Insights into non-projectivity in Hindi

Prashanth Mannem, Himani Chaudhry and Akshar Bharati

LTRC, IIIT
Hyderabad, India 50032

{prashanth,himani}@research.iiit.ac.in

Aug 4, 2009
Outline

1. Introduction
   HyDT

2. Non-projectivity
   Non-projectivity in HyDT
   Non-projectivity Analysis

3. Graph properties
   HyDT’s graph properties

4. Linguistic Analysis
   Classes

5. Summary

P. Mannem, H. Chaudhry and A. Bharati
IIIT, Hyderabad

Insights into non-projectivity in Hindi
Introduction

- Hindi is a verb final, flexible word order language
  - raama baazaara gayaa thaa
    - Ram market go.PAST be.PAST
  - baazaara gayaa thaa raama
  - raama gayaa thaa baazaara
  - baazaara raama gayaa thaa

- Hyderabad Dependency Treebank (HyDT) for Hindi
Introduction

- Hindi is a verb final, flexible word order language
  - raama baazaara gayaa thaa
  - Ram market go.PAST be.PAST
  - baazaara gayaa thaa raama
  - raama gayaa thaa baazaara
  - baazaara raama gayaa thaa

- Hyderabad Dependency Treebank (HyDT) for Hindi
HyDT - Hyderabad Dependency Treebank

- Paninian Grammar
  - Syntactic cues help in determining the type of relation
- Sentences annotated with
  - POS tags
  - Minimal constituents (chunks) and their heads
  - Relations between chunks (inter-chunk)
  - Intra-chunk dependencies left unspecified
  - *Trees can be expanded if needed*
Example

- meraa baDaa bhaaii bahuta phala khaataa hai
  
  "my big brother lots-of fruits eat PRES."

- (( meraa baDaa bhaaii ))_{NP} (( bahuta phala ))_{NP} (( khaataa hai ))_{VG}

- (( meraa_{PRP} baDaa_{JJ} bhaaii_{NN} ))_{NP} (( bahuta_{QF} phala_{NN} ))_{NP} (( khaataa_{VM} hai_{VAUX} ))_{VG}
Paninian Grammatical Model

- A dependency grammar based approach
- Inspired by inflectionally rich language (Sanskrit)
- Better suited for handling Indian Languages
- Provides syntactico-semantic analysis of language
- Various linguistic phenomena handled seamlessly
- The grammar facilitates analysis of the intended meaning as an ‘expression’ of what the speaker wants to communicate (vivaksha) (Bharati et al., 1995)
Dependency relations

- **karaka relations**: Direct participants (karaka) of the action denoted by the verb
  - 6 basic karakas: karta (subject/agent/doer), karma (object/patient), karana (instrument), sampradaan (beneficiary), apaadaan (source), adhikarana (location in place/time/other)

- Other than karaka relations: purpose, genitive, reason etc...

- Relations which are not strictly 'dependency relation' but are used to represent 'co-ordination' and 'complex predicates'

- 40 labels in all
Outline

1. Introduction
   HyDT

2. Non-projectivity
   Non-projectivity in HyDT
   Non-projectivity Analysis

3. Graph properties
   HyDT’s graph properties

4. Linguistic Analysis
   Classes

5. Summary
Non-projectivity

Every word in the span of relation has to be dominated by the head in that relation for it to be projective.

Otherwise, the relation is non-projective.

In a flat representation, crossing arcs indicate non-projectivity.
HyDT and non-projectivity

- 1865 sentences, 16620 chunks, 35787 words
- 14% sentences have non-projective structures
- 1.87% of inter-chunk relations are non-projective
- 0.87% if intra-chunk relations are also considered
- In PDT 2.0 (Czech), 23% (out of 73088) of the sentences are non-projective
HyDT and non-projectivity

- 1865 sentences, 16620 chunks, 35787 words
- 14% sentences have non-projective structures
- 1.87% of inter-chunk relations are non-projective
- 0.87% if intra-chunk relations are also considered
- In PDT 2.0 (Czech), 23% (out of 73088) of the sentences are non-projective
Why is non-projectivity important as a constraint

- Poses problems in parsing with respect to both accuracy and efficiency
- Need special algorithms to handle non-projectivity
- Bharati et al. (2008) showed that a major chunk of errors in their Hindi parser is due to non-projectivity
- A need to analyse non-projectivity in Hindi for a better insight into such constructions
Why is non-projectivity important as a constraint

- Poses problems in parsing with respect to both accuracy and efficiency
- Need special algorithms to handle non-projectivity
- Bharati et al. (2008) showed that a major chunk of errors in their Hindi parser is due to non-projectivity

- A need to analyse non-projectivity in Hindi for a better insight into such constructions
Why is non-projectivity important as a constraint

- Poses problems in parsing with respect to both accuracy and efficiency
- Need special algorithms to handle non-projectivity
- Bharati et al. (2008) showed that a major chunk of errors in their Hindi parser is due to non-projectivity
- A need to analyse non-projectivity in Hindi for a better insight into such constructions
Non-projectivity analysis

- From two perspectives
  - Graph properties constraining non-projectivity (Kuhlmann and Nivre, 2006; Nivre, 2006)
    - Like gap degree, edge degree, planarity, well-nestedness
    - These constraints give an idea of the extent of non-projectivity
  - Linguistic phenomenon giving rise to non-projectivity
    - Provides better understanding and gives insight into what kind of constructions lead to non-projectivity
    - Can be used as features for better learning
Non-projectivity analysis

- From two perspectives
- Graph properties constraining non-projectivity (Kuhlmann and Nivre, 2006; Nivre, 2006)
  - Like gap degree, edge degree, planarity, well-nestedness
  - These constraints give an idea of the extent of non-projectivity
- Linguistic phenomenon giving rise to non-projectivity
  - Provides better understanding and gives insight into what kind of constructions lead to non-projectivity
  - Can be used as features for better learning
Non-projectivity analysis

- From two perspectives
- Graph properties constraining non-projectivity (Kühlmann and Nivre, 2006; Nivre, 2006)
  - Like gap degree, edge degree, planarity, well-nestedness
  - These constraints give an idea of the extent of non-projectivity
- Linguistic phenomenon giving rise to non-projectivity
  - Provides better understanding and gives insight into what kind of constructions lead to non-projectivity
  - Can be used as features for better learning
Outline

1. Introduction
   HyDT

2. Non-projectivity
   Non-projectivity in HyDT
   Non-projectivity Analysis

3. Graph properties
   HyDT’s graph properties

4. Linguistic Analysis
   Classes

5. Summary
Graph constraints restricting non-projectivity

- Gap degree
- Edge degree
- Planarity
- Well-nestedness
Gap degree

- Gap is a pair of adjacent nodes in the projection of a node but not adjacent in the sentence.
- Gap degree of a node is the number of gaps in the projection of a node.
- Gap degree of a sentence is the maximum among gap degrees of nodes in the sentence.
Edge degree

- The number of connected components in the span of an edge which are not dominated by the outgoing node in the edge.
- Edge degree of a sentence is the maximum among edge degrees of edges in the sentence.
Planarity and Well-nestedness

A dependency graph is **planar** if edges do not cross when drawn above the sentence.

A dependency graph is **well-nested** if no two disjoint subgraphs interleave.

Two subgraphs are **disjoint** if neither of their roots dominates the other.

They **interleave** if their projections overlap.
## HyDT w.r.t the graph properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>All structures</td>
<td>1865</td>
<td></td>
</tr>
<tr>
<td>Gap degree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gd(0)</td>
<td>1603</td>
<td>85.9%</td>
</tr>
<tr>
<td>Gd(1)</td>
<td>259</td>
<td>13.89%</td>
</tr>
<tr>
<td>Gd(2)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Gd(3)</td>
<td>3</td>
<td>0.0016%</td>
</tr>
<tr>
<td>Edge degree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ed(0)</td>
<td>1603</td>
<td>85.9%</td>
</tr>
<tr>
<td>Ed(1)</td>
<td>254</td>
<td>13.6%</td>
</tr>
<tr>
<td>Ed(2)</td>
<td>6</td>
<td>0.0032%</td>
</tr>
<tr>
<td>Ed(3)</td>
<td>1</td>
<td>0.0005%</td>
</tr>
<tr>
<td>Ed(4)</td>
<td>1</td>
<td>0.0005%</td>
</tr>
<tr>
<td>Projective</td>
<td>1603</td>
<td>85.9%</td>
</tr>
<tr>
<td>Planar</td>
<td>1639</td>
<td>87.9%</td>
</tr>
<tr>
<td>Non-projective</td>
<td>36</td>
<td>1.93%</td>
</tr>
<tr>
<td>&amp; planar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well-nested</td>
<td>1865</td>
<td>100%</td>
</tr>
</tbody>
</table>

*Table: Results on HyDT*
Outline

1 Introduction
   HyDT

2 Non-projectivity
   Non-projectivity in HyDT
   Non-projectivity Analysis

3 Graph properties
   HyDT’s graph properties

4 Linguistic Analysis
   Classes

5 Summary
Linguistic phenomena leading to non-projectivity

- Classes
  - Relative co-relative constructions
  - Extraposed relative clause constructions
  - Intra-clausal non-projectivity
  - Paired connectives
  - \(ki\) complement clauses
  - Genitive relation split by a verb modifier
  - Phrase split a co-ordinating structure
  - Shared argument splitting the non-finite clause
  - Others
Analysis of the non-projective classes

- Cues to identify non-projectivity
- 
  - *Rigidity*
  - Reorderability of the constructions retaining the gross meaning
  - *Gross meaning* — Meaning of the sentence not taking the discourse and topic-focus into consideration
- What is the *best projective approximation* possible by reordering?
- Is this projective construction more *natural* compared to the non-projective one?
Relative co-relative constructions

- Cues: relative co-relatives like jaba-tabā (when-then), jo-vo (which-that), jahām-vahām (where-there), jisa-usa (which-that)
- Not rigid
- Can be made projective by reordering
- Hard to say which among the projective & non-projective ones is more natural
Extraposed relative clause constructions

- NP and the relative clause are separated by the verb group
- Cues: Relative pronoun following a verb group
- Not rigid
- Extraposed relative clause can be moved next to the noun phrase to make it projective
- Resulting projective construction is less natural than the original non-projective one
- Most common non-projective class

This letter was published in the press at Mumbai where I worked.
Paired connectives

- Cues: Paired connectives like agar-to (if-then), yadi-to (if-then)
- Can be reordered and is not rigid
- The phrase that comes after to followed by yadi clause and then to
  - to is optional here
- Resulting projective construction is not a natural one

If [you] needed money then [you] should have told me
**ki complement clause constructions**

He had such [a] liking for football that he was not able to give it up.

- **Cues:** *ki* comes after words like *yaha* (*this*), *aisaa* (*such*), *is tarah* (*such*), *itana* (*this much*)
  - Takes the pattern *yaha-its property-VP-ki clause*
- **Rigid**
- **If VP has a transitive verb, then the *ki* clause and the referent both modify the verb, making it projective**
Genitive relation split by a verb modifier

He had huge liking for football

- No obvious cues
- Is not rigid
- Move the verb modifier out of the genitive phrase to make it projective
- Projective one is more natural

P. Mannem, H. Chaudhry and A. Bharati

Insights into non-projectivity in Hindi
After this Jaman Shah [got it] and then, in 1795 Shah Shuja got it

- Cues: NONE
- Adverb occurring in the middle of a co-ordinating structure
- Is not rigid
- Projective one is more natural
Shared argument splitting the non-finite clause

We used to talk after placing that writer’s identity proudly before the publisher.

- Cues: NONE
- Is not rigid
- Projective one is more natural
Others

- Few very rare and not natural phenomena
- Annotation errors
- Inconsistent NULL placement
### Non-projective classes in HyDT

<table>
<thead>
<tr>
<th>Non-projective Class</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative co-relatives constructions</td>
<td>18</td>
<td>6.8</td>
</tr>
<tr>
<td>Extraposed relative clause constructions</td>
<td>101</td>
<td>38.0</td>
</tr>
<tr>
<td>Intra-clausal non-projectivity</td>
<td>12</td>
<td>4.5</td>
</tr>
<tr>
<td>Paired connectives</td>
<td>33</td>
<td>12.4</td>
</tr>
<tr>
<td>ki complement clauses</td>
<td>52</td>
<td>19.5</td>
</tr>
<tr>
<td>Genitive relation split by a verb modifier</td>
<td>10</td>
<td>3.8</td>
</tr>
<tr>
<td>Phrase splitting a co-ordinating structure</td>
<td>4</td>
<td>1.5</td>
</tr>
<tr>
<td>Shared argument splitting the non-finite clause</td>
<td>10</td>
<td>3.8</td>
</tr>
<tr>
<td>Others</td>
<td>26</td>
<td>9.8</td>
</tr>
</tbody>
</table>

**Table:** Non-projectivity class distribution in HyDT
Outline

1 Introduction
   HyDT

2 Non-projectivity
   Non-projectivity in HyDT
   Non-projectivity Analysis

3 Graph properties
   HyDT’s graph properties

4 Linguistic Analysis
   Classes

5 Summary
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree ≤ 1 ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree $\leq 1$ ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree $\leq 1$ ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them

P. Mannem, H. Chaudhry and A. Bharati
IIIT, Hyderabad

Insights into non-projectivity in Hindi
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree $\leq 1$ ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree $\leq 1$ ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them
Summary

- Analysed non-projectivity in HyDT from two perspectives
- Gap degree and edge degree $\leq 1$ ensures 99.99% coverage
- Non-projective structures classified into 8 categories
- Around 75% of the non-projective cases can be identified using strong lexical cues
  - Parsers can make use of this information and determine non-projective arcs directly
- The rest are hard to recognize and need extra information (world knowledge!) to identify non-projectivity in them
Thank You
References

- Akshar Bharati, Samar Husain, Bharat Ambati, Sambhav Jain, Dipti Sharma, and Rajeev Sangal. 2008a. Two semantic features make all the difference in parsing accuracy. In Proceedings of the 6th International Conference on Natural Language Processing (ICON-08), Pune, India.
References


References
