

Handling non-projectivity in a dependency-to-phrase-structure conversion algorithm for Hindi

Himanshu Yadav¹ Ashwini Vaidya² Samar Husain²

¹Jawaharlal Nehru University, New Delhi, India

²Indian Institute of Technology Delhi, India

39th International Conference of Linguistic Society of India
(ICOLSI-39) 08-11, December, 2017

Discontinuous phrases

- (1) raam=kaa do din pahle
Raam.M.Sg=GEN two day.M.PL before
chashmaa kho gayaa
spectacles.M.Pl lost go.perf
'Ram's spectacles were lost two days ago'

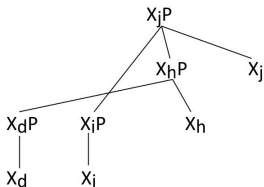
Discontinuous phrases

- (1) raam=kaa do din pahle
Raam.M.Sg=GEN two day.M.PL before
chashmaa kho gayaa
spectacles.M.Pl lost go.perf
'Ram's spectacles were lost two days ago'

- Treebank representation of discontinuous phrases
 - ▶ Constituent structure (phrase structure trees, e.g. Penn Treebank for English)
 - ▶ Dependency structure (modifier-modified relations e.g. Prague Dependency Treebank for Czech)

Constituent Treebanks

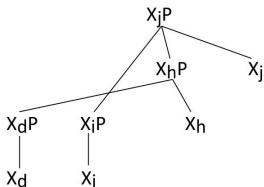
- Discontinuous phrases result in crossing branches,



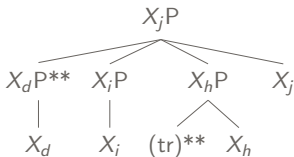
**Usually disallowed

Constituent Treebanks

- Discontinuous phrases result in crossing branches,
- Usually transformed by using null elements and introducing new labels in the grammar



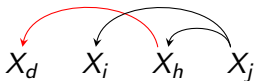
**Usually disallowed



Representation in phrase structure using null node

Dependency Treebanks

- Dependency trees can contain crossing arcs: **non-projective** arcs



Non-projective dependency

$X_h \rightarrow X_d$ is non-projective iff there is at least one node X_i between X_h and X_d that X_h does not dominate

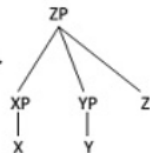
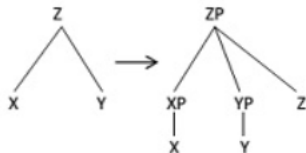
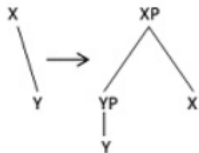
Dependency-to-Phrase structure conversion

- Dependency treebanks (DS) → constituency trees
- Rule-based conversion to produce well-formed PS trees

Dependency-to-Phrase structure conversion

- Dependency treebanks (DS) → constituency trees
- Rule-based conversion to produce well-formed PS trees

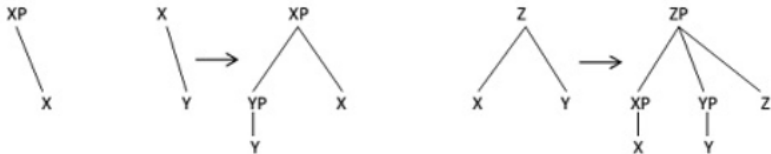
PS representation



Dependency-to-Phrase structure conversion

- Dependency treebanks (DS) → constituency trees
- Rule-based conversion to produce well-formed PS trees

PS representation



- Major challenge: **18.75%** non-projective trees

DS-PS Conversion

Yadav et. al (2017)

- Head computation
 - Argument-adjunct distinction
 - Sub-tree formation
 - Clause formation
-
- No consistent way to convert non-projective DS trees
 - Placement of the null element in the tree

DS-PS Conversion

Example

raam=kaa kal chashmaa kho gayaa

✗ kal Raam=ka chashmaa kho gayaa → correct head projection,
wrong linear order

DS-PS Conversion

Example


raam=kaa kal chashmaa kho gayaa

✗ kal Raam=ka chashmaa kho gayaa → correct head projection, wrong linear order

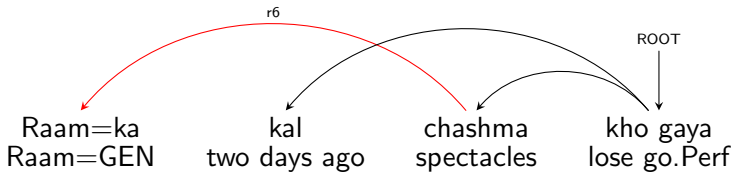
∴ kal Raam=ka chashmaa kho gayaa.



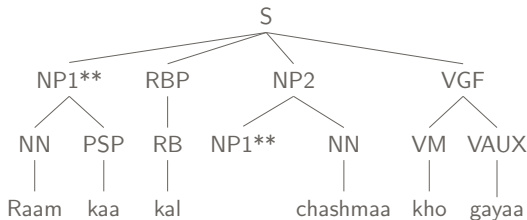
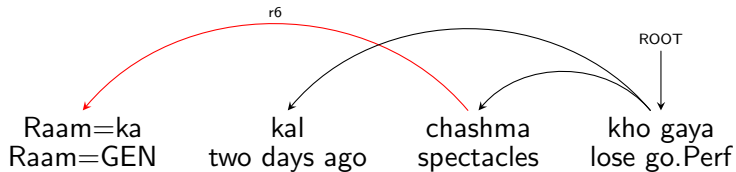
kal Raam=ka ∴ chashmaa kho gayaa.



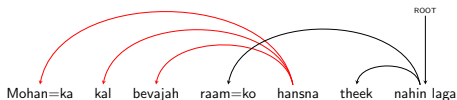
Single intervening constituent



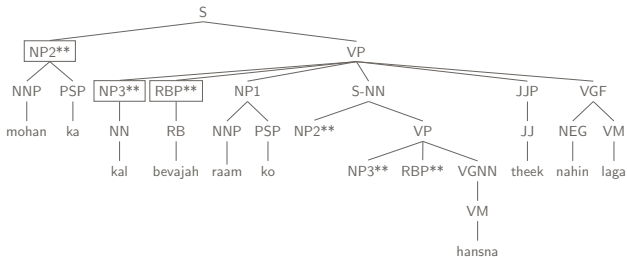
Single intervening constituent



Single intervening constituents comprise 93.6% of non-projective structures

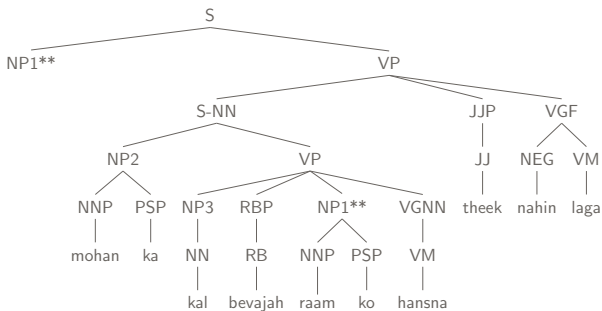


- If all non-projective dependents placed before intervenor → 3 null elements, wrong attachments
- Devise a rule to carry this out consistently



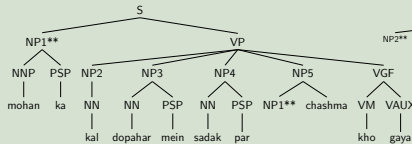
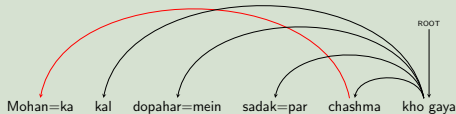
Erroneous attachments

>1 non projective arc

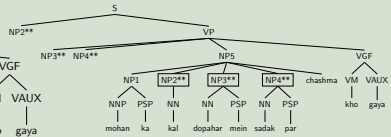


- Non-projective dependents moved after the intervenor.
- One null element, valid heads, right linear order

>1 intervening element



✓(a)



✗(b)

- In (a), one null node, dependent of non-projective arc is moved, correct attachments
- In (b), each intervening phrase (i.e. X_i) is moved after the dependent of the non-projective arcs. Wrong attachments, more nulls.

Projectivizing rule

- We follow a minimality principle, where a tree with least number of empty categories is chosen
- Also reduces the possibility of wrong attachments.
- Terminal nodes in the converted to PS trees matched with source tree with 98.05% accuracy
- Improves coverage in comparison to our previous conversion rules

Further remarks

Using stricter measures (acyclicity, no duplication, argument representation, clausal correspondence) we have an overall accuracy of 95.8% for non-projective structures

Further remarks

Using stricter measures (acyclicity, no duplication, argument representation, clausal correspondence) we have an overall accuracy of 95.8% for non-projective structures

- **Discontinuous PS**

- ▶ Non-projectivity in DS for Hindi, other languages [Bhat and Sharma, 2012] (RC extraposition, Genitives)
- ▶ Discontinuous structures in PS ?
- ▶ Number of constituent boundaries crossed could be informative.
- ▶ Very few trees cross more than 4 constituents

References



Bhat, R. A. and Sharma, D. M. (2012).

Non-projective structures in indian language treebanks.

In *Proceedings of the 11th Workshop on Treebanks and Linguistic Theories (TLT11)*, pages 25–30.