Overview

- Introduction to the nature of syntactic representations. (Rambow, 15 minutes)
- Introduction to the morphology, syntax, and lexical semantics of Hindi and Urdu. (Sharma, 40 minutes)
- The morphological representation for Hindi and Urdu, including encoding issues, tokenization, part-of-speech tags, and morphological representation. (Sharma and Rambow, 20 minutes)
- The dependency representation (DS) for Hindi and Urdu syntax: principles, representation, and examples. (Sharma, 25 minutes)
- The lexical semantic representation (PB) for Hindi and Urdu: principles, representation, and examples. (Vaidya, 25 minutes)
- The phrase structure representation (PS) for Hindi and Urdu syntax: principles, representation, and examples. (Rambow, 25 minutes)
- Sample initial experiments in Hindi and Urdu NLP using the HUTB. (Sharma and Rambow, 15 minutes).

Paninian Grammatical Model and Hindi/Urdu Treebanks

Dipti Misra Sharma
IIIT, Hyderabad
<dipti@iiit.ac.in>

COLING-2012

Outline

- Paninian Grammatical framework : The Grammatical Model used in the Hindi/Urdu treebanks
  - Some basic concepts
- Some Hindi constructions
  - Causatives
  - Co-ordination
  - Unaccusatives
  - Relative clauses
- Conclusions

Introduction

- Treebank - One of the most important linguisticresources.
- Utility in various NLP tasks such as parsing, natural language understanding etc.
- Linguistic information encoded at different levels such as morphological, syntactic, syntactico-semantic (dependency).

Hindi Dependency Treebank

- The Corpus
  - News articles 350k
  - Tourism articles 25-30k
  - Converstational data 25-20k
- Dependency grammar framework : Paninian Grammatical model

Why Paninian Grammar

Indian languages

- Rich morphology
- Relatively flexible word order

For example,

a) baccaa phala khaataa hai
'child' 'fruit' 'eat_hab' 'pres'

b) phala baccaa khaataa hai

c) phala khaataa hai baccaa

d) baccaa khaataa hai phala

Panini's Grammar

- Dated around 500 B.C.
- Seeks to provide a complete, maximally concise and theoretically consistent analysis of Sanskrit grammatical structure
- Based on spoken form

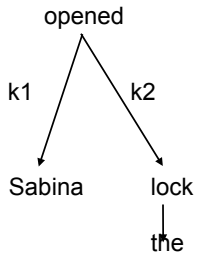
<Kiparsky, 1993>

- Focuses on language as a means of communication

Panini's Grammar contd

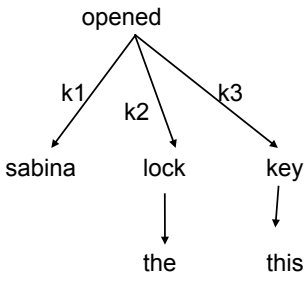
- Treats a sentence as a series of modifier-modified relations
- Every sentence has a primary modified (generally a verb)
- Relations between verbs and their particiapants called 'karaka'
- Other relations – such as reason, prupose, genitive etc
- The relations are expressed through explicit markers called 'vibhakti'

Sabina opened the lock



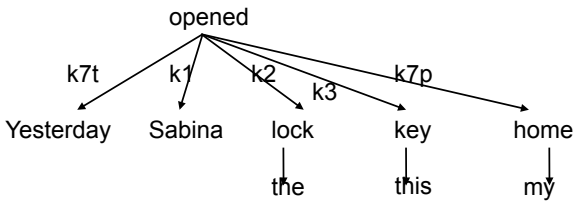
K1 (Karta) : the doer of the action (the locus of activity)
K2 (Karma) : locus of result

Sabina opened the lock with this key



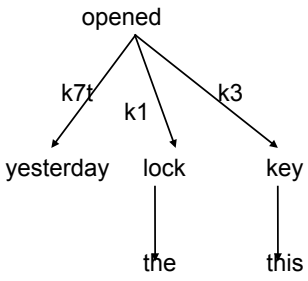
K3 (karaNa) : instrument

Yesterday, Sabina opened the lock with this key at my home



K7t (deshadhikaraNa) : time
K7p (kaladhikaraNa): place

Yesterday, the lock opened with this key



'lock' becomes the 'karta' !!!

Levels of Analysis

L1 – Semantic relations : karakas, eg *raama* karta

L2 – Morphosyntactic : vibhakti, eg *raama* prathamaa

L3 – Morphological representation (abstract) : vibhakti markers, eg raama + su (Sanskrit)
raama + 0 (Hindi)
raama + du (Telugu)

L4 – Phonological form : raamaH (Sans)
raama (Hindi)
raamudu (Telugu)

Our Model

- Morph analysis
- POS tagging
- Identify minimal constituents (chunks/bags) and their heads
- Mark the relations across chunks (head to head relation)
- Chunk-internal dependencies are left unspecified
- The trees are fully expanded automatically

For Example

meraa baDzaa bhaaii bahuta phala khaataa hai

=>

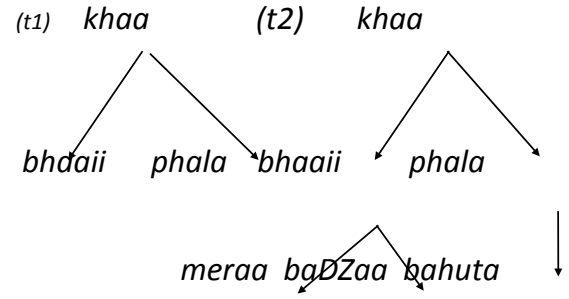
meraa_PRP baDzaa_JJ bhaaii_NN bahuta_QF phala_NN khaataa_VM hai_VAUX

=>

*((meraa_PRP baDzaa_JJ **bhaaii**_NN))_NP*
*((bahuta_QF **phala**_NN))_NP*
*((**khaataa**_VM hai_VAUX))_VG*

Example Contd...

*((meraa_PRP baDzaa_JJ **bhaaii**_NN))_NP*
*((bahuta_QF **phala**_NN))_NP*
*((**khaataa**_VM hai_VAUX))_VG*



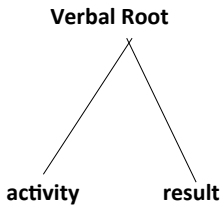
Karaka Relations

- Direct participants in an action/event
- Syntactico-semantic
- The karta and karma of a verb are determined by the verb's semantics
- Verb denotes an action/event
- Any action is a bundle of sub-actions

Sabina opened the lock with the key
The key opened the lock
The lock opened

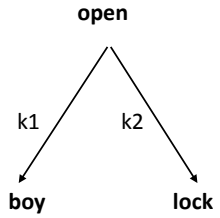
Semantics of the verb

- A verbal root denotes:
  - The activity
  - The result
- Locus of activity : *karta*
- Locus of result : *karma*



karta - karma

- The boy opened the lock
  - k1 – *karta*
  - k2 – *karma*
- *karta, karma* sometimes correspond to agent/theme
  - Not always

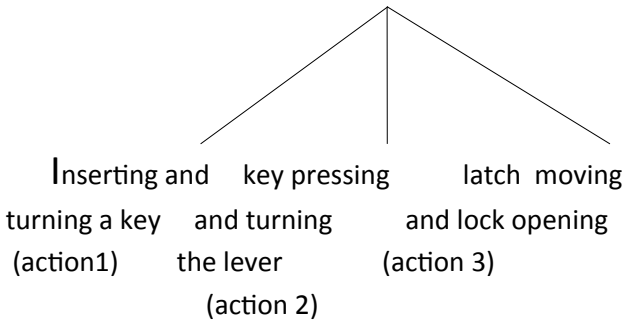


The door opened

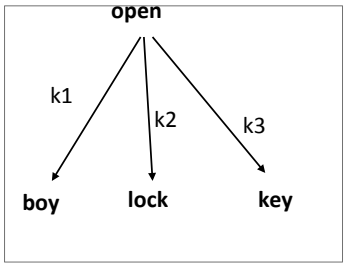
- 'The door' is karta
- The sentence has no explicit karma

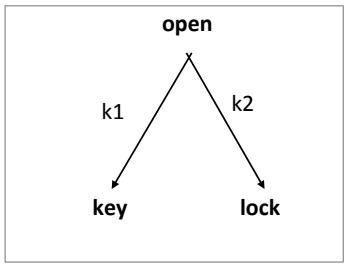
Sub-actions - Opening of lock

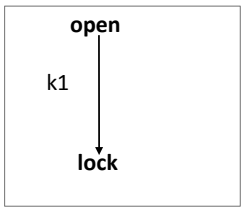
Opening of lock



Sub-actions - Opening of lock







k1 – *karta (doer)*
k2 – *karma (affected)*
k3 – *karana (instrument)*

Thus,

- The action of 'opening' normally requires an agentive participant. So,

Sabina opened the lock

However,

- The speaker may decide not to express the role of the agent. Hence,

The key opened the lock

- The 'karaNa' (instrument) is raised to the role of 'karta' (doer - karaNa-kartri)

The lock opened

- The 'karma' is raised to the role of 'karta' (doer - karma-kartri)

Thus, 'karta' or the other karaka roles can 'shift' depending on what the speaker wants to express (vivaksha)

- Which sub-action the speaker wants to focus on.

Speaker's Intention *(vivakshaa)*

- Every sentence reflects speaker's intention
  - Participants are assigned various relations accordingly

  *(a) '**I** opened the lock with **this key**'*

  *(b) 'I am sure **this key** will open the lock'*

  - 'key' gets assigned *karta (in b), karana (in a)* based on what the speaker wants to express
- Syntax reflects *vivaksha*

The Scheme

- Morph analysis
- POS tagging
- Chunking
- Mark the syntactic relations (dependency relations) across chunks (head to head relation).

Overview

- **Objective**
- **The Scheme**
  - **Morph Analysis**
  - **POS Tagging**
  - **Chunking**
  - **Dependency Relations**
- **Dependency Scheme**
- **Relations in Dependency Scheme**
- **Some Hindi Constructions**

Objective

- To evolve an adequately comprehensive tagging scheme for the purpose of annotating corpora for dependency relations within a sentence.
- We are developing treebanks for Hindi/Urdu.
- Following Paninian framework as the annotation scheme.
- We show how the scheme handles some phenomena such as complex verbs, causatives, relative clauses, conjunctions, etc. in Hindi.

An Example

- **Example:**
  - ***meraa badZaa bhaaii bahuta phala khaataa hai***

    'my' 'elder' 'brother' 'lots' 'fruits' 'eat+HAB' 'PRES'

    'MY elder brother eats lots of fruits.'

An Example (Contd...)

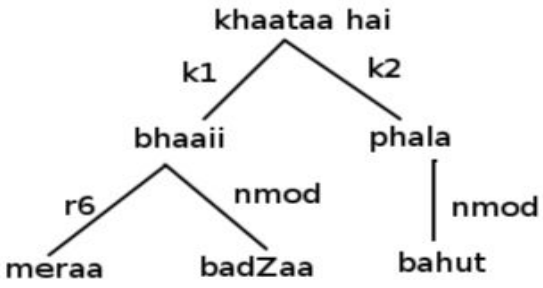
- **Morph Analysis:**
  - *meraa* <fs af= root=*meraa*, cat=pron, gend=any, num=sg, pers=1, case=o>
  - *badZaa* <fs af= root=*badZaa*, cat=adj, gend=m, , , >
  - *bhaaii* <fs af= root=*bhaaii*, cat=n, gend=m, num=sg, pers=3, case=d>
  - *bahuta* <fs af= root=*bahuta*, cat=adj, gend=any, , , >
  - *phala* <fs af= root=*phala*, cat=n, gend=m, num=any, pers=3, case=d>
  - *khaataa* <fs af= root=*khaa*, cat=v, gend=m, num=sg, pers=3, TAM=taa>
  - *hai* <fs af= root=*hai*, cat=v, gend=any, num=any, pers=3, >

An Example (Contd ..)

- **POS Tagging:**
  - *meraa_PRP baDzaa_JJ bhaaii_NN bahuta_QF phala_NN khaataa_VM hai_VAUX*
- **Chunking:**
  - *((meraa_PRP))_NP*
    *((baDzaa_JJ **bhaaii**_NN))_NP*
    *((bahuta_QF **phala**_NN))_NP*
    *((**khaataa**_VM hai_VAUX))_VG*

An Example (Contd...)

- **Dependency Relation**



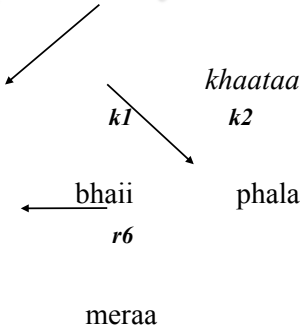
Dependency Scheme

- The Paninian approach treats a sentence as a series of modifier-modified relations.
- Hence, it provides framework for dependency analysis.
- In our dependency tree:
  - each node is a chunk, and
  - the edge represents the relations between the connected nodes labeled with the karaka or other relations.
- Chunk represents a set of adjacent words which are in dependency relations with each other.
- All the modifier-modified relations between the heads of the chunks (inter-chunk relations) are marked in this manner.

Dependency Scheme (Contd..)

- Here, modifier-modified relations are marked between the heads of the chunks:
  - ***meraa 'my'***
  - ***bhaaii 'brother',***
  - ***phala 'fruit', and***
  - ***khaataa 'eats'.***
- ***badZaa 'big'*** *and* ***bahut 'much'*** *are part of the chunks.*

Dependency Scheme (Contd..)



Relations in Dependency Scheme

- **There are 3 types of relations in Dependency Scheme;**
  - *Karaka* relations,
  - Relations other than *karakas*, and
  - Relations which do not fall under 'dependency relation' directly but are required for showing the dependencies indirectly.
- *Karaka* relations are participants directly involved in the action denoted by the verb
- Relations other than *karakas* denote *purpose, reason.*
- Relations which do not fall under 'dependency relation' directly are used for representing 'co-ordination' and 'complex predicates'.

Basic *karaka* relations

- **Only six**
  - *karta* – subject/agent/doer
  - *karma* – object/patient
  - *karana* – instrument
  - *sampradaan* – beneficiary
  - *apaadaan* – source
  - *adhikarana* – location in place/time/other

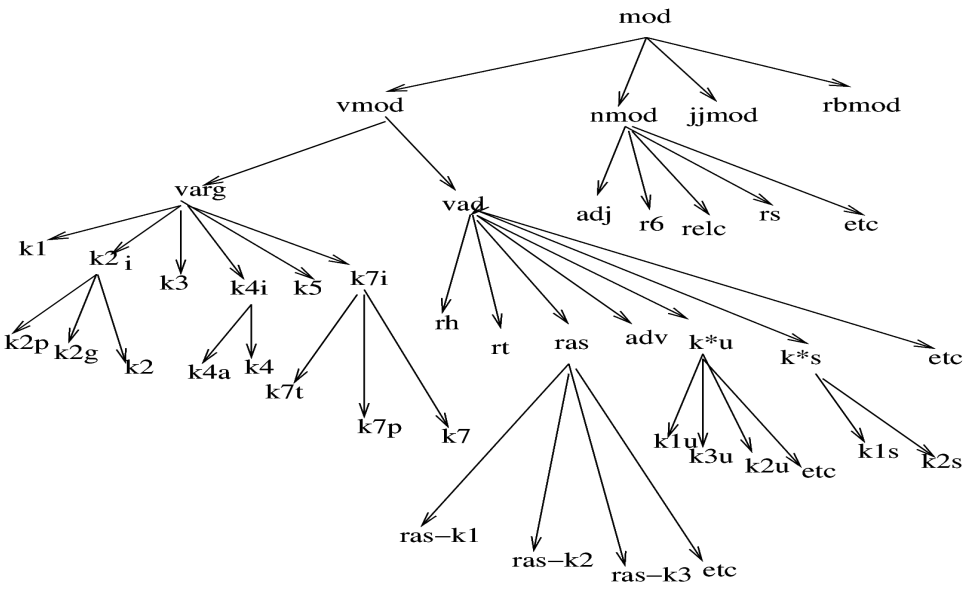
Relations other than *karakas*

- *r6* – Genitive
- *rt* – Purpose
- *rh* – Reason
- *nmod_relc* – Relative clause
- *rad* – Address

Relations which do not fall under 'dependency relation'

- *ccof* – *Conjunction*
- *pof* – *Complex Predicates*
- *fragof* – *Fragment of*

Dependency Relation Types



Some Hindi Constructions

(1) Causative Constructions:

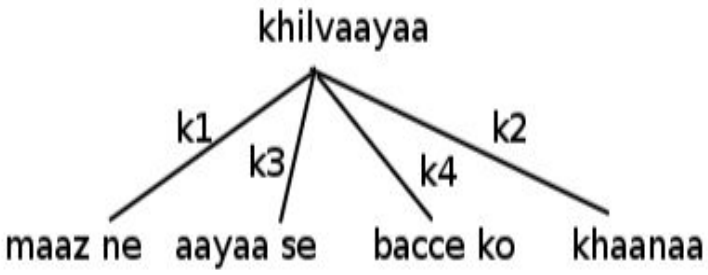
- ***maaz ne aayaa se bacce ko khaaanaa khilvaayaa***
  'mother' 'Erg.' 'maid' 'by' 'child' 'Acc.' 'food' 'eat-Caus.'
  'Mother caused the maid to feed the child.'

- **Issue:**

  - **Possibility-I: Go by syntactic analysis**

    - ***khilvaa*** 'cause to eat' is the verb root.
    - ***maaz ne*** has ***karta*** vibhakti so mark as ***k1***.
    - ***aayaa se*** has ***karana*** vibhakti so mark as ***k3***.
    - ***bacce ko*** has ***sampradan*** vibhakti so mark as ***k4***.

Causative Constructions (Contd …)

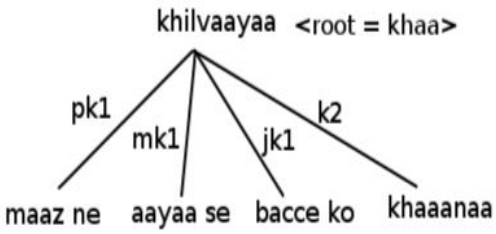


Causative Constructions (Contd …)

- **Possibility-II:**
  - The verb ***khilvaa 'cause to eat'*** is a causative verb and it is morphologically related to the base verb ***khaa 'eat'.***
  - Paninian framework provides the relations:
    - ***prayojaka karta 'causer' (pk1):*** The causer in a causative construction.
    - ***prayojya karta 'causee' (jk1):*** The causee in a causative construction.
    - ***madhyastha karta 'mediator causer' (mk1):*** The mediator-causer in the causative construction.

Causative Constructions (Contd …)

- **Possibility-II:**
  - Do we mark the above dependency roles?
  - If we mark these relations then root will be ***khaa 'eat'.***



Causative Constructions (Contd …)

- ***Ex: maaz ne (k1) cammaca se (k3) bacce ko khaanaa (k2) khilavaayaa.***
  'Mother fed the child with the spoon.'

- ***Ex: maaz ne (pk1) aayaa se (mk1) bacce ko (jk1) khaanaa (k2) khilavaayaa.***
  'Mother made the maid to feed the child'.

- As there is **morphological relatedness** between the base verb ***khaa 'eat'*** and causative verb ***khilvaa 'cause to eat',*** we mark ***pk1, mk1, jk1*** instead of ***k1, k3, k4*** respectively.

- For causatives, our current decision: **Follow Possibility-II.**

(2) Relative Clauses (nmod__relc)

- ***Ex: jo ladZakaa vahaaz khadZaa hai vaha meraa bhaaii hai.***
  'who' 'boy' 'there' 'stand' 'is' 'he' 'my' 'brother' 'is'
  'The boy who is standing there is my brother.'

- **Issue:**

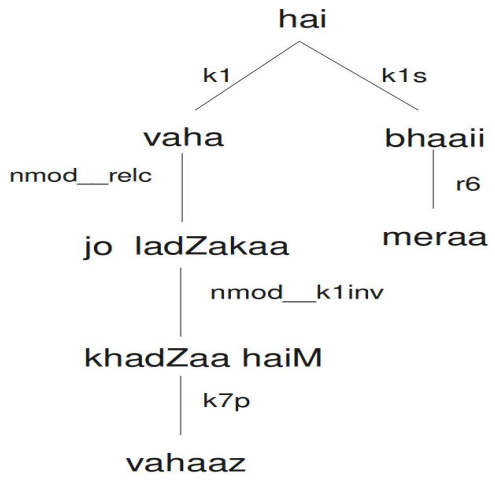
  - **Possibility-I:**

    - Provides relation between ***vaha 'he'*** in main clause and ***jo ladZakaa 'the boy'*** in rel. clause.

    - The dependency of ***jo ladZakaa 'the boy'*** is on ***vaha 'he'.***

    - ***jo ladZakaa 'the boy'*** is the root of the relative clause '***jo ladZakaa vahaaz khadZaa hai'.***

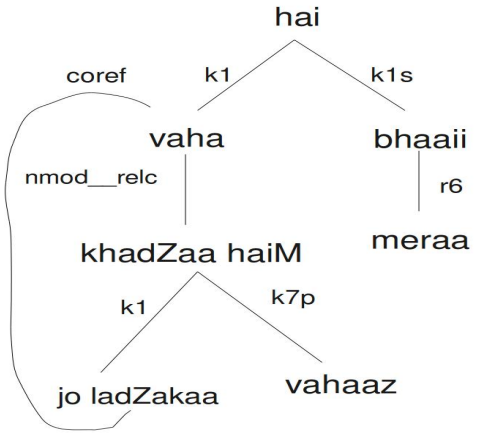
Relative Clause: Possibility-I



Relative Clauses (nmod__relc)

- **Possibility-II**
  - The verb ***khadZaa hai 'is standing'*** is the root of the relative clause.
  - The modifier of ***vaha 'he'*** in main clause is the entire relative clause.
  - Here the relation between ***jo ladZakaa 'the boy'*** in the relative clause and **vaha 'he'** in the main clause is captured by the feature ***coref.***

Relative Clause: Alternative-II



Relative Clauses (Contd…)

- For relative clauses, our current decision: **Follow Possibility-II.**
- In Possibility-II, ***jo ladZakaa 'the boy'*** in the rel. clause attaches with the verb ***khadzaa hai 'is standing'*** of the rel.clause**.**
- The rel.clause attaches with ***vaha 'he'*** of main clause by ***'nmod__relc'*** relation.
- The relation between ***jo ladZakaa 'the boy'*** and **vaha 'he'** is captured by the feature ***coref.***

(3) *anubhava karta – k4a*

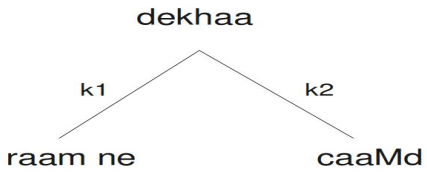
- ***Ex-1: mujhko dukh hai***
  'I.Dat.' 'unhappy 'is'
  'I am unhappy.'
- Here ***ko*** vibhakti in ***mujhko 'to me'*** tells that it is not a ***karta.***
- Here, ***dukh 'unhappy'*** is the ***karta.***
- Here ***mujhko 'to me'*** is a subtype of *sampradan.*
- This *sampradan is different from the sampradan (k4—beneficiary).*
- We call it as ***anubhava karta*** represented by ***k4a.***

anubhava karta – k4a (Contd ..)

- ***Ex-2: raam ne (agent) caaMd dekhaa → Base verb***
  'ram' 'Erg.' 'moon' 'saw'
  'Ram saw the moon.'
- ***Ex-3: raam ko (experiencer) caaMd dikhaa →Derived***
  'ram.Dat' 'moon' 'appeared' **Intransitive** 'Moon was visible to me.' **Verb**
-

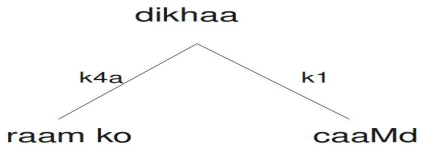
anubhava karta – k4a (Contd…)

- ***Ex-2:***



anubhava karta – k4a (Contd…)

- ***Ex-3:***



(4) Relation samanadhikaran- rs

- ***Ex-1: raam ne kahaa ki vo kal aayegaa.***
  'Ram said that he will come tomorrow.'

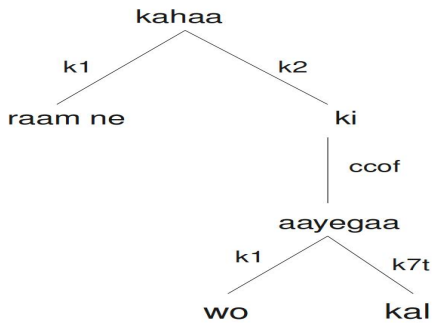
- ***Ex-2: raam ne yaha kahaa ki vo kal aayegaa.***
  'Ram said that he will come tomorrow.'

- In ***Ex-1*****,** the clause ***'ki vo kal aayegaa'*** is the object, i.e.**,** ***karma.***

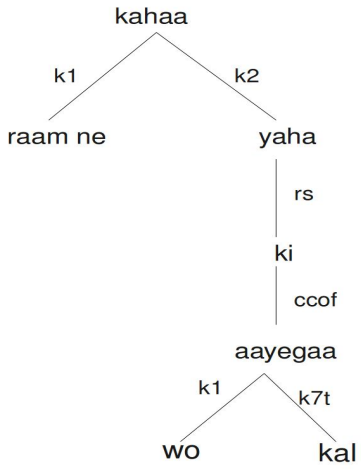
- In ***Ex-2*****,** the clause ***'ki vo kal aayegaa'*** is the complement of the object ***yaha 'this'*** so it attahes to ***yaha*** as ***rs.***

Relation samanadhikaran- rs *(Contd…)*

- ***Ex-1***



Relation samanadhikaran- rs (Contd..) – *Ex-2*

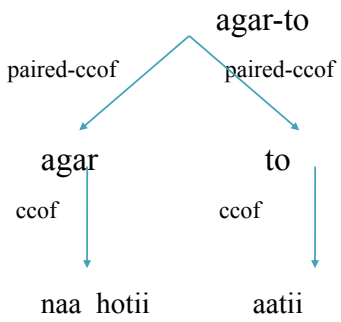


(5) Conditionals

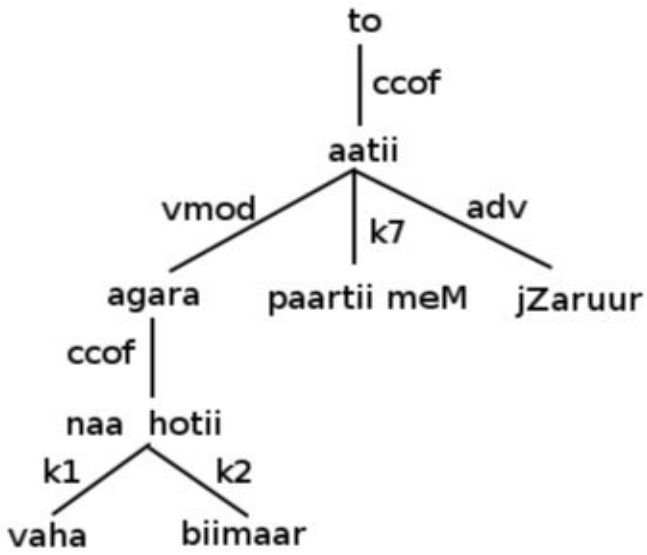
- ***Ex: agara vaha biimaara na hotii to paartii me jZarUra aatii***
  'if' 'she' 'sick' 'not' 'happened' 'then' 'party' 'in' definitely' 'come'
  'Had she been not sick she would have definitely come to the party.'

- **Issue:**
  - **Possibility-I: Abstract node**
  - **Possibility-II: One clause depends on the other clause**

Possibility - I



Possibility - II



Conditionals (Contd..)

- Possibility-I is not possible because ***agar-to*** is the head of the tree which is an abstract node, i.e. it is not a lexical node.
- For conditionals, our current decision: **Follow Possibility-II.**
- In Possibility-II, the ***agar 'if'*** clause is dependent on the ***to 'then'*** clause.
- Here, the ***agar 'if'*** clause is the subordinate clause and ***to 'then'*** clause is the main clause.

(6) Participles (vmod)

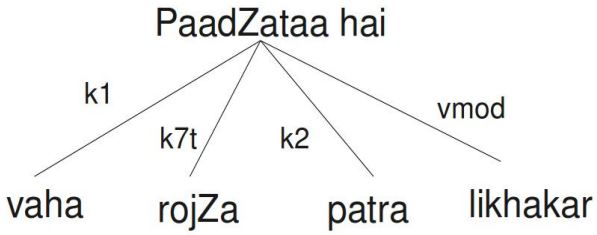
- In non-adjectival partiples, an argument of a verb (main) is shared with another verb(participle).
- The arguments occurs only once in the sentence but is semantically related to both the verbs.
- The shared argument syntactically always attaches with the main verb.
- For the other verb this argument is semantically realized but not syntactically.

Participles (vmod) (Contd ..)

- ***Ex: vaha rojZa patra likhakara PaadZataa hai***

  ’he’ ’daily’ ’letter’ ’having written’ ’tear’ ’is’

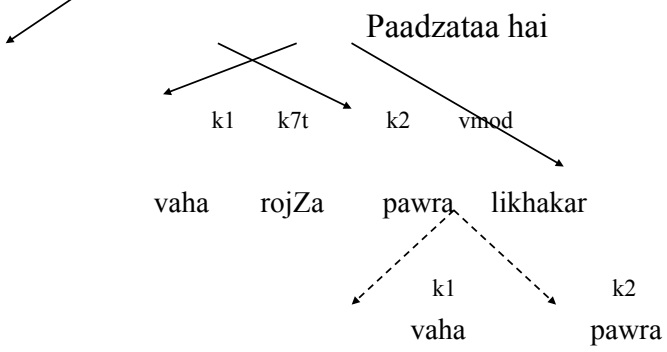
  ‘Having letters written everyday he tears.’



Participles (vmod) (Contd ..)

- The arguments ***vaha 'he'*** and ***pawra 'letter'*** of the verb **PaadZataa 'tears'** is shared with another participle verb ***likhakar 'having written'.***

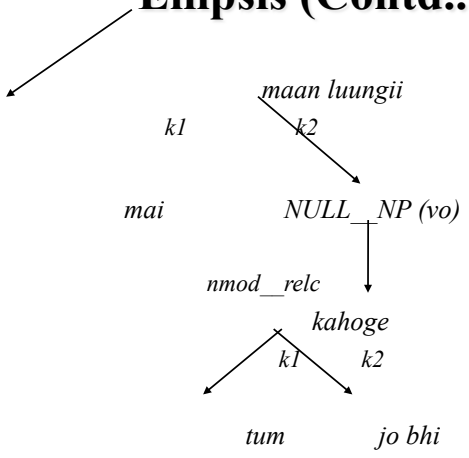
Participles (vmod) (contd..)



(7)Ellipsis

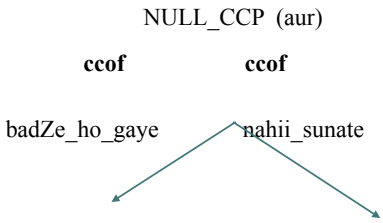
- How to show dependencies when the head is missing ?
- ***Ex: tum jo bhi kahoge (vo) mai maan luungii***
  'you 'whatever' 'will say' 'that' 'I' 'will believe'
  'I will believe whatever you say.'
- In the above example, ***vo 'that'*** is missing which becomes the parent node for relative clause ***'tum jo bhi kahoge'***
- We insert a null element i.e. NULL_NP for ***vo 'that'*** to show the dependency.

Ellipsis (Contd...)



Ellipsis (Contd...)

- ***Ex: bacce badZe ho gaye hai (aur) kisii kii baat nahii sunate***
  'children' 'big' 'happen' 'is' 'no one' 'Gen' 'matter' 'not' 'listen'
  "The children have grown up, they don't listen to anyone"
- No explicit conjunct !
- Insert a NULL element to show the dependencies (if it is essential).



Non-dependency Relations

- *ccof* – *Conjunction*
- *pof* – *Complex Predicates*
- *fragof* -- *Fragment of*

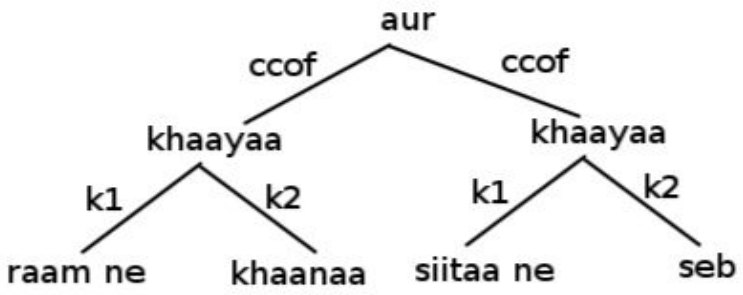
(1) Conjunction (ccof)

- ***ccof*** relation doesn't reflects a dependency relation.
- It is used for coordinating as well as subordinating conjunctions.
- The dependency trees will show the conjuncts as heads.
- In coordinating conjuncts, the conjunct is the head and takes the coordinating elements as its children.
- In subordinating conjunct , it would take the clause to which it is syntactically attached (the subordinate clause) as its child.

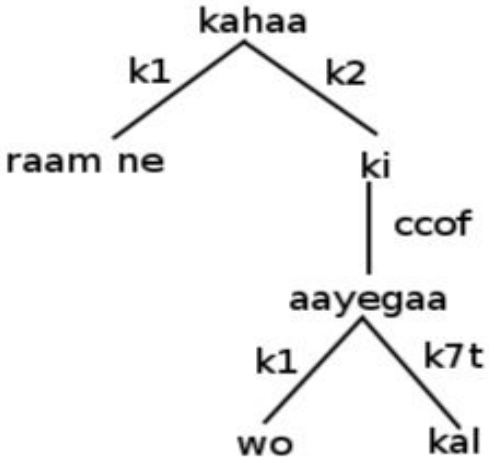
Conjunction (ccof) (Contd…)

- **Coordinate Conjunction**
  - ***Ex: raam ne khaanaa khaayaa aur siitaa ne seb khaayaa***
    'ram' 'Erg.' 'food' 'ate' 'and' 'sita' Erg.' 'apple' 'ate'
    'Ram ate food and Sita ate an apple.'
- **Subordinate Conjunction**
  - ***Ex: raam ne kahaa ki vo kal aayegaa***
    'ram' 'Erg.' 'said' that' 'he' 'tomorrow' 'come-Fut'
    'Ram said that he will come tomorrow.'

Coordinate Conjunction (ccof)



Subordinate Conjunction



(2) Conjunct Verbs

- ***Ex: maine usase ek prashna kiyaa***
  'I-erg' 'him-inst' 'one' 'question' 'did'
  'I asked him a question'

- The noun ***prashna 'question'*** within the conjunct verb sequence ***prashna kiyaa 'questioned'*** is being modified by the adjective ***ek 'one'*** and not the entire noun-verb sequence.

- The annotation scheme should be able to account for this relation in the dependency tree.

- If ***prashna kiyaa*** is grouped as a single verb chunk, it will not be possible to mark the appropriate relation between ***ek*** and ***prashna***.

Conjunct Verbs (Contd..)

- To overcome this problem we break ***ek prashna kiyaa*** into two separate chunks, [***ek prashna]/NP [kiyaa]/VG***.

- The dependency relation of ***prashna*** with ***kiyaa*** will be **POF ('Part OF' relation).**

- It means noun or an adjective in the conjunct verb sequence will have a **POF** relation with the verb.

- This way, the relation between ***ek*** and ***prashna*** becomes an intra-chunk relation as they will now become part of a single NP chunk.

- Conjunct verbs are chunked separately, but semantically they constitute a single unit.

- It captures the fact that the noun-verb sequence is a conjunct verb by linking them with **POF** relation.

Conjunct Verbs (Contd..)