# Mathematical Modeling of Ākāṅkṣā and Sannidhi for Parsing Sanskrit

Amba Kulkarni, Devanand Shukl, and Sheetal Pokar

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad

$15^{th}$ World Sanskrit Conference, Delhi
$6^{th}$ Jan., 2012

# Outline

# Śābdabodhaḥ

Śābdabodha is an understanding that arises from a linguistic utterance.

Three schools of Śābdabodha: vyākaraṇa, nyāya and mīmāṁsā

Main Difference: mukhya-viśeṣya (chief qualificand / Head)

The process of Śābdabodha involves parsing or vākyaviśleṣaṇam as one of the steps.

Parsing: A process of analysing a text to determine its grammatical structure and syntactic relations between various units.

# Non-determinism in Parsing

Grammar typically specifies rules for generation.

Analysis is an inverse process.

Inverse process may involve non-determinism.

Consider for example the following two sūtras:

- anabhihite ( 2.3.1)
- kartṛkaraṇayos tṛtīyā (2.3.18)

Consider for example the following two sūtras:

- anabhihite ( 2.3.1)
- kartṛkaraṇayos tṛtīyā (2.3.18)

While generating a sentence using these two rules, there is no non-determinism.

Consider for example the following two sūtras:

- anabhihite ( 2.3.1)
- kartṛkaraṇayos tṛtīyā (2.3.18)

While generating a sentence using these two rules, there is no non-determinism.

vaktṛ vivakṣā

hanana: kriyā
rāma: kartā
bāṇa: karaṇa
vālī: karma
voice: passive

vibhakti = f(dhātu, voice, kāraka)

Consider for example the following two sūtras:

- anabhihite ( 2.3.1)
- kartṛkaraṇayos tṛtīyā (2.3.18)

While generating a sentence using these two rules, there is no non-determinism.

vaktṛ vivakṣā

hanana: kriyā
rāma: kartā
bāṇa: karaṇa
vālī: karma
voice: passive

vibhakti = f(dhātu, voice, kāraka)

rāmeṇa bāṇena vālīḥ hanyate

However, given a sentence,

rāmeṇa bāṇena vālīḥ hanyate.

the analysis may lead to non-determinism as follows:

rāma and bāṇa, both are in 3rd case, and hence both of them are eligible candidates for kartā and karaṇa.

World Knowledge or yogyatā decides the appropriate role for each of them.

# Śābdabodha Factors

Factors useful for Śābdabodhaḥ

- ākāṅkṣā
- yogyatā
- tātparya
- sannidhi

# Ākāṅkṣā

Literally it means ‘desire’ on part of the listner (jijñāsā).

# Ākāṅkṣā

Literally it means 'desire' on part of the listner (jijñāsā).
Is it Psychological or Syntactic?

# Ākāṅkṣā

Literally it means 'desire' on part of the listner (jijñāsā).
Is it Psychological or Syntactic?
Naiyāyikas: Syntactic

dvāram = dvāra + am
'am' has an expectancy.

dvāram = dvāra + am
'am' has an expectancy.
This expectancy is not one way, but mutual.
The requirement of a karma in a verb such as 'pidhehi' is based on the usage of a verb.

# sannidhi

Tarkasa.ngraha

*padānām avilambena uccāraṇam*,
utterance of words without any gap,

# sannidhi

Tarkasa.ngraha

*padānām avilambena uccāraṇam*,
utterance of words without any gap,

*avyavadhānena padajanya padārthopasthitiḥ*
the presentation of word meanings without any intervention.

Viśvanātha Pañcānan in Nyāyakusumāñjali gives the following examples
Example 1:
**giriḥ bhuktam agnimān devadattena**
gloss: hill is_eaten firy by_Devadatta

Example 2:

**nīlo ghaṭaḥ dravyaṁ paṭaḥ**

gloss: blue pot matter cloth

Two cognitions:

- The pot is blue and the cloth is a matter.
  **nīlo ghaṭaḥ dravyaṁ paṭaḥ**
- The cloth is blue and the pot is a matter.
  **nīlo ghaṭaḥ dravyaṁ paṭaḥ**

# Computational Perspective

What is a parse?
A dependency parse: a tree
Words: nodes
Relations: labelled directed edges

Starting point?
To start with should we assume that potentially every word is related to every other word?

Starting point?
To start with should we assume that potentially every word is related to every other word?

No.

Starting point?
To start with should we assume that potentially every word is related to every other word?

No.

Ākāṅkṡā constraints the initial number of edges.

# Clues for possible relations

Clues for possible relations:

- abhihitatva
- vibhakti
- avyaya
- samānādhikaraṇa
- Tiṅantas (sakarmaka, akarmaka, any special kāraka requirement)

# Clues for possible relations

abhihitatva

tiṅ, kṛt, taddhita, samāsa

rāmaḥ vanam gacchati
rāmeṇa vanam gamyate

# Clues for possible relations

vibhaktiḥ: n-v / n-n

# Clues for possible relations

vibhaktiḥ: n-v / n-n

upapadavibhaktiḥ and upapadam: n-n / n-v
rāmeṇa saha sītā vanaṁ gacchati.
grāmaṁ paritaḥ vṛkṣāḥ santi.

# Clues for possible relations

vibhaktiḥ: n-v / n-n

upapadavibhaktiḥ and upapadam: n-n / n-v
rāmeṇa saha sītā vanaṁ gacchati.
grāmaṁ paritaḥ vṛkṣāḥ santi.

avyayas: a -> v / a -> n

rāmaḥ eva sundaraḥ
rāmaḥ na gacchati

# Clues for possible relations

samānādhikaraṇa:

śvetaḥ aśvaḥ dhāvati.

aśvaḥ śvetaḥ asti.

# Which relations: Explicit or Implicit

Which relations to represent – Explicit or Implicit?

samānakartṛkayoḥ pūrvakāle (3.4.21)

ktvā marks pūrvakālīnatva or kartṛtva or both?

rāmaḥ dugdham pītvā śālām gacchati.

# Explicit(abhihita) or Implicit(ākṣipta)

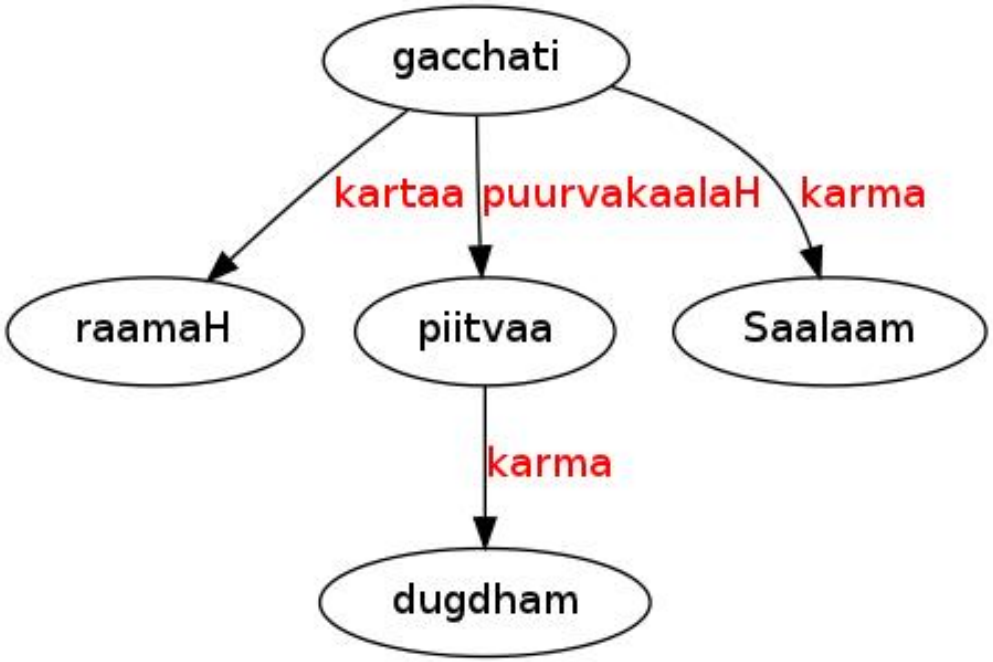
Bhartṛhari in Vākyapadīyam states (3.7.81-82),

pradhānetayor yatra dravyasya kriyayoḥ pṛthak
śaktir guṇāśrayā tatra pradhānam anurudhyate 3.7.81
pradhānaviṣayā śaktiḥ pratyayenābhidhīyate
yadā guṇe tadā tadvad anuktāpi prakāśate 3.7.82

i.e., in case X is an argument of both the main verb as well as the subordinate verb, it is the main verb which assigns the case and the relation of X to the sub-ordinate verb gets manifested even without any other marking.

# Explicit(abhihita) or Implicit(ākṣipta)

rāmaḥ dugdham pītvā śālām gacchati.
rāmeṇa dugdham pītvā śālā gamyate.


gacchati
kartaa
puurvakaalaH
karma
raamaH
piitvaa
Saalaam
karma
dugdham

# Sannidhi (Proximity)

**giriḥ bhuktam agnimān devadattena**

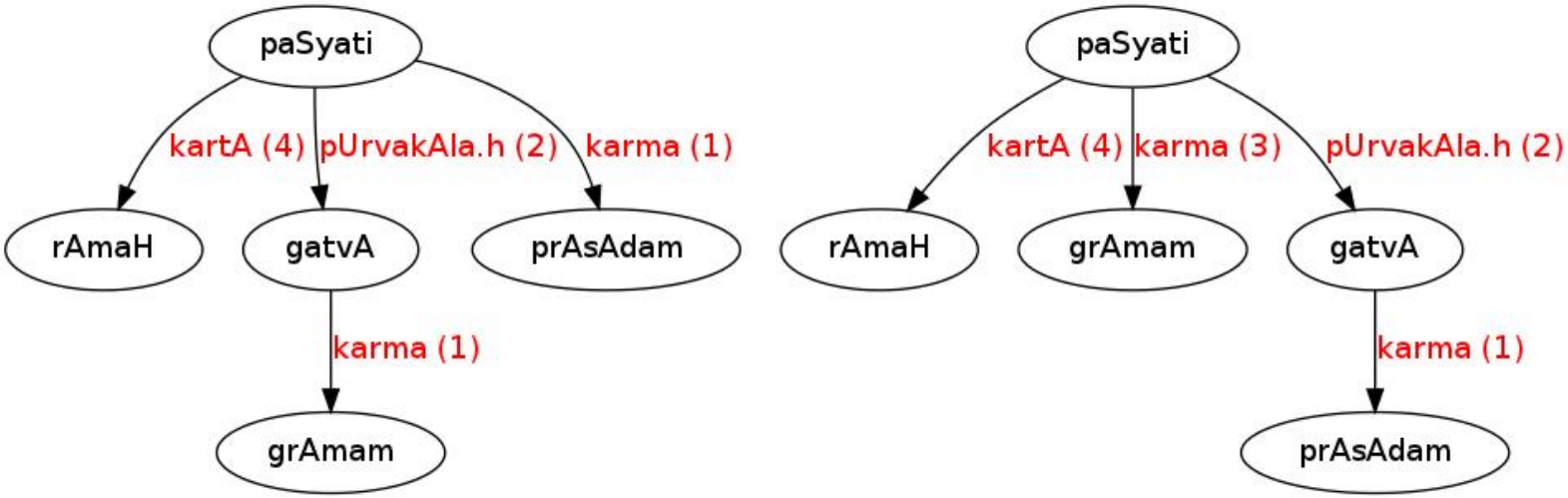
# Sannidhi (Proximity)

**giriḥ bhuktam agnimān devadattena**

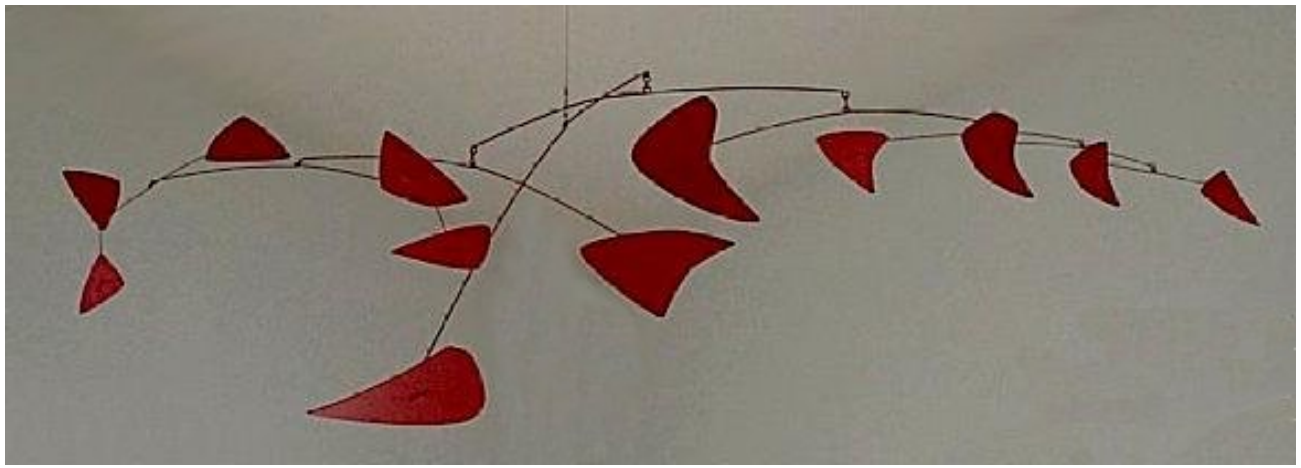
No crossing of edges

# Sannidhi (Proximity)

rāmaḥ grāmam gatvā prāsādam paśyati.



# Sannidhi (Proximity)



Suggested by Staal (1967) and
further worked out by Gillon (1993).

*Ākāṅkṡā*: To draw the potentital edges between nodes.
*Sannidhiḥ*: To forbid crossing of edges.

# Mathematical Model

Words: nodes, and
Relations: directed labelled edges.

Given a Graph G with *n* nodes, the task is to find a sub-graph T which is a directed Tree.[1]

[1] A tree is a graph in which any two vertices are connected by exactly one simple path.

We divide the problem into three subtasks:

1. Task 1: For a given sentence, draw all possible labeled directed edges among the nodes. (ākāṅkṣā)
2. Task 2: Identify a sub-graph $T$ of $G$ such that $T$ is a directed Tree which satisfies the given constriants.(ākāṅkṣā, sannidhiḥ)
3. Task 3: Prioritize the solutions, in case there is more than one possible directed Tree. (sannidhiḥ)

# Mathematical representation

Representation: 5 dimensional Matrix.

C[i,j,k,l,m]

- i: $i^{th}$ word
- j: $j^{th}$ analysis of $i^{th}$ word
- k: Relation
- l: $l^{th}$ word
- m: $m^{th}$ analysis of $l^{th}$ word

# Task 1:

Using abhihitatva, vibhakti, sāmānādhikaraṇya, and the expectancies the matrix $C$ is populated with 0s and 1s.

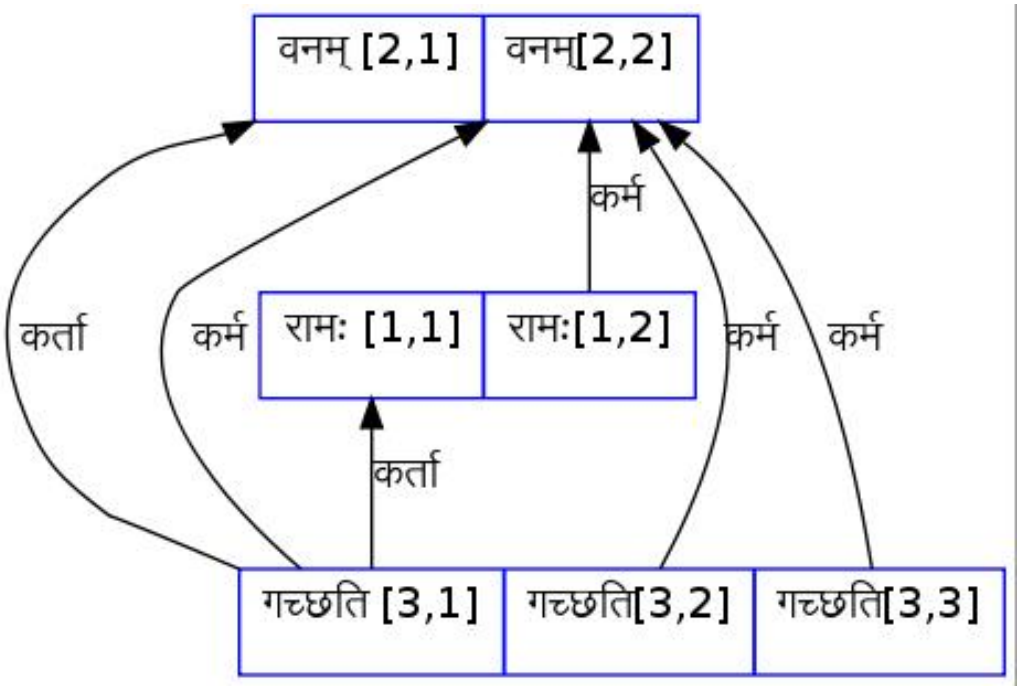
# Task 1:

*rāmaḥ vanaṁ gacchati*.

Morphological Analysis:
[1, 1]: rāma {gender=m, case=1, number=sg},
[1, 2]: rā {gaṇaḥ=*adādi*, lakāra=laṭ, person=1, number=pl, prayogaḥ=kartari, parasmaipadī}.

[2, 1]: vana {gender=n, case=1, number=sg},
[2, 2]: vana {gender=n ,case=2, number=sg}.

[3, 1]: gam {lakāra=laṭ, person=3, number=sg, voice=active, parasmaipadī},
[3, 2]: gacchat (gam śatṛ) {gender=m, case=1, number=sg},
[3, 3]: gacchat (gam śatṛ) {gender=n, case=1, number=sg}.


वनम् [2,1]
वनम्[2,2]
कर्म
कर्ता
कर्म
रामः [1,1]
रामः[1,2]
कर्म
कर्म
कर्ता
गच्छति [3,1]
गच्छति[3,2]
गच्छति[3,3]

# Task 2:

In order to get a Tree from this graph, we impose the following constraints.

1. A vibhaktiḥ marks only one relation.
   I.e., a node can have one and only one incoming arrow.
   $\sum_{j,R,k,l} C[i,j,R,k,l] = 1, \forall i.$
2. Each kāraka relation is marked by a single morpheme.
   There can not be more than one outgoing arrow with the same label from the same cell, if the relation marks a kāraka relation,[2] i.e. there can not be two words satisfying the same kāraka role of the same verb.
   $\sum_{i,j} C[i,j,R,k,l] = 1$, for each tuple $(R,k,l)$.

[2] adhikaraṇam is treated as an exception since one can have more than one adhikaraṇam as in
rāmaḥ adya pañca vādane gṛham agacchat.

1. A morpheme does not mark a relation to itself.
   A word can't satisfy its own expectancy. i.e. a word can't be linked to itself[3]. Or there can not be self loops in a graph.
   $\sum_{j,R,k} C[i,j,R,i,k] = 0, \forall i.$
2. Only one valid analysis for every word per solution
   2.1 If a word has both an incoming arrow as well as an outgoing arrow, they should be through the same cell.
   $\forall i \forall j \sum_{R,l,n} C[i,j,R,l,n] + \sum_{a,b,R,k!=j} C[a,b,R,i,k] \leq 1.$
   2.2 If there is more than one outgoing arrow through a node, then it should be through the same cell.
   if, for some i,j,R,l,m C[i,j,R,l,m] = 1,
   then $\forall a \forall b \forall R \sum_{a,b,R,k!=j} C[a,b,R,l,k] = 0.$
3. All the words in a sentence should be connected.
4. There is no crossing of links
   If all the nodes are plotted in a straight line, then they should not intersect each other. i.e.,
   if $C[i,j,R,k,l] = 1$, then
   $\forall v \forall y C[u,v,w,x,y] = 0$, if i < x < k and u < i or u > k.

[3] in case of some of the taddhita suffixes which are in svārtha, there will be self loops. But we do not consider the meaning of taddhita suffixes in the first step, and thus can avoid the self loops

The resultant graph is a Tree provided:

1. It is connected.
2. It has n-1 edges.

# Task 3:

The solutions are prioritized using the conditions specified below.

For each of the solutions, the cost is calculated as

Cost $= \sum_{i,R,j} c_{iRj}$, where
i) $c_{iRj} = |j - i| * wt_R$, if $C[i, a, R, j, b] = 1$ for some $a$ and $b$.
$= 0$ otherwise.

ii) $wt_R = rank(R)$

This cost ensures the following:

1. ākāṅkṣā (kāraka relation) is preferred over other relations (rank of the relations takes care of this.).
2. The ranking of the solutions on the basis of distance-based weights takes care of sannidhiḥ.

# Implementation

## Modularity

The first task demands the inputs from grammar,

whereas the second and the third tasks are purely mathematical ones, which can be handled by a constraint solver.

The separation of tasks into three sub-tasks makes it not only modular, but also easy for a grammarian to test his/her rules independently.

# Implementation

First task is implemented using an expert shell CLIPS

Second task uses a constraint solver MINION.

The system is available at
http://sanskrit.uohyd.ernet.in/scl/SHMT/shmt.html

Main purpose of this exercise is to have a proof of the concept.

# Performance

## Performance

113 sentences with single finite verb.
Sentence length 2 to 14 words.

Manual tagging for testing

- 97 (86%) sentences had the first parse correct.
- 16 (14%) sentences had one relation wrong.
  - wrong label: 10
  - wrong attachments: 3
  - wrong label and wrong attachments: 3

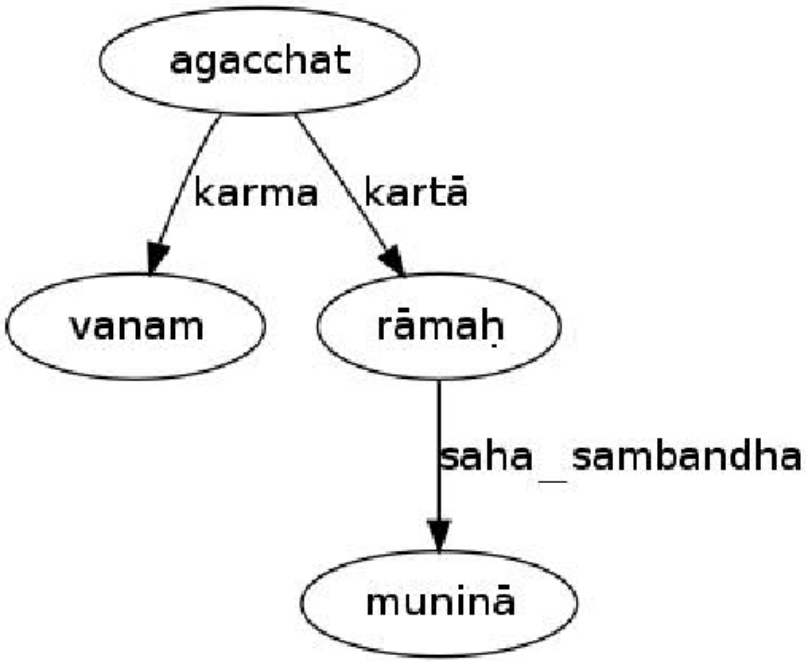
# Diagnosis

Reasons for wrong analysis:

- Fine grain / Coarse grain distinction
  - adhikaraṇa / deśādhikaraṇa / kālādhikaraṇa
  - mukhya karma / gauṇa karma
  - hetu / karaṇa
- verbs in the curādi ($10^{th}$) gaṇa.

# Diagnosis

Upapada a function word (dyotaka) or a content word (vācaka)?



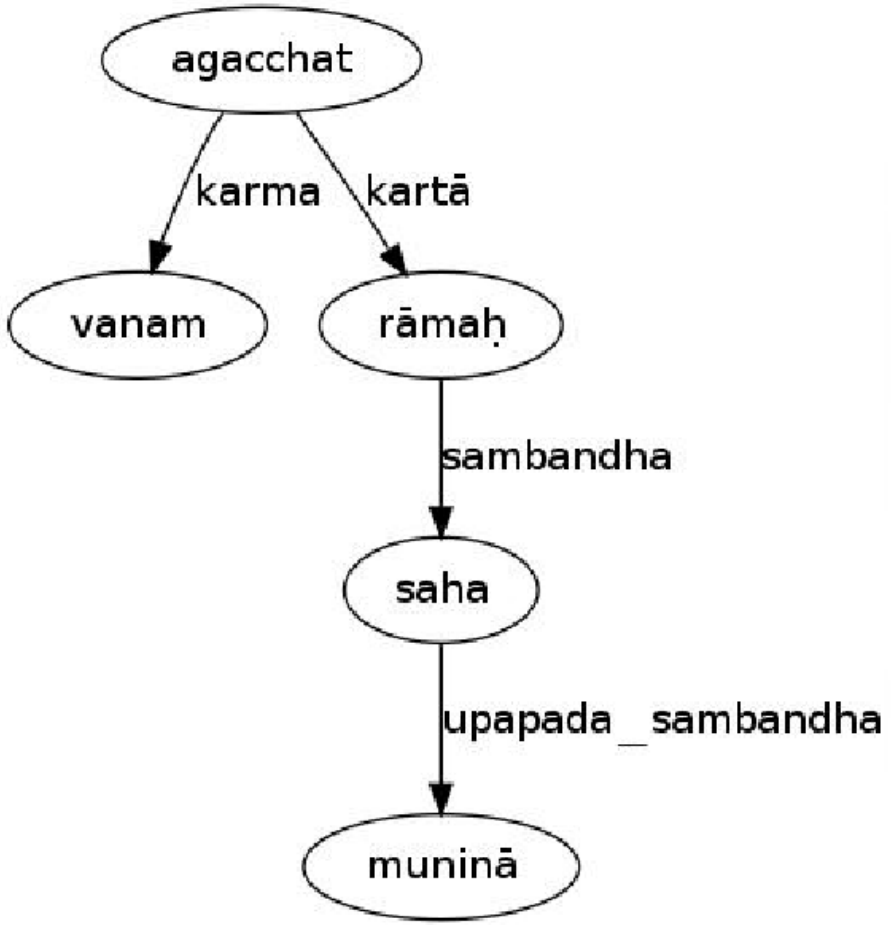




Figure: saha-function

# Diagnosis

Need to have dictionaries rich with semantic content

The second case suffix denotes the meaning of

- kriyāviśeṣaṇa (manner)
- kāla (time)
- adhvan (path)
- karma

For disambiguation, one should appeal to yogyatā.

# Real Text Challenges

Since the parser does the analysis 'mechanically', it detects the problems of 'violation' of the rules more easily.

> *guhena lakṣmaṇena sītayā ca sahitaḥ rāmaḥ vanena vanaṁ gatvā bahūdakāḥ nadīḥ tīrtvā bharadvājasya śāsanāt citrakūṭam anuprāpya vane ramyam āvasathaṁ kṛtvā devagandharvasaṅkāśāḥ te trayaḥ ramamāṇāḥ sukhaṁ nyavasan. (Saṁ. rā.:30-32)*

# Real Text Challenges

This sentence poses the following problems:

- Whom does the phrase 'te trayaḥ' refer to?
- *rāmaḥ* does not agree with the finite verb *nyavasan*. Is it not a violation of *samānakartṛkayoḥ pūrvakāle*?
- Does *gatvā* precede *tīrtvā* or *nyavasan*?
- In case of *vanena vanaṁ* what should be the meaning of the third case?

# DEMO