Parsing Indian Languages

Computational Paninian Grammar (CPG)

- Based on Panini's Grammar (500 BC)
- Inspired by Inflectionally rich language (Sanskrit)
- A dependency based analysis
- Good for paring Indian Langauges

Computational Paninian Grammar (The Basic Framework)

Treats a sentence as a set of modifier-modified relations

- Sentence has a primary modified or the root (which is generally a verb)

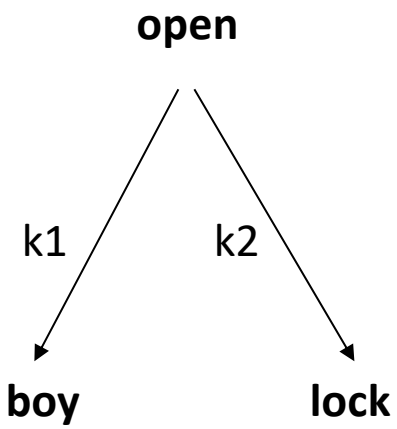
Gives us the framework to identify these relations

- Relations between noun constituent and verb called '*karaka*'
- *karakas* are syntactico-semantic in nature
- Syntactic cues help us in identifying the *karakas*

karta – *karma* karaka

The boy opened the lock	
k1 – *karta*	k2 – *karma*

karakas are direct participants in the activity denoted by the verb



Basic *karaka* relations

karta – agent/doer/force

- Relation label – k1

karma – object/patient

- Relation label – k2

karana – instrument

- Relation label – k3

sampradaan – beneficiary

- Relation label – k4

apaadaan – source

- Relation label – k5

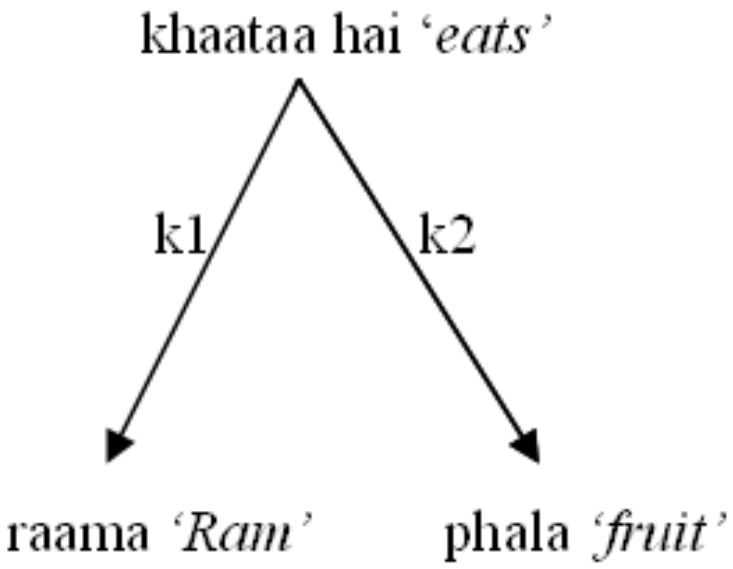
adhikarana – location in place/time/other

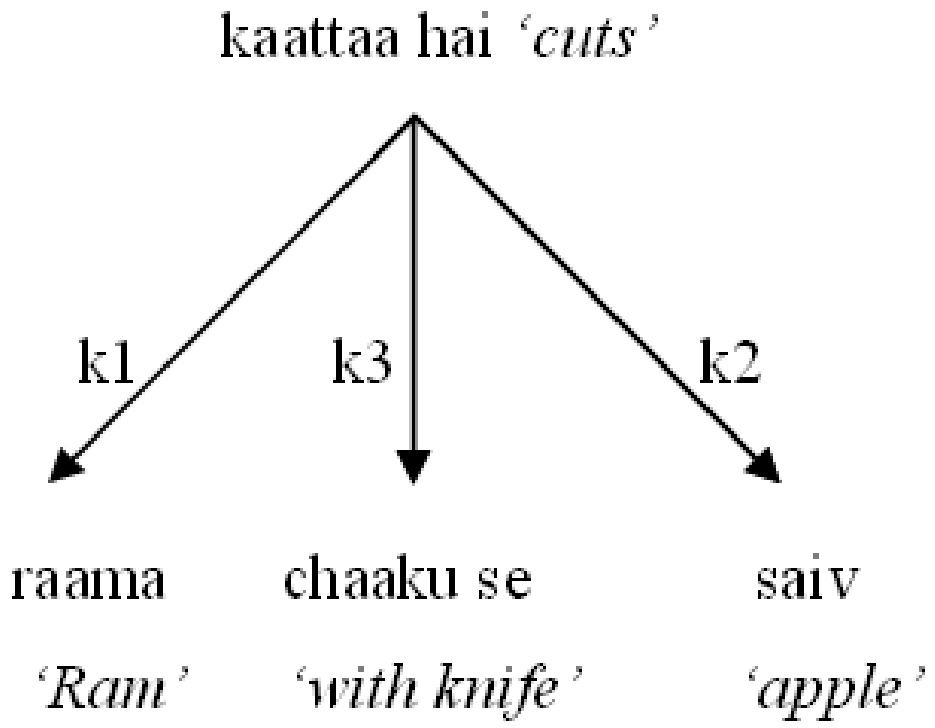
- Relation label – k7p/k7t/k7

For complete list of dependency relations: (Begum et al., 2008)

Basic *karaka* relations

- *raama phala khaataa hai*
- ‘Ram eats fruit’



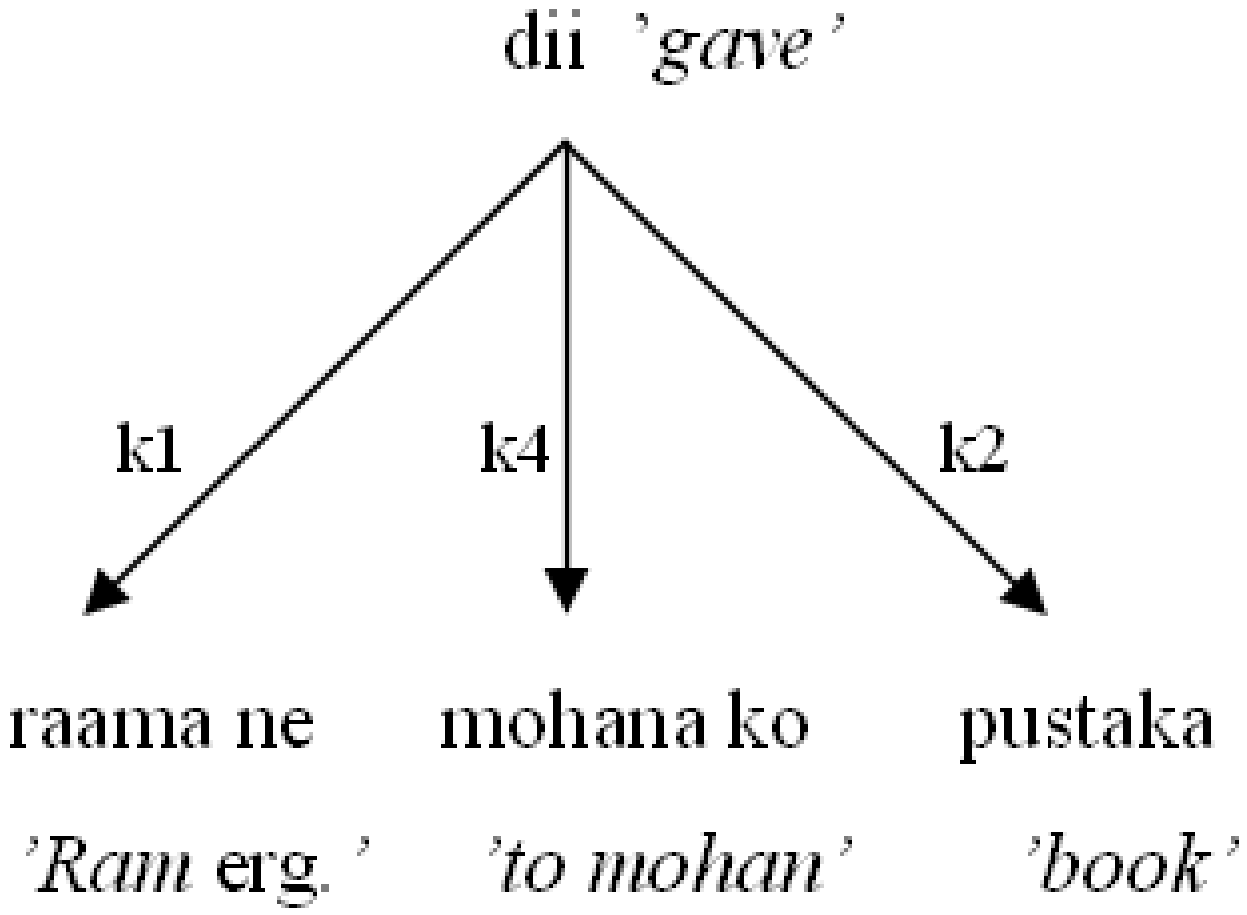


Basic *karaka* relations

- *raama chaaku se saiv kaatataa hai*
- 'Ram cuts the apple with knife'

Basic *karaka* relations

- *raama ne mohana ko pustaka dii*
- ‘Ram gave a book to Mohan’



Why Paninian Labels

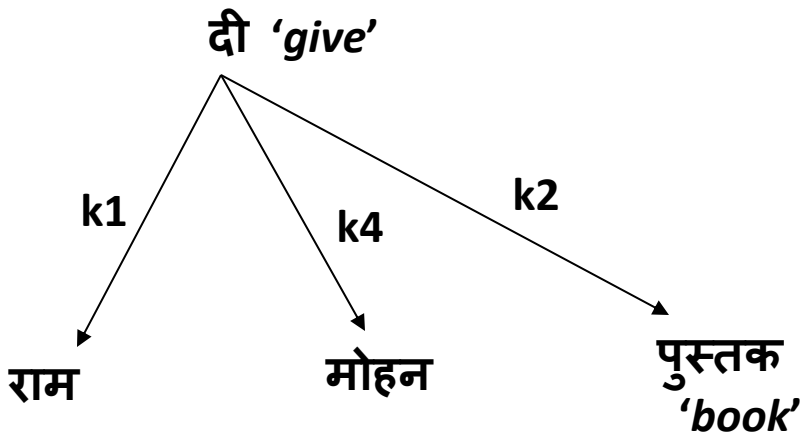
- Other choices for labels could be
  - Grammatical relations
    - Subject, Object, etc.
    - Behavioral tests (Mohanan, 1994)
  - Thematic roles
    - Agent, patient, etc.
    - No concrete cues

  - Difficult to extract them automatically
  - Karakas can be computationally exploited
    - Syntactically grounded, Semantically loaded
    - Gives a level of interface

Example

- राम ने मोहन को पुस्तक दी |

Example – Parsed Output



Parser

Two stage strategy	Appropriate constraints formed
Stage I (Intra-clausal relations)	Dependency relations marked Relations such as k1, k2, k3, etc. for each verb
Stage II (Inter-clausal relations & conjunct relations)	Conjuncts, relative clauses, kriya mula, etc

Demand Frame for Verb

A demand frame or karaka frame for a verb indicates the demands the verb makes

It depends on the verb and its tense, aspect and modality (TAM) label.

A mapping is specified between karaka relations and vibhaktis (post-positions, suffix).

Karaka Frame

It specifies what karakas are mandatory or optional for the verb and what vibhaktis (post-positions) they take respectively

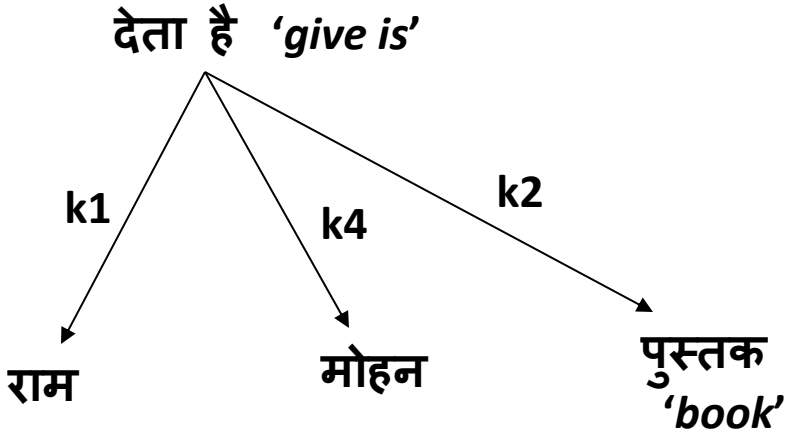
Each verb belongs to a specific verb class

- Each class has a basic karaka frame

Each TAM specifies a transformation rule

Example

- राम मोहन को पुस्तक देता है |



Parsed Dependency Tree

Transformations

- Based on the TAM of the verb
  - *राम ने मोहन को खिलौना दिया* |
  - *राम को मोहन को खिलौना देना पड़ा* |
  - Appropriate transformation applied

Example:

- राम ने मोहन को खिलौना दिया।

Constraints

C1: For each of the mandatory demands in a demand frame for each demand group, there should be exactly one outgoing edge labeled by the demand from the demand group.

C2: For each of the optional demands in a demand frame for each demand group, there should be at most one outgoing edge labeled by the demand from the demand group.

C3: There should be exactly one incoming arc into each source group.

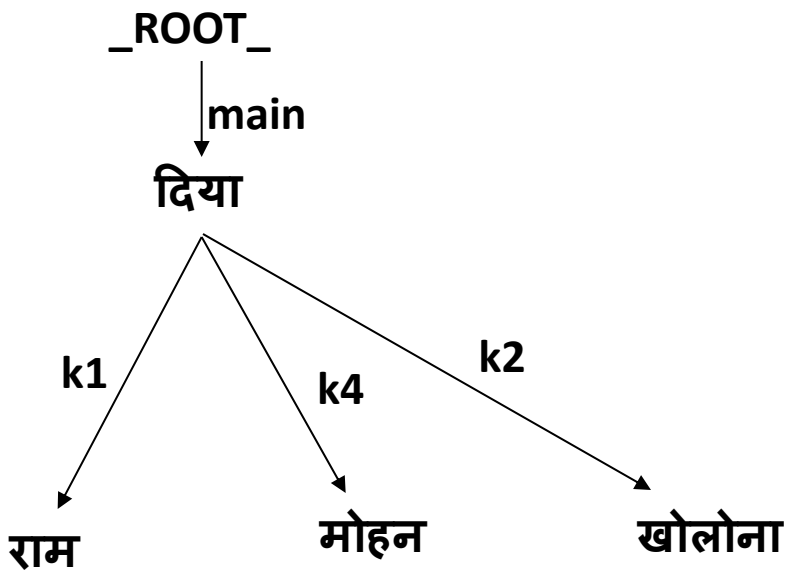
Constraints

A parse of a sentence is obtained by satisfying all the above constraints

Ambiguous sentences have multiple parses

Ill formed sentences have no parse.

Parse - I



Complex Sentences

Common karaka frame

- Attached to each karaka frame
- Preference given to main frame if there are clashes

Fallback karaka frame

- required karaka frame is missing
- Graceful degradation

Example (Complex Sentence)

- राम ने फल खा कर मोहन को खिलौना दिया

  Ram 'ERG' fruit 'having eaten' Mohan 'DAT' toy gave

'Having eaten the fruit Ram gave the toy to Mohan'

Solution Graph