
Two stage constraint based hybrid approach to free word order language dependency parsing

Samar Husain
LTRC, IIIT-Hyderabad,
India.

Outline

- Introduction
 - Grammatical framework
 - Two stage parsing
 - Two stage hybrid parsing
 - Evaluation
-

Introduction

- Broad coverage parser for Hindi
 - Very crucial
 - MT systems, IE, co-reference resolution, etc.
 - Attempt to make a hybrid parser

 - Grammatical framework: Dependency
-

Introduction

- Levels of analysis before parsing
 - Morphological analysis (Morph Info.)
 - Analysis in local context (POS tagging, Chunking, case markers/postpositions computation)
 - We parse after the above processing is done.
-

Computational Paninian Grammar (CPG)

- Based on Panini's Grammar
 - Inspired by inflectionally rich language (Sanskrit)
 - A dependency based analysis (Bharati et al. 1995a)
 - Earlier parsing approaches for Hindi (Bharati et al, 1993; 1995b; 2002)
-

CPG (The Basic Framework)

- Treats a sentence as a set of modifier-modified relations
 - Sentence has a primary modified or the root (which is generally a verb)
 - Gives us the framework to identify these relations
 - Relations between noun constituent and verb called '*karaka*'
 - *karakas* are syntactico-semantic in nature
 - Syntactic cues help us in identifying the *karakas*
-

karta – *karma* karaka

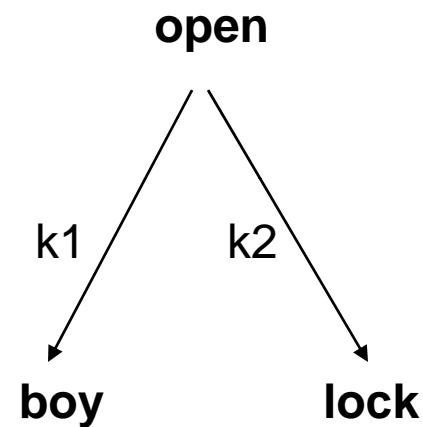
‘The boy opened the lock’

- k1 – *karta*
- k2 – *karma*

- *karta*, *karma* usually correspond to agent, theme respectively
 - But not always (Vaidya et. al, 2009)

- *karakas* are direct participants in the activity denoted by the verb

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



Two stage parsing

- Basic idea

- There are two layers (stages)
 - The 1st stage handles intra-clausal relations, and the 2nd stage handles inter-clausal relations,
 - The output of each stage is a linguistically sound partial parse that becomes the input to the next layer
-

Stage 1

- Identify intra-clausal relations
 - the argument structure of the verb,
 - noun-noun genitive relation,
 - infinitive-verb relation,
 - infinitive-noun relation,
 - adjective-noun,
 - adverb-verb relations, etc.
-

Stage 2

- Identify inter-clausal relations
 - subordinating conjuncts,
 - coordinating conjuncts,
 - relative clauses, etc.
-

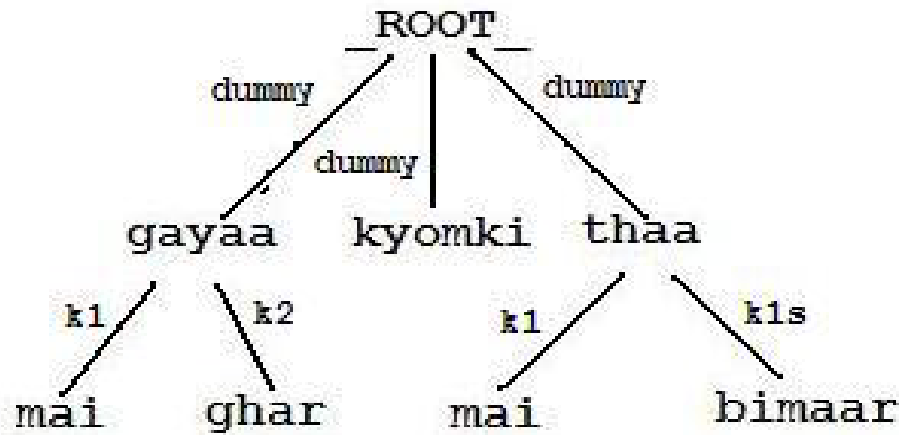
How do we do this?

- Introduce a dummy `__ROOT__` node as the root of the dependency tree
 - Helps in giving linguistically sound partial parses
 - Keeps the tree connected
 - Classify the dependency tags into two sets
 1. Tags that function within a clause,
 2. Tags that relate two clauses
 - Clausal boundaries
-

An example

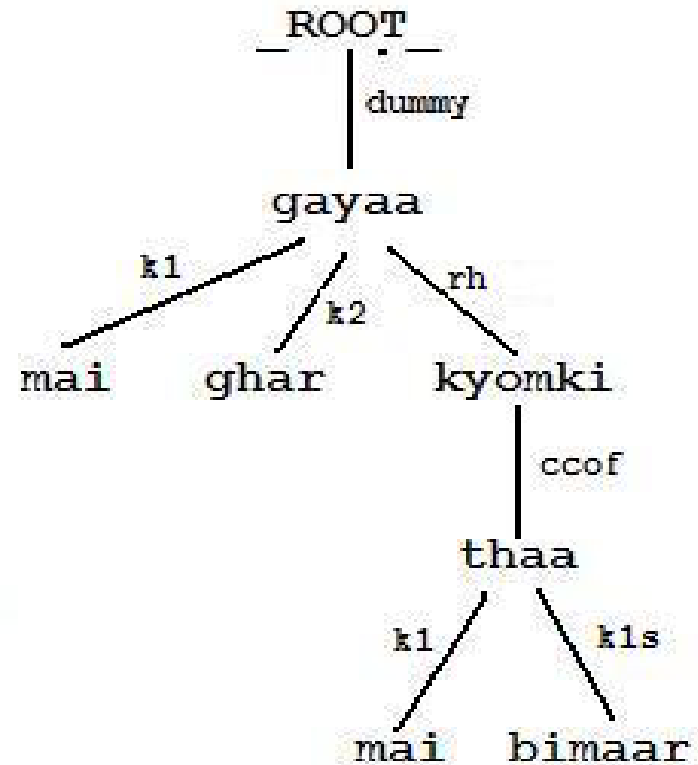
mai ghar gayaa kyomki mai bimaar thaa
'I' 'home' 'went' 'because' 'I' 'sick' 'was'
'I went home because I was sick'

The parses



(a) Intra-Clausal

(a): 1st stage output,



(b) Inter-Clausal

(b): 2nd stage final parse

2 stage parsing

- 1st stage
 - All the clauses analyzed
 - Analyzed clauses become children of __ROOT__
 - Conjuncts become children of __ROOT__
 - 2nd stage
 - Usually, does not modify the 1st stage analysis
 - Identifies relations between 1st stage parsed sub-trees
 - Selective resolution of demands
 - Repair
 - Partial Parses
-

Constraint based hybrid parsing

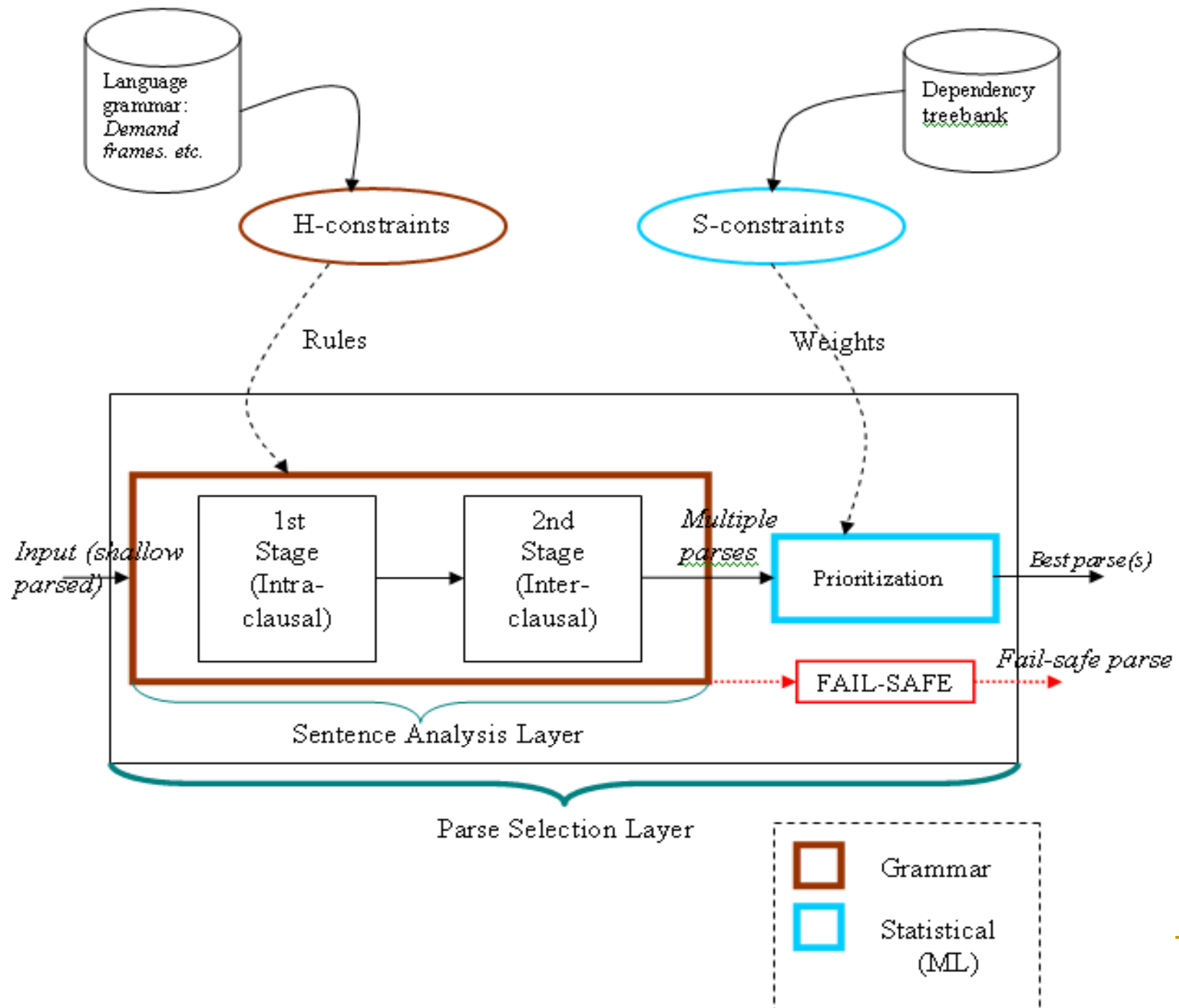
- **Hard constraints**

- Cannot be broken without a sentence being called ungrammatical
- Rule based
- Constraint satisfaction using integer programming.

- **Soft constraints**

- Used as preferences
- ML using dependency treebank
- Learn parameters weight that maximize the overall accuracy.

- Both H & S constraints reflect the linguistic realities of language and together can be thought as the grammar of a language.
-



Overall performance

	UA	LA	L
CBP	86.1	63	65
CBP'	87.69	69.67	72.39
CBP''	90.1	75	76.9
MST	87.8	70.4	72.3
Malt	86.6	68.0	70.6

UA: unlabeled attachments accuracy,

L : labeled accuracy

LA: labeled attachment accuracy

Error analysis

- Reasons for low LA
 - Less verb frames
 - Some phenomena not covered
 - Prioritization errors
 - Ambiguities
-

Advantages

- Linguistic generalizations can be easily converted as constraints,
 - Use of H-constraints and S-constraints reflects the grammar of a language,
 - IP formulation allows for handling non-projective parsing (Riedel and Clarke, 2006)
 - Complex linguistic cues can easily be encoded as part of various constraints,
 - Two-stage parsing lends itself seamlessly to parsing complex sentences by modularizing the task of overall parsing,
 - The problem of label bias faced by the data driven Hindi parsers (Bharati et al., 2008a) for some cases does not arise here as contextually similar entities are disambiguated by tapping in hard to learn features,
-

-
- Use of clauses as basic parsing units reduces the search space at both the stages,
 - Parsing closely related languages will become easy.
-

References

- R. Begum, S. Husain, A. Dhvaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. *In Proceedings of IJCNLP-2008*.
 - A. Bharati and R. Sangal. 1993. Parsing Free Word Order Languages in the Paninian Framework. *Proc. of ACL:93*.
 - A. Bharati, V. Chaitanya and R. Sangal. 1995a. *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, New Delhi.
 - A. Bharati, A. Gupta and Rajeev Sangal. 1995b. Parsing with Nested Constraints. *In Proceedings of 3rd NLP Pacific Rim Symposium. Seoul*.
 - A. Bharati, R. Sangal and T. P. Reddy. 2002. A Constraint Based Parser Using Integer Programming *In Proc. of ICON-2002*.
 - A. Bharati, S. Husain, D. Sharma, and R. Sangal. 2008a. A two-stage constraint based dependency parser for free word order languages. In Proceedings of the COLIPS IALP, Chiang Mai, Thailand.
 - A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal. 2008b. Two semantic features make all the difference in parsing accuracy. *In Proceedings of the 6th ICON, Pune, India*.
 - A. Bharati, S. Husain, S. P. K. Gadde, B. Ambati, and R. Sangal. 2009. A modular cascaded partial parsing approach to complete parsing. *In Proceedings of the COLIPS International Conference on Asian Language Processing 2009 (IALP). Singapore. 2009*.
-

-
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. *In Proc. of HLT/EMNLP*, pp. 523–530.
 - J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2), 95-135.
 - S. Riedel and J. Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. *In Proc. EMNLP*.
 - A. Vaidya, S. Husain, P. Mannem, D. M. Sharma. 2009. A karaka-based dependency annotation scheme for English. *In Proceedings of the CICLing-2009, Mexico City, Mexico*.
-



Thanks!!



Non-projective sentences

- Relative co-relative constructions
 - Extraposed relative clause constructions
 - Paired connectives
 - Genitive relation split by a verb modifier
 - Shared argument splitting the non-finite clause

 - Issues:
 - A phrase splitting a coordinating structure
 - semantics of the conjoined elements
-