CS626: Speech, NLP and the Web

Dependency Parsing

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week of 11th October, 2021

Example: raw sentence

The strongest rain shut down the financial hub of Mumbai

(from: Stanford parser https://nlp.stanford.edu/software/lex-parser.shtml)

Example: POS Tagged sentence

The/DT strongest/JJS rain/NN shut/VBD down/RP the/DT financial/JJ hub/NN of/IN Mumbai/NNP

Constituency parse

```
(S
  (NP
      (DT The)
       (JJS strongest)
         (NN rain))
   )

(VP
    …
```

```
(VP
  (VP
     (VBD shut)
     (PRT (RP down))
  (NP
     (NP
       (DT the) (JJ financial)
       (NN hub))
     (PP (IN of)
          (NP (NNP Mumbai)))))
```

Dependency Parse

root(ROOT-0, shut-4)

nsubj(shut-4, rain-3)

prt(shut-4, down-5)

det(rain-3, the-1)

amod(rain-3, strongest-2)

dobj(shut-4, hub-8)

det(hub-8, the-6)

amod(hub-8, financial-7)

prep(hub-8, of-9)

pobj(of-9, Mumbai-10)

Note: dependency parsing chooses to remain shallow; prepositions are NOT Disambiguated wrt their semantic roles.

Examples to illustrate difference between DP and Semantic Role Labeling (SRL)

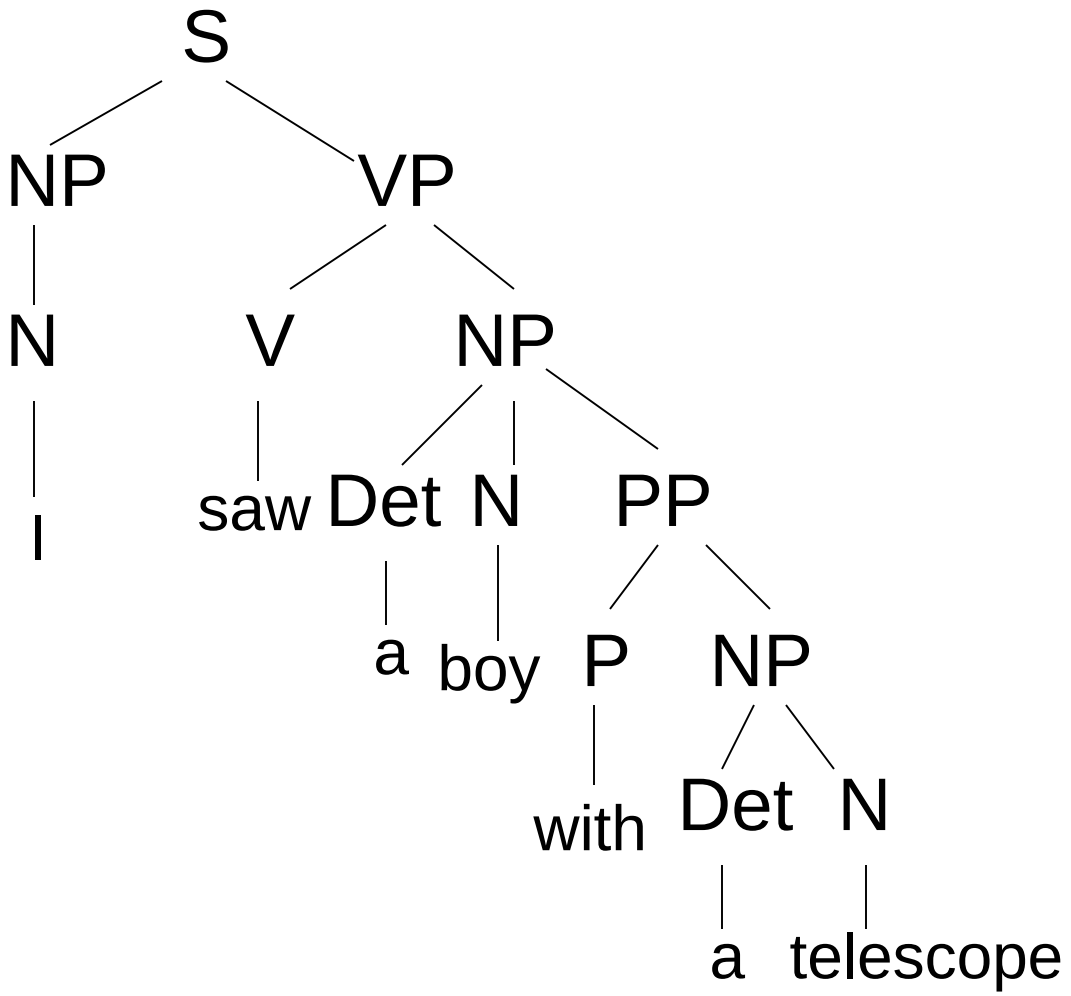
Sentence	Shallow relation from Dependency Parsing	Deeper relation from Semantic Role Labeling
John broke the window	nsubj	Agent
The stone broke the window	nsubj	Instrument
The window broke	nsubj	Object
1947 saw the freedom of India	nsubj	Time
Delhi saw bloodshed when Nadir Shah attacked Delhi	nsubj	Place

Disambiguation is needed to convert shallow DP relations to semantic roles.

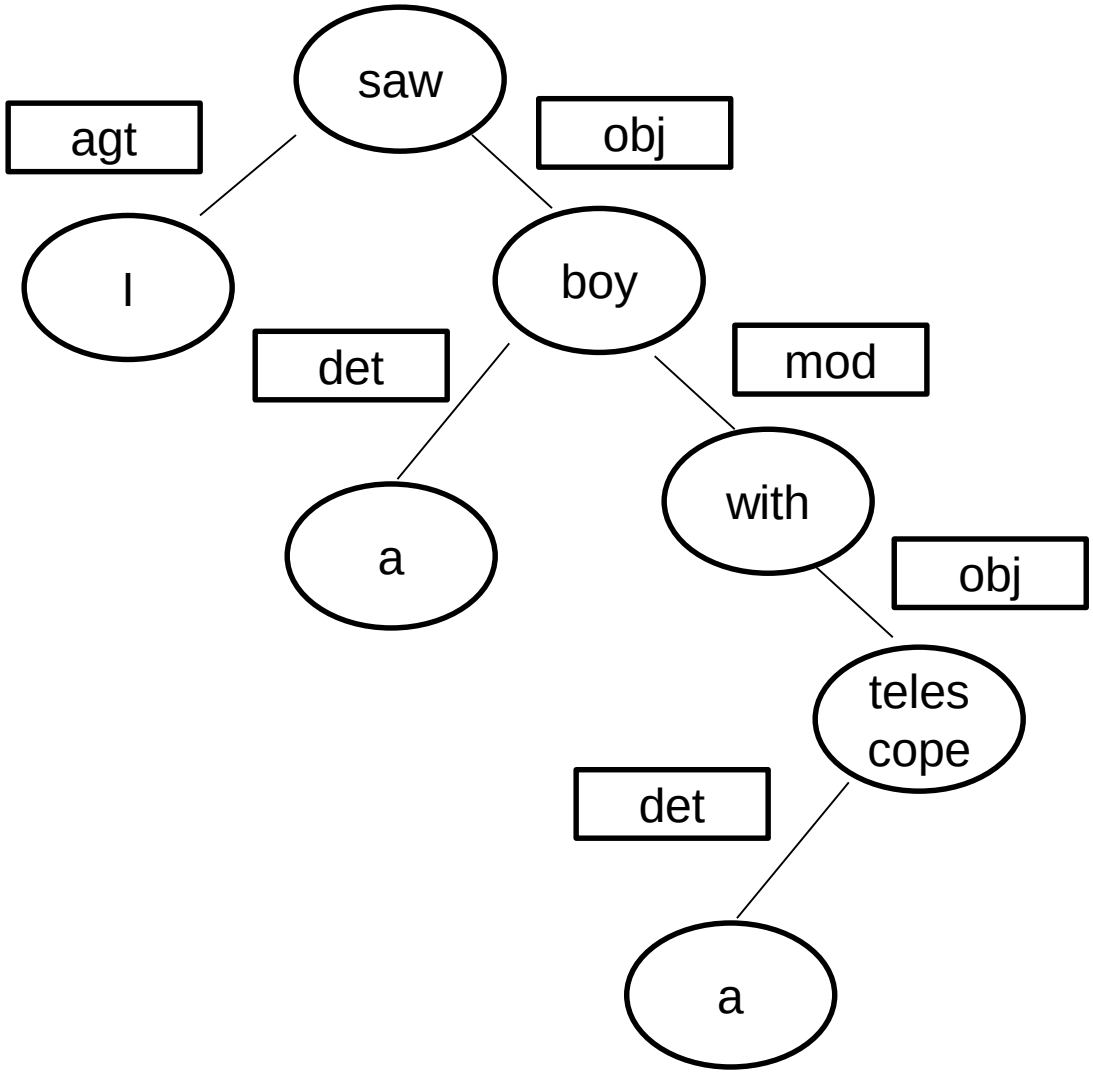
Two kinds of parse representations: Constituency Vs. Dependency

- Penn Constituency Treebank
  - http://www.cis.upenn.edu/~treebank/
- Prague Dependency Treebank
  - http://ufal.mff.cuni.cz/pdt2.0/

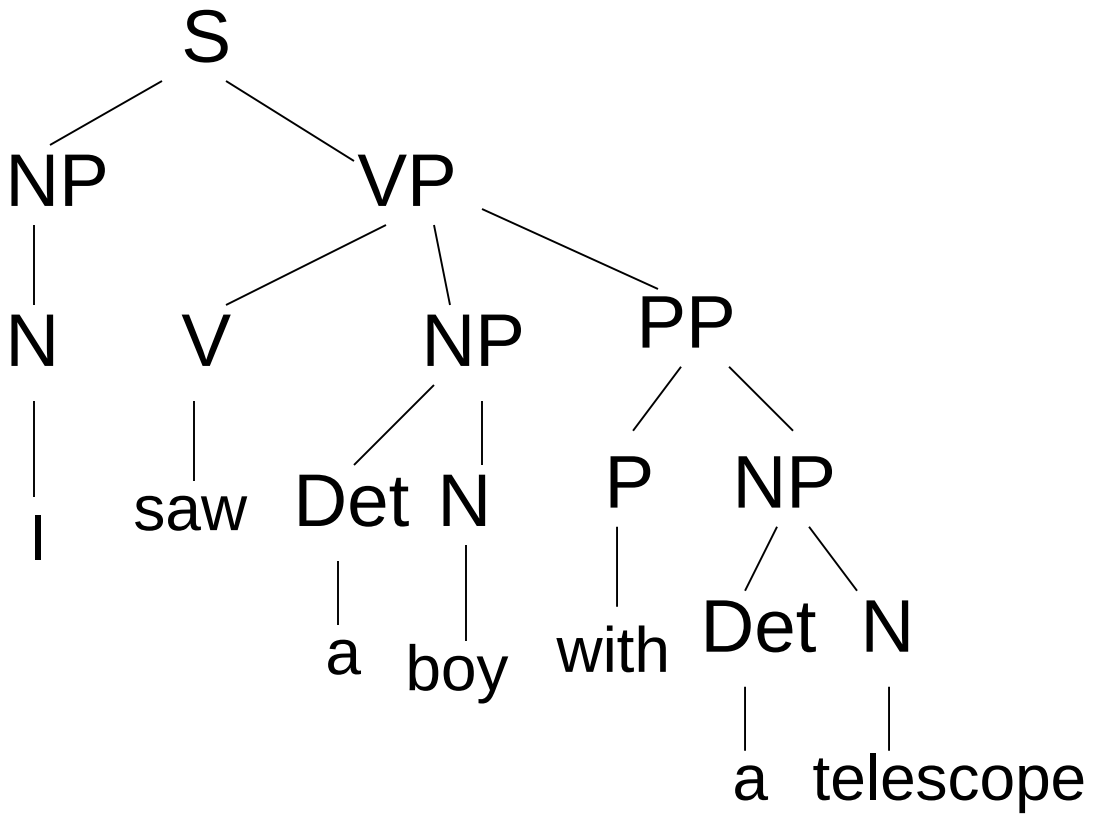
“I saw the boy with a telescope”: Constituency parse-1: *telescope with boy*



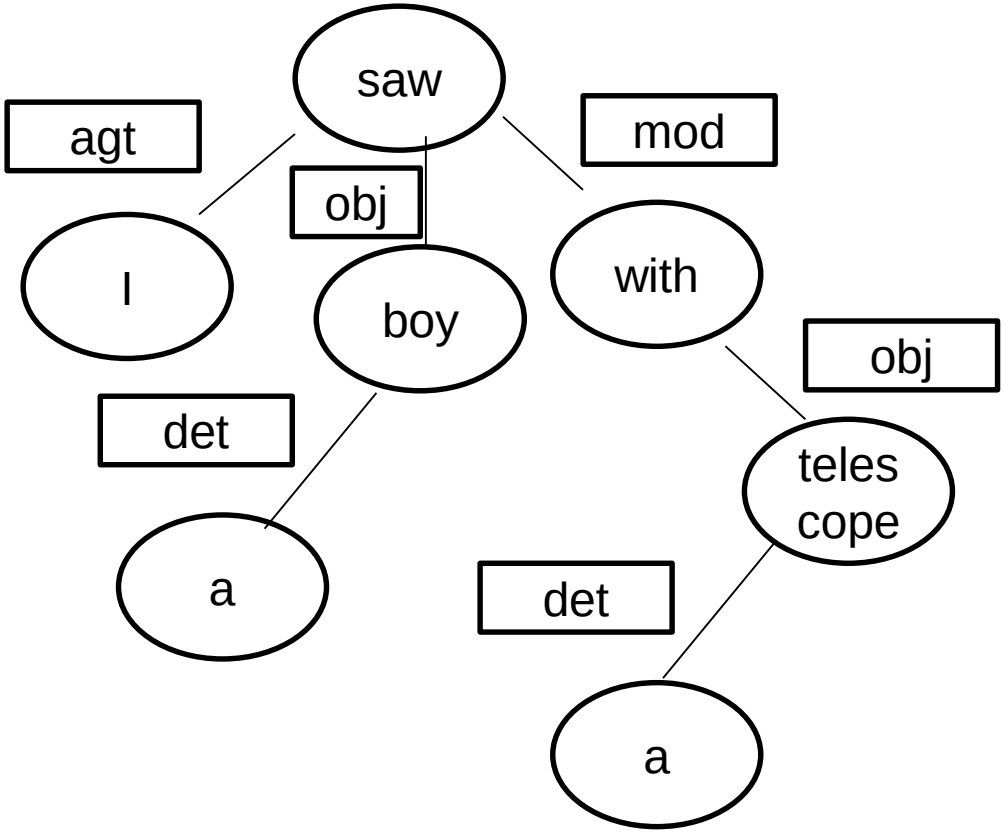
“I saw the boy with a telescope”: Dependency Parse Tree-1



Constituency Parse Tree-2: *telescope with me*



Dependency Parse Tree-2



Advantage of DP over CP

- Related entities are closer in DP than in CP: in terms of path length
- Free word order does not affect DP; CP needs additional rules
- Additional rules may overgeneralize!!

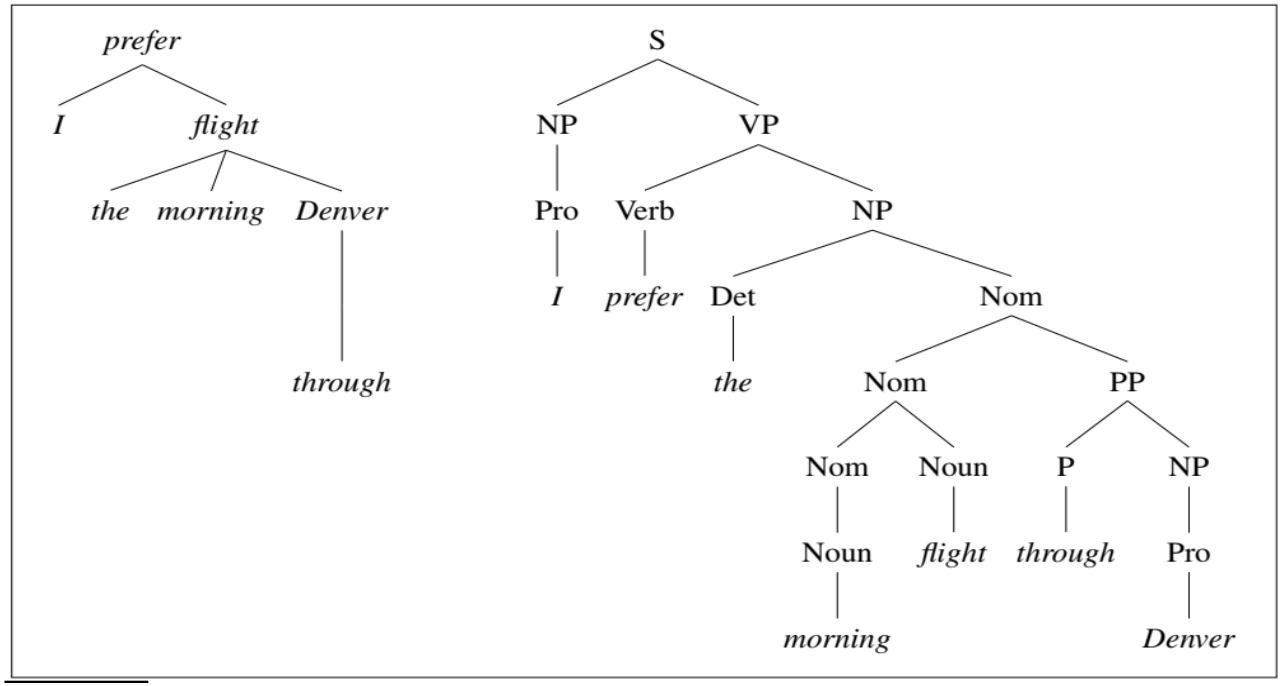
…CP needs additional rules

- *I saw the boy with a telescope*
  - *S → NP VP*
  - *VP → VBD NP PP*
- *With a telescope I saw the boy*
  - *S → NP VP*
  - *S → PP NP VP ???*

Impact of free word order on constituency parsing

- Constituency parse fundamentally use adjacency information.
- Word order disturbs the adjacency
- Chomsky normal form demands that
  - The deduction should happen by linking together two adjacent entities.
- Example:
  - राम ने श्याम को देखा | (Ram ne Shyam ko dekha)
    - श्याम को देखा =VP
  - श्याम को राम ने देखा | (Shyam ko Ram ne dekha)
    - VP is discontinuous
    - Constituency parsing fails here
  - The agent and object is reversed in the above example.
  - CP needs additional rules

Arguments are immediately linked



J & M, Chapter 15, 3rd Edition

Prefer: who prefers? "I"; what is preferred?: "flight".

On the other hand, phrases are like *suitcases* that put all related things **at one place**: "The morning flight through Denver"

Subset of Dependency Relations: from Universal Dependency Project (Nivre et all 2016)

Clausal Argument Relations	**Description**
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Examples to illustrate Dependency Relations

- NSUBJ, DOBJ, IOBJ- "*Ram gave a book to Shyam*"
  - Main Verb (MV): *gave*
  - NSUBJ: *Ram;* DOBJ: *book;* IOBJ: *Shyam*
- CCOMP, XCOMP: "I said that he should go", "I told him to go"
  - CCOMP: *said→go*
  - XCOMP: *told→go*

A note on CCOMP and XCOMP

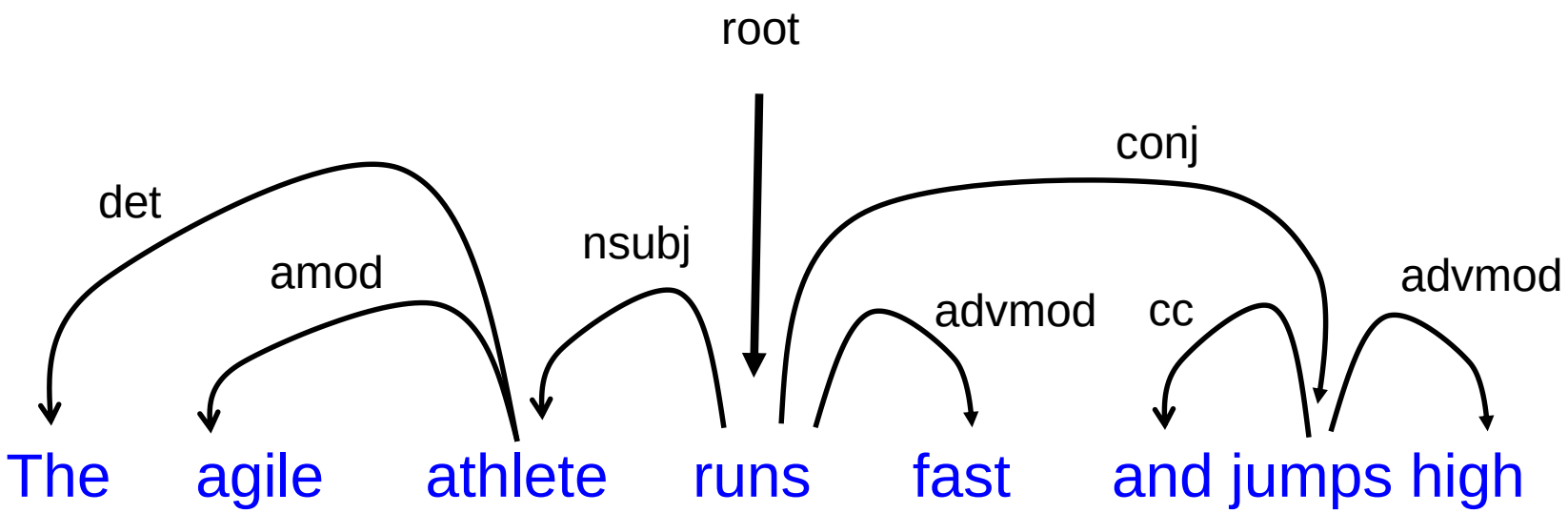
- CCOMP links the main verb with the finite verb
- XCOMP links main verb with an infinite verb
- Finite verb means: “takes GNPTAM marking”
- Infinite verb: remains in lemma form
- E.g. “told him to *go”:* ‘go’ will not change form (infinite form)
- “said he should go/be_going”: ‘go’ can change form

Illustration of DRs cntd.

- NMOD (nominal modifier), AMOD (adjective modifier), NUMMOD (numerical modifier), APPOS (appositional modifier)
  - NMOD: *The bungalow of the Director: Director ← bungalow*
  - AMOD: *The large bungalow: large ← bungalow*
  - NUMMOD: *Three cups: three ← cups*
  - APPOS: *covid19, the pandemic: covid19 ← pandemic*

Illustration of DRs cntd.

- DET (determiner), CASE (preposition, postposition and other case markers), CONJ (conjunct), CC (coordinating conjuct)
  - DET: *The bungalow: The→bungalow*
  - CASE: *The bungalow of Director: of→Director*
  - CONJ: *He is sincere and honest: sincere→honest*
  - CC: *He is sincere and honest: honest→and*


root
conj
det
amod
nsubj
advmod
cc
advmod
The
agile
athlete
runs
fast
and
jumps
high

Dependency Tree

- (1) There is a single designated root node that has no incoming arcs.
- (2) With the exception of the root node, each vertex has exactly one incoming arc.
- (3). There is a unique path from the root node to each vertex in *V*.

Projectivity

Definition

- An arc from a head to a modifier is said to be projective if there is a path from the head to every word that lies between the head and the modifier in the sentence
- A dependency tree is then said to be projective if all the arcs that make it up are projective
- ***Intuition***- the dependency graph can be drawn on a plane w/o crossing of arcs (condition: all arc must be on ONE side of the sentence: upper or lower, but not both)

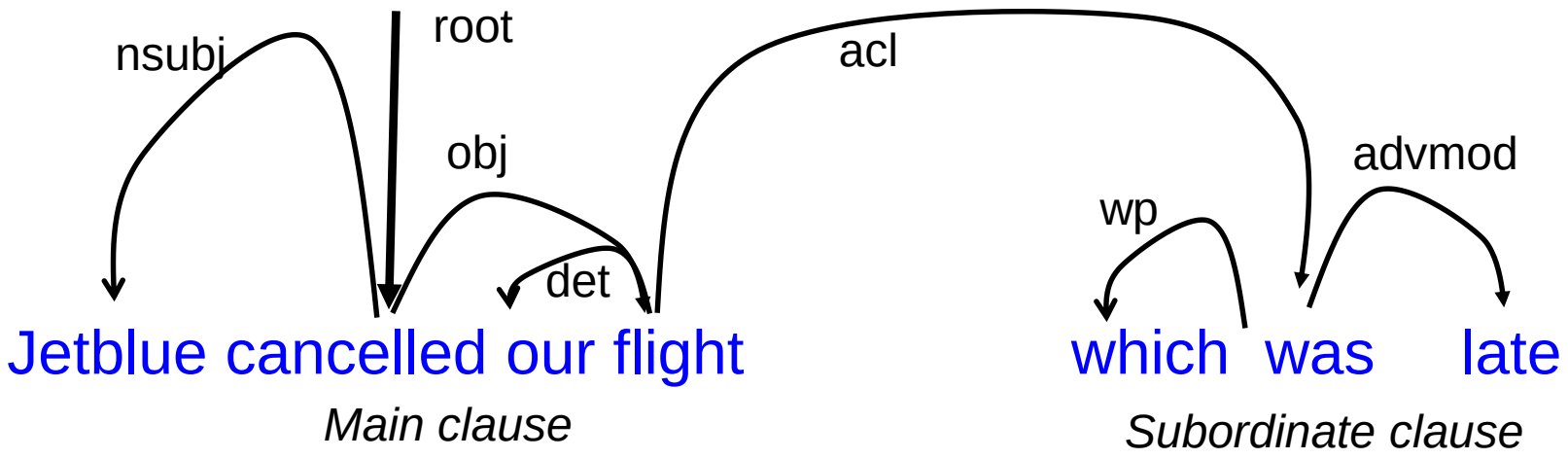
Conditions for projective dependency tree

1. All arcs are on ONE side (above or below) of the sentence.

2. There is NO crossing of arcs.

Equivalent: for EVERY Head-Modifier pair in the sentence, there is a path from the said Head to EVERY word in between the said Head and the said modifier.

Example (from J & M, 2019)



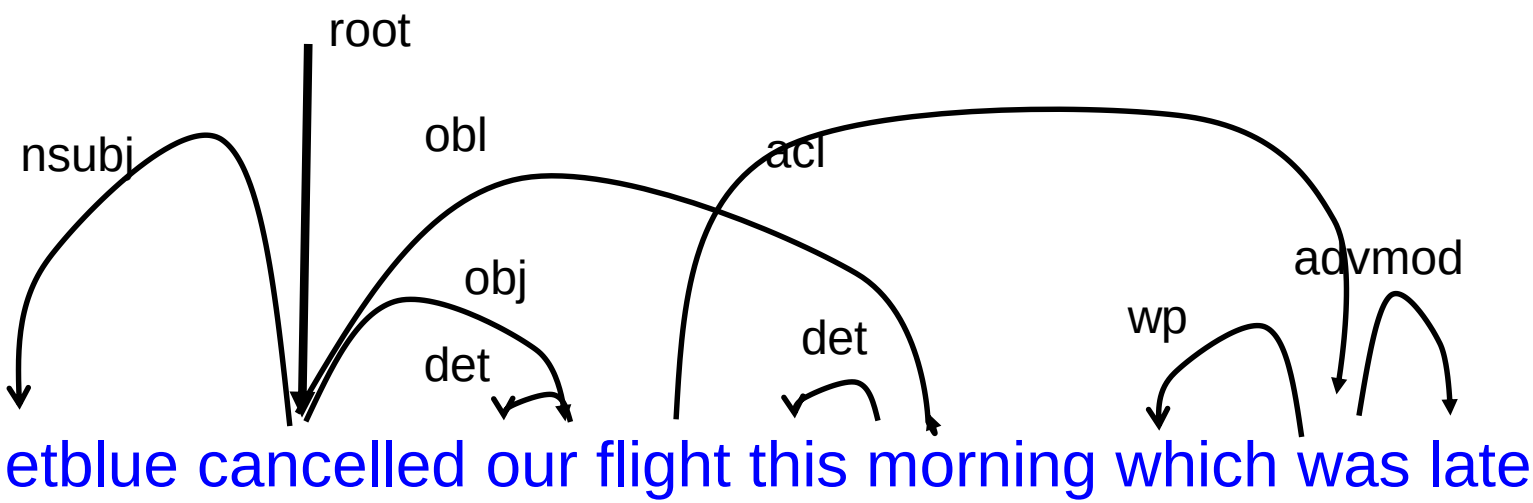
Uses Universal Dependency

wp- relative pronoun

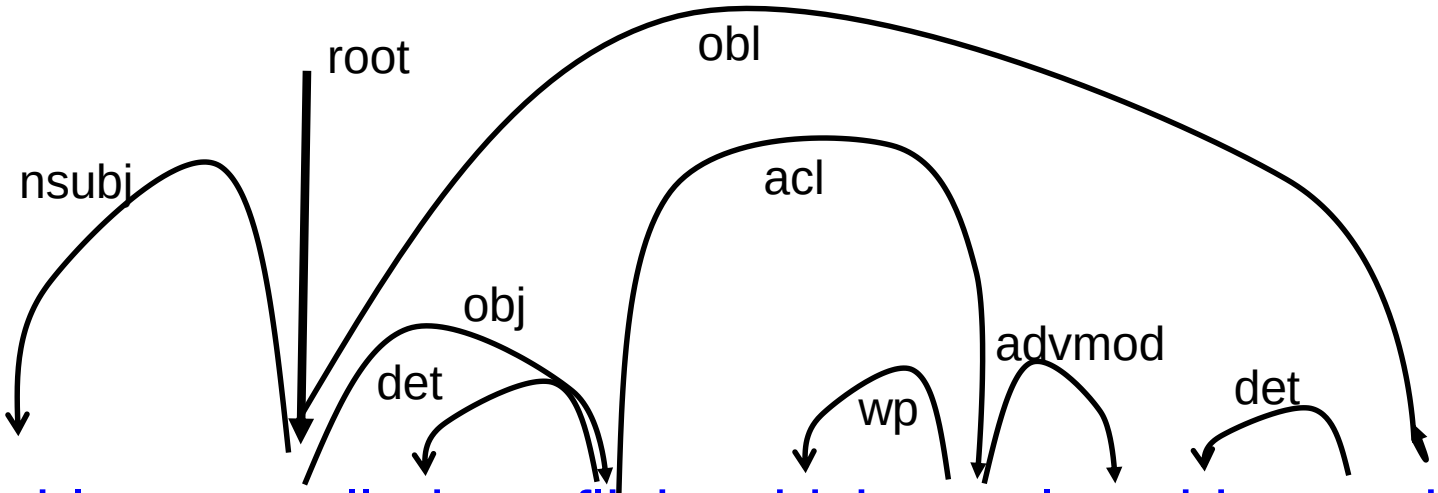
acl- clausal modifier of noun

The head of the acl relation is the noun that is modified, and the dependent is the head of the clause that modifies the noun

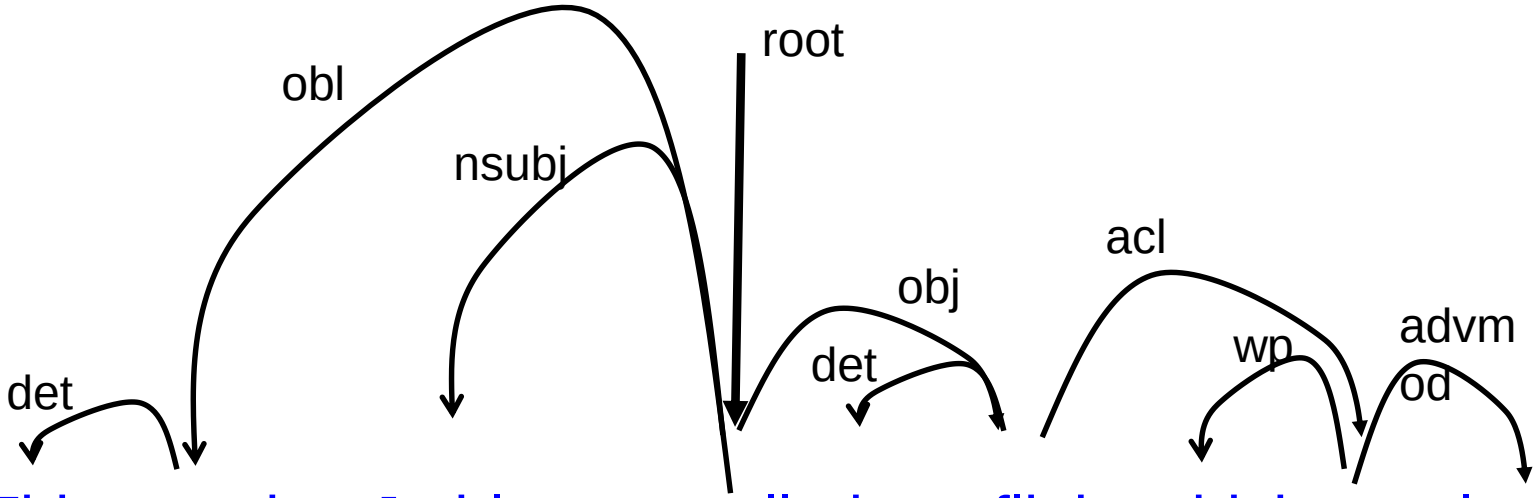
Example cntd. (from J & M, 2019):
insert “this morning”



Move around “this morning”

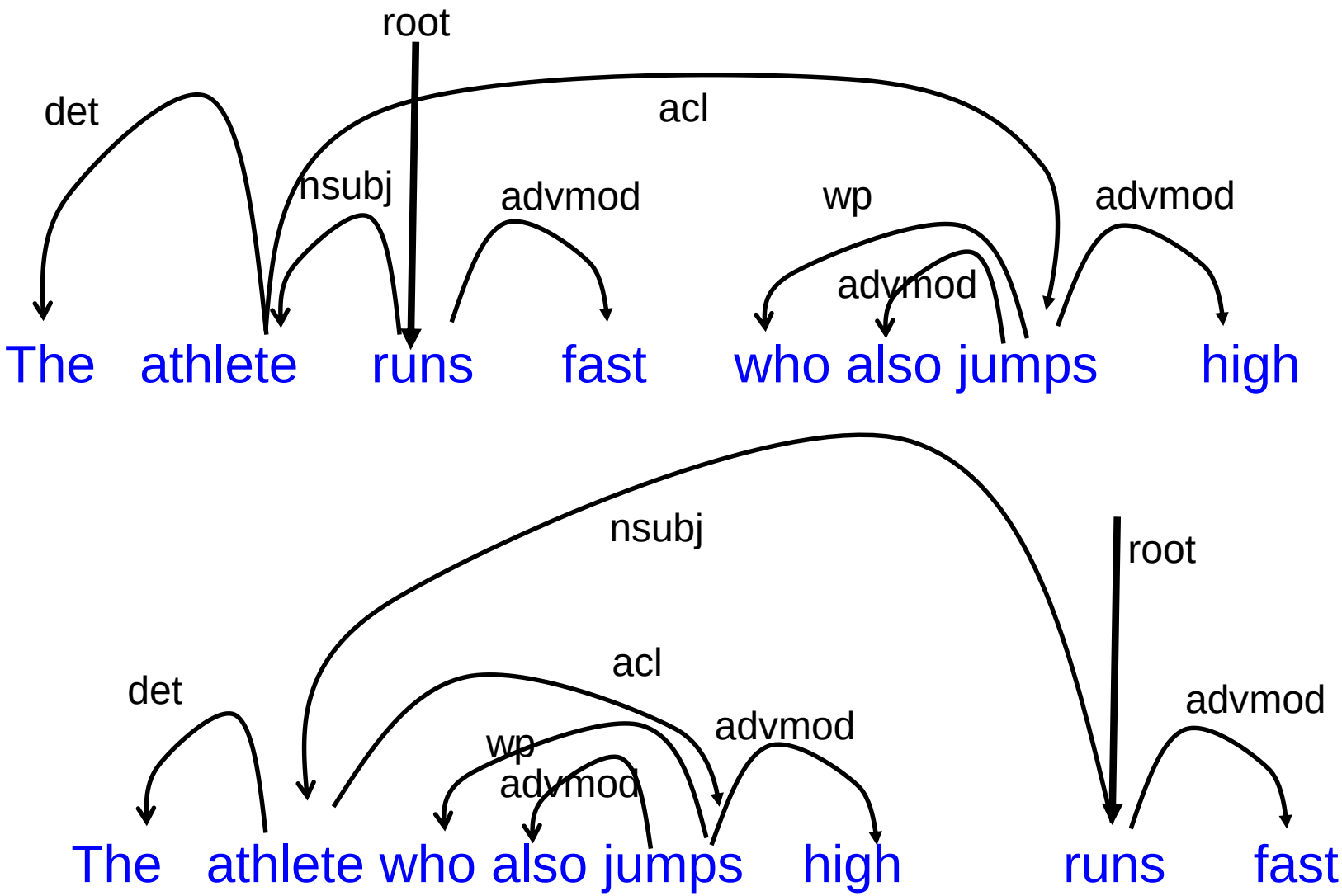


Jetblue cancelled our flight which was late this morning

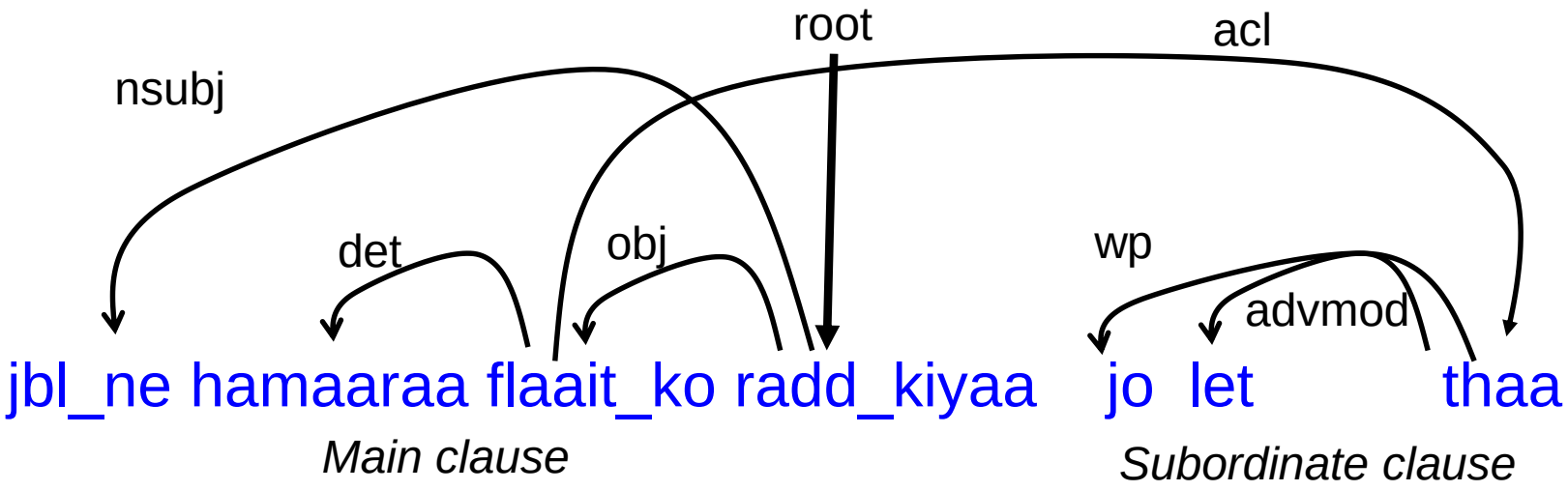


This morning Jetblue cancelled our flight which was late

Another Example

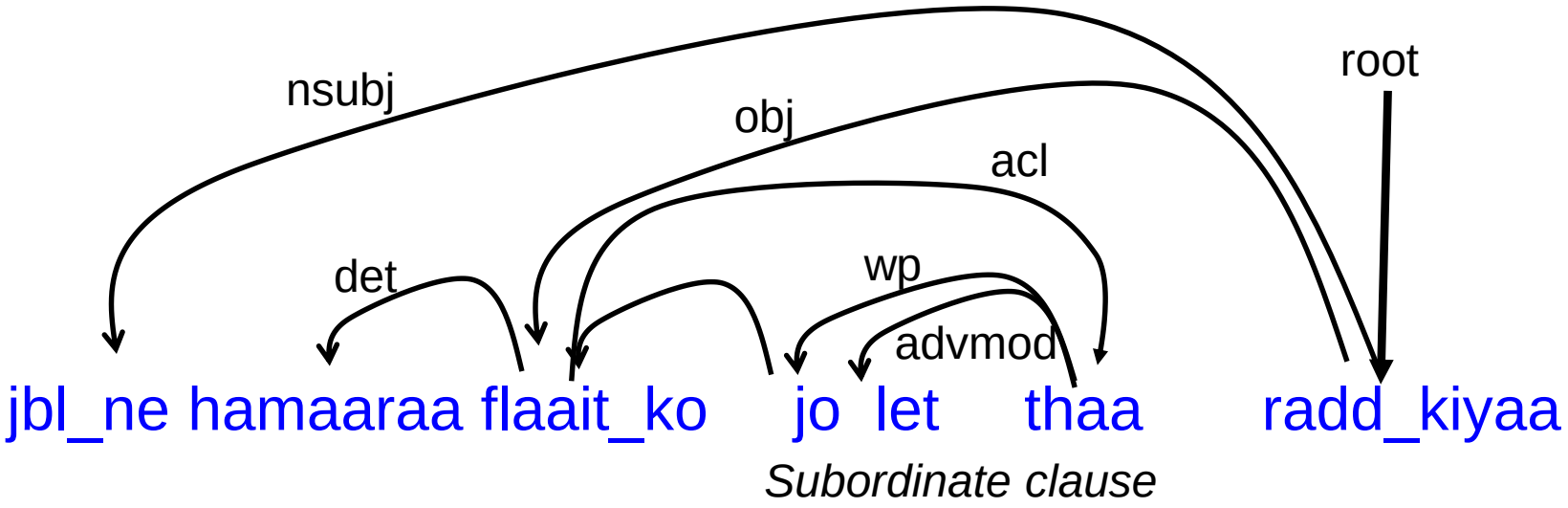


DP for other languages (Indian languages)



Uses Universal Dependency

DP for other languages (Indian languages)



Uses Universal Dependency

Pragmatic considerations

- The most sensitive parts of a sentence are the beginning and end parts
- Example: “Jetblue cancelled our flight which was late **this morning**” → “**This morning** Jetblue cancelled our flight which was late”
    - Emphasis on “this morning”
    - Restores projectivity

- **Pragmatic difference** is coming because of
    - Speech act
    - Topicalization
    - Emphasis
    - Focus

Important Insight and Principle

- Keep the Head and Modifier adjacent to each other

- Entities which are SEMANTICALLY CLOSE should be SYNTACTICALLY CLOSE too.

Vulnerability

The **greedy transition based dependency parsing** algorithm is constrained to make **projective tree**

Hence it might give wrong (semantically odd and/or wrong) parses because of projectivity constraint

Exercise: run the example “This morning…” by moving around the adjunct “this morning” on stanford online parser and observe results

Exercise

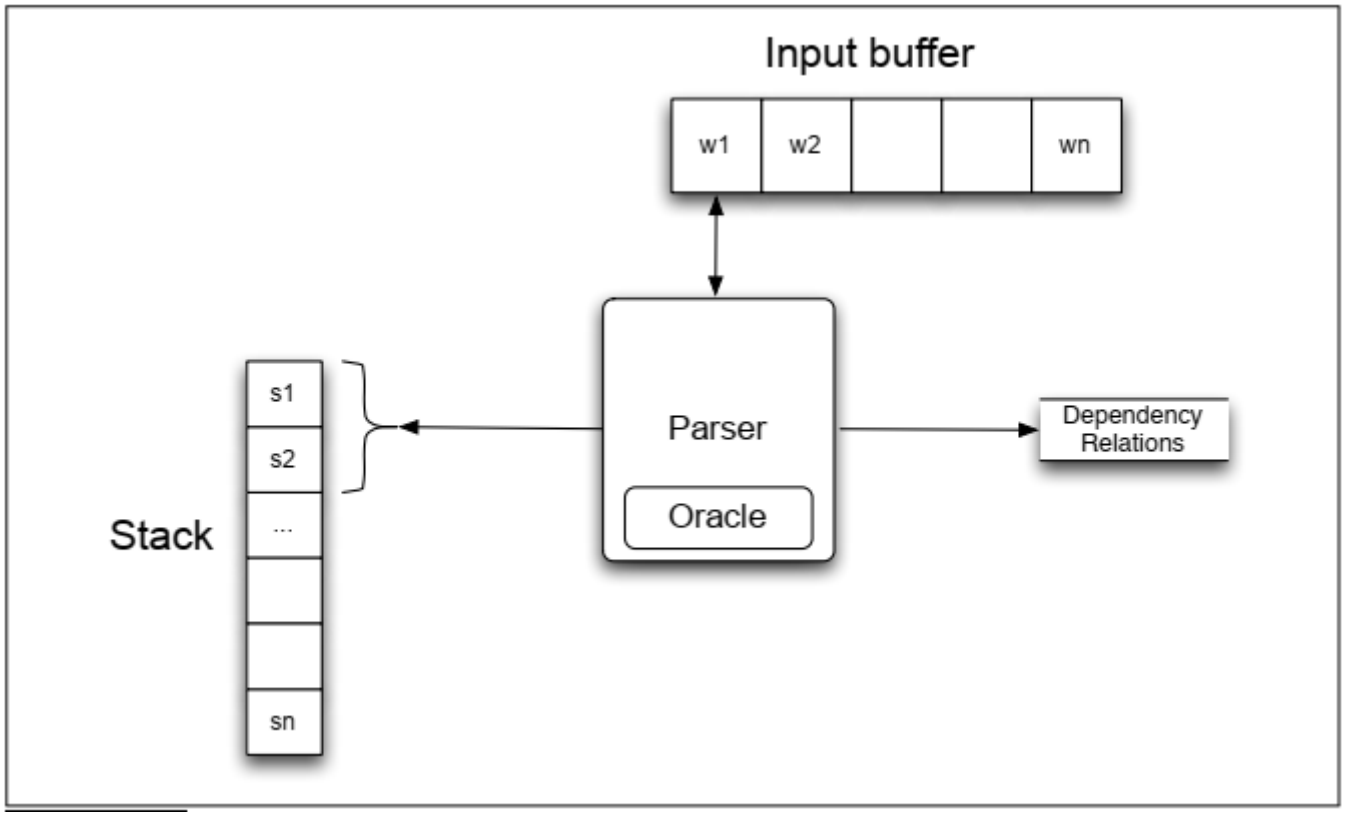
- Draw DP for
  - “*hamaara jo flaait let thaa usko jbl_ne radd_kiya*”
  - “*hamaara us flaait_ko jo let thaa jbl_ne radd_kiya*”
  - “*hamaara us flaait_ko jbl_ne radd_kiya jo let thaa*”
- Take any other language you know and repeat the above

Data Driven Algorithms for Dependency Tree Construction

Two Data Driven Approaches

- Transition-based
  - State machine for mapping a sentence to its dependency graph
  - Inducing a model for predicting the next transition, given the current state and the transition history so far.
- Graph-based
  - Induce a model for assigning scores to the candidate dependency graphs for a sentence
  - Find the maximum-scoring dependency Tree
  - Maximum spanning tree (MST) parsing

Basic Transition Based DP



Examines top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

Speech and Language Processing, Jurafksy & Martin, Ch-15, 2019

Example: transition based

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Trace of a transition-based parse

Speech and Language Processing, Jurafksy & Martin, Ch-15, 2019

ORACLE

- Decides whether to “shift” or to “reduce”
- If “reduce”, whether to set up “rightarc” or to set up “leftarc”
- Can be controlled by DEPENDENCY GRAMMAR RULES
- Or, by rules learnt from data
- Or, by a neural network

A neural transition based parser
(chen and Manning 2014)

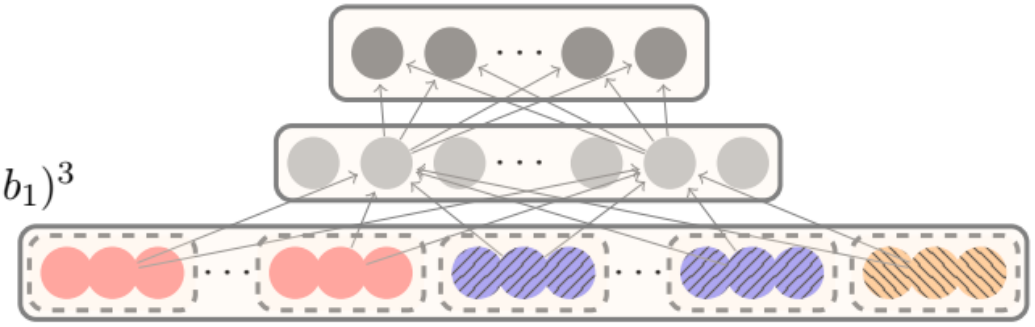
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

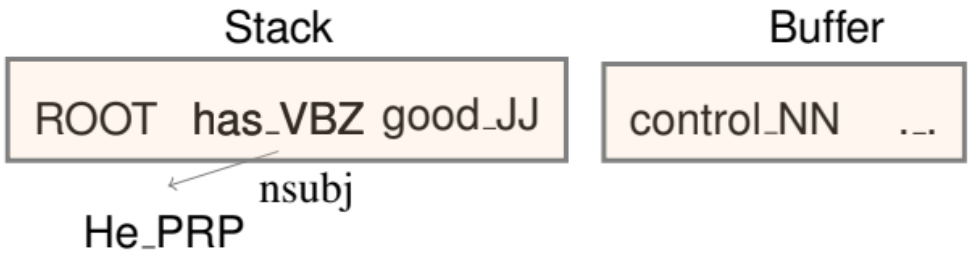
Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



Configuration



How to get the Neural Net Trained and how to get the training data

- The training data will come from Dependency Trees
- For example, given “the morning flight” and “the←flight”, “morning←flight”, it is possible to ***simulate*** the parser generate training data (next slide)
- Such trees come from ***Prague Dependency Tree Bank***

Learning of transitions

Speech and NLP, J & M, Ch 15, 2019.

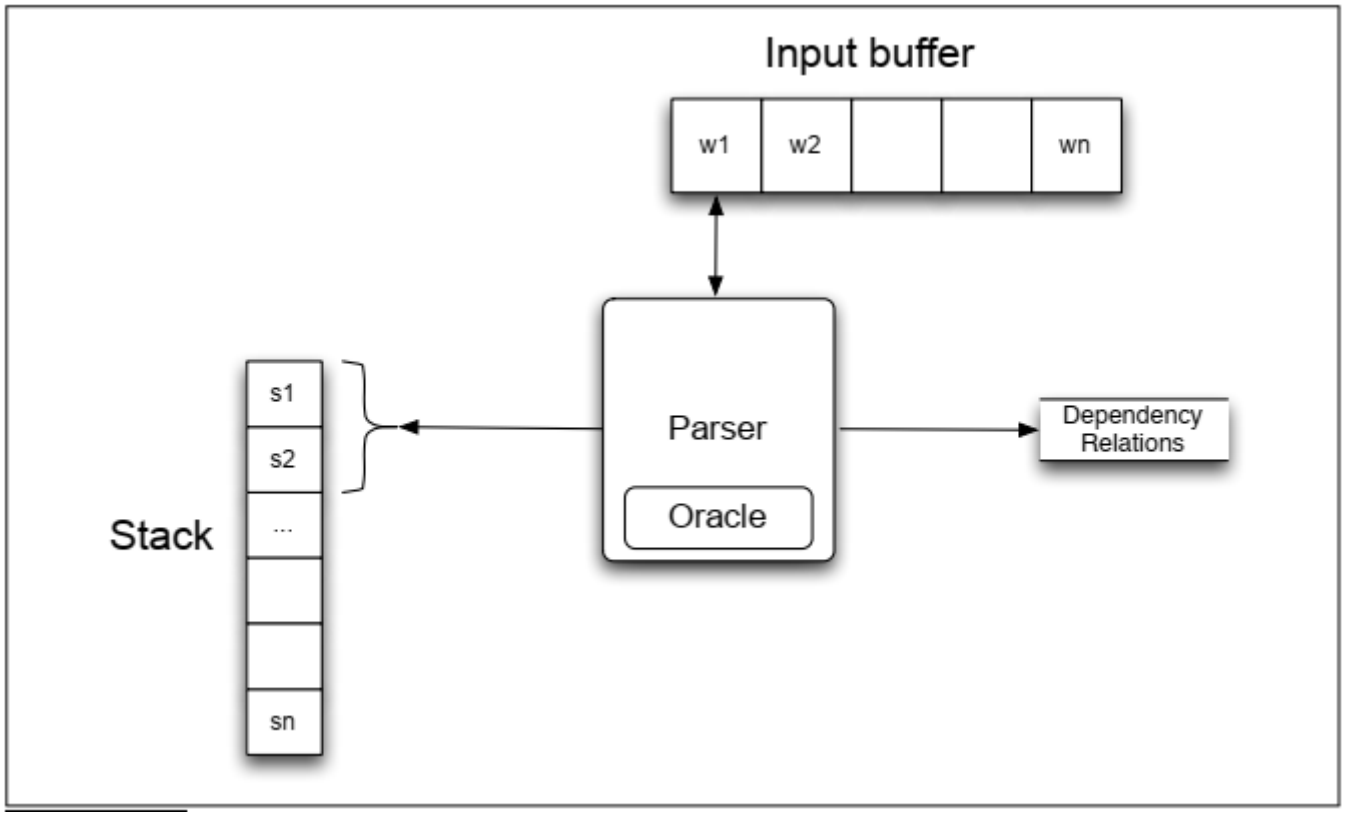
Recall: transition based DP

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Trace of a transition-based parse

Speech and Language Processing, Jurafksy & Martin, Ch-15, 2019

Basic Transition Based DP



Examines top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

Speech and Language Processing, Jurafksy & Martin, Ch-15, 2019

Operators: shift, leftarc, rightarc

function DEPENDENCYPARSE(*words*) **returns** dependency tree

state ← {[root], [*words*], [] } ; initial configuration
while *state* **not final**
t ← ORACLE(*state*) ; choose a transition operator to apply
state ← APPLY(*t*, *state*) ; apply it, creating a new state
return *state*

Generation of Training Data

Step	Stack	Word List	Predicted Action
0	[root]	[book, the, flight, through, houston]	SHIFT
1	[root, book]	[the, flight, through, houston]	SHIFT
2	[root, book, the]	[flight, through, houston]	SHIFT
3	[root, book, the, flight]	[through, houston]	LEFTARC
4	[root, book, flight]	[through, houston]	SHIFT
5	[root, book, flight, through]	[houston]	SHIFT
6	[root, book, flight, through, houston]	[]	LEFTARC
7	[root, book, flight, houston]	[]	RIGHTARC
8	[root, book, flight]	[]	RIGHTARC
9	[root, book]	[]	RIGHTARC
10	[root]	[]	Done

Training data

How are operators generated

LEFTARC(r): **if** $(S_1\ r\ S_2) \in R_p$

RIGHTARC(r): **if** $(S_2\ r\ S_1) \in R_p$ **and** $\forall r', w\ s.t. (S_1\ r'\ w) \in R_p$ **then** $(S_1\ r'\ w) \in R_c$

SHIFT: **otherwise**

Generation of Training Data

Step	Stack	Word List	Predicted Action
0	[root]	[book, the, flight, through, houston]	SHIFT
1	[root, book]	[the, flight, through, houston]	SHIFT
2	[root, book, the]	[flight, through, houston]	SHIFT
3	[root, book, the, flight]	[through, houston]	LEFTARC
4	[root, book, flight]	[through, houston]	SHIFT
5	[root, book, flight, through]	[houston]	SHIFT
6	[root, book, flight, through, houston]	[]	LEFTARC
7	[root, book, flight, houston]	[]	RIGHTARC
8	[root, book, flight]	[]	RIGHTARC
9	[root, book]	[]	RIGHTARC
10	[root]	[]	Done

Training data

A neural transition based parser
(chen and Manning 2014)

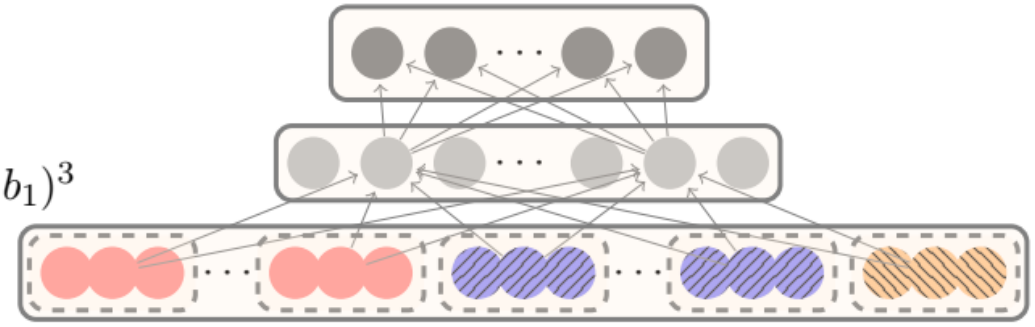
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

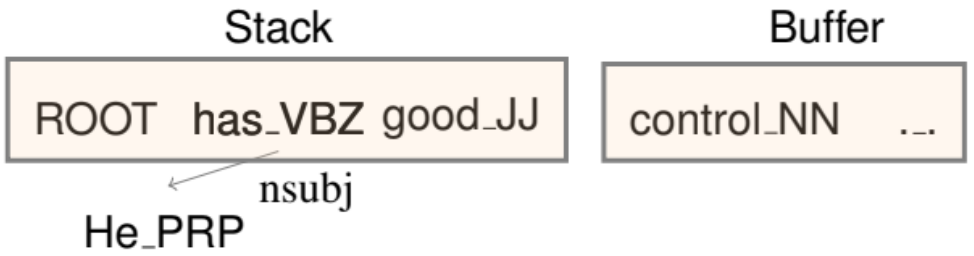
Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



Configuration



Input to Neural Net

Single-word features (9)
$s_1.w$; $s_1.t$; $s_1.wt$; $s_2.w$; $s_2.t$;
$s_2.wt$; $b_1.w$; $b_1.t$; $b_1.wt$
Word-pair features (8)
$s_1.wt \circ s_2.wt$; $s_1.wt \circ s_2.w$; $s_1.wts_2.t$;
$s_1.w \circ s_2.wt$; $s_1.t \circ s_2.wt$; $s_1.w \circ s_2.w$
$s_1.t \circ s_2.t$; $s_1.t \circ b_1.t$
Three-word feaures (8)
$s_2.t \circ s_1.t \circ b_1.t$; $s_2.t \circ s_1.t \circ lc_1(s_1).t$;
$s_2.t \circ s_1.t \circ rc_1(s_1).t$; $s_2.t \circ s_1.t \circ lc_1(s_2).t$;
$s_2.t \circ s_1.t \circ rc_1(s_2).t$; $s_2.t \circ s_1.w \circ rc_1(s_2).t$;
$s_2.t \circ s_1.w \circ lc_1(s_1).t$; $s_2.t \circ s_1.w \circ b_1.t$

- The feature templates used for analysis
  - *lc*1(*si*) and *rc*1(*si*) denote the leftmost and rightmost children of s_i
  - *w* denotes word
  - *t* denotes POS tag

Features: example sentence "*cancelled flights to Houston*"

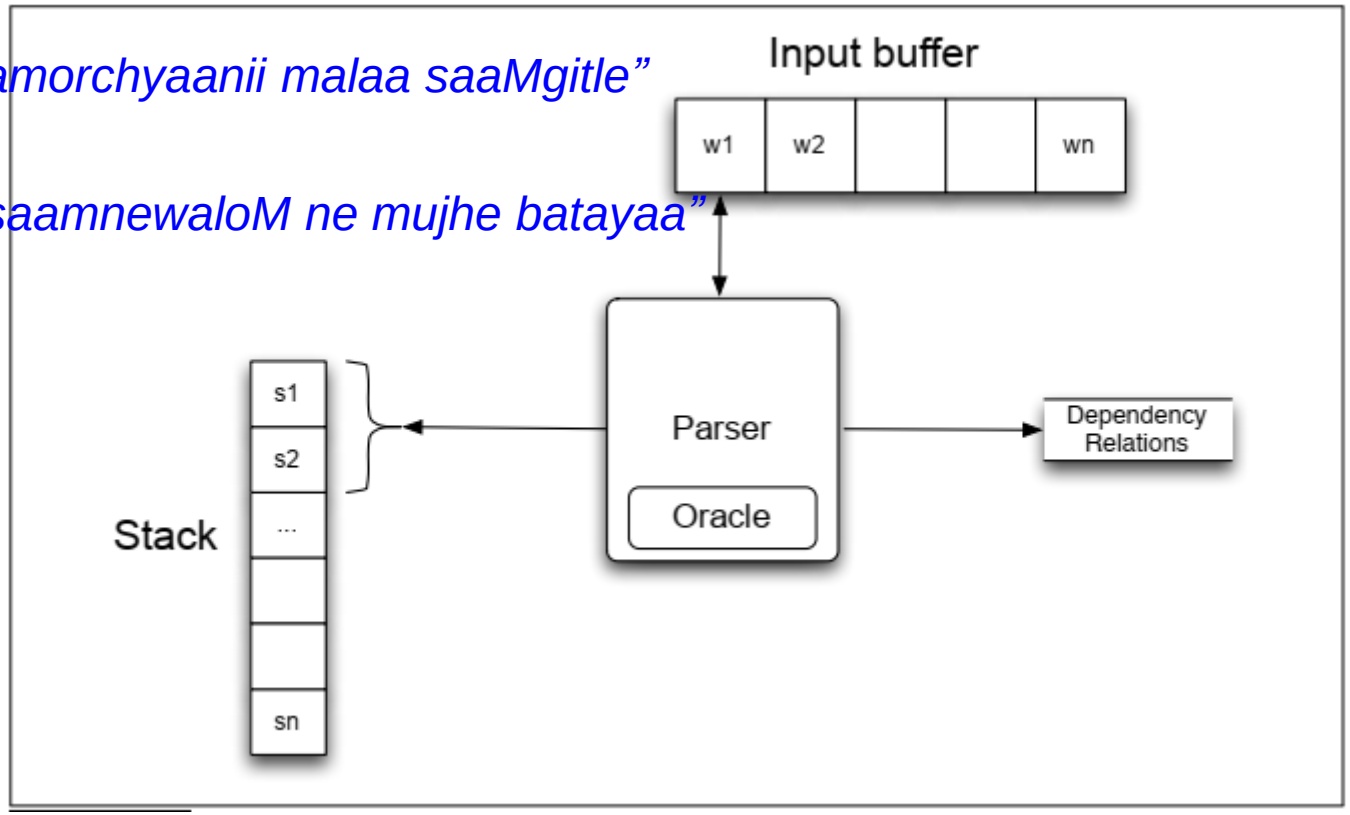
$$\langle s_1.w = flights, op = shift \rangle$$
$$\langle s_2.w = canceled, op = shift \rangle$$
$$\langle s_1.t = NNS, op = shift \rangle$$
$$\langle s_2.t = VBD, op = shift \rangle$$
$$\langle b_1.w = to, op = shift \rangle$$
$$\langle b_1.t = TO, op = shift \rangle$$
$$\langle s_1.wt = flightsNNS, op = shift \rangle$$

DP across languages

- *“people in front of the house told me”*
- *“gharaasamorchyaanii malaa saaMgitle”*
- *“ghar ke saamnewaloM ne mujhe batayaa”*

Multilingual DP

- *“people in front of the house told me”*
- *“gharaasamorchyaanii malaa saaMgitle”*
- *“ghar ke saamnewaloM ne mujhe batayaa”*



Examines top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

Speech and Language Processing, Jurafksy & Martin, Ch-15, 2019

Essence of DP

- Cannot pop a *head* out of the stack if any of its dependents remains on the stack

- The above works if the sentence's semantics is consistent with projectivity

Recall: CYK Algo

- S → NP VP 1.0
- NP → DT NN 0.5
- NP → NNS 0.3
- NP → NP PP 0.2
- PP → P NP 1.0
- VP → VP PP 0.6
- VP → VBD NP 0.4

- DT → the 1.0
- NN → gunman 0.5
- NN → building 0.5
- VBD → sprayed 1.0
- NNS → bullets 1.0

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	sprayed	the	building	with	bullets
The		L					
gunman							
sprayed							
the							
building							
with							
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	**the**	**building**	**with**	**bullets**
The		L					
gunman			L				
sprayed							
the							
building							
with							
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	the	**building**	**with**	**bullets**
The		L					
gunman			L				
sprayed							
the					L		
building							
with							
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	the	buildi ng	**with**	**bullets**
The		L					
gunman			L				
sprayed					R		
the					L		
building							
with							
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	the	buildi ng	**with**	bullets
The		L					
gunman			L				
sprayed					R		
the					L		
building							
with							R
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	the	buildi ng	**with**	bullets
The		L					
gunman			L				
sprayed					R		
the					L		
building							
with							R
bullets							

CYK based DP: 1/

The gunman sprayed the building with bullets (unlabeled DT)

Head→ Modifier	The	gunman	**sprayed**	the	building	with	bullets
The		L					
gunman			L				
sprayed					R	R	
the					L		
building							
with							R
bullets							

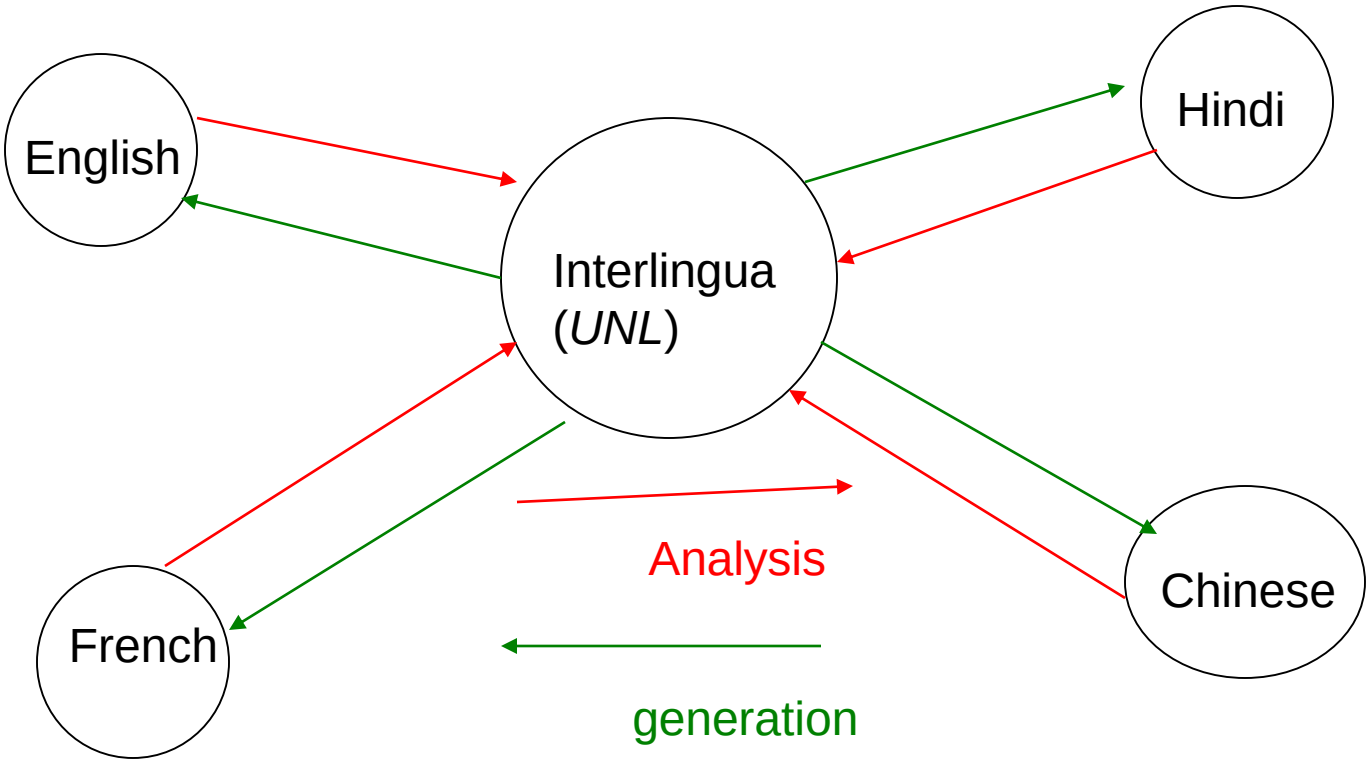
Universal Networking Language: Foundations and Applications

Introduction

Motivation

- Extraction of *semantics, i.e., deep meaning* is important for many applications.
  - Machine Translation, Meaning-based IR, CLIR
- Robust, scalable & efficient methods of knowledge extraction required
- Machine Translation and Cross Lingual IR: a need of the hour for crossing language barrier

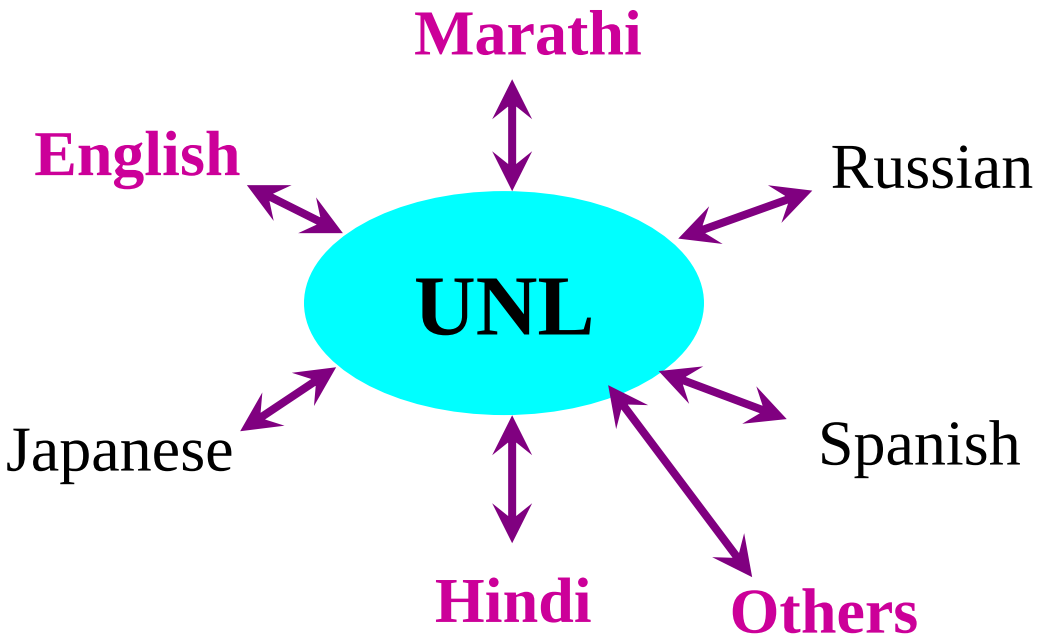
Interlingua: *a vehicle for machine translation*



UNL: a United Nations project

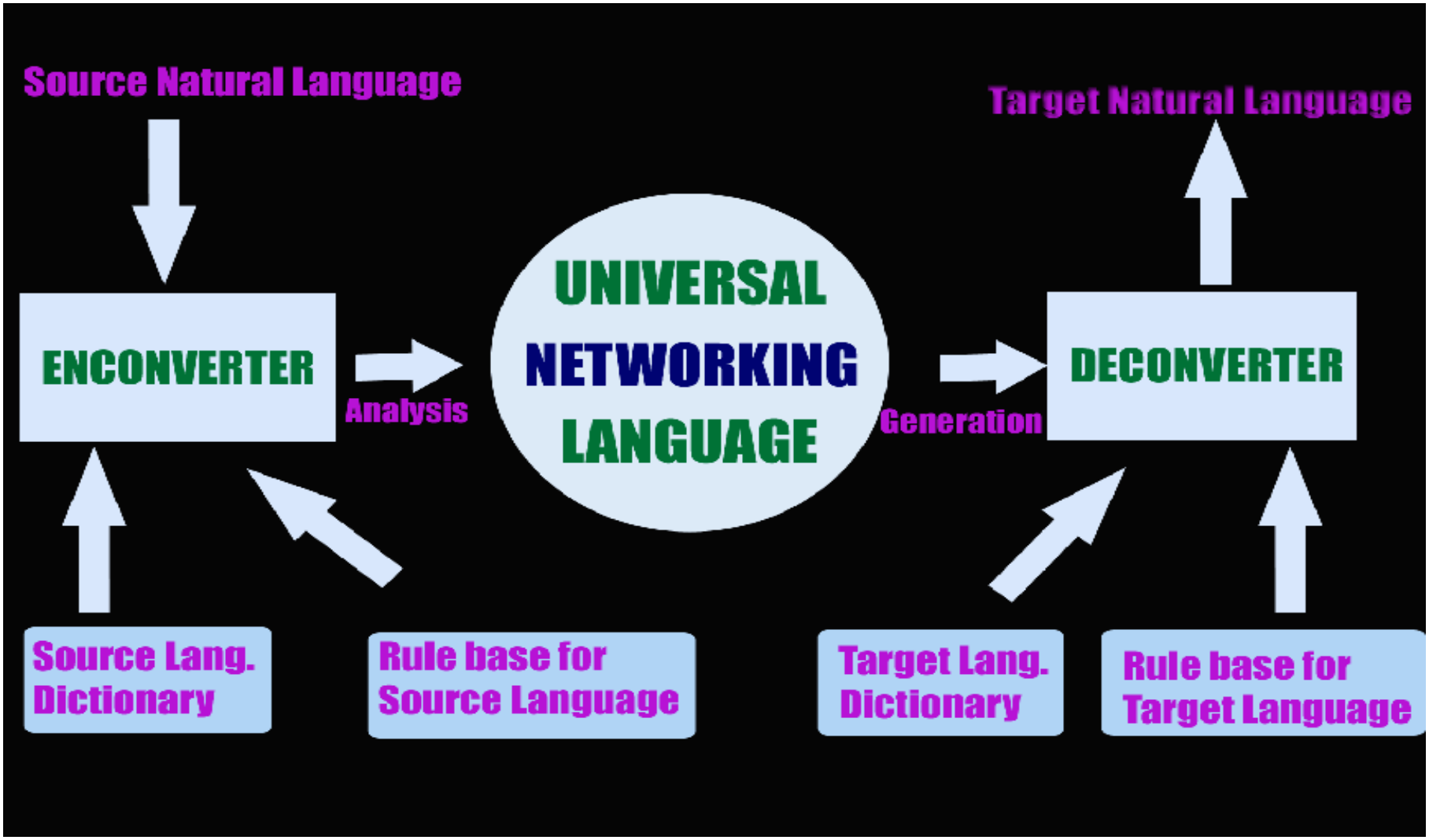
- Started in 1996
- 10 year program
- 15 research groups across continents
- First goal: generators
- Next goal: analysers (needs solving various ambiguity problems)
- Current active language groups
  - UNL_French (GETA-CLIPS, IMAG)
  - UNL_Hindi (IIT Bombay with additional work on UNL_English)
  - UNL_Italian (Univ. of Pisa)
  - UNL_Portugese (Univ of Sao Paolo, Brazil)
  - UNL_Russian (Institute of Linguistics, Moscow)
  - UNL_Spanish (UPM, Madrid)

World-wide
Universal Networking Language (UNL) Project

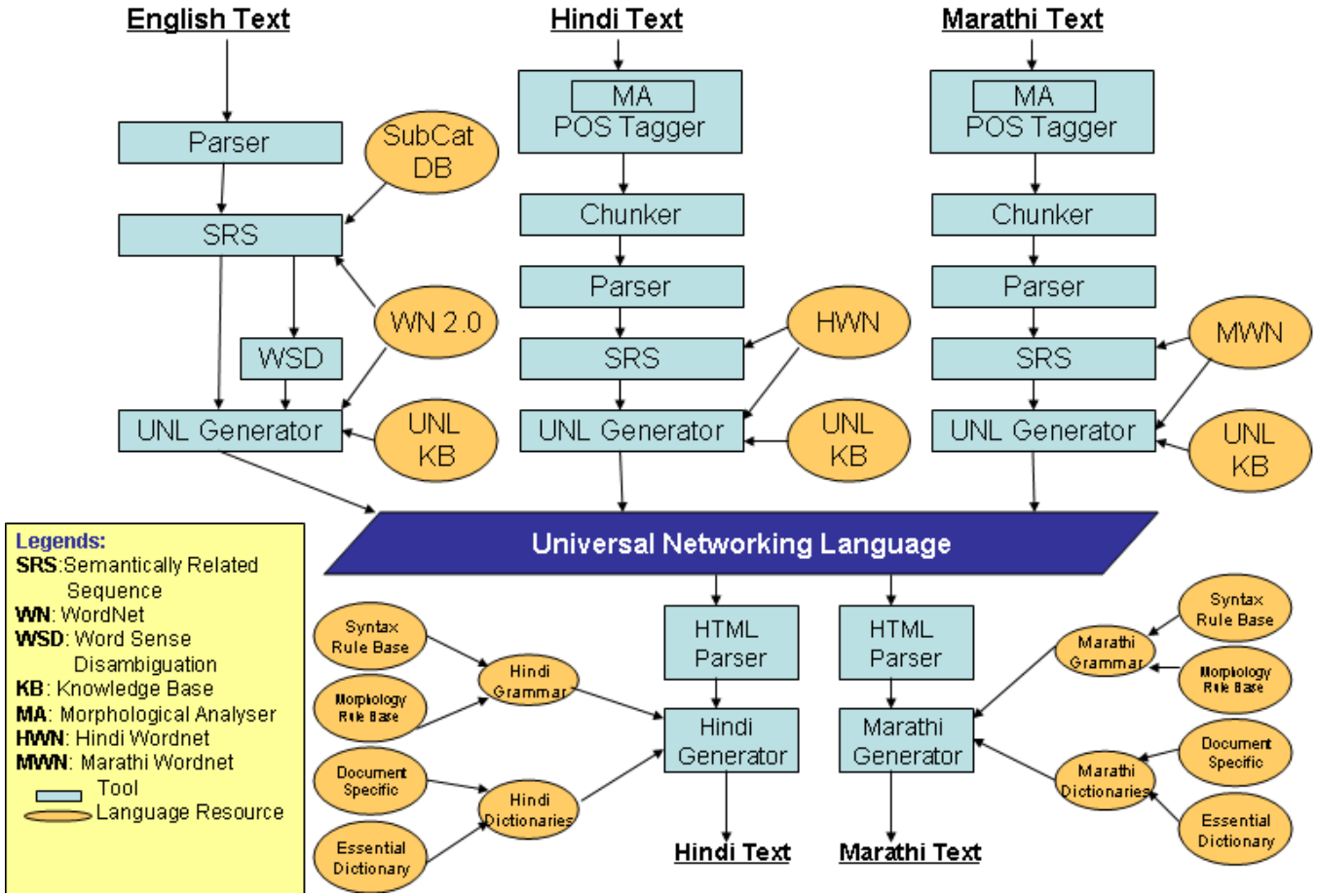


- Language independent meaning representation.

The UNL MT System: an Overview



NLP@IITB


English Text
Hindi Text
Marathi Text
Parser
SubCat DB
SRS
WN 2.0
WSD
UNL Generator
UNL KB
MA
POS Tagger
Chunker
Parser
HWN
SRS
UNL Generator
UNL KB
MA
POS Tagger
Chunker
Parser
MWN
SRS
UNL Generator
UNL KB
Universal Networking Language
Legends:
SRS:Semantically Related Sequence
WN: WordNet
WSD: Word Sense Disambiguation
KB: Knowledge Base
MA: Morphological Analyser
HWN: Hindi Wordnet
MWN: Marathi Wordnet
Tool
Language Resource
Syntax Rule Base
Morphology Rule Base
Document Specific
Essential Dictionary
Hindi Grammar
Hindi Dictionaries
HTML Parser
Hindi Generator
Hindi Text
HTML Parser
Marathi Generator
Marathi Text
Syntax Rule Base
Marathi Grammar
Morphology Rule Base
Document Specific
Marathi Dictionaries
Essential Dictionary

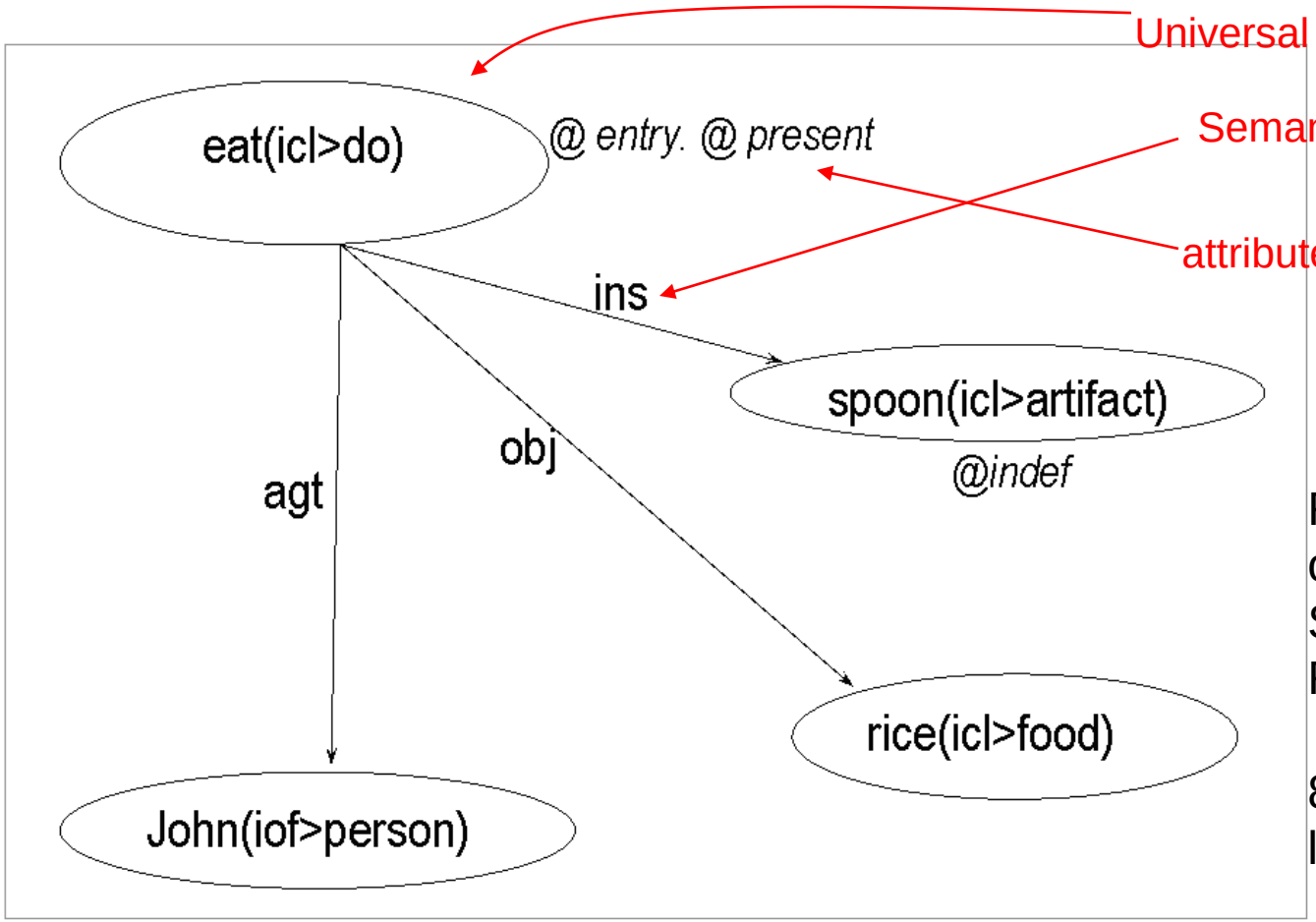
Roadmap and timemap

- UNL Foundations
  - Semantic Relations (0.5 hr)
  - Universal Words (0.5 hr)
  - Attributes (0.25 hr)
  - How to write UNL expressions (0.25 hr)
- UNL Applications
  - Machine Translation (1.5 hr)
  - Search (0.5 hr)
  - Text Entailment (0.5 hr)

The UNL System

UNL represents knowledge:

John eats rice with a spoon



Repository of 42 Semantic Relations and 84 attribute labels

Sentence embeddings

Deepa claimed that she had composed a poem.

[UNL]

agt(claim.@entry.@past, Deepa)

obj(claim.@entry.@past, :01)

agt:01(compose.@past.@entry.@complete, she)

obj:01(compose.@past.@entry.@complete,
poem.@indef)

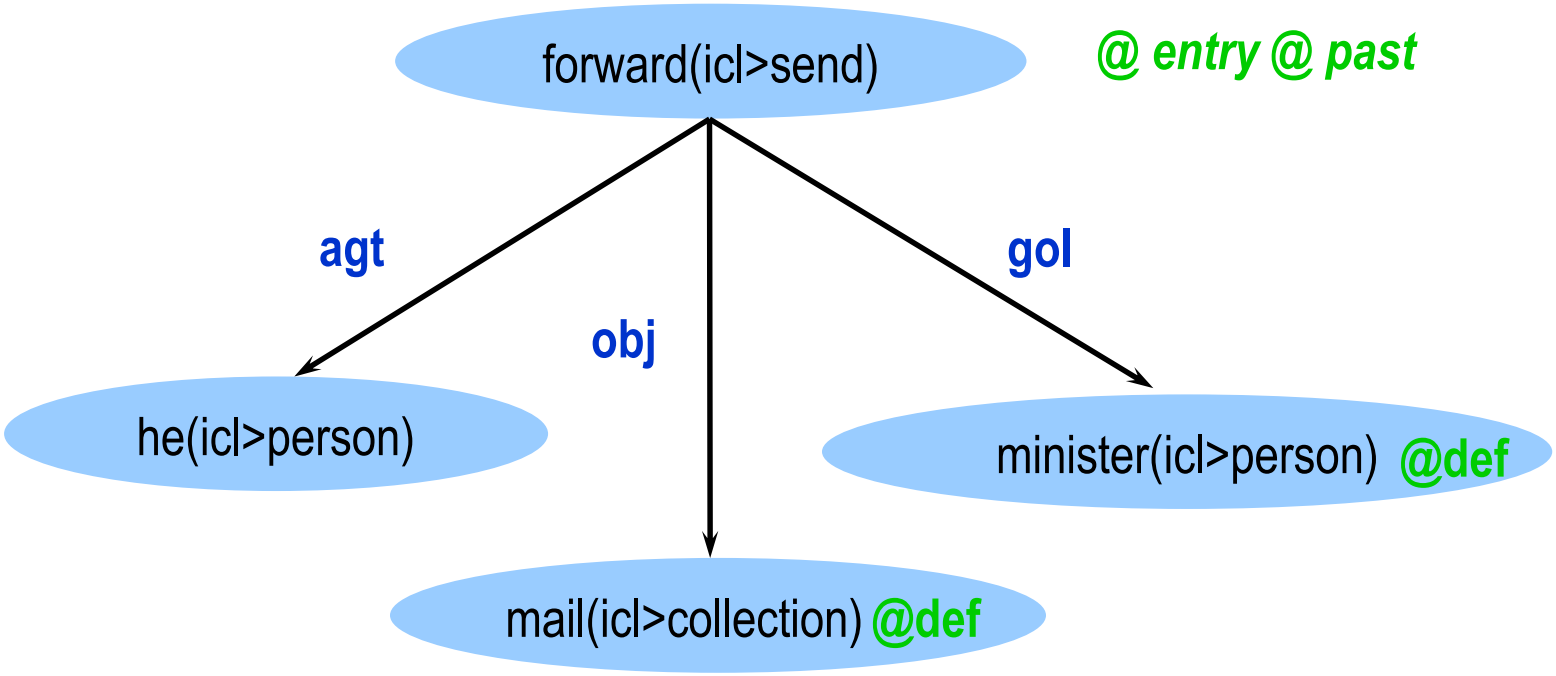
[\UNL]

Universal Networking Language

- Universal Words (UWs)
- Relations
- Attributes
- Knowledge Base

UNL Graph

He forwarded the mail to the minister.



UNL Expression

agt (forward(icl>send).@ ***entry*** @ ***past***, he(icl>person))

obj (forward(icl>send).@ ***entry*** @ ***past***, minister(icl>person))

gol (forward(icl>send).@ ***entry*** @ ***past***, mail(icl>collection). @**def**)

What is a Universal Word (UW)?

- Words of UNL
- Constitute the UNL vocabulary, the syntactic-semantic units to form UNL expressions
- A UW represents a concept
  - Basic UW (an English word/compound word/phrase with no restrictions or Constraint List)
  - Restricted UW (with a Constraint List)
- Examples:

  “crane(icl>device)”

  “crane(icl>bird)”

The Lexicon

Format of the dictionary entry

[headword] {} “Universal word“ (Attribute list);

e.g., [minister] {} “minister(icl>person)” (N,ANIMT,PHSCL,PRSN);

- Head word
- Universal word
- Attributes
  - Morphological - Pl(plural), V_ed(past tense form)
  - Syntactic - V(verb),VOA(verb of action)
  - Semantic - ANIMT(animate), PLACE, TIME

The Lexicon (cntd)

He forwarded the mail to the minister.

Content words:

[forward] {} “forward(icl>send)” (V,VOA) <E,0,0>;

[mail] {} “mail(icl>message)” (N,PHSCL,INANI) <E,0,0>;

[minister] {} “minister(icl>person)” (N,ANIMT,PHSCL,PRSN) <E,0,0>;

The Lexicon (cntd)

He forwarded the mail to the minister.

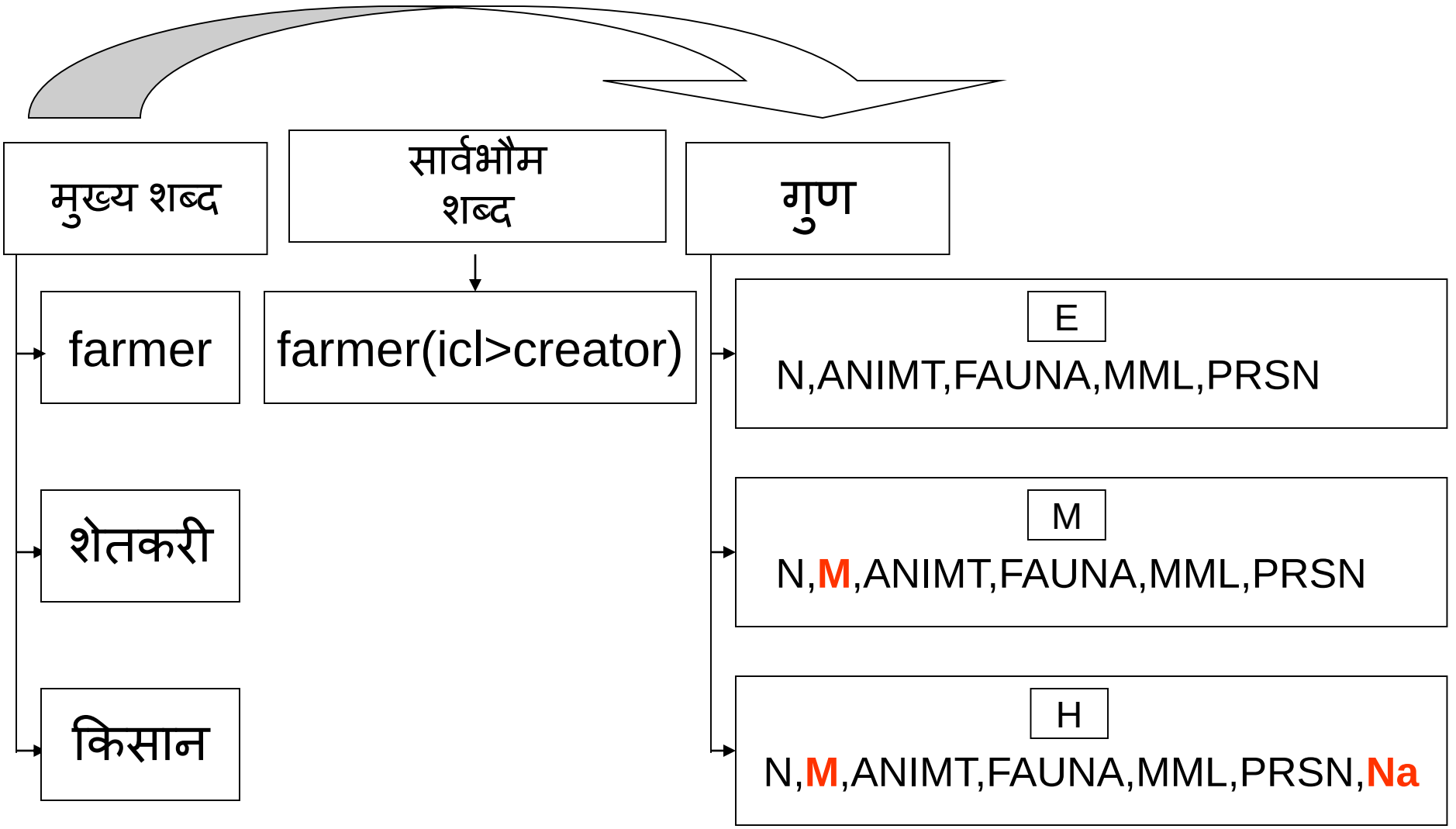
function words:

[he] {} “he” (PRON,SUB,SING,3RD) <E,0,0>;

[the] {} “the” (ART,THE) <E,0,0>;

[to] {} “to” (PRE,#TO) <E,0,0>;

Hindi example: संज्ञा का उदाहरण १/२



The Features of a UW

- Every concept existing in any language must correspond to a UW
- The constraint list should be as small as necessary to disambiguate the headword
- Every UW should be defined in the UNL Knowledge-Base

Restricted UWs

- Examples
  - *He will hold office until the spring of next year.*
  - *The spring was broken.*
- Restricted UWs, which are Headwords with a constraint list, for example:

  “spring(icl>season)”

  “spring(icl>device)”

  “spring(icl>jump)”

  “spring(icl>fountain)”