

# The Hindi/Urdu Treebank: New Frontiers in Hindi and Urdu Natural Language Processing

Dipti Misra Sharma

LTRC, IIIT, Hyderabad, India, dipti@iiit.ac.in

Owen Rambow

CCLS, Columbia, New York City, USA, rambow@ccls.columbia.edu

Ashwini Vaidya

Linguistics, University of Colorado, Boulder, USA, Ashwini.Vaidya@colorado.edu

Dec 8, 2012

COLING 2012

## 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).

## 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).

## The Hindi Treebank

- **3 Representations**
  - DS: Dependency Structure
  - PB: PropBank (lexical predicate-argument structure)
  - PS: Phrase Structure
- **Why have three levels of representation? What does “level of representation” mean, in fact?**

## What is a Syntactic Representation?

1. Syntactic phenomena (“what”), e.g.:
  - Subject of a verb
  - Relative clause
  - Small clause

Linguists tend to agree on what phenomena exist
2. Mathematical representation type (“basic how”), e.g.:
  - Phrase structure tree
  - Dependency tree
  - Or something more complicated: graph, LFG, TAG, ...
3. Formal syntactic description (“detailed how”):
  - a. Mapping from phenomena to representations (in particular type)
  - b. Chosen representation for a specific phenomenon also called **analysis**
  - c. Phenomena extracted in representation are the **interpretation**
  - d. Formal description is a **syntactic theory** if it makes predictions

## Representation Types: Dependency and Phrase Structure

- Dependency Tree (DS):
  - One label alphabet, words (= words in a sentence)
  - All nodes labeled with words or empty strings
- Phrase Structure Tree (PS):
  - Two disjoint label alphabets, terminals (= words in sentence) and nonterminals
  - All and only interior nodes are labeled with nonterminals
  - Leaves are labeled with terminals or empty strings
- Nothing else is part of the definition!

## Example: Small Clauses

- Hindi
  - आतिफ ने सीमा को बेवकूफ समझा
  - Atif ne Seema ko bewakuuf samjhaa
  - Atif Erg Seema Acc stupid consider.Pfv
  - ‘Atif considered Seema stupid.’
- English
  - Atif considered Seema stupid
  - Atif considered her stupid

## What is the Phenomenon?

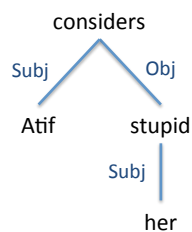
- Syntactically and semantically, *consider* takes a clausal complement
  - Atif considered [<sub>clause</sub> that she is stupid]
  - Atif considered [<sub>clause</sub> her stupid]
- But two problems:
  - No verb
  - *her* is semantically subject of *stupid* but has accusative case, which is unusual (subjects are usually nominative)
- So:
  - Atif considered [<sub>small clause</sub> her stupid]

## What is the Representation Type?

- For this example, we will show dependency trees and phrase structure trees

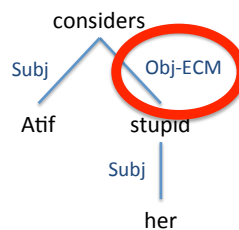
## Analysis 1a for Small Clauses: No Accusative Case Marking

- Structure represents *her* as subject but not accusative case marking of *her*



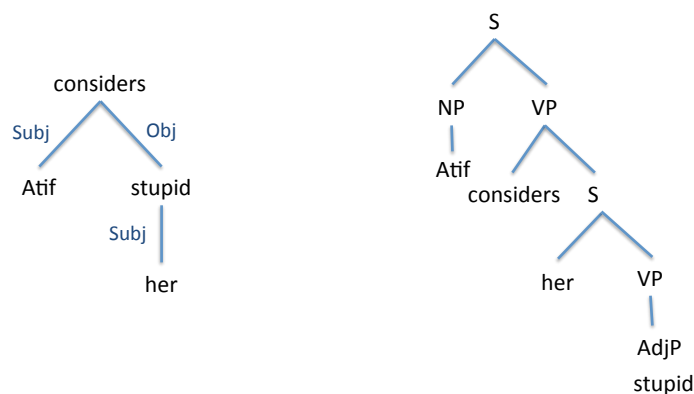
## Analysis 1b for Small Clauses: Exceptional Case Marking

- Structure represents *her* as subject and accusative case marking through node label



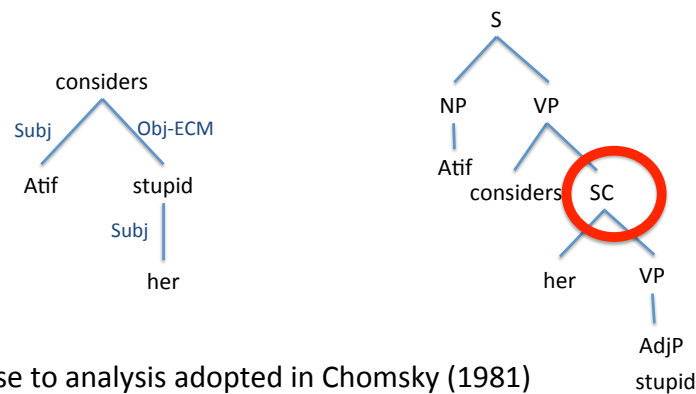
## Analysis 1a for Small Clauses: No Accusative Case Marking

- Structure represents *her* as subject but not accusative case marking of *her*



## Analysis 1b for Small Clauses: Exceptional Case Marking

- Structure represents *her* as subject but not accusative case marking of *her*

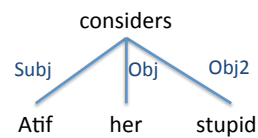


## Note on DS and PS

- These analyses are intuitively very similar
- Formal notion: “consistency” (Fei Xia, see Bhatt, Rambow & Fei 2011)
  - Intuition: very simple and general algorithm can transform consistent DS to PS and *vice versa*

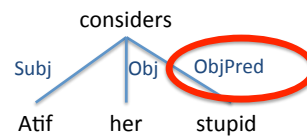
## Analysis 2a for Small Clauses: General Monoclausal Analysis

- Structure represents accusative case marking of *her* (as object of matrix verb) but not *her* as semantic subject



## Analysis 2b for Small Clauses: Syntactic Monoclausal Analysis

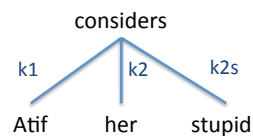
- Structure represents accusative case marking of *her* (as object of matrix verb) and *her* as semantic subject using node label





## Analysis 2b for Small Clauses: Syntactic Monoclausal Analysis

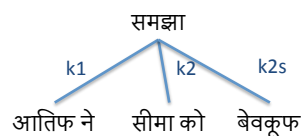
- Structure represents accusative case marking of *her* (as object of matrix verb) and *her* as semantic subject using node label



Neo-Paninian analysis

## Analysis 2b for Small Clauses: Syntactic Monoclausal Analysis

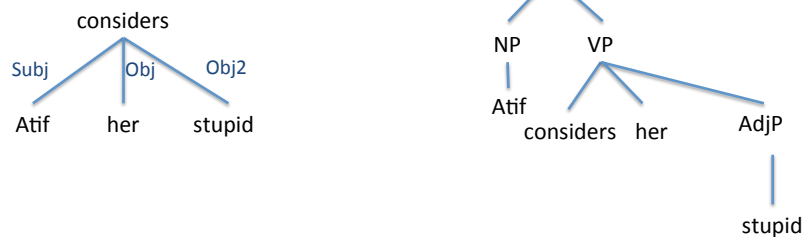
- Structure represents accusative case marking of *her* (as object of matrix verb) and *her* as semantic subject using node label



Neo-Paninian analysis from IIIT Hyderabad,  
Used for DS in Hindi-Urdu Treebank

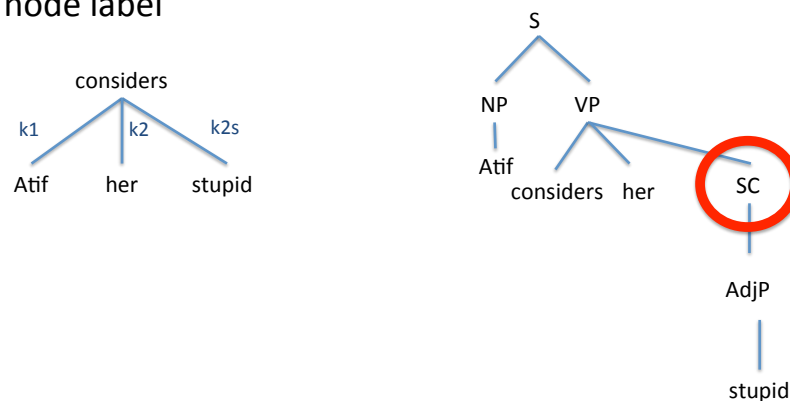
## Analysis 2a for Small Clauses: General Monoclausal Analysis

- Structure represents accusative case marking of *her* (as object of matrix verb) but not *her* as semantic subject



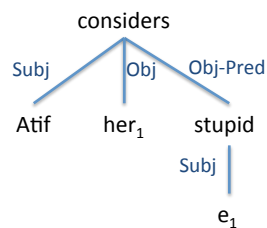
## Analysis 2b for Small Clauses: Syntactic Monoclausal Analysis

- Structure represents accusative case marking of *her* (as object of matrix verb) and *her* as semantic subject using node label



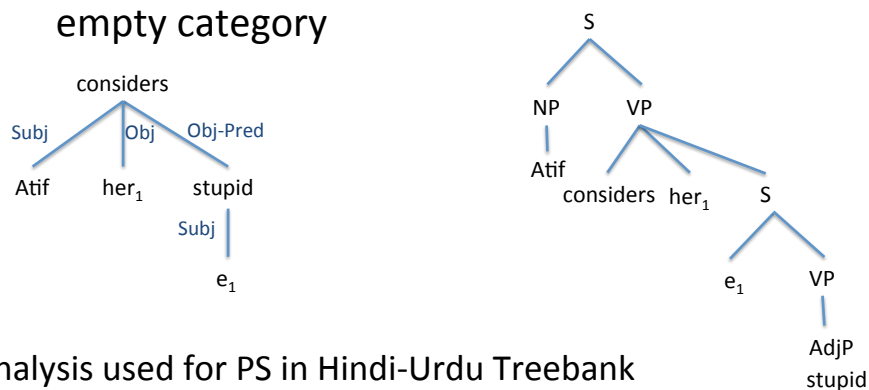
## Analysis 3 for Small Clauses: Raising to Object

- Structure represents accusative case marking of *her* and *her* as semantic subject but requires empty category



## Analysis 3 for Small Clauses: Raising to Object

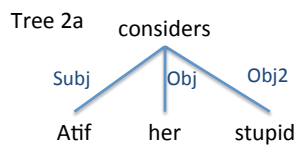
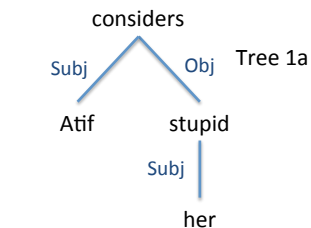
- Structure represents accusative case marking of *her* and *her* as semantic subject but requires empty category



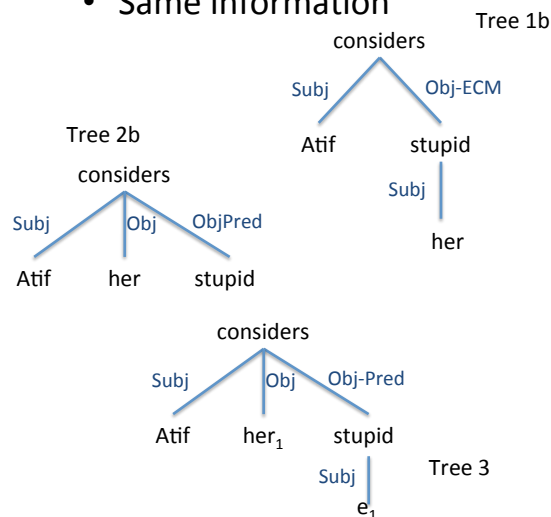
Analysis used for PS in Hindi-Urdu Treebank

## Comparison of Representations

- Less Information



- Same information

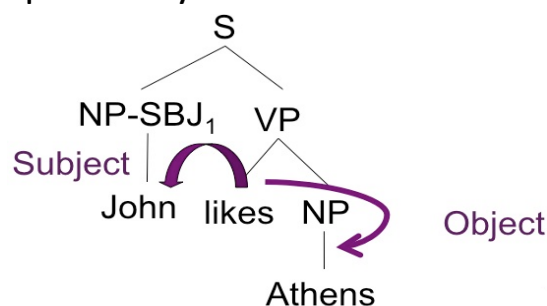


## Summary: Syntactic Phenomena, Representation Types, Analyses

- Syntactic phenomena are the empirical data of syntax as part of the science of language
  - Can be very similar across languages
- There can be several possible analyses
  - Some have less information
  - But there can be different analyses that represent the same information differently
- The analyses can be similar in DS and PS
- Lots of choices in treebank design!

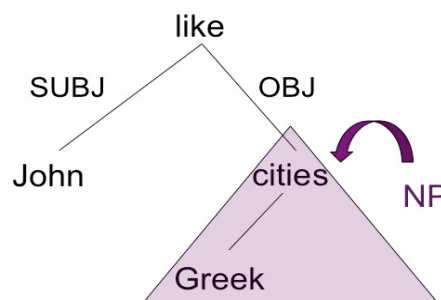
## Aren't DS and PS Representations Complementary? NO!

- Syntactic dependency can be encoded in PS, and typically is
- Usual convention: attachment in projection shows type of dependency



## Aren't DS and PS Representations Complementary? NO!

- Syntactic constituency is represented in DS
- Usual convention: each node is the word, and the head of the phrase containing it and all descendents



## What Does This Mean for NLP?

- Treebanks are not naturally occurring data
- The guidelines are painstakingly produced by linguists and represent a formal description of the language
- Annotators understand a sentence, determine what syntactic phenomena exist, and use the guidelines to choose an analysis for the sentence (a structure)
- Users of the treebank can use the guidelines to interpret the structures and get back the syntactic phenomena present
- These phenomena, and not their representation in the treebank, can be used for NLP in *whatever representation chosen by the researcher!*
- There is already lots of linguistics in our resources, we just need to make use of that linguistic information!

## The Hindi Treebank

- DS: dependency, annotated by hand
- PB: annotated by hand on top of DS, adds information about lexical semantics
  - Does not change trees
  - Adds labels to arcs and features to nodes
- PS: phrase structure, derived automatically from DS+PB
  - Contains less information than DS+PB
  - DS and PS contain different information

## Comparison of DS, PB, PS (Sample)

		DS	PB	PS
How?	Dependency	✓	✓	
	Phrase Structure			✓
What?	Distinguish unergative/unaccusative		✓	✓
	Distinguish temporal/locative adjuncts	✓	✓	
	Distinguish unaccusative/transitive with empty agent	✓		✓

## 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).

# **Introduction to Morphology, Syntax and Lexical Semantics of Hindi and Urdu**

Dipti Misra Sharma  
<[dipti@iiit.ac.in](mailto:dipti@iiit.ac.in)>  
LTRC, IIIT, Hyderabad  
India

Dec 8, 2012  
COLING 2012

## **Outline**

- Introduction
  - Some facts about Hindi and Urdu
- Linguistic properties
  - Morphology
  - Some basic Syntax
  - Lexical semantics



## Hindi: Some facts

- A major language of Indo-Aryan family
- Official language of 11 Indian states: Uttar Pradesh, Uttarakhand, Bihar, Delhi, Jharkhand, Chhattisgarh, Himachal Pradesh, Haryana, Rajasthan and Madhya Pradesh
- Also spoken outside India in Fiji, Mauritius, Guyana etc
- Number of speakers who returned Hindi as their mother tongue (in India) : 422,048,642 (41.03%) (2001, Census of India report)
- A large population in India who speak Hindi as their second language
- Script : Devanagari – a syllabic script

## Urdu : Some facts

- An Indo-Aryan language
- Evolved in India around eight-tenght centuries from 'khariboli', a dialect spoken in and around Delhi
- Significant borrowings from Arabic and Persian
- It was also known as 'rekhta' (mixed language)
- Official language of Pakistan
- Official language of ?? states of India
- Also spoken in Fiji, Bangladesh etc
- Number of speakers in India 51,536,111 (5.01%) (2001, Census of India report)
- Script Perso-Arabic

## Hindi-Urdu (Hindustani)

- Hindi and Urdu are mutually intelligible
- Linguists consider them as two registers of the same language
- Similar in grammatical structures
- Differ in vocabulary, particularly in the formal written varieties
- A mixed variety of the two is used as a lingua franca in India and is also known as Hindustani

### Some Basic characteristics of Hindi/Urdu

- Hindi/Urdu have relatively free word order
- The unmarked word order in both the languages is *subject-object-verb* (SOV).
- Auxiliary verbs follow the main verb.
- Nouns are followed by postpositions.
- Adjectives precede the nouns they modify.
- In Urdu, sometimes, adjectives follow the noun (*ezafe* constructions).
- Large use of *participles*, *complex predicates*, and *causatives*.
- Reduplication and echo-compounding are productively used in Hindi/Urdu (in fact almost all the Indian languages).

# Morphology

Hindi and Urdu have following morphological properties

- Grammatical gender: masculine and feminine
- Number: singular and plural
- Person: first, second and third
- Case: direct, oblique and vocative
- Adjectives inflect for number, gender and case
  - Some adjectives do not decline

## Nouns

- Nouns in Hindi/Urdu are inflected for *number* and *case*
  - Gender  
All nouns have inherent gender : *pankhaa* (fan.masc), *lataa* (creeper.fem), *ghar* (house.masc)
  - Number  
Singular : *pankhaa* (fan), *lataa* (creeper), *ghar* (house)  
Plural : *pankhe* (fans), *lataeM* (creepers), *ghar* (houses)
  - Case  
The case roles in Hindi are normally marked by postpositions. However, Hindi nouns reflect two cases morphologically,  
*Direct* and *Oblique*

## Case

- *Direct* nouns are in nominative and are not followed by a postposition
  - Occur denoting *subject* and/or *object*.

*LaRkaa aayaa* 'the boy came'  
<ladkaa,n,m,sg,3,d,,> <aa,v,m,sg,3,,yaa,>

*laRkii aayii* 'the girl came'  
<ladkii,n,f,sg,3,d,,> <aa,v,f,sg,3,,yaa,>

- *Oblique* nouns are objects of a *postposition* such as *ne (erg)*, *ko (acc,dative)*, *se (instr)*, *meM (loc)*, *par (loc)*, and *kaa (gen)*.

*laRke ne roTii khaayii* 'the boy ate bread'  
<laRkaa,n,m,sg,3,obl,,> <roTi,n,f,sg,3,d,,> <khaa,v,f,sg,3,,yaa,>

*laRke ne roTii ko zamiin se uThaayaa* 'the boy picked the bread from the floor'  
<laRkaa,n,m,sg,3,obl,,> <roTi,n,f,sg,3,obl,,> <zamiin,n,f,sg,3,obl,,> <uThaa,v,m,sg,3,,yaa,>

# Pronouns

Morphologically, like nouns, the pronouns also inflect for *number* and *case*.

**Sg - dir :** *yaha* (this), *vaha* (that), *jo* (who/which), *kaun* (who.interro), *kyaa* (what) and *kuch* (some)

**Sg – Obl :** *isa* (this), *usa* (that), *jisa* (which), *kisa* (which.interro), *koi* (someone), and *kisii* (someone)

**Pl - dir :** *ye* (these), *ve* (those), *jo* (who.pl)

**Pl – Obl**

**a. except before ne (erg) :** *ina* (these), *una* (those), *jina* (whoever), *kina* (who.interro), *kinhiiM* (who.indef)

**b. :** *inhoM* (these), *unhoM* (those), *jinhoM* (who), *kinhoM* (who) and *kinhiiM* (some people.indef)

## Pronouns (Contd...)

- **Before ‘ne’:** *maiM* (I) and *tuu* (you.sg) don’t change. For example, *maiMne* (I.erg) and *tuune* (you.erg)
- **Before other postpositions:** *maiM* (I) and *tuu* (you.sg) change to the oblique forms, *mujh* (me) and *tujh* (you.sg). Thus, *mujhko* (to me) and *tujhko* (to you.sg)
- **Before all postpositions:** *ham* (we), *tum* (you.sg/pl) and *aap* (you-hon) don’t change form. *hamne* (we.erg), *tumne* (you.erg), *aapne* (you-hon.erg), *humko* (to us), *tumko* (to you.sg/pl) *aapko* (to you-hon).
- *maiM* (I), *tuu* (you), *ham* (we), and *tum* (you) don’t attach to *kaa* (*gen*) postposition. Instead they have irregular forms *meraa* (my), *teraa* (your), *hamaaraa* (our), and *tumhaaraa* (your)

# Adjectives

- Morphologically, an adjective is inflected for *gender*, *number*, and *case* as it agrees with the following noun.
- Postpositions are attached only to the nouns. Adjectives preceding these nouns also have oblique form. *acche laRke ne* 'good boys erg'
- The transformations that an adjective (eg *acchaa* 'good') undergoes with regard to *number*, *gender* and *case* are given below

Case →	Ditect		Oblique	
Number → Gender ↓	Sg	Pl	Sg	Pl
Masc	acchaa	acche	acche	acche
Fem	acchii	acchii	acchii	acchii

# Verbs

- Verbs in Hindi/Urdu are inflected for tense, aspect, mood (TAM) and the agreement features of gender, number and person.
- Tense, aspect and mood are mostly expressed by auxiliaries in Hindi/Urdu. Thus, only certain moods, aspects and tense are marked in the verb forms.

Given below are some examples of various verb forms from Hindi/Urdu:

Root	<i>ro</i>	cry
Infinitive	<i>ronaa</i>	to cry
Habitual	<i>rotaa</i>	cry.hab
Perfective	<i>royaa</i>	cried
Causative	<i>ru<del>l</del>aa</i>	cause someone to cry
	<i>ru<del>l</del>vaa</i>	make someone to cause someone to cry

## Auxiliaries

- Auxiliaries mark Tense, Aspect and Modality information on verbs

(a) *bacce khaanaa khaa rahe haiM*

Children\_d meal eat prog be.pl.pres

'The children are having a meal'

(b) *bacce khaanaa khaate rah sakte haiM*

Children\_d meal eat\_nf prog ablit be.pl.pres

'The children can continue to have their meal'

- Auxiliaries also carry the gender, number and person information

## Postpositions

- Postpositions largely mark the case relations

*baccoM ne raat meM mez se khaanaa le liyaa*  
*children\_obl erg night in table ablat food take refl.pst*

- Hindi also has compound postpositions

## Compound Post-positions

*Compound postpositions* are formed by connecting the postpositions *ke*, *kii*, and *se* with other words as follows:

<i>ke anusaar</i>	‘according to’
<i>ke alaavaa</i>	‘in addition to’
<i>ke kaaran</i>	‘because of’
<i>ke dvaaraa</i>	‘through’
<i>ke saamne</i>	‘in front of’
<i>ke liye</i>	‘for’
<i>kii ora/taraf</i>	‘towards’
<i>kii tarah</i>	‘like’
<i>kii jagah</i>	‘in place of’
<i>se baahar</i>	‘out of’
<i>se pahle</i>	‘before’

## Urdu Specific Features

Prepositions in Urdu  
'ezafe' in Urdu



## Urdu has Prepositions

- Unlike Hindi, Urdu has prepositions as well.
  - Some Urdu examples with prepositions are:
- (a) *qabl* 'before'  
qabl az\_ayn (qabl azeen)      qabl az\_waqt  
'before' 'from\_this'      'before' 'from\_time'  
'before this'      'before time'
- (b) *dar* 'in, inside, amidst'  
dar\_ayn asnah (dariin asnah)  
'in\_this' 'moment'
- (c) *az* 'from, since, for'  
az raahe hamdardi,      az sare nau,      khaarij az bahes  
'from' 'way' 'empathy'      'from' 'beginning' 'new'      'beyond' 'from' 'discussion'  
'for the sake of courtesy'
- (d) *ta* 'to, until, till'  
ta waqt  
'until' 'time'

## 'ezafe' in Urdu

- Urdu has what is referred to as 'ezafe'.
- Normally marks a genitive but is not restricted to genitive alone

### **EZ: N+N**

*daur-e-hukumat*  
'period-of-rule'

*hukumat-e-hind*  
'government-of-India'

### **EZ: N+Adj**

*nasl-e-insani*  
'race-of-humanity'

*lamha-e-aakhar*  
'The last moment'

### **EZ: Adj+N**

*qabil-e-rahem*  
'qualified for sympathy'

### **EZ: Adj+Adj**

*qabil-e-qubul*  
'qualified-for-acceptance'

## Reduplication: A morphological processes

- Words belonging to various categories can be reduplicated
- These expressions are often hyphenated
- Reduplication has various morphological functions depending on the lexical category which is reduplicated. For example,
  - Nouns : it adds the sense of 'every',
  - Verbs : it brings the sense of adverbial participle
  - Adjectives and adverbs: it adds intensity
- Hindi has three types of reduplication: *full*, *partial* and *redundant*
- Reduplication is highly productive in these languages

## Full Reduplication

If the word is *X* then its reduplicated form is *X-X*.

<i>raam-raam</i>	‘Ram-Ram’ (proper noun)
<i>baccaa-baccaa</i>	‘child-child’ (common noun)
<i>garam-garam</i>	‘hot-hot’ (adjective)
<i>dhiire-dhiire</i>	‘slowly-slowly’ (adverb)
<i>jaa-jaa</i>	‘go-go’ (verb)
<i>naa-naa</i>	‘not-not’ (negative particle)
<i>kyaa-kyaa</i>	‘what-what’ (question word)
<i>jaate-jaate</i>	‘going-going’ (participle)

## Partial Reduplication (Echo words)

- In partial reduplication, an expression X is repeated partially
- Only a part of a given word is repeated which gives the meaning of '*X etc.*'
- In Hindi/Urdu The *first consonant of X* is replaced by *v-*.

For example, *khaanaa-vaanaa* 'food-etc.'

- *vaanaa* is not a valid word in Hindi

Some more examples of partial reduplication in Hindi are:

*jaanaa* 'going' → *jaanaa-vaanaa* 'going etc.'  
*aaloo* 'potato' → *aaloo-vaaloo* 'potato etc.'  
*aisaa* 'like this' → *aisaa-vaisaa* 'like this etc.'

There are also examples which do not fall in this pattern

The meaning of such words changes substantially and does not have the sense of 'etc.'

*bhaag* 'run' → *bhaagambhaag* 'rush'  
*jhuuth* 'lie' → *jhuuth-muuth* 'just like that (without meaning it)'  
*dekh* 'see' → *dekhaa-dekhii* 'in imitation'

## Some Basic Syntax

- Hindi and Urdu are both relatively free word order SOV languages
- For case marking, Hindi primarily uses postpositions.
- The verb agrees either with subject or with object
- Adjective agrees with the noun it modifies

## Simple Transitive

- trans-1: आतिफ़ किताब पढ़ेगा  
Atif kitab paRhegaa  
Atif book.f read.m.sg.fut  
'Atif will read the book'
- trans-2: आतिफ़ ने किताब पढ़ी  
Atif ne kitaab paRhii  
Atif erg book.f read.f.sg.pst  
'Atif read the book'
- trans-3: आतिफ़ को किताब पढ़नी पड़ी  
Atif ko kitaab paRhnii paRii  
Atif dat book.f read.f.inf compel.f.pst  
'Atif had to read the book'

## Intransitive: Unergative

- Unerg-1: आतिफ़ सोएगा  
Atif soyegaa  
Atif sleep.m.sg.fut  
'Atif will sleep'
- unerg-2: \*आतिफ़ ने सोया  
\*Atif ne soyaa  
Atif erg sleep.m.sg.pst  
'Atif slept'
- unerg-3: आतिफ़ को सोना पड़ेगा  
Atif ko sonaa paRegaa  
Atif dat sleep.inf compel.fut  
'Atif will have to sleep'

## Intransitive: Unaccusative

- unacc-1: दरवाज़ा खुलेगा  
darvaazaa khulegaa  
door.m.sg.d open.m.sg.fut  
'The door will open'
- unacc-2: \*दरवाज़े ने खुला  
\*darvaaze ne khulaa  
door.m.sg.obl erg open.pst  
'The door opened'
- unacc-3: दरवाज़े को खुलना पड़ेगा  
darvaaze ko khulnaa paRegaa  
door.m.sg.obl dat open.inf compel.fut  
'The door will have to open'

## Existential

- exist-1: उस कमरे में चूहे हैं  
us kamre meM cuuhe haiM  
that room in rats be.pres.pl  
'There are rats in that room'

## Dative Subject

unacc-4: कल रात बादलों में चाँद दिखा  
kal raat baadaloM meM caaMd dikhaa  
yesterday night clouds in moon see(unacc).pst  
'Yesterday night, the moon was seen behind the clouds'

dat-subj-1: कल रात बादलों में मुझको चाँद दिखा  
kal raat baadaloM meM mujhko caaMd dikhaa  
yesterday night clouds in me.dat moon see(unacc).pst  
'Yesterday night, I saw the moon behind the clouds'

## Ditransitive

ditrans-1: राम मोहन को किताब देगा  
raam mohan ko kitaab degaa  
Ram Mohan dat book.f give.m.sg.fut  
'Ram gave a book to Mohan'

ditrans-2: राम ने मोहन को किताब दी  
raam ne mohan ko kitaab dii  
Ram erg Mohan dat book.f give.f.sg.pst  
'Ram gave a book to Mohan'

## Complement Clause

compl-cl-1: राम जानता है कि सीता देर से आएगी  
raam jaantaa hai ki siita der se aayegii  
Ram know.hab.m.sg be.sg.pres that Sita late part come.f.sg.fut  
'Ram knows that Sita will arrive late'

## Relative Clause

rel-cl-1: मेरी बहन जो दिल्ली में रहती है कल आ रही है  
merii bahan jo dillii meM rahtii hai kal aa  
My sister who Delhi in stay.hab.f.sg be.sg.pres tomorrow come  
rahii hai  
prog.f.sg be.sg.pres  
'My sister who stays in Delhi is coming tomorrow'

## Relative Clause

rel-cl-2: मैंने वह किताब जो तुमने दी थी पढ़ ली  
maiMne vah kitaab jo tumne dii thii paRh lii  
I.erg that book.f which you.erg give.f.sg.pst be.f.sg.pst read refl.f.sg.pst  
'I have read the book which you gave me'

rel-cl-3: मैंने वह किताब पढ़ ली जो तुमने दी थी  
maiMne vah kitaab paRh lii jo tumne dii thii  
I.erg that book.f read refl.f.sg.pst which you.erg give.f.sg.pst be.f.sg.pst  
'I have read the book which you gave me'

rel-cl-4: जो किताब तुमने दी थी वह मैंने पढ़ ली  
jo kitaab tumne dii thii vah maiMne paRh lii  
which book.f you.erg give.f.sg.pst be.f.sg.pst that I.erg read refl.f.sg.pst  
'I have read the book which you gave me'

## Complex Predicate

compl-pred-1: राम रवि की प्रतीक्षा कर रहा था  
raam ravi kii **pratikshaa kar rahaa thaa**  
Ram Ravi gen wait do prog.m.sg be.m.sg.pst  
'Ram was waiting for Ravi'

compl-pred-2: राम रवि को याद कर रहा था  
raam ravi ko **yaad kar rahaa thaa**  
Ram Ravi acc remember do prog.m.sg be.m.sg.pst  
'Ram was remembering Ravi'



## Causatives

- Unerg-1: आतिफ़ सोएगा  
Atif soyegaa  
Atif sleep.m.sg.fut  
'Atif will sleep'
- causative-1: आया ने आतिफ़ को सुलाया  
aayaa ne Atif ko sulaayaa  
maid erg Atif acc sleep.caus.pst  
'The maid caused the child to sleep'
- causative-2: माँ ने आया से आतिफ़ को सुलवाया  
maaN ne aayaa se Atif ko sulvaayaa  
mother erg maid by Atif acc sleep.caus.pst  
'The mother made the maid to cause the child to sleep.'

## Lexical Semantics

Semantic properties of certain verb types seem to affect the case selection for certain arguments. For example,

➤ Experiencer verbs

- The experiencer argument takes dative case

**raam ko bukhaar hai**, **raam ko caand dikhaa**, **raam ko dukh hai**  
'Ram' 'dat' 'fever' 'be.pres', 'Ram' 'dat' 'moon' 'see-unacc.pst', 'Ram' 'dat' 'sorrow' 'be.pres'

➤ Participatory verbs

- The second argument of the participatory verbs takes 'se' postposition

**raam ravi se carcaa karega**, **siitaa raam se shaadi karegii**,  
'Ram' 'Ravi' 'to' 'discussion' 'do.m.sg.fut', 'Sita' 'Ram' 'with' 'marriage' 'do.f.sg.fut',

**ravi mohan se milegaa**

'Ravi' 'Mohan' 'with' 'meet.m.sg.fut'

## References

- ❑ Agnihotri, Rama K. 2007. *Hindi, An Essential Grammar*. Routledge, London and New York.
- ❑ Kachru, Yamuna. 2006. *Hindi*. London Oriental and African Language Library.
- ❑ McGregor, R. S. 1995. *Outline of Hindi Grammar*. Oxford University Press.
- ❑ Dr. Sharma, A. 1975. *A Basic Grammar of Modern Hindi*. New Delhi.

## 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).

## Representing Tokens, Morph Analysis, POS and Chunks in The Hindi/Urdu Treebanks

## Outline

- Tokenization
- Morphological Representation
- POS tagging
- Chunking
- Inter-chunk dependency annotation
- Intra-chunk dependencies

## Tokenization

- Automatic
- Issues
  - Compounds
  - Punctuations

For example,

<i>usa</i>	<i>ladake</i>	<i>ne</i>	<i>kelaa</i>	<i>khaayaa</i>	<i>thaa</i>
that	boy	erg	banana	eat-perf	past

# Tokenization

*Represented in SSF*

**ADDR    TOKEN**

1	usa
2	laDake
3	ne
4	kelA
5	khAyA
6	thA
7	.

## Tokenization: Issues

- Punctuations

All punctuations to be tokenized

- Compounds

BAI-bahana (brother-sister), bAlIkA-vixyAlaya (girl-school)

- Compounds internally contain a punctuation
- Are productive
- Morphological analysis of the members of the compounds
- The issue, whether to create a single token
- Decision
- Create three tokens
- Mark the hyphen as 'JOIN'

## Morph Analysis and its Representation

**'af'** defines the composite attribute consisting of root, category, gender, number, person, case, tam (tense, aspect, modality)/vibhakti(case marker), suffix

ADDR_	TKN_	OTHR
1	usa	<fs af='vaha,pr,,, '>
2	laDake	<fs af='laDakaa,n,m,sg,3,o,, '>
3	ne	<fs af='ne,psp,,,,, '>
4	kelaa	<fs af='kelaa,n,m,sg,3,o,,,,, '>
5	khaayaa	<fs af='khaa,v,m,sg,any,,yaa, '>
6	thaa	<fs af='kelaa,v,e,,, '>
7	.	<fs as='&STOP,punc,,,,, '>

## POS Tagging

- ILMT POS Tagsets adopted
- Total 26 tags

ADDR	TKN	CAT	OTHR
1	usa	PRP	<fs af='vaha,pron... '>
2	laDake	NN	<fs af='laDakA,noun... '>
3	<u>ne</u>	PSP	<fs af='ne,psp... '>
4	kelA	NN	<fs af='kelA,noun... '>
5	khAyA	VM	<fs af='KA,verb... '>
6	thA	VAUX	<fs af='kelA,verb... '>
7	.	SYM	<fs as='&STOP,punc,,,,, '>

## Chunking

- Chunking is introduced to save the effort in manual tagging
- Dependency relations are marked between the chunk heads
- Chunking restructures the tree, i.e.,

ADDR_	TKN_	CAT_	OTHR
1	((	NP	
1.1	usa	PRP	<fs af='vaha,pron... '>
1.2	laDake	NN	<fs af='laDakA,noun... '>
1.3	<u>ne</u>	PSP	<fs af='n&sp how
	)		
2	((	NP	
2.1	kelA	NN	<fs af='kelA,noun... '>
	)		
3	((	VG	
3.1	khAyA	VM	<fs af='KA,verb... '>
3.2	thA	VAUX	<fs af='kelA,verb... '>
4.	((	BLK	
4.1	.	SYM	<fs as='&STOP,punc,,,, '>
	)		

## 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](mailto: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 linguistic resources.
- 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
  - Conversational 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*

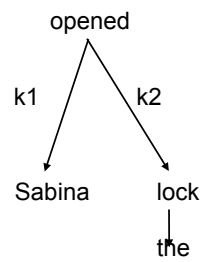
## Panini's Grammar

- 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

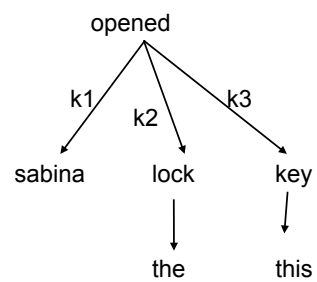
- Treats a sentence as a series of modifier-modified relations
- Every sentence has a primary modified (generally a verb)
- Relations between verbs and their participants called 'karaka'
- Other relations – such as reason, purpose, genitive etc
- The relations are expressed through explicit markers called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

Sabina opened the lock



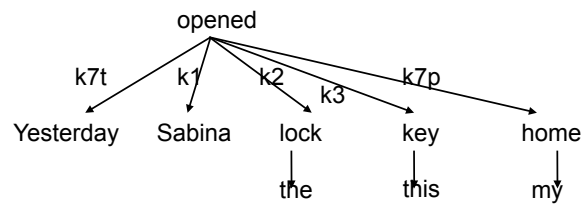
K1 (Karta) : the doer of the action (the locus of activity)  
 K2 (Karma) : locus of result

Sabina opened the lock with this key



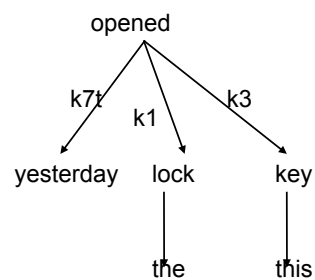
K3 (karaNa) : instrument

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 :  
*raama*H (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*

=>

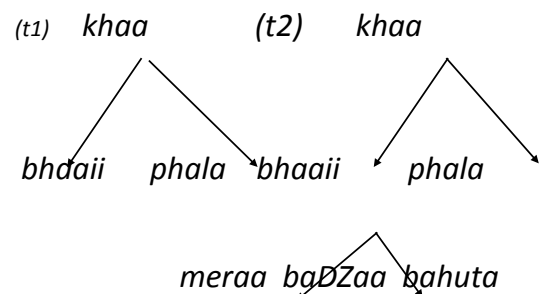
*meraa*<sub>PRP</sub> *baDzaa*<sub>JJ</sub> *bhaaii*<sub>NN</sub>  
*bahuta*<sub>QF</sub> *phala*<sub>NN</sub> *khaataa*<sub>VM</sub> *hai*<sub>VAUX</sub>

=>

*((meraa*<sub>PRP</sub> *baDzaa*<sub>JJ</sub> ***bhaaii***<sub>NN</sub>*))*<sub>NP</sub>  
*((bahuta*<sub>QF</sub> ***phala***<sub>NN</sub>*))*<sub>NP</sub>  
*((khaataa*<sub>VM</sub> *hai*<sub>VAUX</sub>*))*<sub>VG</sub>

## Example Contd...

*((meraa*<sub>PRP</sub> *baDzaa*<sub>JJ</sub> ***bhaaii***<sub>NN</sub>*))*<sub>NP</sub>  
*((bahuta*<sub>QF</sub> ***phala***<sub>NN</sub>*))*<sub>NP</sub>  
*((khaataa*<sub>VM</sub> *hai*<sub>VAUX</sub>*))*<sub>VG</sub>



## 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*

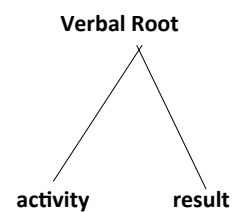
*The lock opened*

## Semantics of the verb

- A verbal root denotes:

- The activity
- The result

- Locus of activity : *karta*
- Locus of result : *karma*

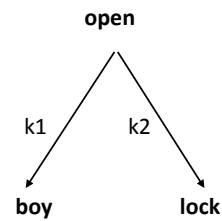




## *karta - karma*

- The boy opened the lock

- k1 – *karta*
- k2 – *karma*



- *karta*, *karma* sometimes correspond to agent/theme

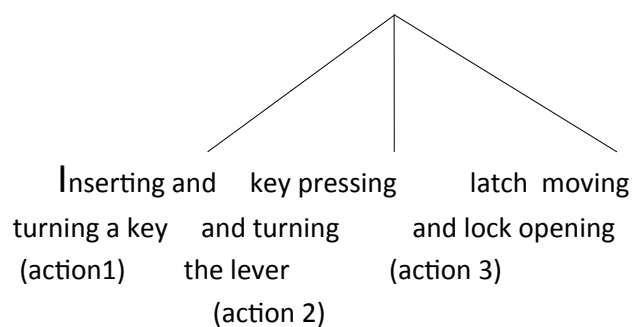
- Not always

*The door opened*

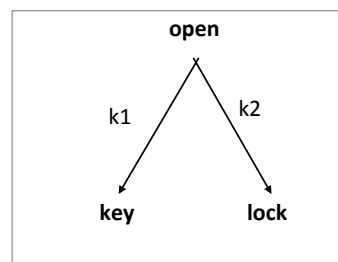
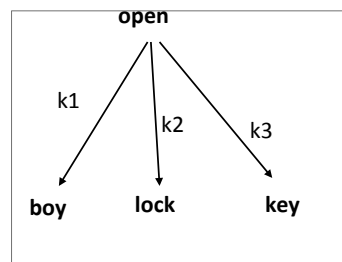
- 'The door' is *karta*
- The sentence has no explicit *karma*

## Sub-actions - Opening of lock

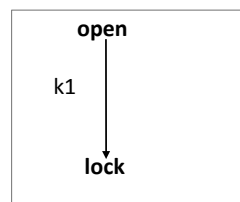
### Opening of lock



## 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:

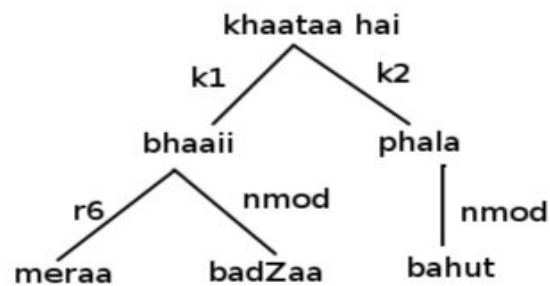
□ *meraa* *PRP*   *baDzaa* *JJ*   *bhaaai* *NN*   *bahuta* *QF*  
*phala* *NN*   *khaataa* *VM*   *hai* *VAUX*

### ➤ Chunking:

□ ((*meraa* *PRP*)) *\_NP*  
 ((*baDzaa* *JJ*   *bhaaai* *NN*)) *\_NP*  
 ((*bahuta* *QF*   *phala* *NN*)) *\_NP*  
 ((*khaataa* *VM*   *hai* *VAUX*)) *\_VG*

## An Example (Contd...)

### ➤ Dependency Relation



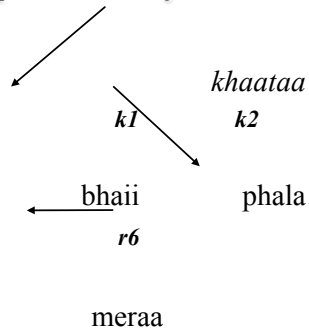
## Dependency Scheme

- 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’
  - ❑ *bhaaai* ‘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

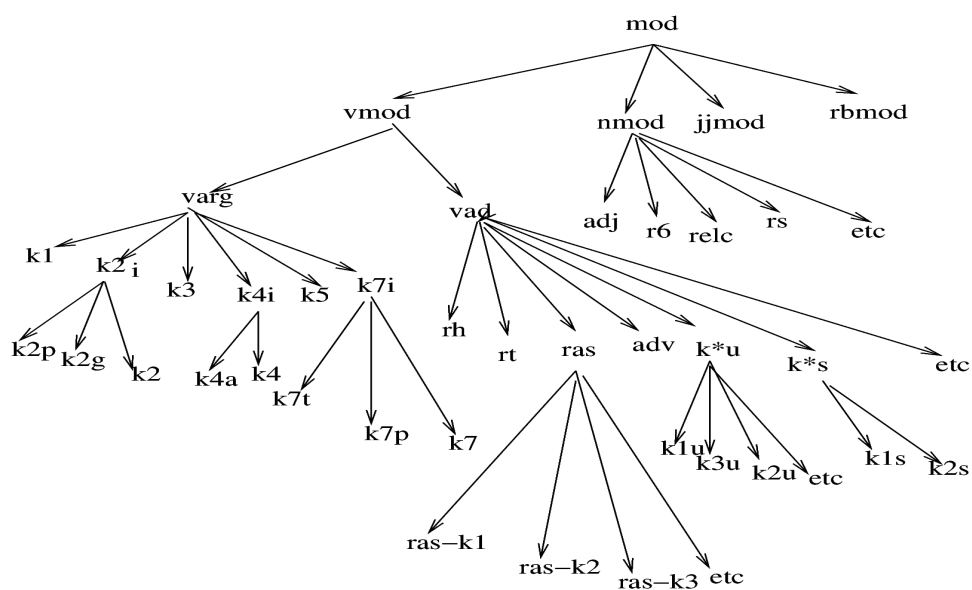
## Relations other than *karakas*

- *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:

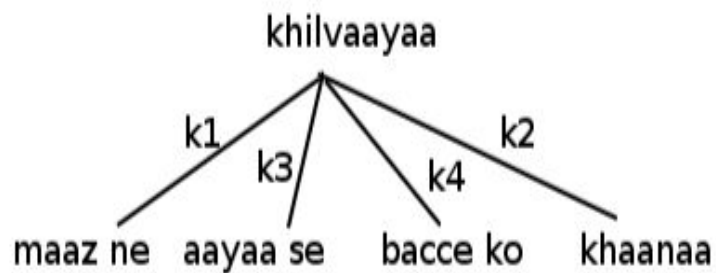
- *maaz ne aayaa se bacce ko khaanaa khilvaayaa*  
 'mother' 'Erg.' 'maid' 'by' 'child' 'Acc.' 'food' 'eat-Caus.'  
 'Mother caused the maid to feed the child.'

#### ➤ 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 ...)

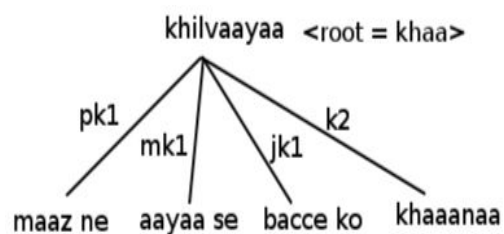
### ➤ Possibility-II:

- 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'.



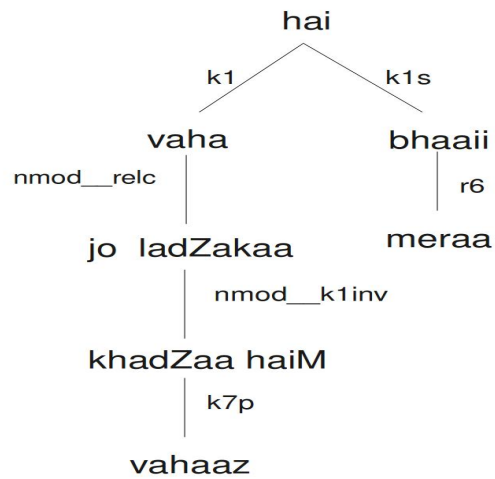
### Causative Constructions (Contd ...)

- *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 bhaai 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

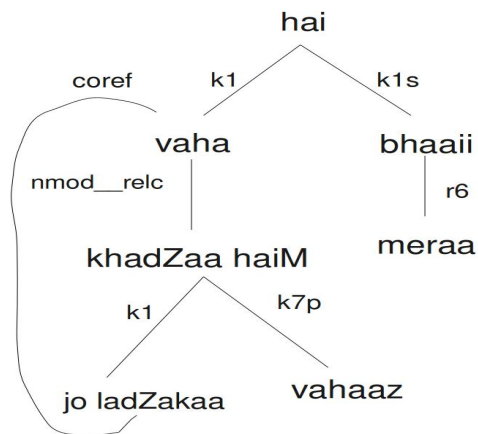


### Relative Clauses (nmod\_\_relc)

#### ➤ 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*.

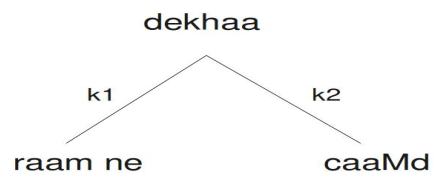
### *anubhava karta – k4a (Contd ..)*

- **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*
-



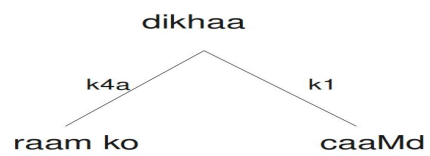
*anubhava karta – k4a (Contd...)*

➤ *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.*

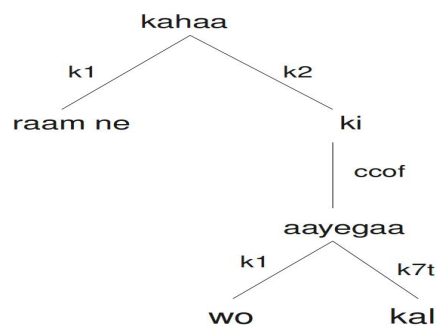
‘Ram said that he will come tomorrow.’

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

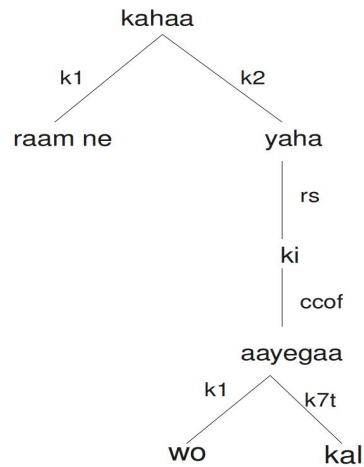
➤ In *Ex-2*, the clause ‘*ki vo kal aayegaa*’ is the complement of the object *yaha* ‘*this*’ so it attaches to *yaha* as *rs*.

#### Relation samanadhikaran- rs (Contd...)

➤ *Ex-1*



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



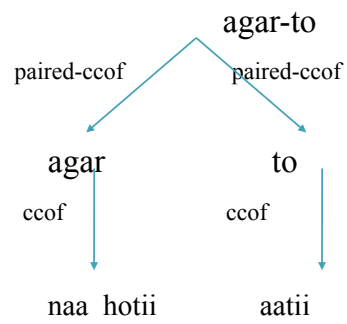
### (5) Conditionals

- *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.'

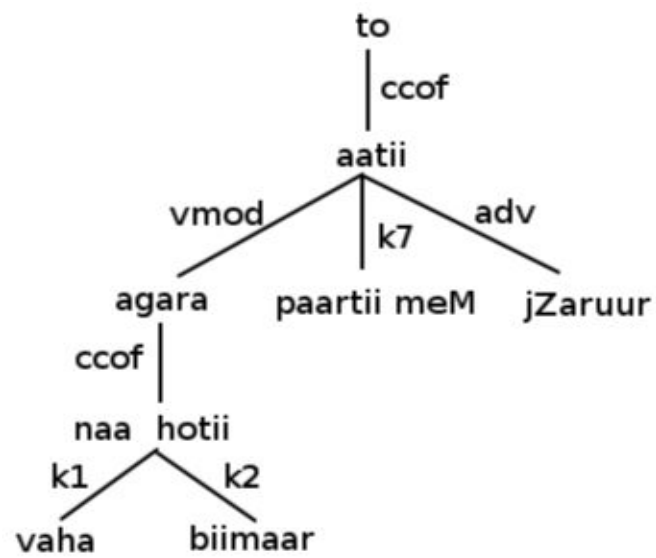
#### ➤ Issue:

- ❑ 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)

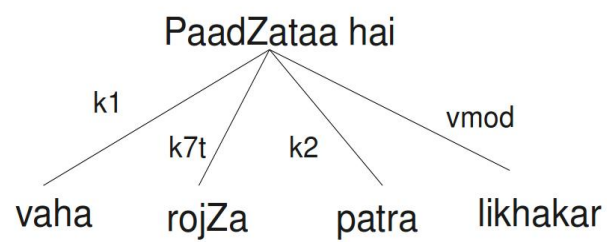
- In non-adjectival participles, an argument of a verb (main) is shared with another verb(participle).
- The arguments occurs only once in the sentence but is semantically related to both the verbs.
- The shared argument syntactically always attaches with the main verb.
- For the other verb this argument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

## Participles (vmod) (Contd ..)

➤ *Ex: vaha rojZa patra likhakara PaadZataa hai*

'he' 'daily' 'letter' 'having written' 'tear' 'is'

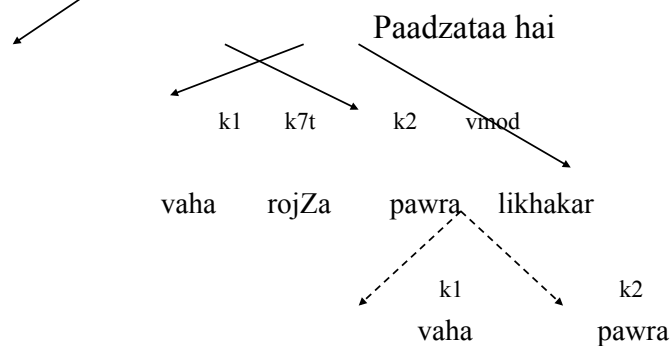
'Having letters written everyday he tears.'



## Participles (vmod) (Contd ..)

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

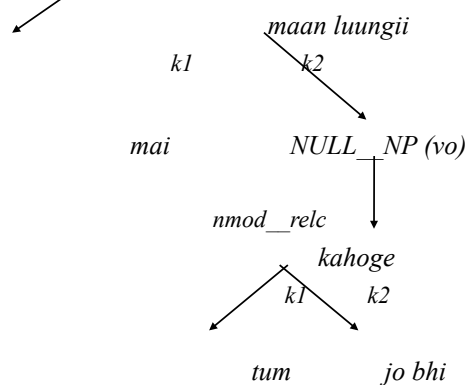
## Participles (vmod) (contd..)



## (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.

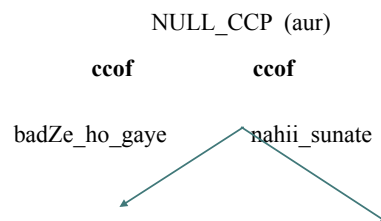
## Ellipsis (Contd...)





### 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 reflect 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...)

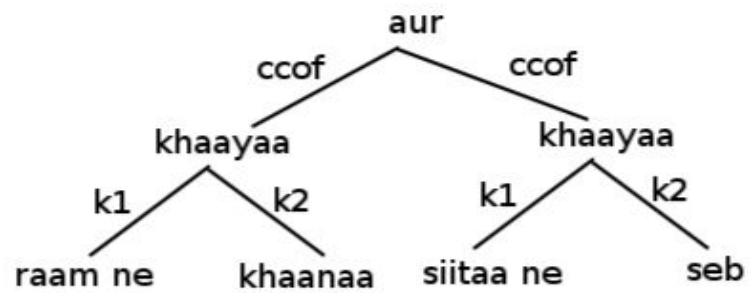
### ➤ Coordinate Conjunction

- ❑ *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.'

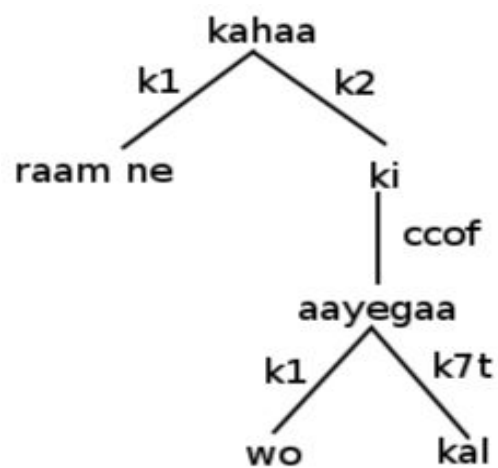
### ➤ Subordinate Conjunction

- ❑ *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



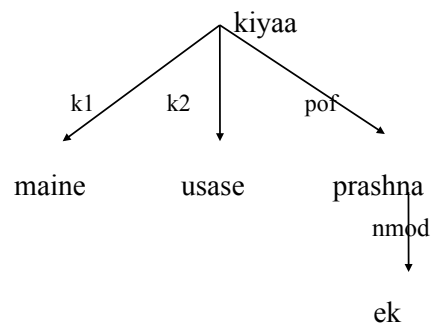
## (2) Conjunct Verbs

- *Ex: maine usase ek prashna kiya*  
‘I-erg’ ‘him-inst’ ‘one’ ‘question’ ‘did’  
‘I asked him a question’
- The noun *prashna* ‘question’ within the conjunct verb sequence *prashna kiya* ‘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 kiya* 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 kiya* into two separate chunks, [*ek prashna*]/*NP* [*kiya*]/*VG*.
- The dependency relation of *prashna* with *kiya* 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..)



## 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).

## Lexical Semantic Representation for Hindi & Urdu : principles, representation and analysis

Ashwini Vaidya  
University of Colorado, Boulder

## Contents

1. Motivation
2. Introducing PropBank
3. Frame file definition
4. Hindi PropBank
5. Linguistic Phenomena

## Why is semantic information important?

- Imagine an automatic question answering system
- Who created the first effective polio vaccine?
- Two possible choices:
  - Becton Dickinson created the first disposable syringe for use with the mass administration of the first effective polio vaccine
  - The first effective polio vaccine was created in 1952 by Jonas Salk at the University of Pittsburgh

## Word Matches

- Who created the first effective polio vaccine?
  - Becton Dickinson created the first disposable syringe for use with the mass administration of the first effective polio vaccine
  - The first effective polio vaccine was created in 1952 by Jonas Salk at the University of Pittsburgh

## Parsing

- Who created the first effective polio vaccine?
  - [Becton Dickinson] created the [first disposable syringe] for use with the mass administration of the first effective polio vaccine
  - [The first effective polio vaccine] was created in 1952 by [Jonas Salk] at the University of Pittsburgh



## Semantic Role labelling

- Who created the first effective polio vaccine?
  - [Becton Dickinson<sub>agent</sub>] created the [first disposable syringe<sub>theme</sub>] for use with the mass administration of the first effective polio vaccine
  - [The first effective polio vaccine<sub>theme</sub>] was created in 1952 by [Jonas Salk<sub>agent</sub>] at the University of Pittsburgh

## SRL gives us the right answer

- We need semantic information to prefer the right answer
- The theme of create should be 'the first effective polio vaccine'
- The theme in the first sentence was 'the first disposable syringe'
- We can filter out the wrong answer

## We need semantic information

- To find out about events and their participants
- To capture semantic information across syntactic variation

## We need semantic information

- To find out about events and their participants
- To capture semantic information across syntactic variation

## Semantic information

- Semantic information about verbs and participants expressed through semantic roles
- Agent, Experiencer, Theme, Result etc.
- However, difficult to have a standard set of thematic roles

## Proposition Bank

- Proposition Bank (PropBank) provides a way to carry out general purpose Semantic role labelling
- A PropBank is a large **annotated** corpus of predicate-argument information
- A set of semantic roles is defined for *each* verb
- A syntactically parsed corpus is then tagged with verb-specific semantic role information

## PropBank Frame files

- PropBank defines semantic roles on a verb-by-verb basis
- This is defined in a verb lexicon consisting of *frame files*
- Each predicate will have a set of roles associated with a distinct usage
- A polysemous predicate can have several rolesets within its frame file

## An example

- John rings the bell

ring.01	Make sound of bell
Arg0	Causer of ringing
Arg1	Thing rung
Arg2	Ring for


## An example

- John rings the bell
- Tall aspen trees ring the lake

ring.01	Make sound of bell
Arg0	Causer of ringing
Arg1	Thing rung
Arg2	Ring for

ring.02	To surround
Arg1	Surrounding entity
Arg2	Surrounded entity

## An example

- [John] **rings** [the bell]  Ring.01
- [Tall aspen trees] **ring** [the lake]  Ring.02

ring.01	Make sound of bell
Arg0	Causer of ringing
Arg1	Thing rung
Arg2	Ring for

ring.02	To surround
Arg1	Surrounding entity
Arg2	Surrounded entity

## An example

- [John<sub>ARG0</sub>] rings [the bell<sub>ARG1</sub>] ← Ring.01
- [Tall aspen trees<sub>ARG1</sub>] ring [the lake<sub>ARG2</sub>] ← Ring.02

ring.01	Make sound of bell
Arg0	Causar of ringing
Arg1	Thing rung
Arg2	Ring for

ring.02	To surround
Arg1	Surrounding entity
Arg2	Surrounded entity

## Hindi PropBank

- Annotating Hindi PropBank involves three steps:
  - Creation of frame files
  - Empty argument insertion
  - Semantic role labelling

## Frame files for Hindi

- Two types of frame files:
  - Frames for simple verbs [385 frames; 703 predicates ]
  - Frames for nominals in complex predicates [1875; 1902 predicates]

## Empty Arguments

- PropBank inserts 4 types of empty arguments
  - **pro** :dropped null arguments; recoverable from discourse context
  - **PRO**: empty subjects of non-finite complement and adjunct clauses
  - **RELPRO**: gaps in participial relative clauses
  - **GAP**: elided arguments in co-ordinated clauses
- **PRO** and **RELPRO** are inserted automatically
- **GAP** and **pro** are inserted manually

## PropBank Tagset

Numbered Arguments	Numbered Arguments with function tags
ARGA: Causer	ARGA-MNS: Indirect causer
ARG0: Agent, experiencer	ARG0-MNS: Induced causer
ARG1: Theme, patient	ARG0-GOL: Causee with a 'recipient' role
ARG2: Recipient	ARG2-ATR: Attribute
ARG3: Instrument	ARG2-GOL: Goal
	ARG2-SOU: Source
	ARG2-LOC: Location
	ARG2-DIR: Direction

## PropBank Tagset

Modifier Arguments
ARGM-TMP : Temporal
ARGM-MNR : Manner
ARGM-LOC : Location
ARGM-PRP: Purpose
ARGM-CAU : Cause
ARGM-DIS : Discourse
ARGM-ADV : Adverb
ARGM-MNS : Means



## Linguistic phenomena

- Simple transitive
- Unaccusative and Unergative
- Existential
- Dative subject
- Ditransitive
- Causatives
- Complex Predicates

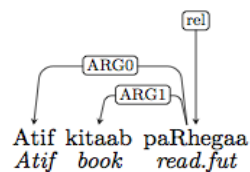
## Simple Transitive

trans-1: आतिफ़ किताब पढ़ेगा

Atif kitab paRhegaa

Atif book.f read.m.sg.fut

'Atif will read the book'

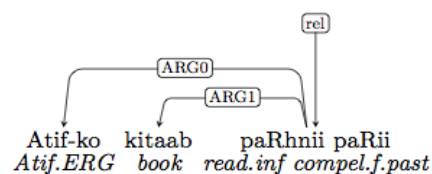


trans-2: आतिफ़ ने किताब पढ़ी

Atif ne kitaab paRhii

Atif erg book.f read.f.sg.pst

'Atif read the book'



trans-3: आतिफ़ को किताब पढ़नी पड़ी

Atif ko kitaab paRhonii paRii

Atif dat book.f read.f.inf compel.f.pst

'Atif had to read the book'

## Unaccusative & Unergative

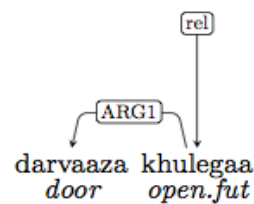
- Distinction between intransitive verbs:
  - unaccusatives e.g. Kula (open), Puta (explode)
  - Unergatives e.g. haMsa (laugh),
- Single argument of unaccusatives takes Arg1, unergatives take Arg0
- Diagnostic tests are used to distinguish unaccusatives from unergatives
  - E.g. animacy test, cognate object test among others

## Intransitive: Unaccusative

unacc-1: दरवाजा खुलेगा  
 darvaazaa khulegaa  
 door.m.sg.d open.m.sg.fut  
 'The door will open'

unacc-2: \*दरवाजे ने खुला  
 \*darvaaze ne khulaa  
 door.m.sg.obl erg open.pst  
 'The door opened'

unacc-3: दरवाजे को खुलना पड़ेगा  
 darvaaze ko khulnaa paRegaa  
 door.m.sg.obl dat open.inf compel.fut  
 'The door will have to open'



## Intransitive: Unergative

Unerg-1: आतिफ़ सोएगा

Atif soyegaa

Atif sleep.m.sg.fut

'Atif will sleep'

unerg-2: \*आतिफ़ ने सोया

\*Atif ne soyaa

Atif erg sleep.m.sg.pst

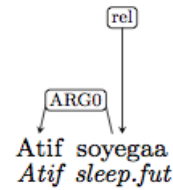
'Atif slept'

unerg-3: आतिफ़ को सोना पड़ेगा

Atif ko sonaa paRegaa

Atif dat sleep.inf compel.fut

'Atif will have to sleep'



## Existential

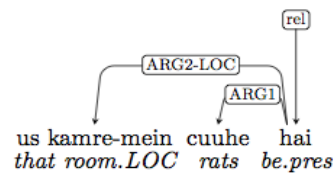
exist-1: उस कमरे में चूहे हैं

us kamre meM cuuhe haiM

that room in rats be.pres.pl

'There are rats in that room'

- We distinguish between existential and copula sentence types by means of different roleset IDs



## Dative Subject

unacc-4: कल रात बादलों में चाँद दिखा

kal raat baadaloM meiM caaMd dikhaa

yesterday night clouds in moon see(unacc).pst

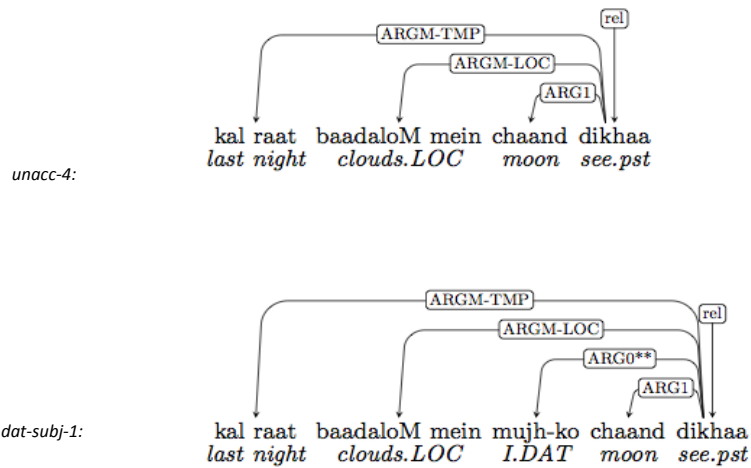
'Yesterday night, the moon was seen behind the clouds'

dat-subj-1: कल रात बादलों में मुझको चाँद दिखा

kal raat baadaloM meiM mujhko caaMd dikhaa

yesterday night clouds in me.dat moon see(unacc).pst

'Yesterday night, I saw the moon behind the clouds'



\*\*The ARG0 analysis of dative subjects may change in future PB annotation

## Ditransitive

ditrans-1: राम मोहन को किताब देगा

raam mohan ko kitaab degaa

Ram Mohan dat book.f give.m.sg.fut

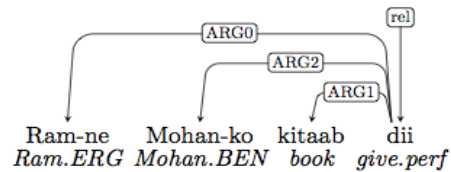
'Ram gave a book to Mohan'

ditrans-2: राम ने मोहन को किताब दी

raam ne mohan ko kitaab dii

Ram erg Mohan dat book.f give.f.sg.pst

'Ram gave a book to Mohan'



## Causatives

- Hindi has two ways of forming the causative:
  - Add -aa
    - (so → sulaa) sleep → make someone sleep
  - Add -vaa
    - (sulaa → sulvaa) make someone sleep → cause someone to fall asleep
- We introduce the label ARG<sub>A</sub> to analyze causers
- Subtypes of ARG<sub>0</sub> (ARG<sub>0</sub>-GOL, ARG<sub>0</sub>-MNS) for causees
- ARG<sub>A</sub>-MNS for intermediate causers

## Causatives

Unerg-1: आतिफ़ सोएगा

Atif soyegaa

Atif sleep.m.sg.fut

'Atif will sleep'

causative-1: आया ने आतिफ़ को सुलाया

aayaa ne Atif ko sulaayaa

maid erg Atif acc sleep.caus.pst

'The maid caused the child to sleep'

causative-2: माँ ने आया से आतिफ़ को सुलवाया

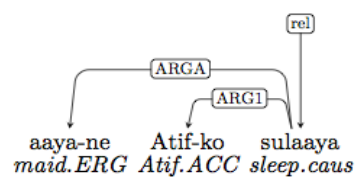
maaN ne aayaa se Atif ko sulvaayaa

mother erg maid by Atif acc sleep.caus.pst

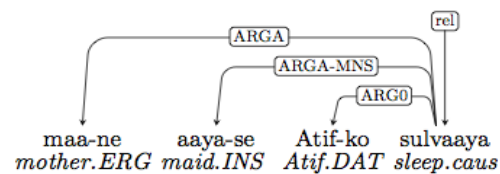
'The mother made the maid to cause the child to sleep.'

## Causatives

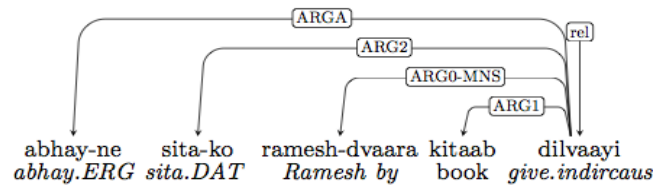
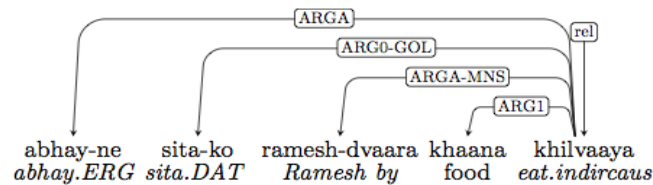
Causative-1



Causative-2



## Causatives: classes



## Complex predicates

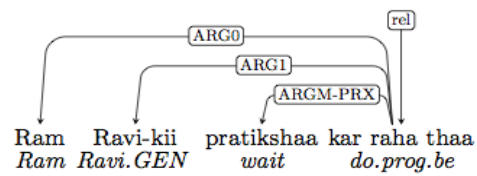
- These are cases such as *bharosaa karnaa* 'trust(n) do(v)'; trust
- Such cases are handled using a noun frame for *bharosaa*

Frame file for Barosa-n	
BarosA.01: trust-n	light verb 'do' to trust
Arg0	person who trusts
Arg1	thing trusted
BarosA.02: trust-n	light verb 'give' to ensure
Arg0	person who ensures
Arg2	recipient of the thing ensured
Arg1	thing that is ensured

[abhay ne ARG0] [sitaa par ARG1] bharosaa kiya

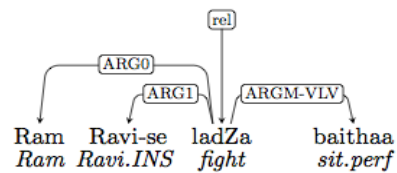
## Complex Predicate

compl-pred-1: राम रवि की प्रतीक्षा कर रहा था  
 raam ravi kii *pratikshaa kar rahaa thaa*  
 Ram Ravi gen wait do prog.m.sg be.m.sg.pst  
 'Ram was waiting for Ravi'



## Complex predicate

compl-pred-2: राम रवी से लड़ बैठा  
 raam ravi se *ladZa baithaa*  
 Ram Ravi inst fight sit.perf  
 'Ram regrettably fought with Ravi'





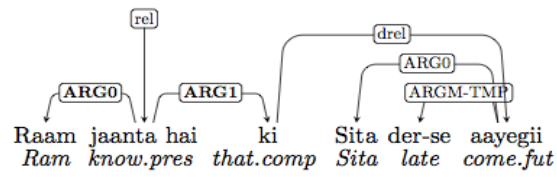
## Complement Clause

compl-cl-1: राम जानता है कि सीता देर से आएगी

raam jaantaa hai ki siita der se aayegii

Ram know.hab.m.sg be.sg.pres that Sita late part come.f.sg.fut

'Ram knows that Sita will arrive late'



# Phrase Structure Representation

Owen Rambow  
CCLS, Columbia University  
rambow@ccls.columbia.edu

## Phrase Structure (PS) Representation in the Hindi and Urdu Treebanks

- Devised by Rajesh Bhatt, University of Massachusetts, Amherst
  - Assisted by Annahita Farudi and Owen Rambow
- Developed in conjunction with DS and PB
- Inspired by Chomskyan tradition

## Background for PS

- Chomskyan program:
  - Motivated by claims about language acquisition in children
  - Develop a theory of syntax such that syntax of a language can be explained by
    - Language-universal principles
    - Language-specific parameters
- PS for Hindi inspired by Chomskyan program, but not following any specific Chomskyan approach

## Basic Principles of PS

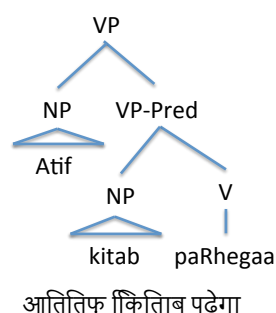
- PS represents relation between **lexical predicate-argument structure** (interface to lexicon) and **surface word order** (interface to phonology and semantics, roughly speaking)
- These two levels are related by derivations:
  - Words and constituents move and leave **traces**
    - Transformational grammar
- Monostratal representation
- Not unlike English Penn Treebank!

## Specific Assumptions about Representation Made by PS

- Phrase structure
- Notion of lexical heads with projections (X-bar theory, sort of) and associated functional projections
  - Nouns with postpositions
  - Verbs with auxiliaries and complementizers (*ki*)
- Binary branching
  - Theoretical reasons
  - To be different from DS

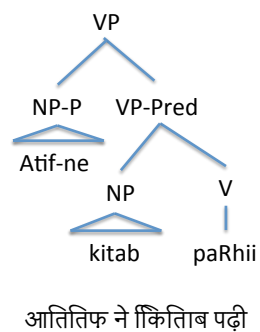
## Basic Transitive Clause (1)

- There are two privileged positions in the verbal projection, corresponding usually to DS's k1 and k2



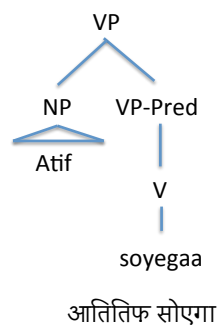
## Basic Transitive Clause (2)

- The representation is maintained when we have an ergative construction



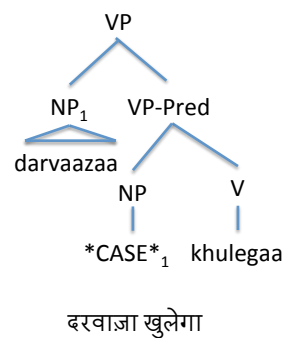
## Intransitive Clause: Unergative

- PS makes a distinction between unergative and unaccusative
- In unergative, there simply is no object



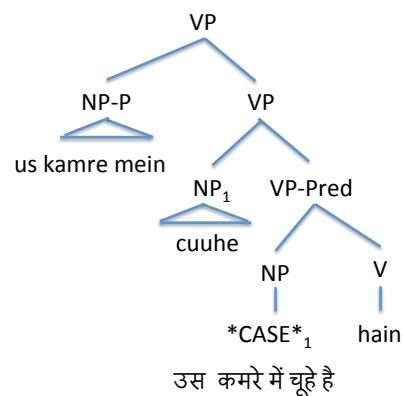
## Intransitive Clause: Unaccusative

- Argument starts in lower position (because of lexical semantics), and moves to higher position (because higher position has no occupant)



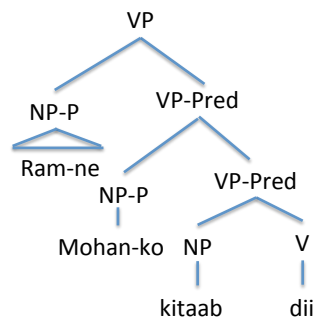
## Existentials

- Existential *ho* 'be' is unaccusative (because agent-free), and location is an adjunct



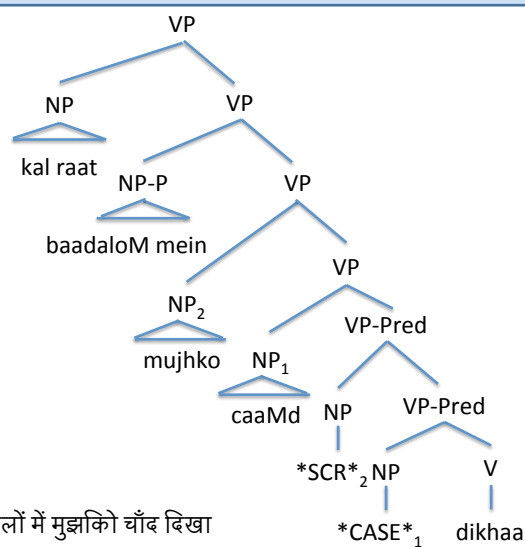
## Ditransitive

- The recipient is introduced as adjoined to the VP-Pred: a fixed, but not structural position



राम ने मोहन को किताब दी

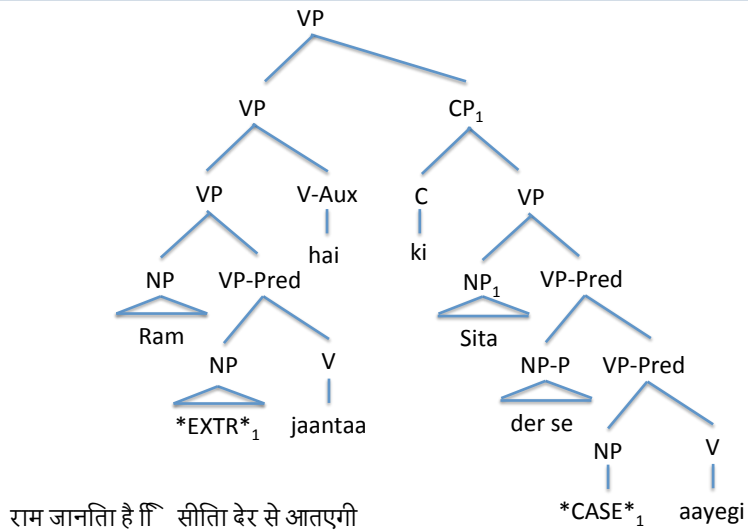
## Putting it All Together: Dative Subjects



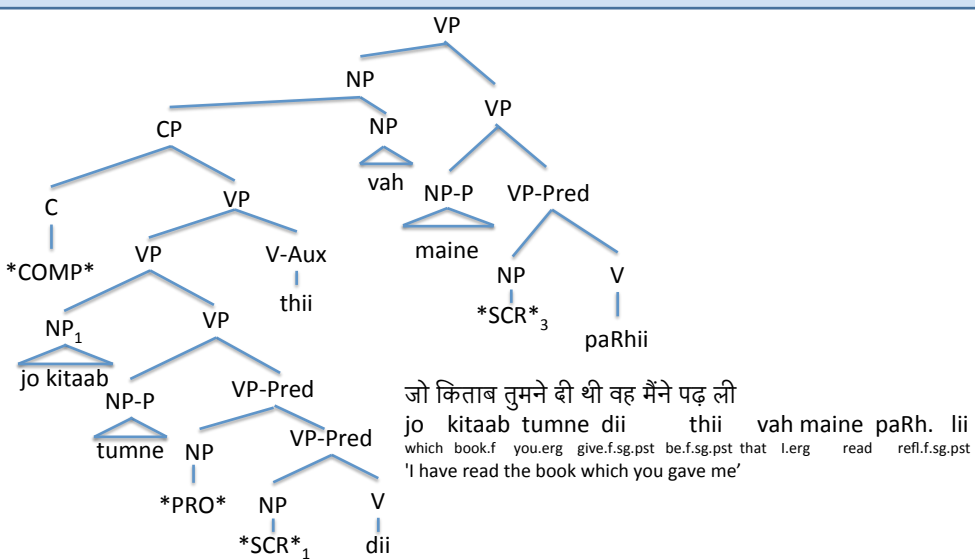
कल राति बादलों में मुझको चाँद दिखा

- Dikhaa* is interpreted semantically as a ditransitive: someone makes something appear to someone
- Since the agent is absent, the lower argument raises to the higher position (like unaccusative)
- The dative beneficiary is base generated in the fixed dative position (adjoined to VP-Pred) and then scrambles elsewhere

## Complement Clauses with *ki*

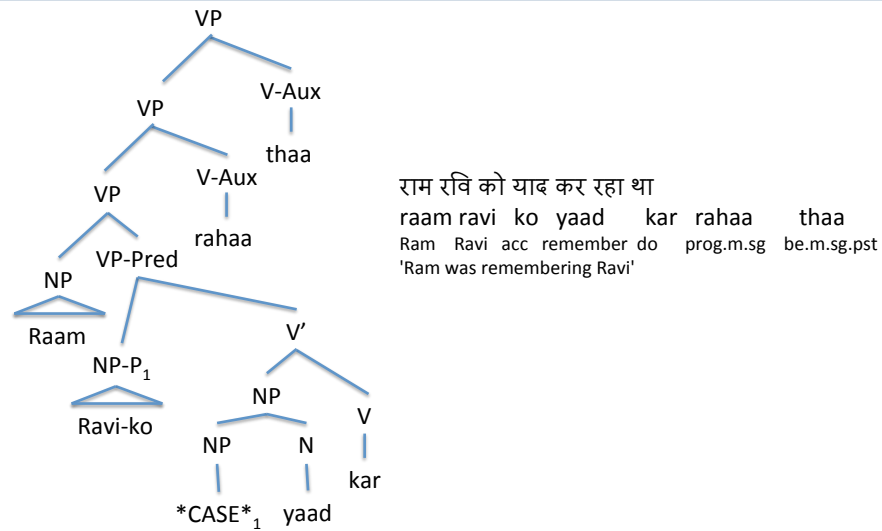


## Relative Clause





## Complex Predicate



## Causative

