Introduction to Natural Language Processing

“Ko ko de ha ki mo no wo nu gu ko to”

Hiroshi Nakagawa

(Information Technology Center; Mathematical Informatics, Graduate School of Information Science and Technology; Graduate School of Interdisciplinary Information Studies, The University of Tokyo)

nakagawa@dl.itc.u-tokyo.ac.jp

http: //www.r.dl.itc.u-tokyo.ac.jp/~nakagawa/

“*Ko ko de ha ki mo no wo nu gu ko to*”

- The sentence “*Ko ko de ha ki mo no wo nu gu ko to*” can be interpreted in two meanings: *Koko-de-hakimono-wo-nugu-koto* 'Take off your shows here', or *koko-de-wa-kimono-wo-nugu-koto* 'Take off your cloth here'.
- The sentence “*Ni wa ni ha ni wa ga a ru*” can be also interpreted in two meanings: *Niwa-ni-wa-niwa-ga-aru* 'There is a garden in a yard' or *Niwa-ni-haniwa-ga-a-ru* 'There is a clay figure in a yard'.
- Morphological Analysis is to find word break points.
- Morpheme is a minimal unit of language that is larger than a character though unbreakable into even smaller structures.

Why Morphological Analysis?

- The identification of words is prerequisite for constituting the meaning of the sentences.
- Does the use of *Kanji,* rather than the writing in *Kana* (sound symbolism), make it easy to understand the sentence?
- Word processor employs *Kana-Kanji* conversion for Japanese.
- Phonetic recognition is presented by processing phonetic sounds (*Kana*), not *Kanji,* or *Katakana* (the other Japanese writing system along with *Kanji* and *Hiragana*).

Differences between Japanese & English

- In English, word break points are clear due to space in writing. (However, the phonetic recognition in English involves in the same issue as in Japanese.)
- Inflectional Language:

  e.g. apples -> apple

  studied -> study
- Root words and parts of speech (POS) must be identified from inflectional expressions.
- In Japanese, word break points are unclear in writing.
- For agglutinative language, it is necessary to break into words and identify POS.

Morphological Analysis of Japanese Language

- Heuristic Methods for Morphological Analysis of Japanese Language
- Longest Match Method:

  To form the longest morpheme from the remaining characters, which is listed in a dictionary, starting from the beginning of the sentence.

  e.g. *Zen Koku To Doh Fu Ken Gi Kai Gi Tyo Seki* 'A seat for chairperson of a national meeting held by prefectural government members' -> *Zen-Koku-To-Doh-Fu-Ken-Gi-Kai-Gi-Tyo-Seki*
- Minimum Number of Segment Method

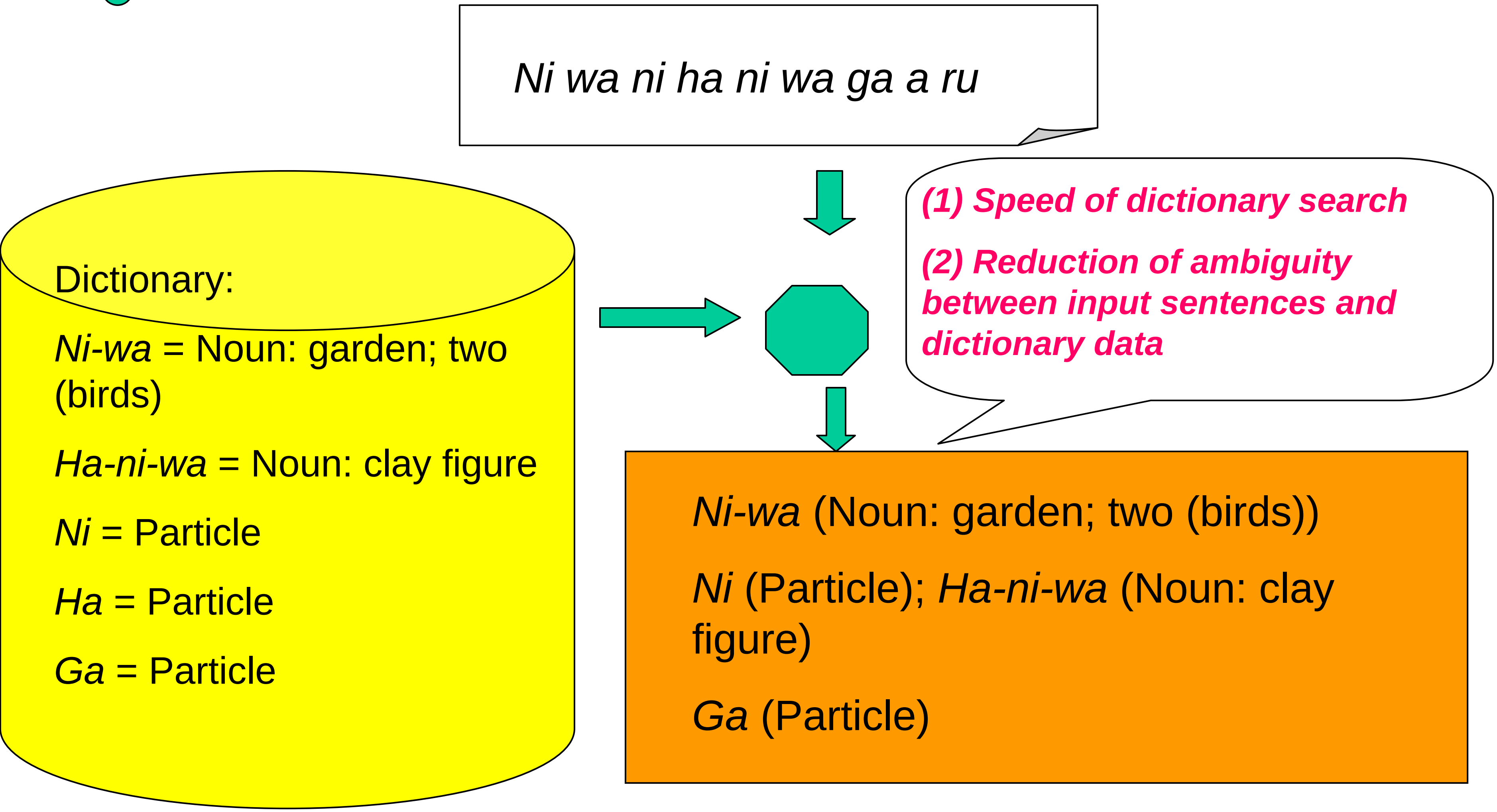
  To reduce into every breakable units as listed in a dictionary, and determine the minimal number of break points.

  e.g. *Zen Koku To Doh Fu Ken Gi Kai Gi Tyo Seki* 'Seats for chairpersons of prefectural government assemblies in Japan' -> *Zen-Koku-To-Doh-Fu-Ken-Gi-Kai-Gi-Tyo-Seki*
- Character Type Method

  To identify word break points when character types (or writing systems) change.

  e.g. *Ka Ra Fu Ru Na Den Shi Me I Ru* 'A colorful e-mail' -> *Karafuru (Katakana) Na (Hiragana) Denshi (Kanji) Meiru (Katakana)*

Framework for Morphological Analysis based on Grammatical Information



Ni wa ni ha ni wa ga a ru

Dictionary:

Ni-wa = Noun: garden; two (birds)

Ha-ni-wa = Noun: clay figure

Ni = Particle

Ha = Particle

Ga = Particle

(1) Speed of dictionary search

(2) Reduction of ambiguity between input sentences and dictionary data

Ni-wa (Noun: garden; two (birds))

Ni (Particle); *Ha-ni-wa* (Noun: clay figure)

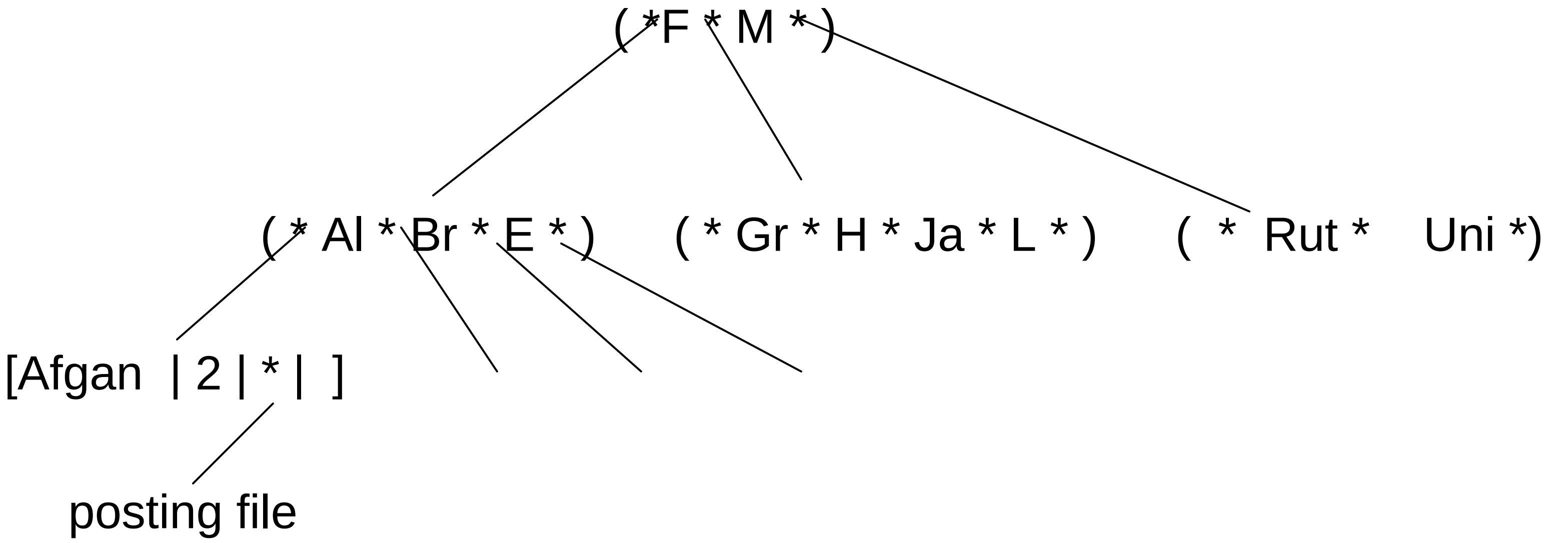
Ga (Particle)

Speed of Dictionary Search

- Important to achieve a high-speed search in memory containing numerous words to match with a part of input sentences.
- Several approaches for data structure:
  - B-Tree
  - Trie
  - PATRICIA Tree

B-Tree

- Featured for its high-speed keyword search in memory containing numerous words.



(*F * M *)

(* Al * Br * E *)

(* Gr * H * Ja * L *)

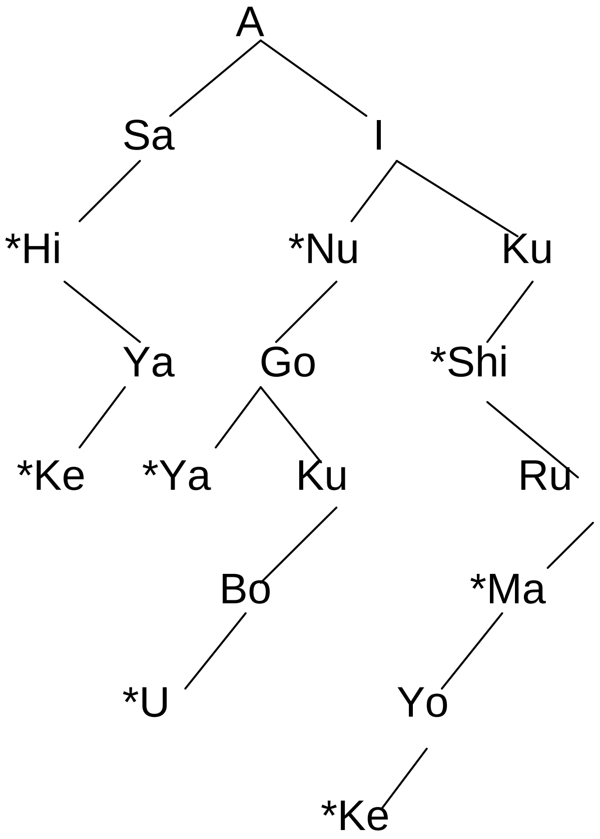
(* Rut *   Uni *)

[Afgan | 2 | * |]

posting file

Trie

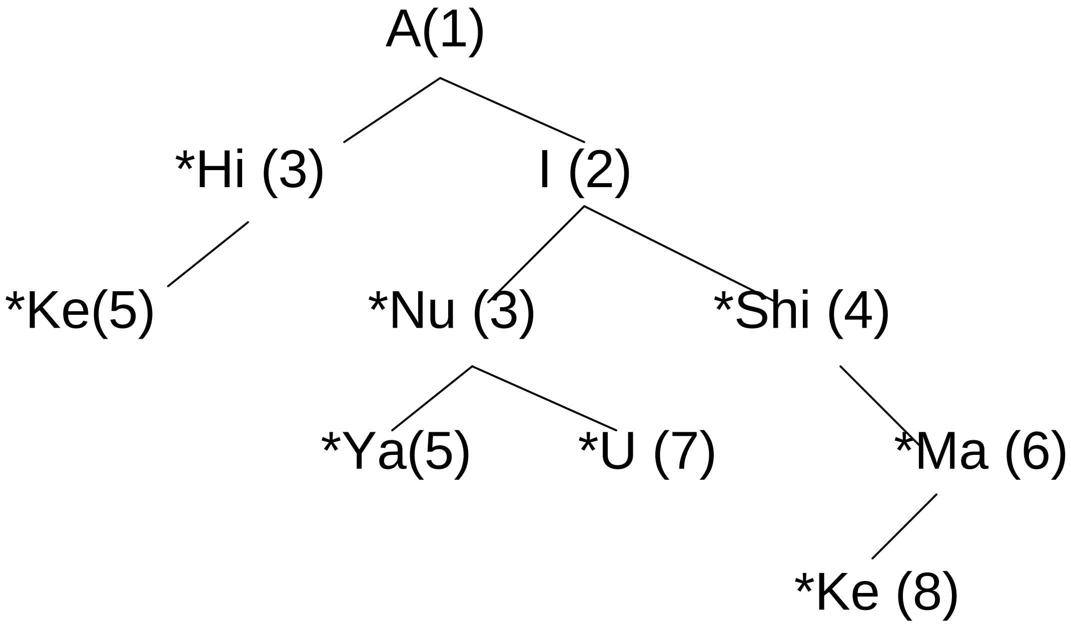
- "Trie" has comparison based tree structures. Left pointers represent the second leftmost character whose parent words are listed in memory. Searching for the next possible words in alphabetical order, right pointers are created to represent the first initial of the next word given.



a-sa-hi; a-sa-ya-ke; i-nu; i-nu-go-ya; i-nu-ku-bo-u; ku-shi; ku-ru-ma; ku-ru-ma-yo-ke

PATRICIA Tree

- In “PATRICIA Tree”, some branches in Trie are eliminated if they have no child branch. PAT Tree gives each node the counter for the number of the depth of tree.



A(1)

*Hi (3)

I (2)

*Ke(5)

*Nu (3)

*Shi (4)

*Ya(5)

*U (7)

*Ma (6)

*Ke (8)

a-sa-hi; a-sa-ya-ke; i-nu; i-nu-go-ya; i-nu-ku-bo-u; ku-shi; ku-ru-ma; ku-ru-ma-yo-ke

Solve Ambiguity Issue In Matching Dictionary

- Morphological analysis based on grammatical connectivity of parts of speech:
- To structure a sentence, grammar is determiners as to which parts of speech come in a sentence in what order. This characteristic is called Grammatical Connectivity of Parts of Speech.

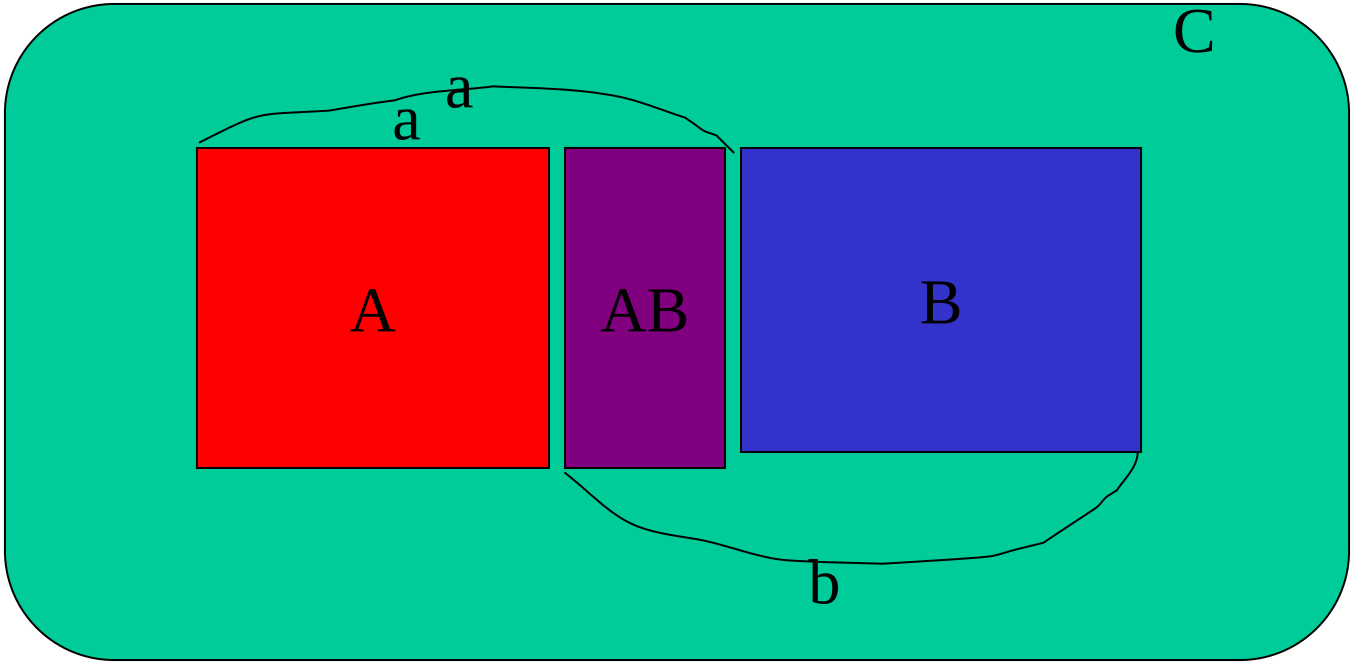
  e.g. Noun can be followed by noun, particle, or auxiliary verb, but not by adjective or adverb. No verb can follow noun unless in spoken language.

  e.g. *Ki Ga Tsu Ku* -> OK: *Ki* (Noun) *Ga* (Particle) *Tsu-ku* (Verb) ->*Ki-ga* (Noun)*Tsu-ku* (Verb)
- Connectivity Cost Method:
- Grammatical connectivity is not expressed by a simple True or False. It is represented by continuous numbers to determine the probability of grammatical connection of POS sequence. In this approach, a POS sequence with the highest probability is to be selected.

Conditional Probability

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

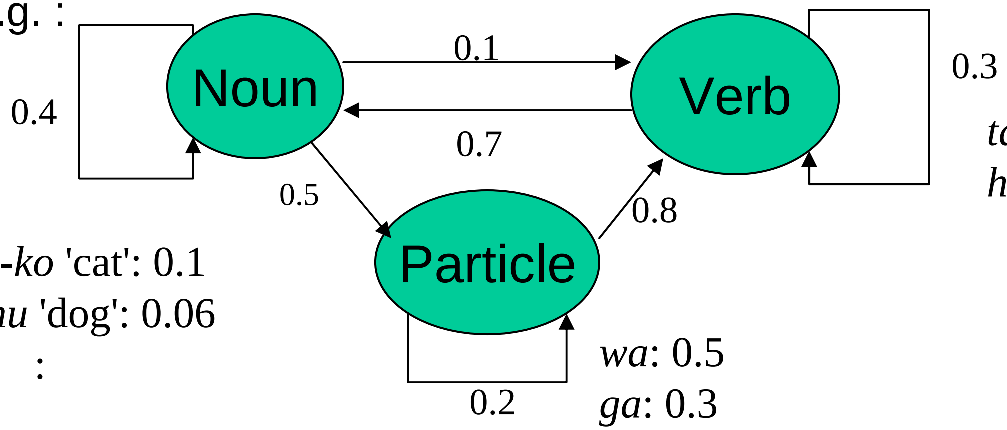
$$C + A + B + AB = N$$



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

Hidden Markov Model (HMM): Morpheme and Parts of Speech

- Probability Process: In this process, a string of symbols are obtained from abstract machines pertaining several inner states.
- Markov Process: The probability of an observation symbol is determined according to the associated probability of its previous state. Transition to the next state is also governed by the previous state.
- e.g. :



ne-ko 'cat': 0.1
i-nu 'dog': 0.06
:

wa: 0.5
ga: 0.3

ta-be-ru 'eat': 0.1
ha-shi-ru 'run': 0.08
:

- Hidden Markov Model (HMM):

  Symbols are observable from outer state although the inner state is unobservable (no direct observation). e.g. When a string of symbols *I Nu Wa Ha Shi Ru* 'A dog runs.' is given, how can we confirm the transition of inner state (Noun->Particle->Verb)?
- Transition matrix of parts-of-speech p(ti|ti-1): e.g. p(Particle|Noun) = 0.5 p(Verb|Noun) = 0.1
- Probability of word emergence p(w|t): e.g. p(*wa*|Particle) = 0.5 p(*ha-shi-ru*|Verb) = 0.08

Formulation of Morphological Analysis by Dynamic Programming (DP) -No. 1

- Input Sentence *S* represented by String *S* = $(c_1 c_2 .....c_m)$
- Word String *W* = $(w_1 w_2 ....w_n)$
- POS String *T* = $(t_1 t_2 .....t_n)$
- In morphological analysis for Japanese, since it do not indicate apparent word breaks with space, the most probable combination of word breaks with POS *(W',T')* must be obtained so that the joint probability of word string and POS string *P(W,T)* becomes maximum given the input sentence.

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

- Dictionary: A pair of word and POS is obtained (Dictionary *D)* from the given string.

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

- Therefore, the string is converted to *w,t* by *D*. The question is how to maximize *P(W,T).*
- *P(W,T)* is now obtained as a product of the occurrence probability of POS 2-gram (two consecutive POS) $p(t_i | t_{i-1})$ and the probability of word and its correspondent POS $p(w_i | t_i)$.

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

Formulation of Morphological Analysis by Dynamic Programming (DP) -No. 2

- □ Obtaining max *P(W,T)* represents huge computational requirements.
- □ So, take the following step:
- □ Define $P(w_1,...,w_i,t_1,...,t_i)=\phi(w_i,t_i)$
- □ Then, $\phi(w_i,t_i)=\max_{w_{i-1},t_{i-1}} \phi(w_{i-1},t_{i-1})p(t_i|t_{i-1})p(w_i|t_i)$
- □ Obtain $\phi(w_{i-1},t_{i-1})$ to the *i-1* order.

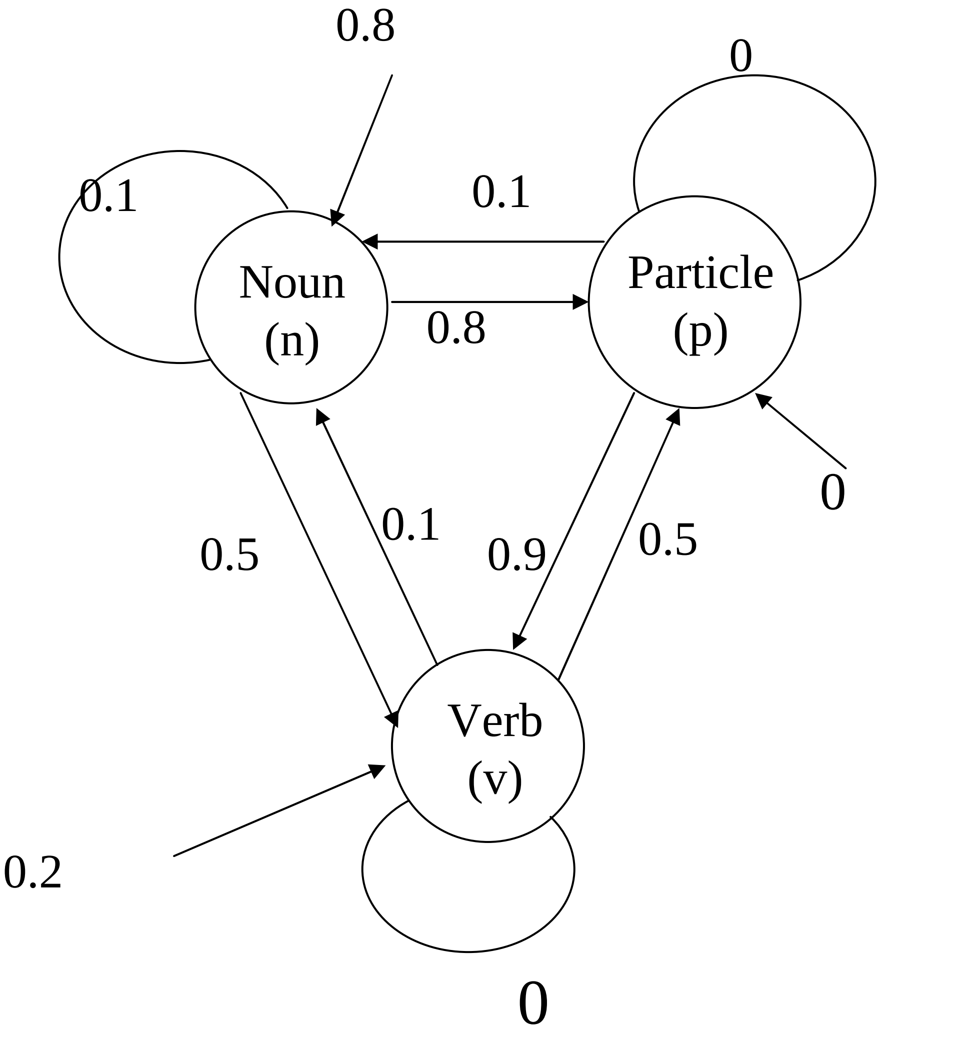
  Use resulting value to obtain $\phi(w_i,t_i)$
- □ Continue till *i = 1,..n* to obtain max *P(W,T)*.
  - □ In the argument above, words are extracted from the input sentence…
- □ In reality, computer reads characters one by one in the input sentence to obtain the result.
- □ The speed of dictionary search for Dictionary *D* is a crucial element which affects the entire performance. Therefore, high-speed digital tree structures, such as Trie and PATRICIA Tree, are employed.

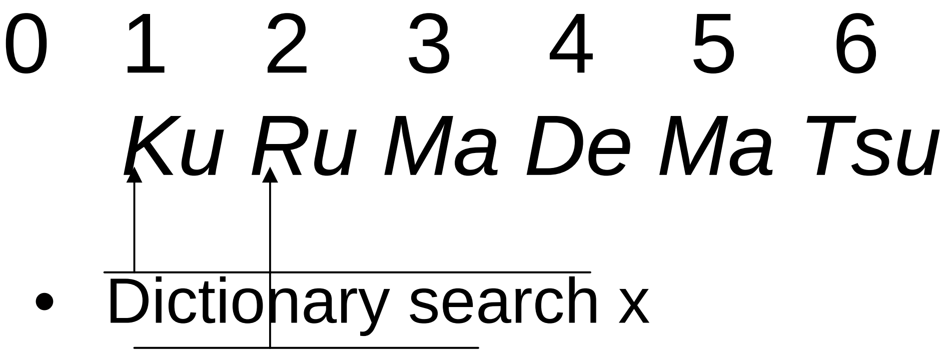
Algorithm of Morphological Analysis by Dynamic Programming (DP)

```
T0 =  {(w0,t0) };  $\phi$(w0,t0) = 1;
for q = 0 to m     % Read characters one by one from the beginning of an input sentence.
    foreach (wi-1,ti-1) in Tq      % Set of partial analysis to the q-th character.
        foreach (wi,ti) in D(Cq,Cq+1,…Cr) where q<r = <m
                                                % Search the q-th through r-th characters in Dictionary.
            begin
                if (wi,ti) is not in Tr then   % Create for unregistered partial analysis process
                    begin
                        Tr  = Tr U  {(wi,ti) };  $\phi$(wi,ti) = 0;
                    end
                % Obtain max  $\phi$.
                newP  =   $\phi$(wi - 1,ti - 1)p(ti|ti-1)p(wi|ti)
                if newP > newp
                  %  $\phi$  for word whose endpoint is at the identical location in the above formula.
                  then  $\phi$(wi,ti) = newP;
            end
```

Examples: Operation of Algorithm for Morphological Analysis

- *Ku-ru-ma* ->'Car' n 0.5
- *Ku-ru* ->'come' v 0.5
- *Ma-de* ->'until' p 0.5
- *Ma-tsu* ->'wait' v 0.5
- *Ma-tsu* ->'pine tree' n 0.5
- *De* ->'at' p 0.5





0 1 2 3 4 5 6

Ku Ru Ma De Ma Tsu

- Dictionary search x
- (*Ku-ru* 'come' v) Inner foreach can be applied only to (*Ku-ru* 'come' v).
- q = 0
  - T0 = {(w0,t0) } ϕ(w0,t0) = 1
- q = 1
  - for (w0,t0)
    - For {(*Ku-ru* 'come' v), (*Ku-ru-ma* 'car' n)}
    - T2 = {(*Ku-ru* 'come' v)} ϕ(*Ku-ru* 'come', v) = 0
    - newp = ϕ(w0,t0)p(v|ϕ)p(*Ku-ru* 'come'|v) = 0.1
    - 0.1 1 0.2 0.5
  - ϕ(*Ku-ru* 'come', v) = 0.1

  - T3 = {(*Ku-ru-ma* 'car', n)} ϕ(*Ku-ru-ma* 'car', n) = 0
  - newp = ϕ(w0,t0) p(n|ϕ) p(*Ku-ru-ma* 'car'|n) = 0.4
  - 1 0.8 0.5
  - ϕ(*Ku-ru-ma* 'car',n) = 0.4
- q = 2
  - for(*Ku-ru* 'come',v) for {(*Ma-de* 'until',p) }
    - T4 = {(*Ma-de* 'until',p), ϕ(*Ma-de* 'until',p) = 0
    - newp = ϕ(*Ku-ru* 'come',v) p(P|v) p(*Ma-de* 'until'|p) = 0.025
    - 0.1 0.5 0.5
    - ϕ(*Ma-de* 'until',p) = 0.025

- q = 3
- for (*Ku-ru-ma* 'car', n)
- for(*De* 'at',p)
  - T4 = {(*De* 'at',p)} ϕ(*De* 'at',p) = 0
  - newp = $\underset{0.4}{\phi(\textit{Ku-ru-ma}\ \text{'car'},n)}\ \ \underset{0.8}{p(p|n)}\ \ \underset{0.5}{p(\textit{De}\ \text{'at'}|p)}$

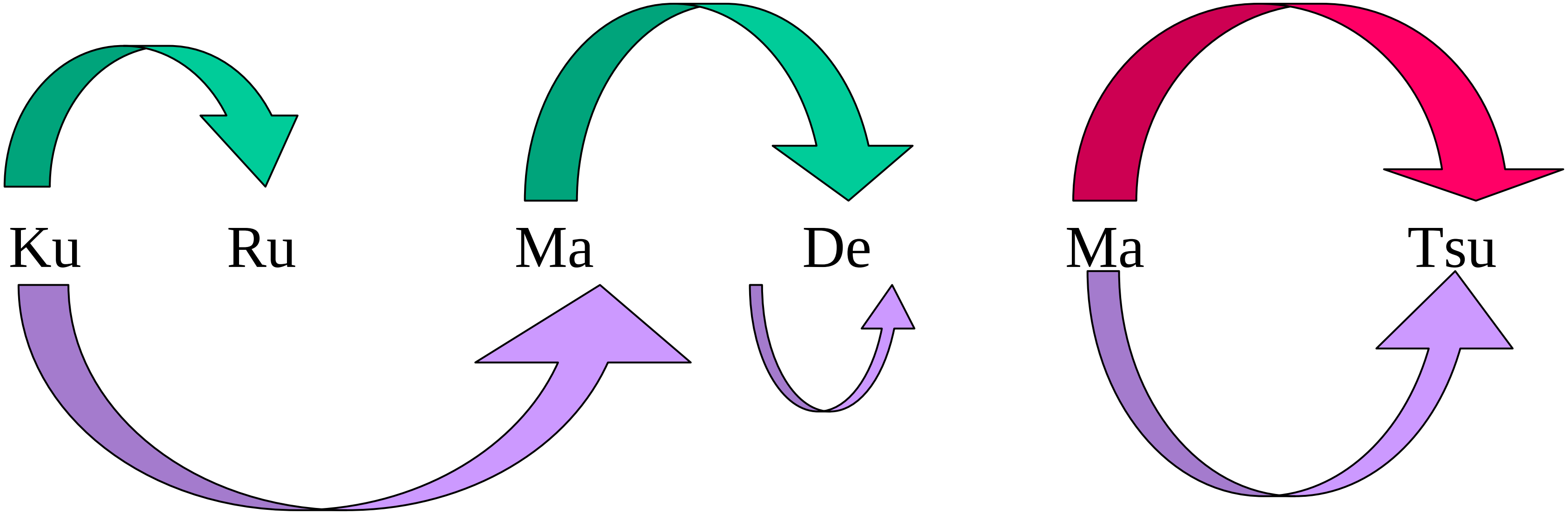
    = 0.16 > ϕ(*Ma-de* 'until',p)
  - ϕ(*De* 'at',p) = 0.16

- q = 4
- for {(*Ma-de* 'until',p)(*De* 'at', p)}
- for {(*Ma-tsu* 'wait',v)(*Ma-tsu* 'pine tree', n)}
  - T6 = {(*Ma-tsu* 'wait',v)} ϕ (*Ma-tsu* 'wait',v) = 0
  - newp = ϕ (De 'at',p) p(v|p) p(*Ma-tsu* 'wait'|v) = 0.072
  - 0.16 0.9 0.5
  - T6 = {(*Matsu* 'wait',v)} ϕ (*Matsu* 'wait',v) = 0
  - newp = ϕ (De 'at',p) p(n|p) p(*Ma-tsu* 'pine tree'|v) = 0.008
  - 0.16 0.1 0.5
- -> ϕ (*Matsu* 'wait',v) = 0.072

ϕ (*Ku-ru* 'come', n)
= ϕ (w0,t0)p(v| ϕ)
p(*Ku-ru* 'come'|v)
= 1 x 0.2 x 0.5
= 0.1

ϕ (*Ma-de* 'until',p)
= ϕ (*Ku-ru* 'come',v)
p(P|v) p(*Ma-de*
'until'|p)
= 0.1 x 0.5 x 0.5
= 0.025

ϕ (*Ma-tsu* 'pine tree',n)
= ϕ (*De* 'at',p) p(n|p)
p(*Ma-tsu* 'pine tree'|v) =
0.16 x 0.1 x 0.5
= 0.008



ϕ (*Ku-ru-ma* 'car',n)
= ϕ (w0,t0) p(n| ϕ)
p(*Ku-ru-ma* 'car'n)
= 1x 0.8 x 0.5
= 0.4

ϕ (*De* 'at',p)
= ϕ (*Ku-ru-ma*
'car',n) p(p|n)
p(*De* 'at'|p)
= 0.4 x 0.8 x 0.5
= 0.16

ϕ (*Matsu* 'wait',v)
= ϕ (*De* 'at',p)
p(v|p) p(*Matsu*
'wait'|v)
= 0.16 x 0.9 x 0.5
= 0.072

Morphological Analysis by Dynamic Programming (DP) using Statistical Data

- □ An issue is how to obtain $p(t_i|t_{i-1})$ $p(w_i|t_i)$ in the algorithm for a max *P(W,T).*
- □ As discussed, $p(t_i|t_{i-1})$ corresponds to POS connectivity and the probability of POS connectivity in Connectivity Cost Method.
- □ A language corpus can be used to obtain statistical data for the algorithm.
- □ *C* as frequency in a language corpus:

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

Fusion of Forward and Backward Approaches

- Morphological analysis by dynamic programming (DB) proceeds with a forward approach as described. However, the longer the input sentence, the heavier the computational requirements, which would increase at an exponential rate.
- To some extent, the forward approach should be applied for the results of partial analyses to be stored.
- Then, analyze from the end of the sentence to the beginning.
- By combining both approaches, determine the max *P(W,T).*
- This method returns the highest to the N-th probability in order.

Sample

- juman
- *Koko-de-hakimono-wo-nugu-koto* 'Take off your shows here'.
- *koko* (*koko*) *koko* noun determinar
- *de* (*de*) *de* case particle
- *hakimono* (*hakimono*) *hakimono* common noun
- *wo* (*wo*) *wo* case particle
- *nugu* (*nugu*) *nugu* verb basic form of a consonantal verb in *ga-gyō*
- EOS

Sample

- *koko-de-wa-kimono-wo-nugu-koto* 'Take off your cloth here'.

koko	(*koko*)	*koko* noun determinar
de	(*de*)	*de* case particle
wa	(wa)	wa adverbial particle
、	(、)	、 *to-ten* ‘reading mark’
kimono	(*kimono*)	*kimono* common noun
wo	(*wo*)	*wo* case particle
nugu	(*nugu*)	*nugu* verb basic form of a consonantal verb in *ga-gyō*

- EOS

Niwa-ni-<u>haniwa</u>-ga-a-ru 'There is a clay figure in a yard'.

ni	(*ni*)	*ni-ru* verb vocalic verb continuative
wa	(*wa*)	*wa* adverbial particle
ni	(*ni*)	*ni-ru* verb vocalic verb continuative
haniwa	(*haniwa*)	*haniwa* common noun
ga	(*ga*)	*ga* case particle
a-ru	(*a-ru*)	*a-ru* verb basic form of a consonantal verb in *ra-gyō*
EOS		