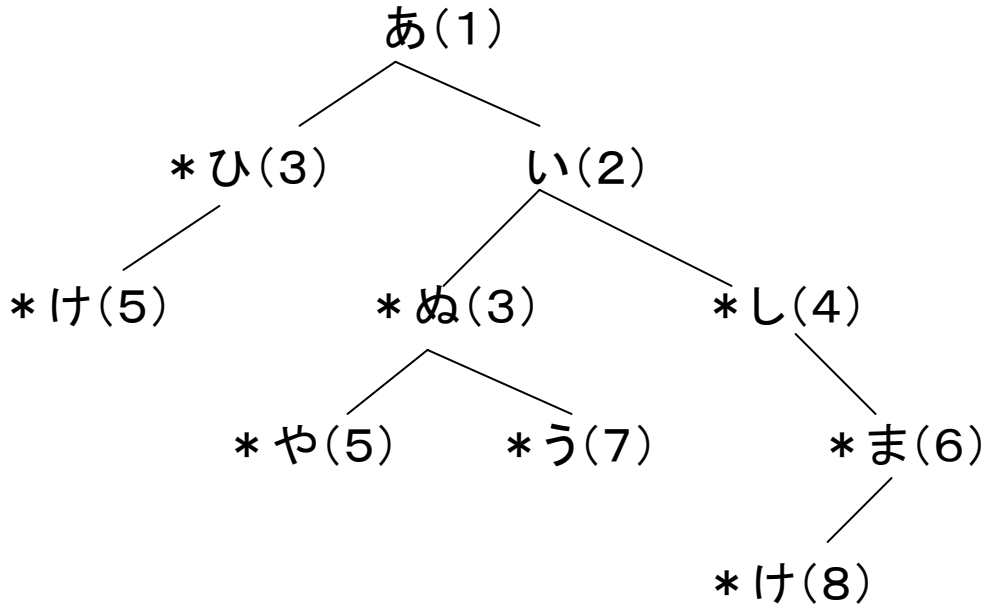
- トライは2 分木で、左ポインタは登録単語の内部の次の文字、右ポインタは左部分木に入らなかった登録単語のうちアルファベット順の次の登録単語の先頭文字



あさひ、あさやけ、いぬ、いぬごや、いぬくぼう、くし、くるま、くるまよけ

PATRICIA木

- PAT 木とは、トライにおいて、非分岐ノードを省略し、その代わりにノードに木の深さ情報を追加したもの



あさひ、あさやけ、いぬ、いぬごや、いぬくぼう、くし、くるま、くるまよけ

辞書との突合せの曖昧さ解消

- 品詞の文法的接続可能性による形態素解析法
- 文においてある品詞の次にくる品詞は文法的に限定されている。これを品詞接続可能性という。例えば、名詞の後には、名詞、助詞、助動詞はくるが形容詞、副詞は来ない。話言葉でなければ動詞も来ない。
- 例 「きがつく」→「き（名詞） が（助詞） つく（動詞）」OK
- →「きが（名詞） つく（動詞）」
- 接続コスト法
- 接続可能性を可能、不可能の２種類ではなく、確率のような連続数値を用いて表すと、ある品詞列とみなした場合の接続確率が計算できる。接続確率の最大の品詞列を選ぶ方法。

少し数学（確率論）の準備をします。

- 事象 e とは、ある確率変数 X の値が　x であること:　e: X=x
- 事象 e　の確率を　p(e),　p(X=x),　p(x)　などと書く。　$\sum_{i=1}^{N} p(e_i) = 1$
- 全事象 e1, e2, ….. , eN　（つまり事象数）とすると

- X=a とX=b　が同時に起こった場合の確率を同時確率といい、p(a,b)と書く。

- 条件付確率　$p(a \mid b) = \frac{p(a, b)}{p(b)}$

- ベイズの定理　$p(b_j \mid a) = \frac{p(a \mid b_j)\, p(b_j)}{p(a)} = \frac{p(a \mid b_j)\, p(b_j)}{\sum_{i=1}^{N} p(a \mid b_i)\, p(b_i)}$

- 従属性　p(A,B)>p(A)p(B) が普通だが、これはAが起これればある確からしさでBが起こるような場合もあるから。　←→ 排反性

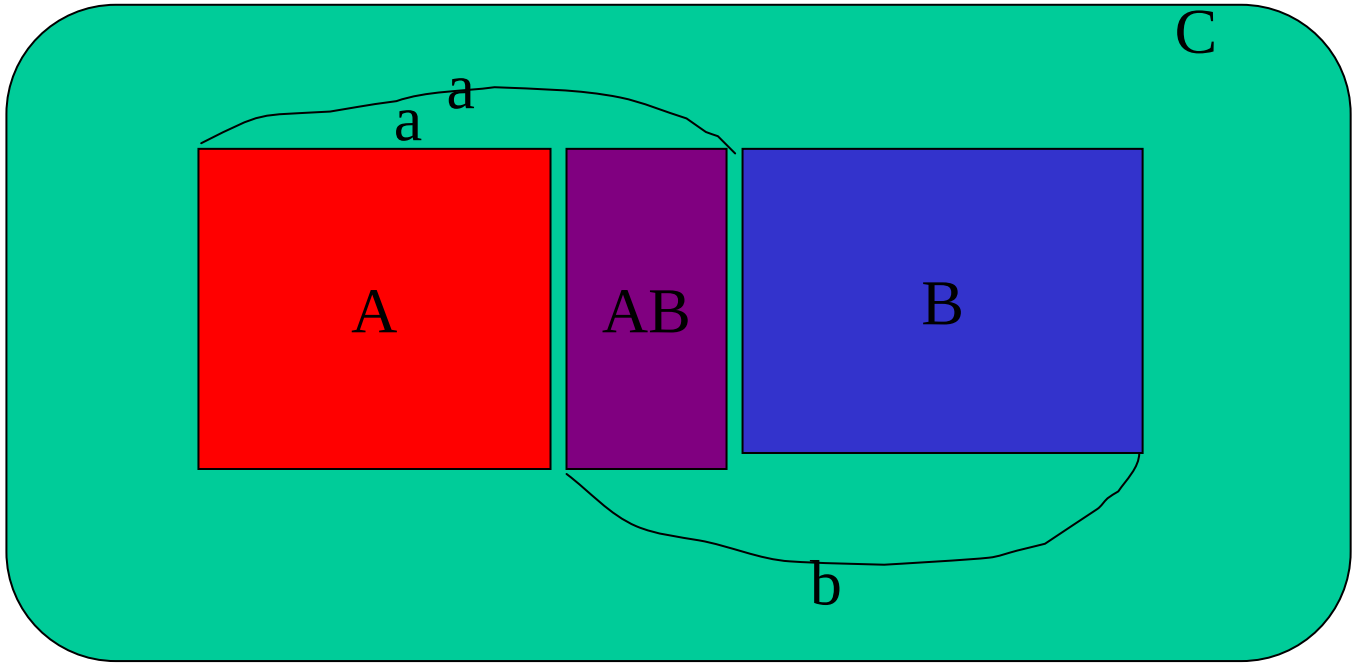
  仮にAは起これば必ずBも起こるならBはAに従属するといい、 p(A,B)=p(A)=p(B)
- 独立性　Aが起こっても次にBが起こるかどうかは影響されない場合、AとBは独立といい、。

  p(A,B)=p(A)p(B)

条件付確率

$$p(a \mid b) = \frac{p(a,\ b)}{p(b)}$$

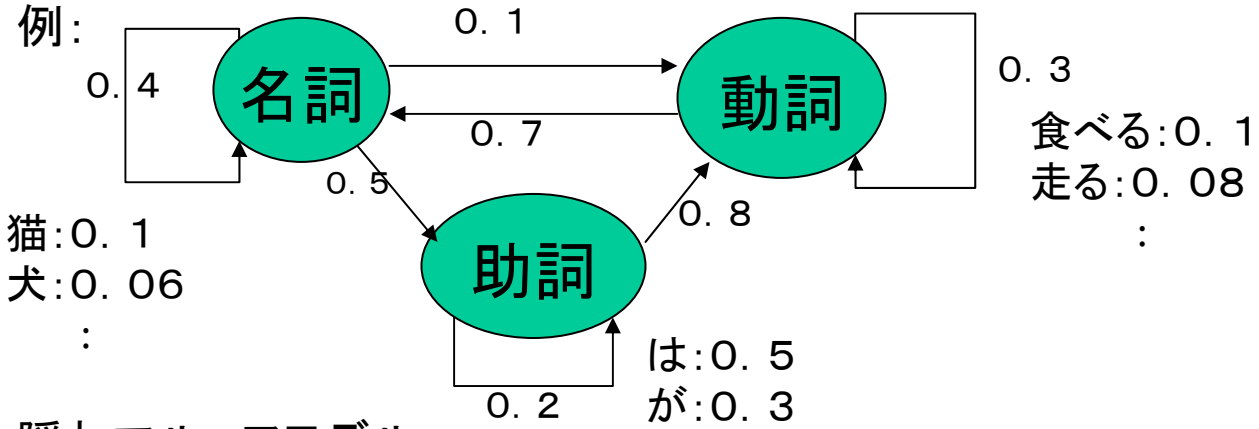
C+A+B+AB=N



$$p(a \mid b) = \frac{AB}{B + AB} = \frac{\frac{AB}{N}}{\frac{B + AB}{N}} = \frac{p(a,b)}{p(b)}$$

隠れマルコフモデルによる形態素と品詞

- 確率過程：いくつかの内部状態を持つ（抽象的機械）から種々の記号が次々と現れる過程。
- マルコフ過程： ある記号が現れる確率はその直前の状態にだけ依存する。また、次の状態への遷移確率も直前の状態によって決まる。
- 例：



- 隠れマルコフモデル：

外部から記号は観測できるが、内部状態は観測できない（直接分からない）

例えば、「犬が走る」という記号列が得られたとき、内部状態が

名詞→助詞→動詞 と遷移したことを知りたい。

- 品詞の遷移確率 p(ti|ti-1): 例 p(助詞|名詞)=0.5 p(動詞|名詞)=0.1
- 単語出現確率 p(w|t): 例 p(は|助詞)=0.5 p(走る|動詞)=0.08

動的計画法による形態素解析の定式化　その1

- 入力文Sを文字列 $S=(c_1 c_2 .....c_m)$
- 単語列　　$W=(w_1 w_2 ....w_n)$
- 品詞列　　$T=(t_1 t_2 .....t_n)$
- 単語の境界が与えられていない日本語の形態素解析は入力文を条件としたときの単語列と品詞列の同時確率P(W,T)を最大にする単語分割と品詞付与の組（W’，T’)を求めること。

$$(W',T')=\arg\max_{W,T} P(W,T|S)$$

- 辞書：　文字列から単語と品詞の対を求めるのが辞書Dである。すなわち、

$$D(c_k c_{k+1}...c_{k+l-1})=\{(w_1,t_1)(w_2,t_2)...\}$$

- よって、文字列からw,tへの変換はDによって行われ、その結果のP(W,T)を最大化する問題になる。
- さて、P(W,T)は品詞2グラム（連続する二つの品詞）$p(t_i|t_{i-1})$の生起確率と品詞と単語の対応する確率　　$p(w_i|t_i)$
の積である。

$$P(W,T)=\prod_{i=1}^{n} p(t_i|t_{i-1})\, p(w_i|t_i)$$

動的計画法による形態素解析の定式化　その2

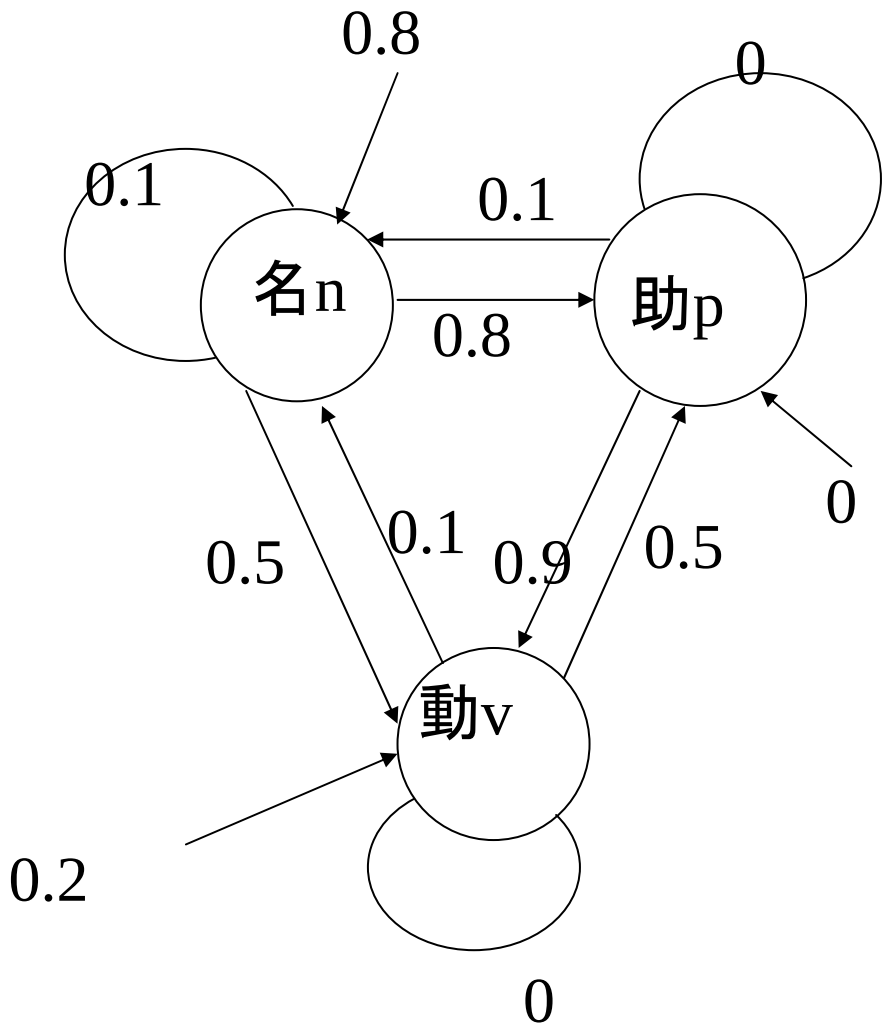
- P(W,T)を最大にする計算は全部の場合を計算すると膨大
- そこで、次のように順々に計算する。
- まず　$P(w_1,...,w_i,t_1,...,t_i)=\phi(w_i,t_i)$　と定義。
- すると　$\phi(w_i,t_i)=\max_{w_{i-1},t_{i-1}} \phi(w_{i-1},t_{i-1})p(t_i|t_{i-1})p(w_i|t_i)$
- まず　$\phi(w_{i-1},t_{i-1})$　を *i-1* 単語めまでの計算で求め、この値をつかって　$\phi(w_i,t_i)$　を計算する。
- これを *i=1,..n* まで繰り返してmax P(W,T) を求める。
  - これだと単語が切り出された状態での議論なので、
- 実際は入力文から１文字づつ読みながら、この計算を行う。
- 辞書Dの辞書引き速度は全体のパフォーマンスを左右する。Trie，PATRICIA木などの高速なディジタル木構造が使われる。

動的計画法による日本語形態素解析アルゴリズム

```
T0＝{（w0, t0）};   φ(w0,t0)=1;
for q＝0 to m     %文頭から1文字づつ読む
   foreach (wi-1,ti-1) in Tq     %q文字目までの部分解析結果の集合
      foreach (wi,ti) in D(Cq, Cq+1,…Cr) where q<r=<m
                                    %qからr文字目までの辞書引き
         begin
            if (wi,ti) is not in Tr then   %未登録な部分解析経路の追加
               begin
                  Tr =Tr U {(wi,ti)}; φ(wi,ti)=0;
               end
            %   最大のφを順次計算する。
            newP = φ（wi－1, ti－1）p(ti|ti-1)p(wi|ti)
            if newP > newpの計算と同一位置で終わる単語
                              に対するφ
              then φ(wi,ti)=newP;
         end
```

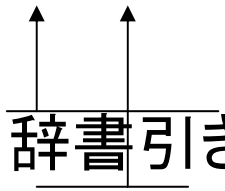
形態素解析アルゴリズムの動作例

- くるま →車　n　0.5
- くる →来る　v　0.5
- まで　→マデ p　0.5
- まつ →待つ　v　0.5
- まつ →松 n 0.5
- で → デ p 0.5



0 1 2 3 4 5 6

く る ま で ま つ



- 辞書引き ×
- (来る v) 内側のforeachは(来る v) に対してのみ動く
- q=0
  - T0={(w0,t0)} φ(w0,t0)=1
- q=1
  - for (w0,t0)に対して
    - For{(来るv)、(車n)}
    - T2= {(来るv)} φ(来る、v)=0
    - newp=φ(w0,t0)p(v|φ)p(来る|v)=0.1
    - 0.1 1 0.2 0.5
  - φ(来る、v)= 0.1

- T3={(車, n)} ϕ（車, n)=0
- newp=ϕ(w0,t0) p(n|ϕ) p(車|n)=0.4
- 1 0.8 0.5
- ϕ(車,n)=0.4

- q=2
  - for(来る,v)
  - for{（まで,p)}
    - T4={(まで,p), ϕ(まで,p)=0
    - newp=ϕ(来る,v) p(P|v) p(まで|p)=0.025
    - 0.1 0.5 0.5
    - ϕ(まで,p)=0.025

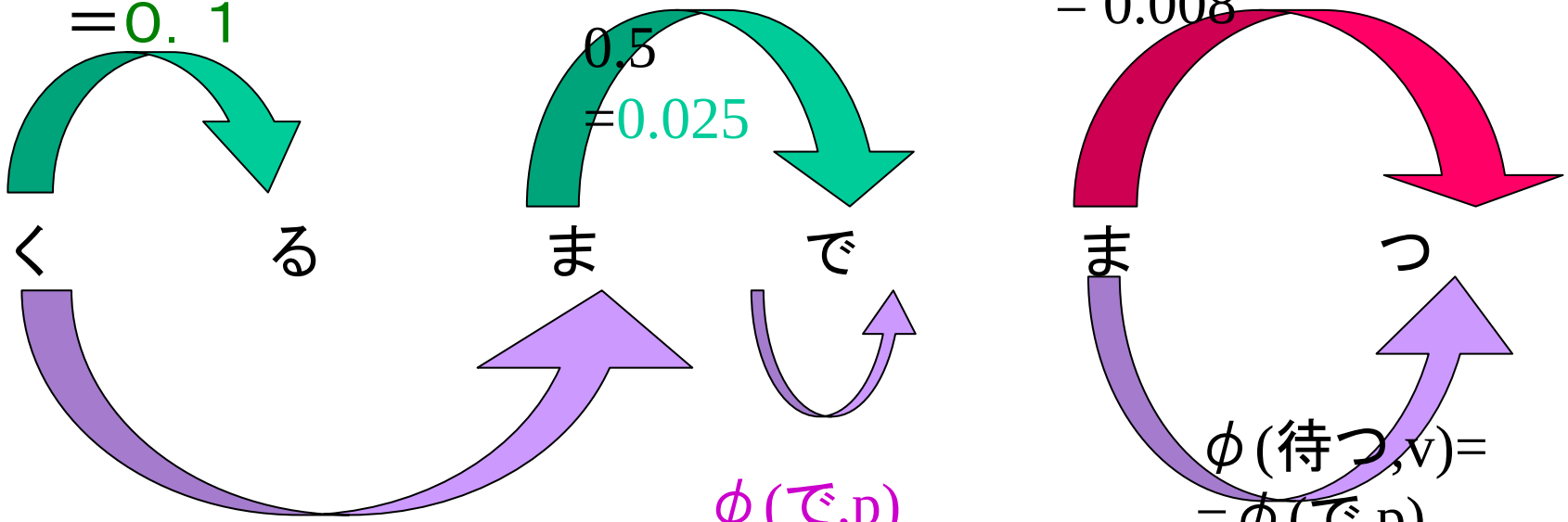
- q=3
- for (車, n)
- for(で,p)
  - T4={(で,p)}　ϕ(で,p)=0
  - newp=ϕ(車,n) p(p|n) p(で|p) =0.16>ϕ(まで,p)
  - 0.4 0.8 0.5
  - ϕ(で,p)=0.16

- q=4
- for{(まで,p)(で, p)}
- for{(待つ,v)(松, n)}
  - T6= {(待つ,v)} ϕ(待つ,v)=0
  - newp= ϕ(で,p) p(v|p) p(待つ|v)=0.072
  - 0.16 0.9 0.5
  - T6={(待つ,v)} ϕ(待つ,v)=0
  - newp= ϕ(で,p) p(n|p) p(松|v)=0.008
  - 0.16 0.1 0.5
- → ϕ(待つ,v)=0.072

φ(来る、n)
=φ(w0,t0)p(v|φ)
p(来る|v)
=1x 0.2 x 0.5
＝0．1

φ(まで,p)
= φ(来る,v)
p(P|v)
p(まで|p)
= 0.1 x 0.5 x 0.5
=0.025

φ(松,n)
=φ(で,p) p(n|p)
p(松|v)
=0.16 x 0.1 x 0.5
= 0.008



φ(車,n)
= φ(w0,t0) p(n|φ) p(車n)
= 1 x 0.8 x 0.5
=0.4

φ(で,p)
=φ(車,n)
p(p|n) p(で|p)
= 0.4 x 0.8 x 0.5
=0.16

φ(待つ,v)=
=φ(で,p)
p(v|p) p(待つ|v)
=0.16 x 0.9 x 0.5
=0.072

統計データからの動的計画法による形態素解析

- P(W,T)を最大化するアルゴリズムにおいて、$p(t_i|t_{i-1})$ $p(w_i|t_i)$ をどうやって入手するかが問題。
- すでに述べたように品詞の接続可能性や、接続コスト法における品詞接続確率が $p(t_i|t_{i-1})$ に相当する。
- 言語コーパスからの統計データによりこれらを計算することもできる。
- コーパスにおける頻度をCとすると

$$p(t_i \mid t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \qquad p(w_i|t_i) = \frac{C(w_i, t_i)}{C(t_i)}$$

前向きと後ろ向きの融合

- 今まで説明してきた動的計画法による形態素解析は前向きに計算を進めた。文の長さが長くなると計算量が指数関数的に増える。
- ある程度、前向きで計算し可能な部分解析結果を保持。
- 次に文末から文頭へ向かって解析する。
- 両者を組み合わせ、最大確率のP(W,T)を選ぶ。
- この方法だと、確率の大きい順に上位N個の解析結果を列挙できる。

実例

- juman
- ここではきものをぬぐ
- ここ　(ここ)　ここ　名詞形態指示詞
- で　(で)　で　格助詞
- はきもの　(はきもの)　はきもの　普通名詞
- を　(を)　を　格助詞
- ぬぐ　(ぬぐ)　ぬぐ　動詞　子音動詞ガ行　基本形
- EOS

実例

- ここでは、きものをぬぐ
- ここ　(ここ)　ここ　名詞形態指示詞
- で　(で)　で　格助詞
- は　(は)　は　副助詞
- 、　(、)　、　読点
- きもの　(きもの)　きもの　普通名詞
- を　(を)　を　格助詞
- ぬぐ　(ぬぐ)　ぬぐ　動詞　子音動詞ガ行　基本形
- EOS

にはにはにわがある

に　(に)　にる　動詞　母音動詞　基本連用形
は　(は)　は　副助詞
に　(に)　にる　動詞　母音動詞　基本連用形
はにわ　(はにわ)　はにわ　普通名詞
が　(が)　が　格助詞
ある　(ある)　ある　動詞　子音動詞ラ行　基本形
EOS

東京大学全学自由ゼミ言語を情報する

東京　(とうきょう) 東京　地名

大学　(だいがく)　大学　普通名詞

全学　(ぜんがく)　全学　普通名詞

自由　(じゆう)　自由　普通名詞

ゼミ　(ゼミ)　ゼミ　カタカナ

言語　(げんご)　言語　普通名詞

を　(を)　を　格助詞

情報　(じょうほう) 情報　普通名詞

する　(する)　する　動詞　サ変動詞　基本形

EOS