NATURAL LANGUAGE PROCESSING
Assignment Demonstration

Group:9
Kritika Jain
Vinita Sharma
Rucha Kulkarni

Outline

- Part 1: Pos-Tagger
  - Bigram
  - Trigram
- Part 2: Discriminative Vs Generative model
- Part 3: Next Word Prediction
- Part 4: Pos-Tagging with A*
- Part 5: Parser Projection
- Part 6: Yago Path finding

Part 1:

Pos-Tagger

PoS Tagger

- Data Sparsity (in Trigrams) : Handled by smoothing using linear interpolation of unigram, bigram and trigram probabilities

- Unkown words : Handled by Suffix matching

5-folds Cross Validation

- Bigrams:

	Case 1	Case 2	Case 3	Case 4	Case 5
Accuracy	90.55	90.01	90.54	91.50	90.26

- Trigrams:

	Case 1	Case 2	Case 3	Case 4	Case 5
Accuracy	91.318	91.439	91.617	91.69	91.5

Bigram Vs Trigram

- AVP: 70% to 73.5%
- NPO: 70.85% to 81.09%
- UNC: 14.9% to 44.64%

Handling Interjections

- These are often followed by “!”
- eg: hi! , what! etc
- Words followed by exclamation marks are tagged as ITJ.
- Per PoS Accuracy of ITJ:
  - Without handling: 40%
  - With Handling: 68.9%

Statistical Comparision

Property	Bigram Assumption	Trigram Assumption
Accuracy	90.57%	91.51%
F-Score	0.90021	0.9152

Part 2:

Discriminative Vs Generative Model

Per PoS Accuracy

Poor Accuracy tags (Mostly confused):

- ITJ (AJ0)
- PNI (CRD)
- VDI, VHI, VHN (VDB, VHB, VHD)
- VVB, VVD, VVI, VVN (AJ0, NN1)
- UNC (NN1)
- ZZ0 (AT0)

Result

- Accuracy: 81.948%
- F-Score: 0.8179

Discriminative Vs Generative

- Accuracy
- Discriminative (unigram) model: 81.948 %
- Generative (bigram) model: 90.57%

Comparison of all Tagging Algorithms

	To	be	or	not	to	be	.
Bigram	PRP	VBI	CJC	XX0	TO0	VBI	PUN
Trigram	PRP	VBI	CJC	XX0	TO0	VBI	PUN
AStar	PRP	VBB	CJC	XX0	TO0	VBI	PUN
Discr	TO0	VBI	CJC	XX0	TO0	VBI	PUN

Comparison of all Tagging Algorithms

	I	bank	in	the	bank	near	the	river	bank	.
Bigram	PNP	VVB	PRP	AT0	NN1	PRP	AT0	NN1	NN1	PUN
Trigram	PNP	VVB	PRP	AT0	NN1	PRP	AT0	NN1	NP0	PUN
Asrtar	PNP	VVB	PRP	AT0	NN1	PRP	AT0	NN1	NN1	PUN
Discriminative	PNP	NN1	PRP	AT0	NN1	PRP	AT0	NN1	NN1	PUN

Part 3:

Next Word Prediction

Tagged Vs Untagged Model

Sentences	Untagged Model Perplexity	Tagged Model Perplexity
hello how are you	17.720760717994096	2.1509372427449653
malaria kills man	16.433234948778548	4.555750584375696
i am going to the market	11.388537335061606	0.2599606247443197
have you done your homework	354.9266847751312	120.36482153064017
sun rises in the east	3.8204977844532206	1.4585256119892978

Tagged Vs Untagged Model

Sentence	tagged	untagged
you have to write	to	the
I don't drink:	and	.
I am tired	of	and

Part 4:

PoS Tagging with AStar

Description

- Cost of each edge: -log(p(t2/t1)*p(t2->w2))

  where the edge is from t1 to t2

- F(x)=H(x)+G(x)
  - H(x)=Distance to the target node (i.e “$”)
  - G(x)=Sum of costs from the start node (i.e “^”)

- Unknown words are handled by Suffix matching.

Viterbi Vs Astar

- Accuracy of Pos-Tagger

  Viterbi: 90.57%(Bigram), 91.83%(Trigram)

  Astar: 90.6%

Part 5:

Parser Projection

Parser Projection

- Mapping from syntactic rules of English to corresponding rule in Hindi
- Input: Sentence to be parsed, in Hindi, English
- Output: Parse tree in Hindi
- Steps:
  - English sentence is parsed using Standford Parser
  - Dependencies are converted into rules in Hindi
  - Hindi sentence is mapped to the generated tree

Mapping of Dependencies

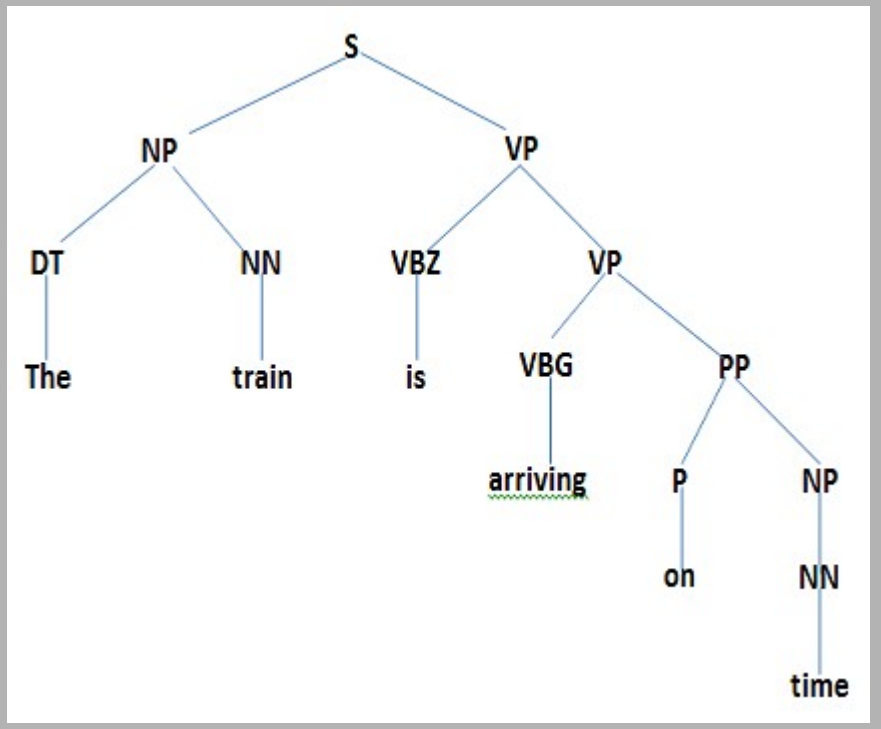
Constraints to be considered in the conversion:

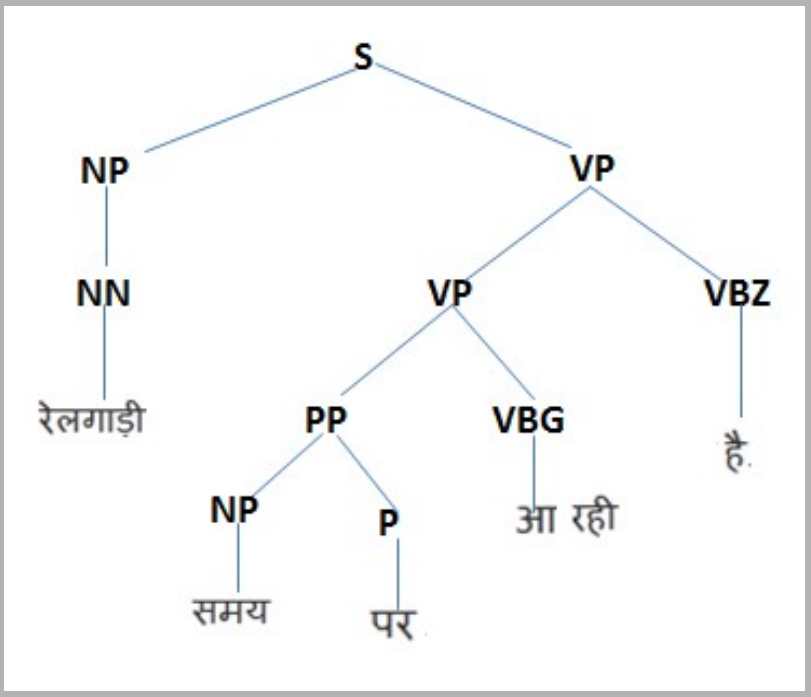
- Inversion of a few rules.
- Elimination of the words like “to”, “the” etc.
- Mapping of phrases into a single word and vice-versa.

Mapping of Dependencies- Examples

- The train is arriving on time.

  DT → The is mapped to DT → NULL

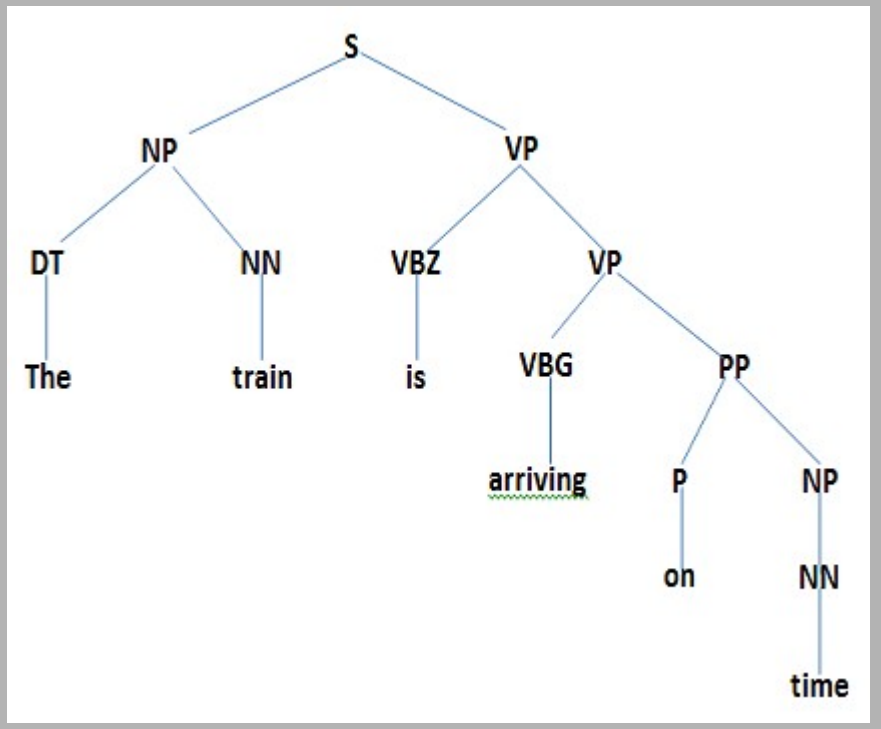


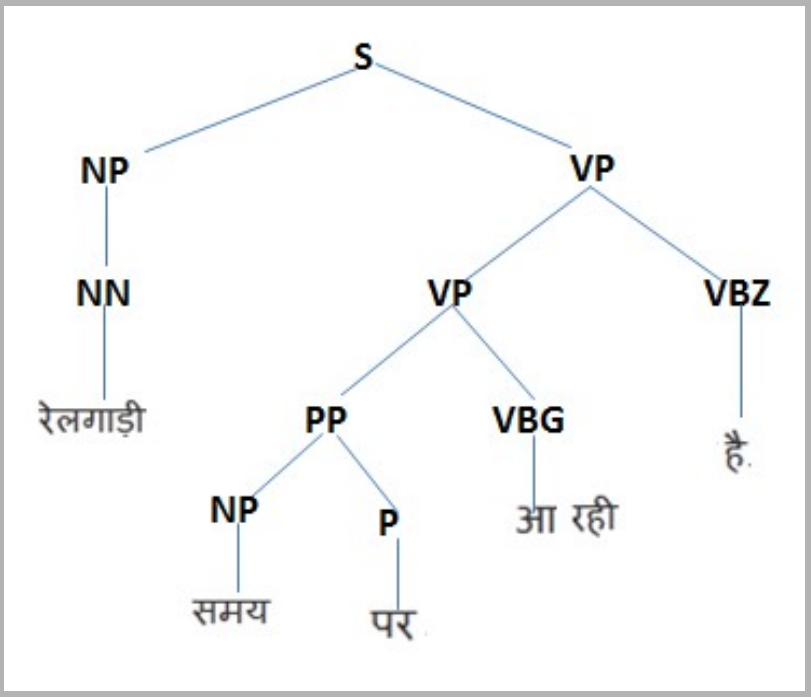


Mapping of Dependencies- Examples

- The train is arriving on time.

  DT → The is mapped to DT → NULL

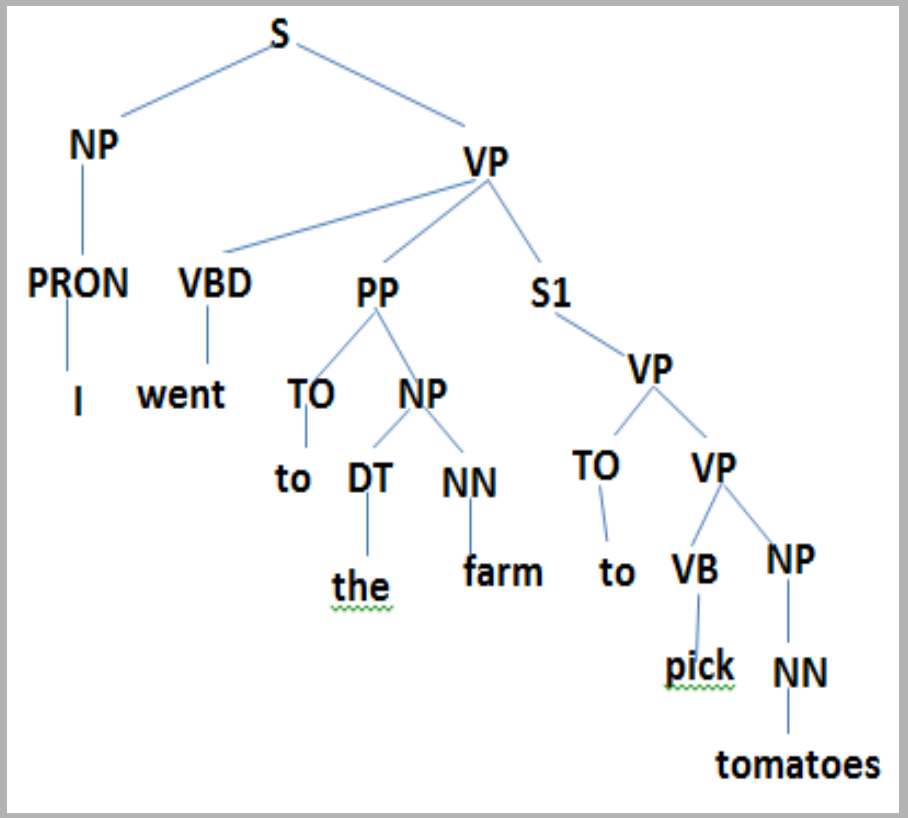


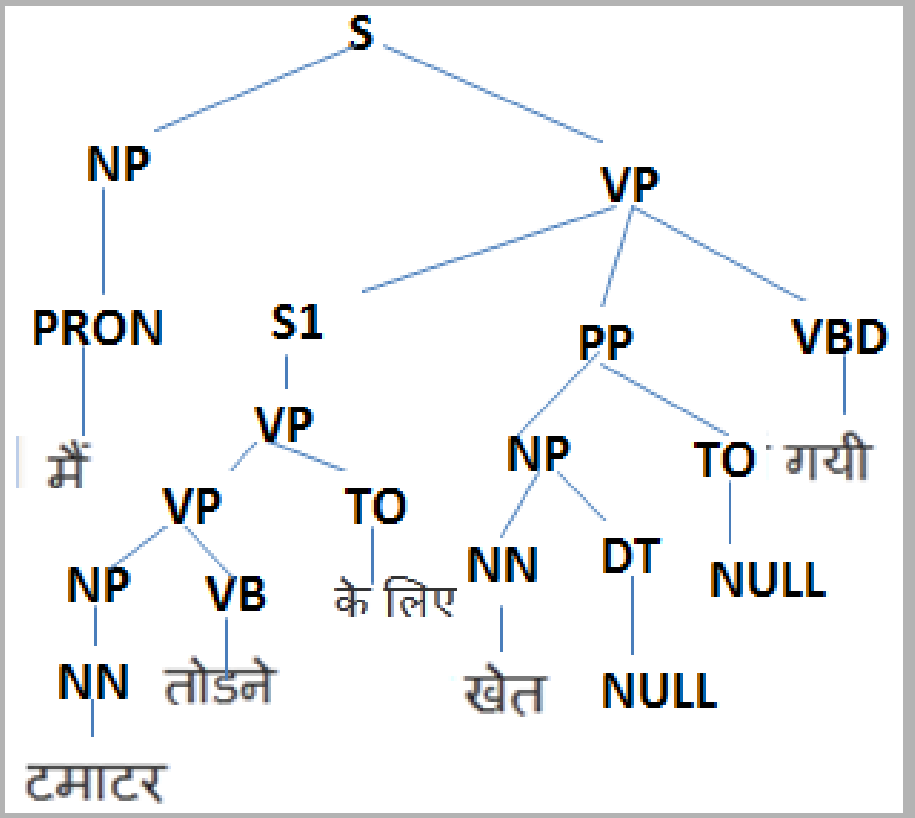


Mapping of Dependencies- Examples

- I went to the farm to pick tomatoes

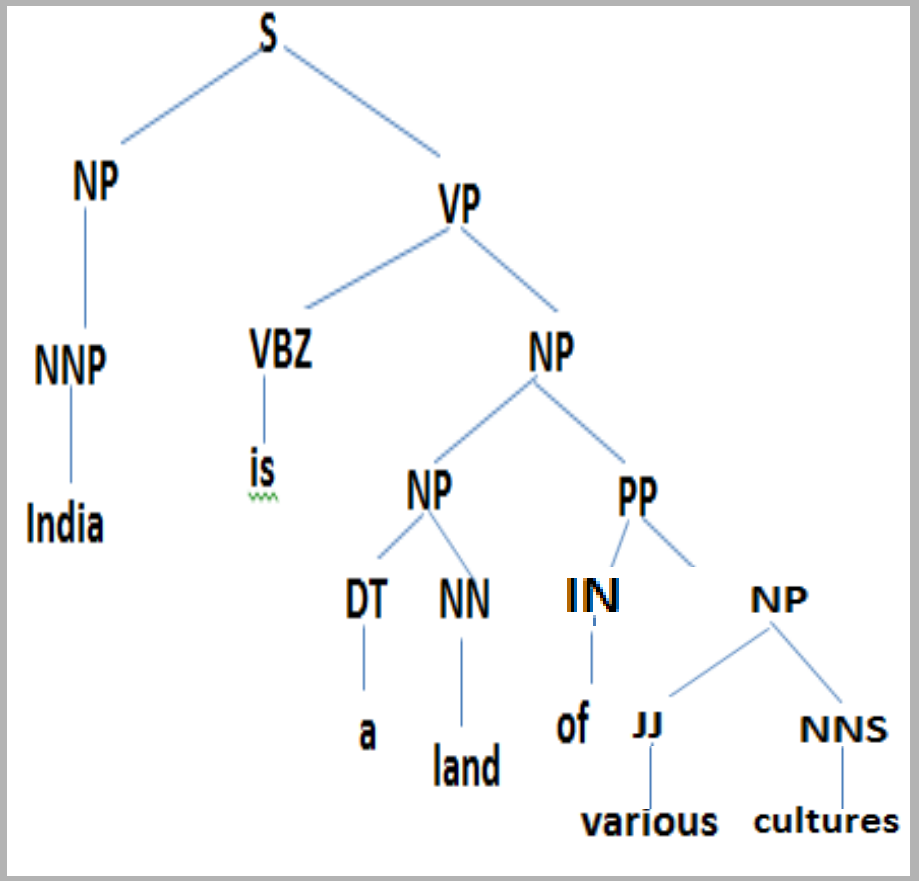
TO → to is mapped to TO → NULL & TO → के लिए

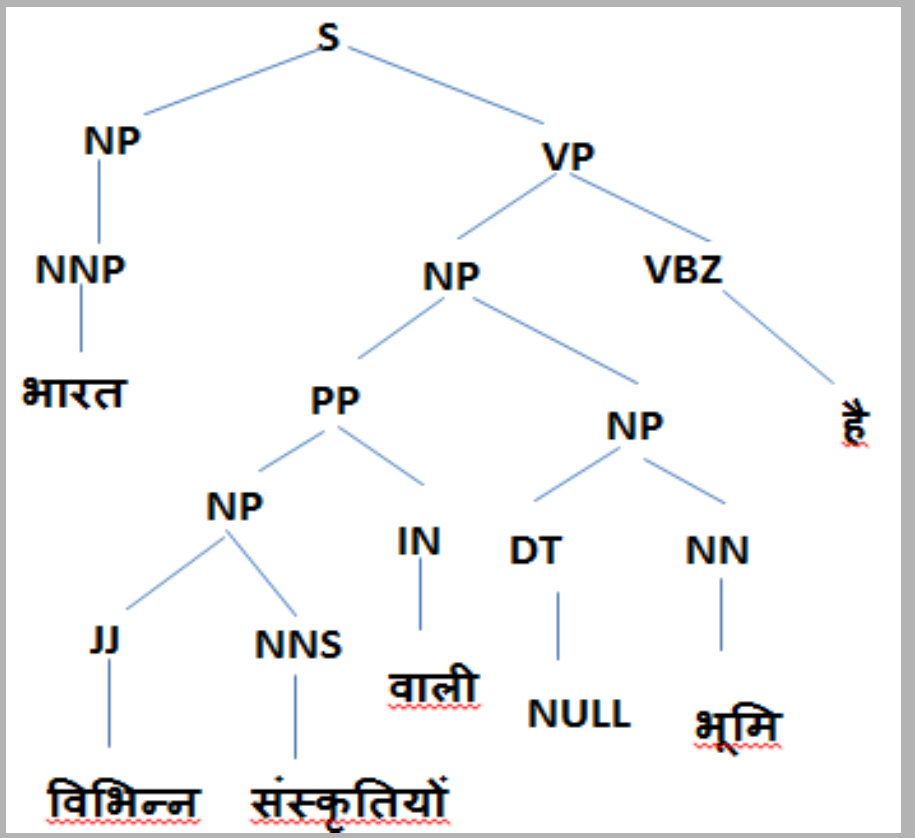




Mapping of Dependencies- Examples

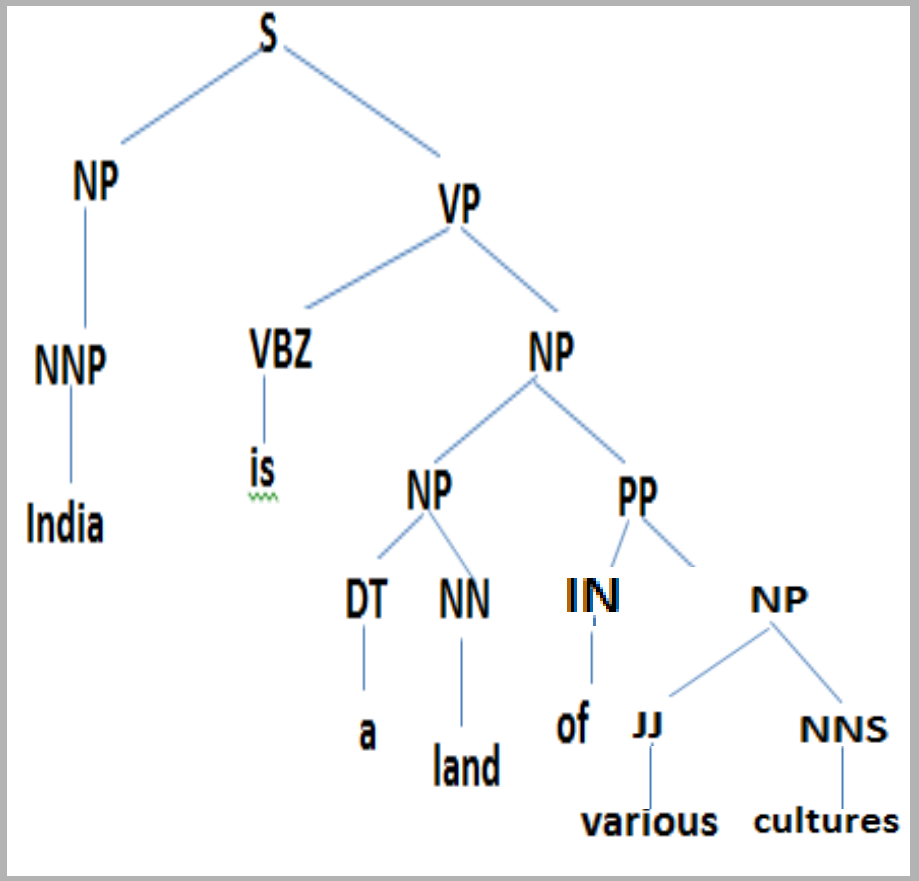
- India is a land of various cultures.

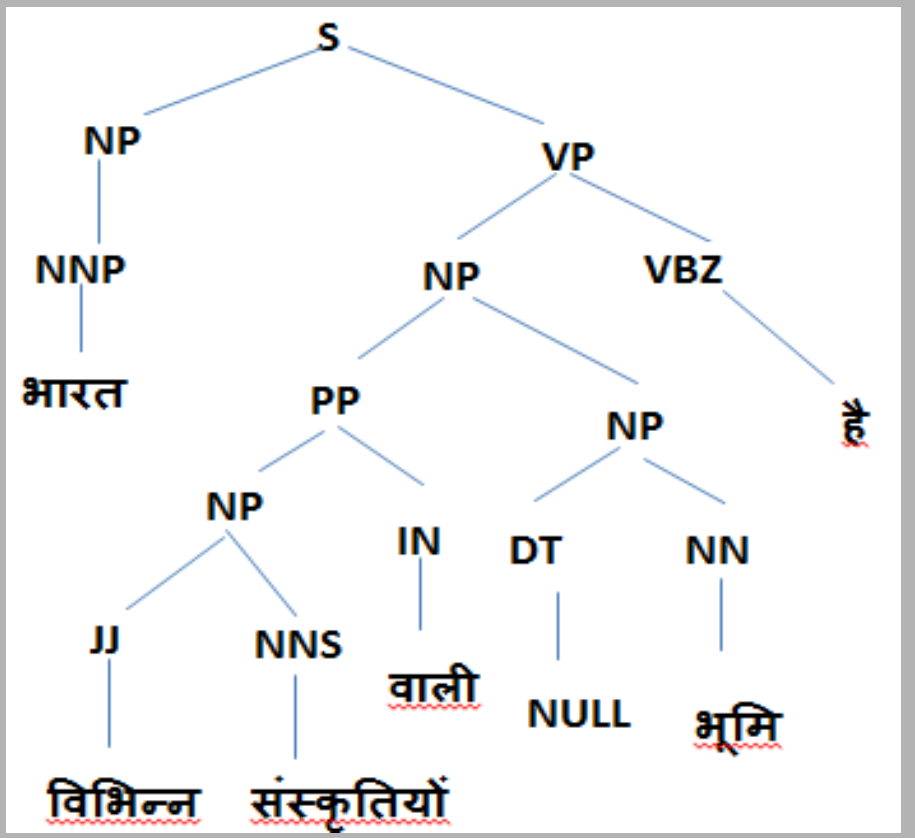




Mapping of Dependencies- Examples

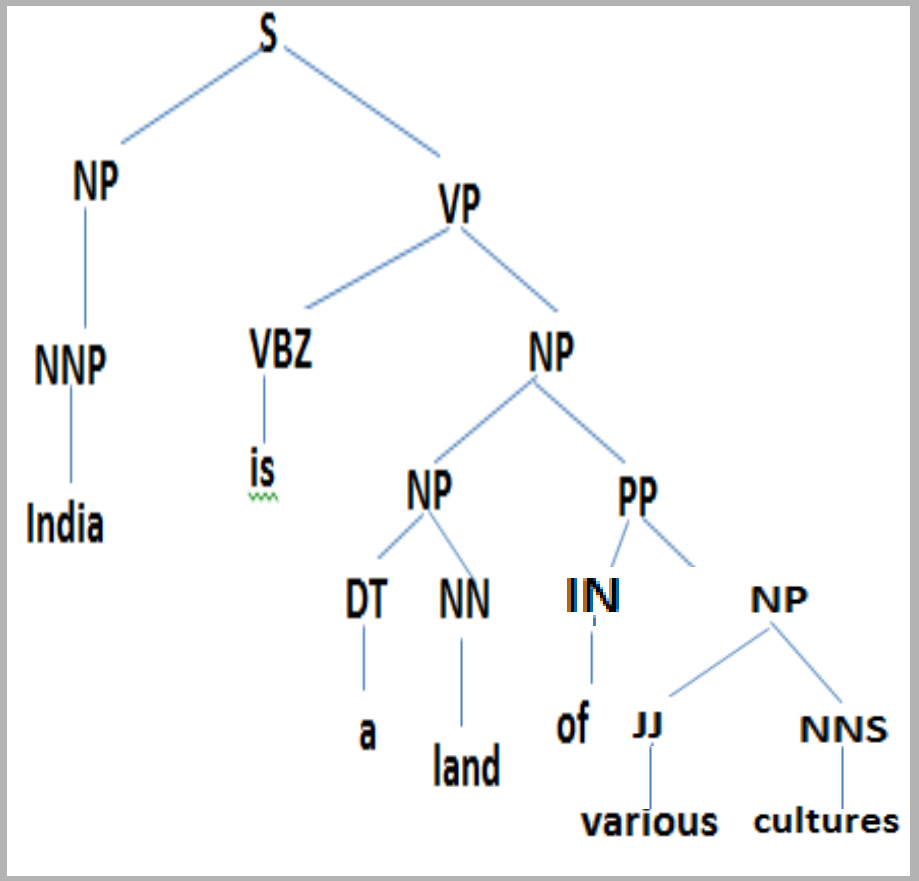
NP → JJ NN remains the same.

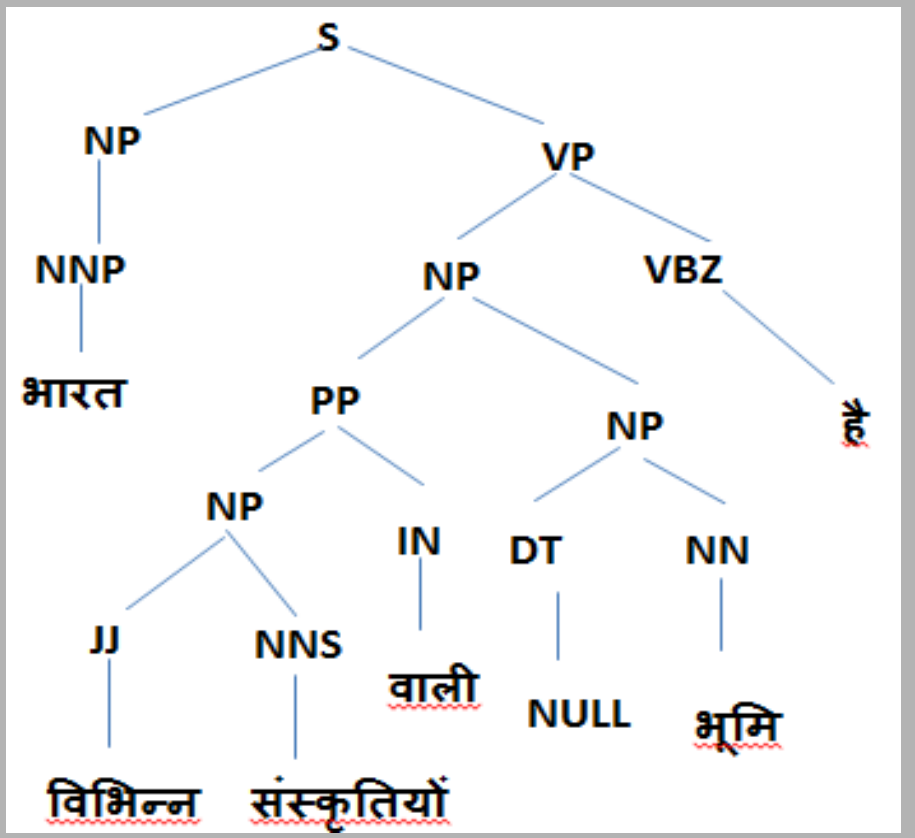




Mapping of Dependencies- Examples

NP → JJ NN remains the same.

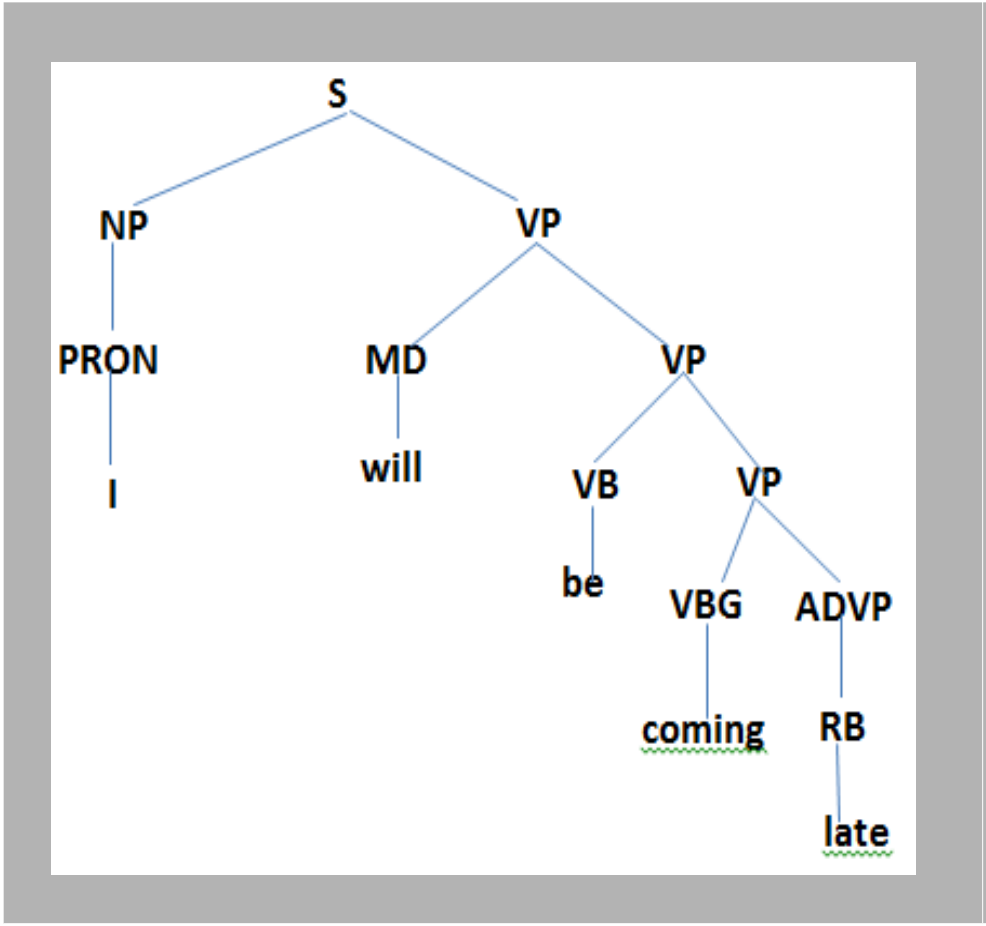


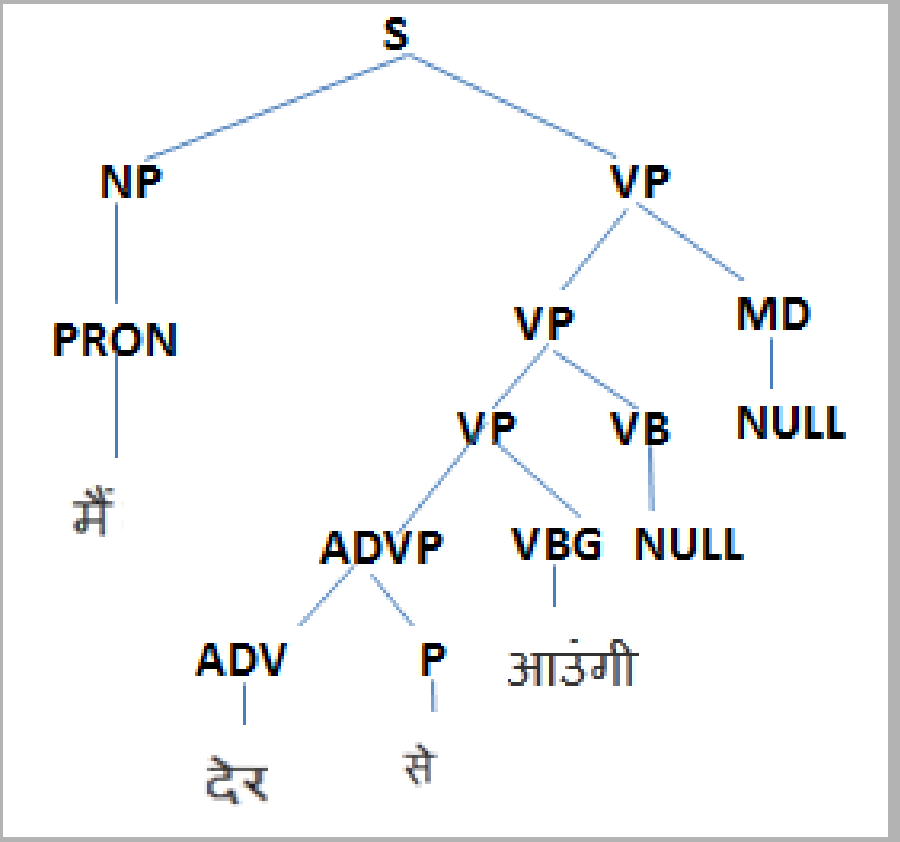


Mapping of Dependencies- Examples

- I will be coming late.

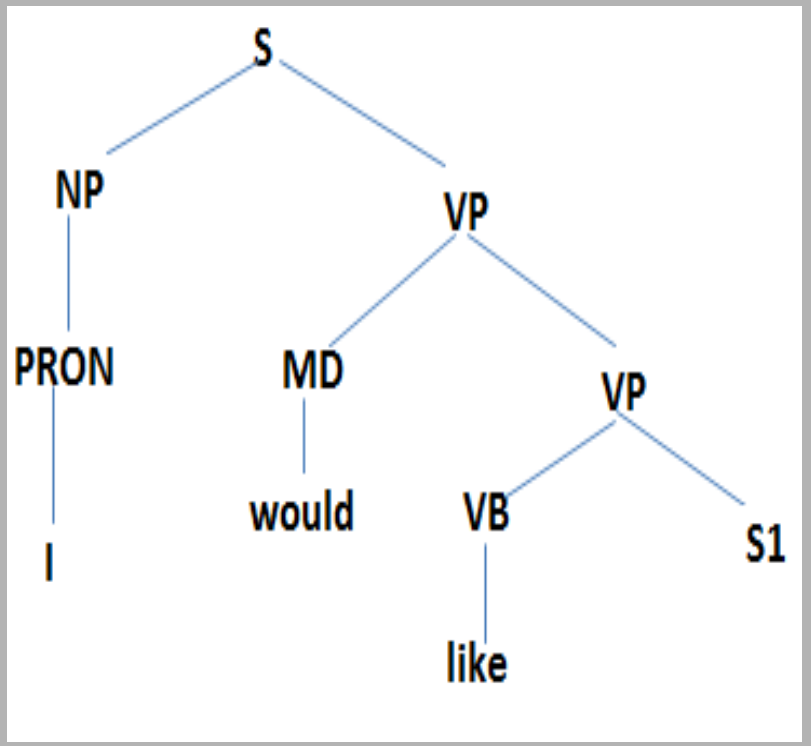
ADVP → RB becomes ADVP → ADV P

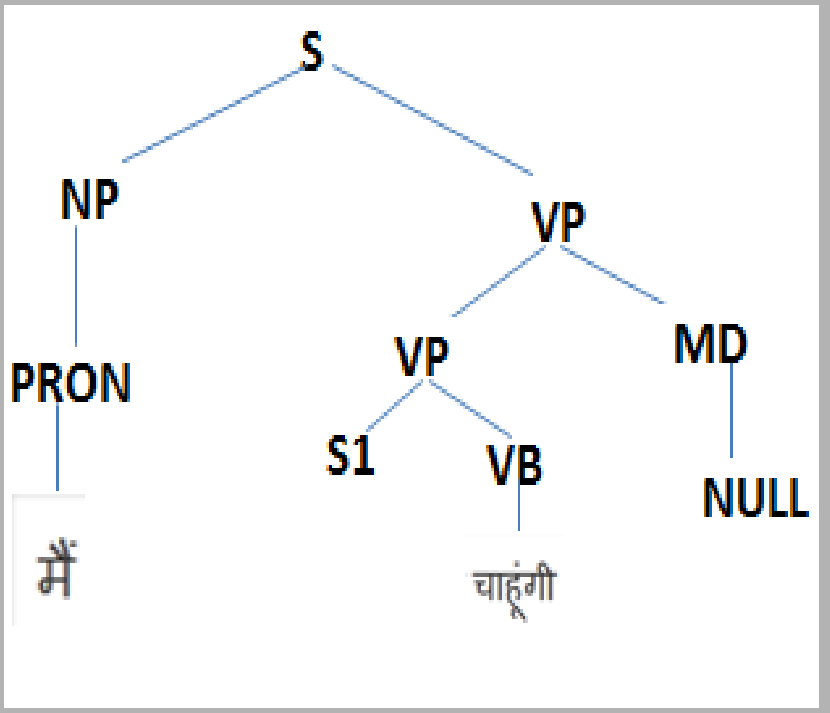




Mapping of Dependencies- Examples

- I would like to go for a walk by the beach .
  - Verb modifiers become NULL
  - MD → would becomes MD → NULL

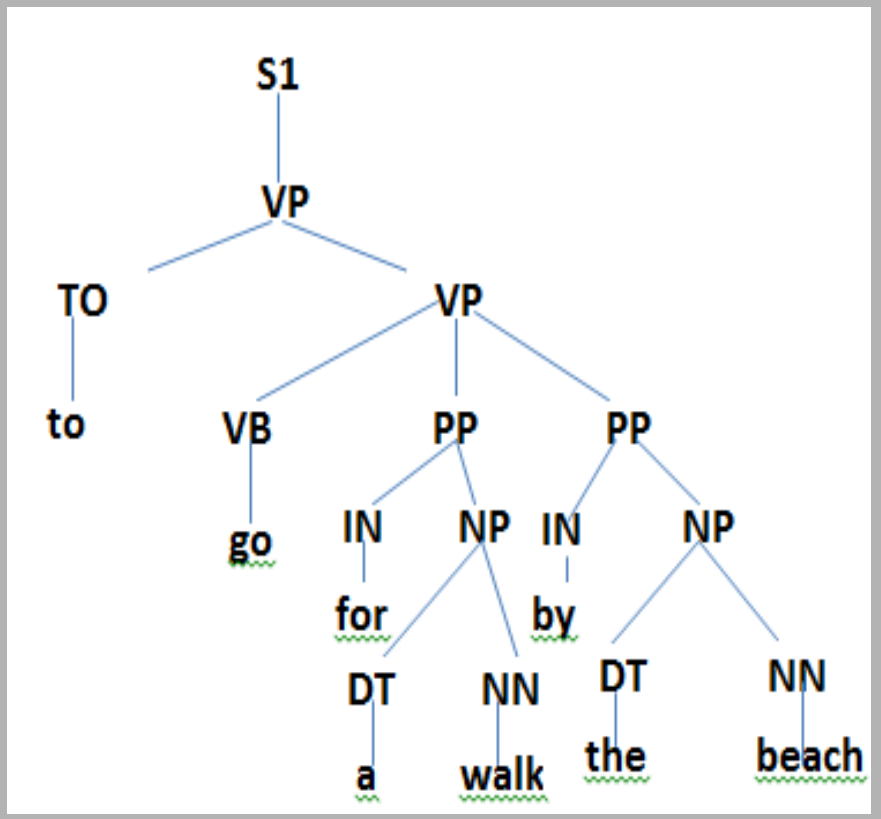


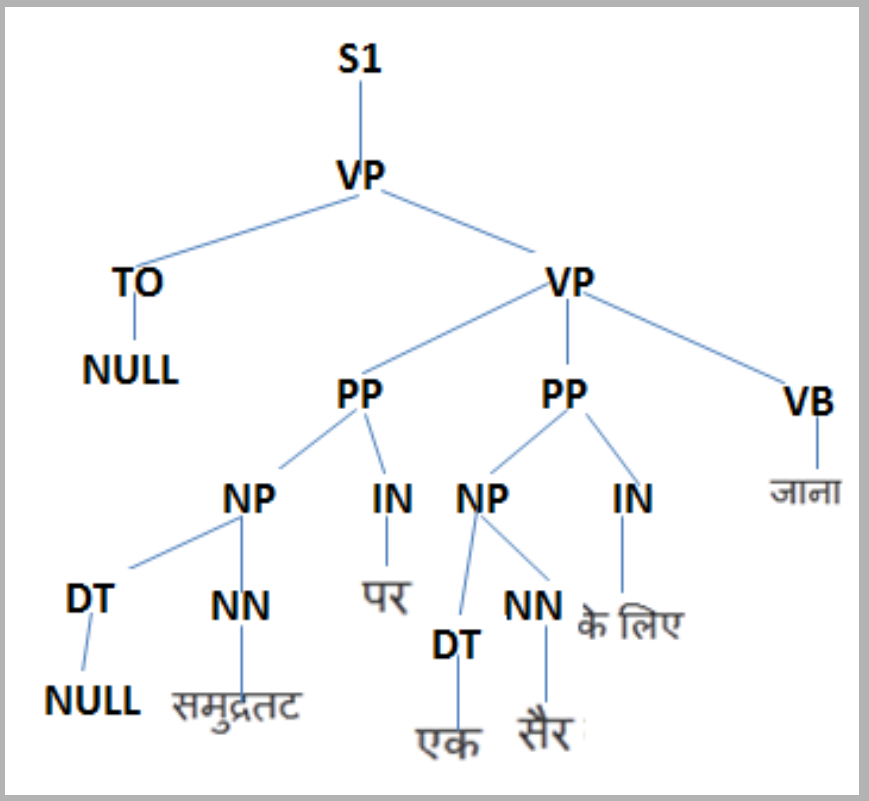


Mapping of Dependencies- Examples

- I would like to go for a walk by the beach ..

DT → a becomes DT → एक



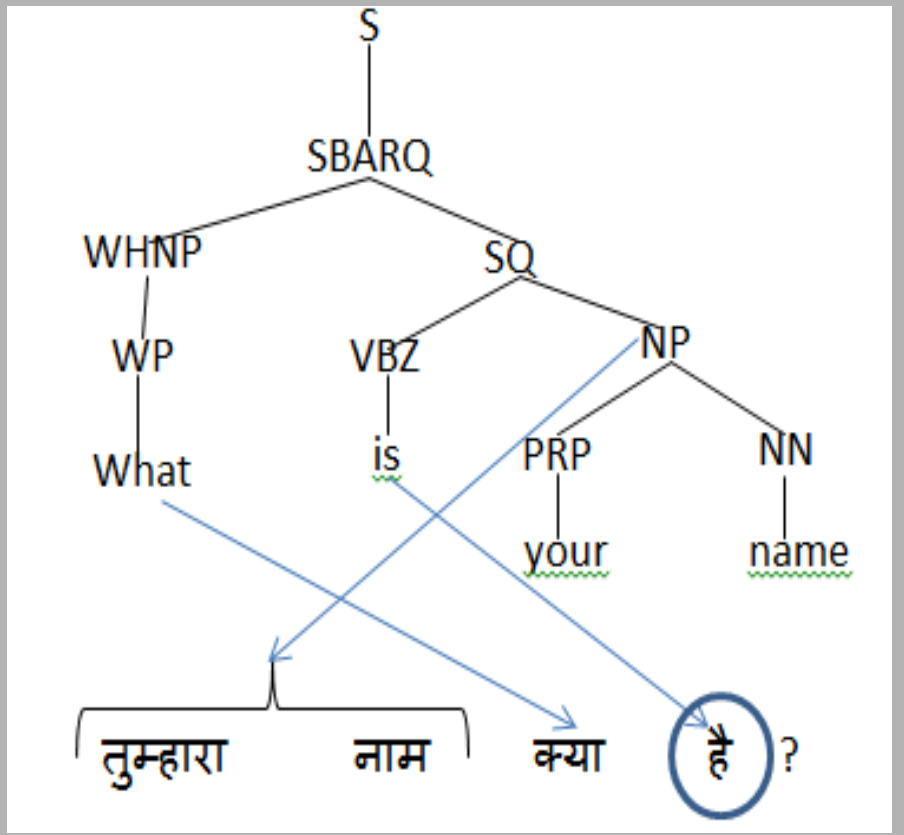


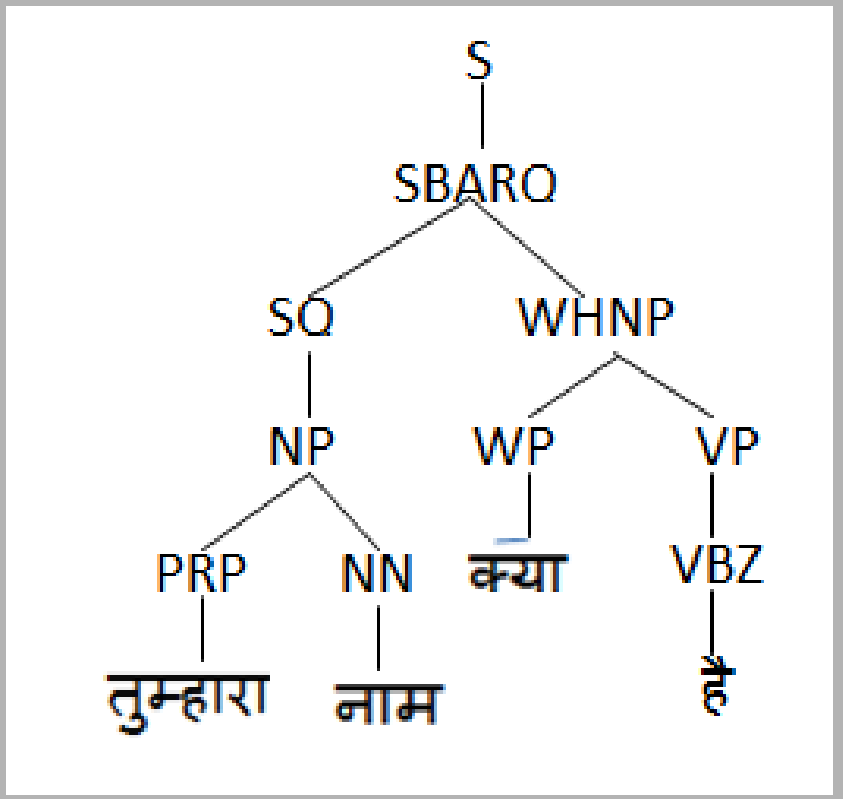
Mapping of Dependencies- Examples

- What is your name ?

  WHNP → WP becomes WHNP → WP VP

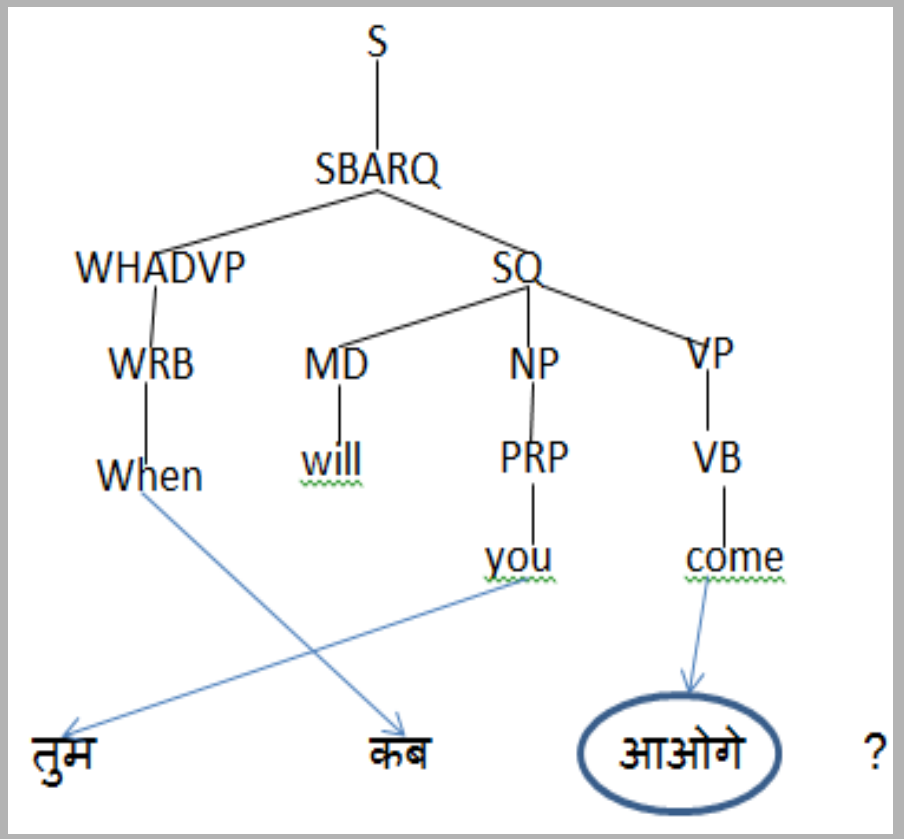
  SQ → VBZ NP becomes SQ → NP

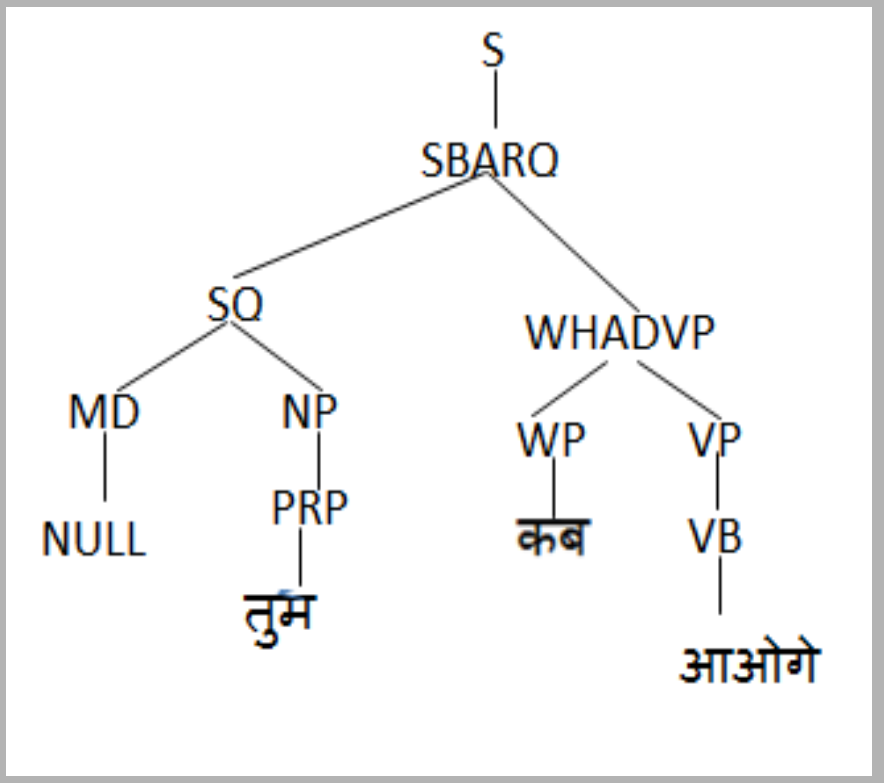




Mapping of Dependencies- Examples

- When will you come ?



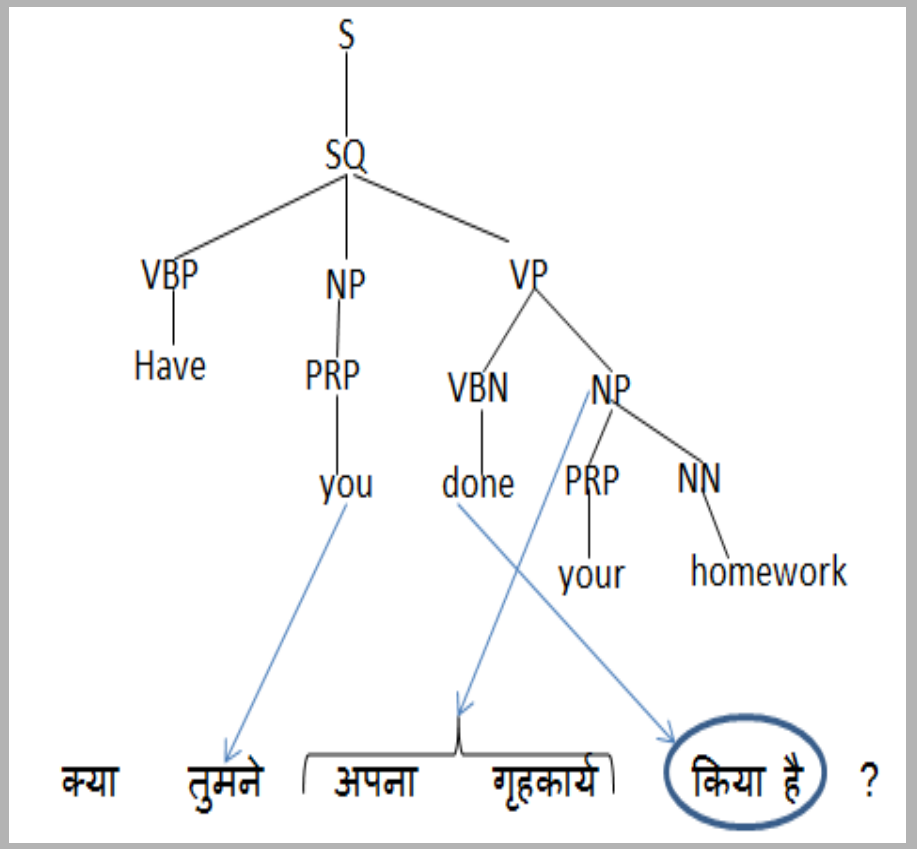


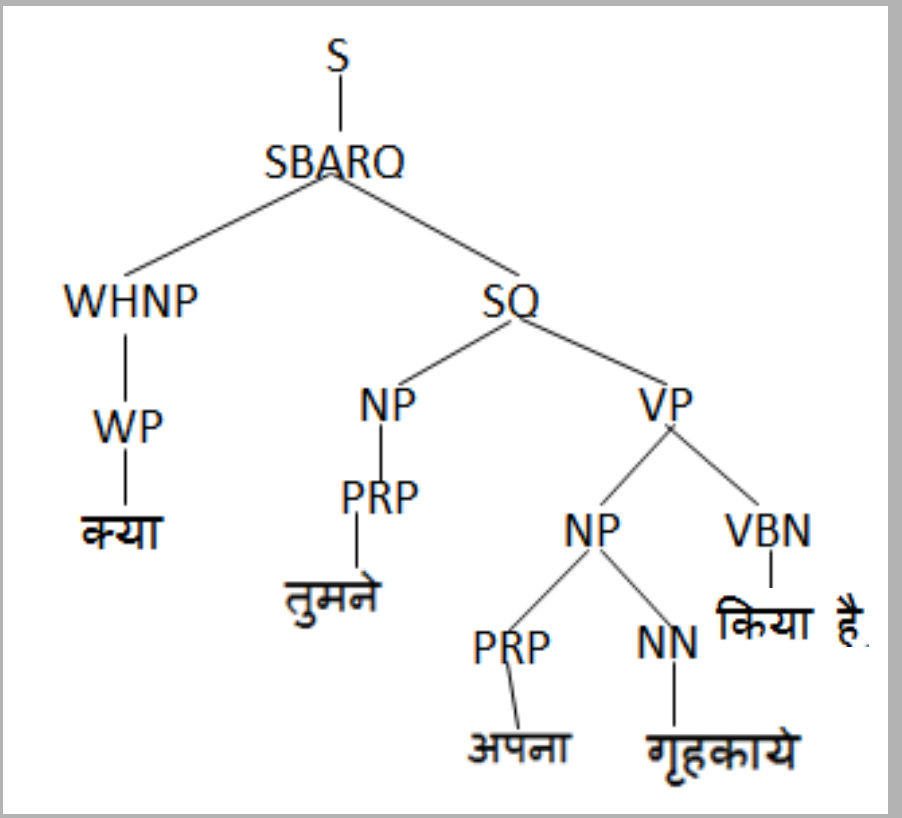
Mapping of Dependencies- Examples

- Have you done your home-work ?

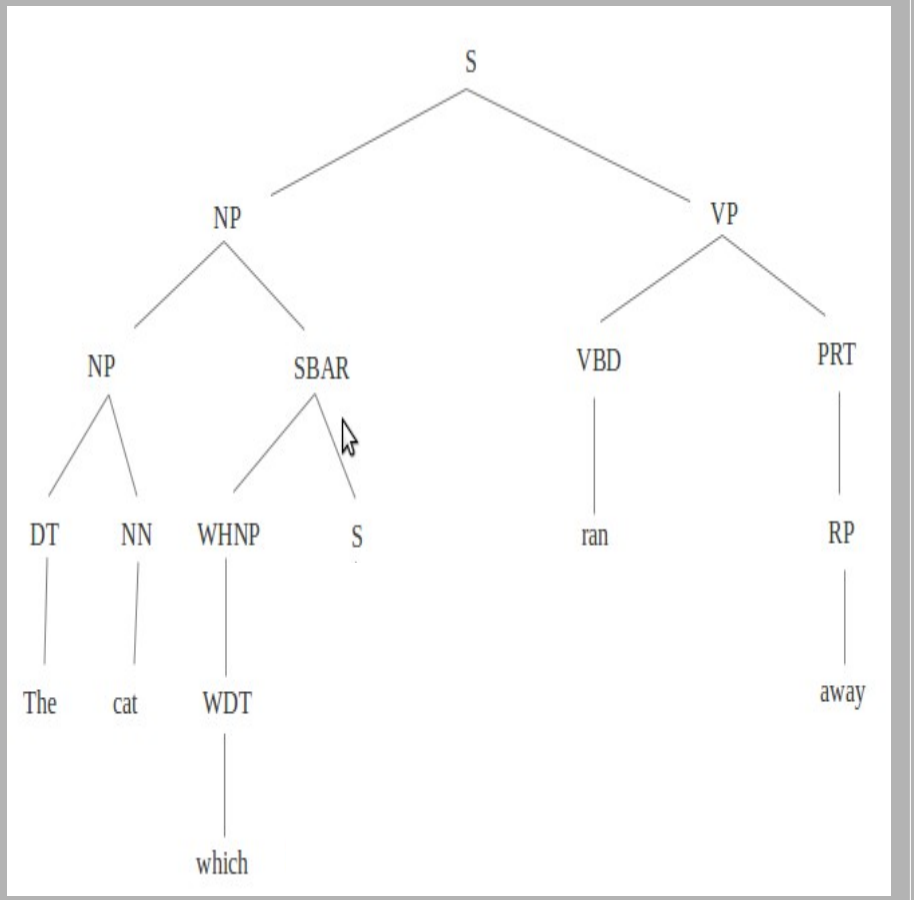
  S → SQ becomes S → SBARQ,

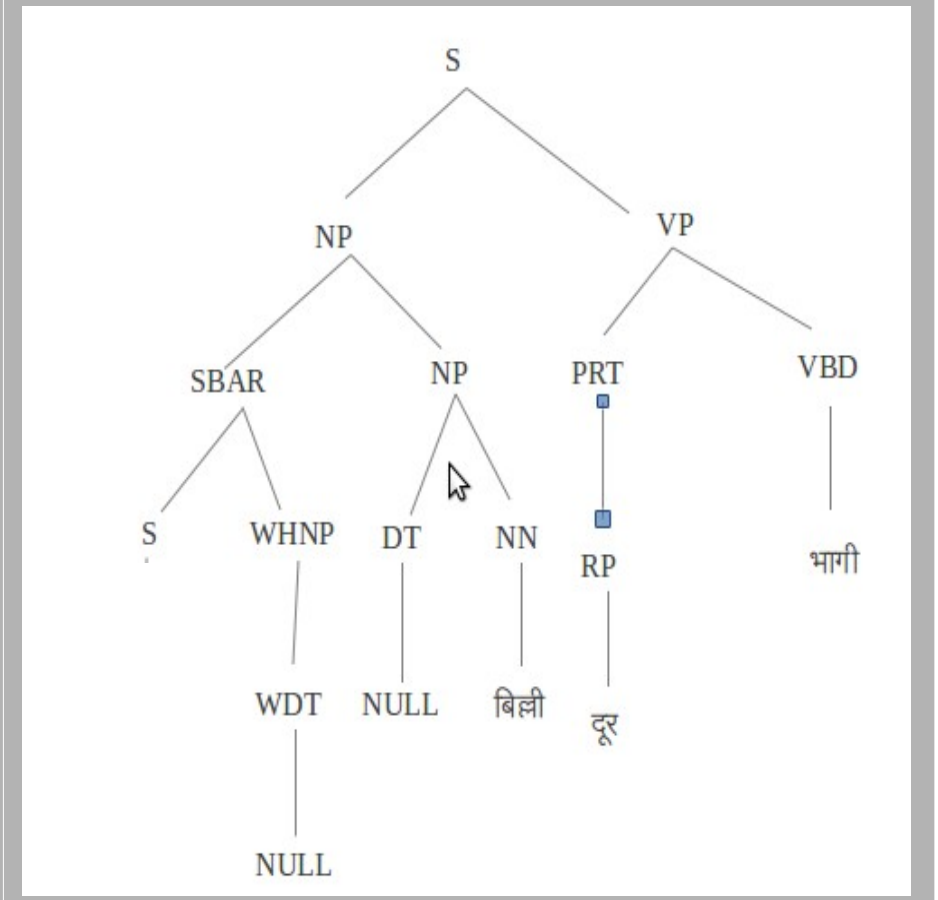
  and then SBARQ → WHNP SQ



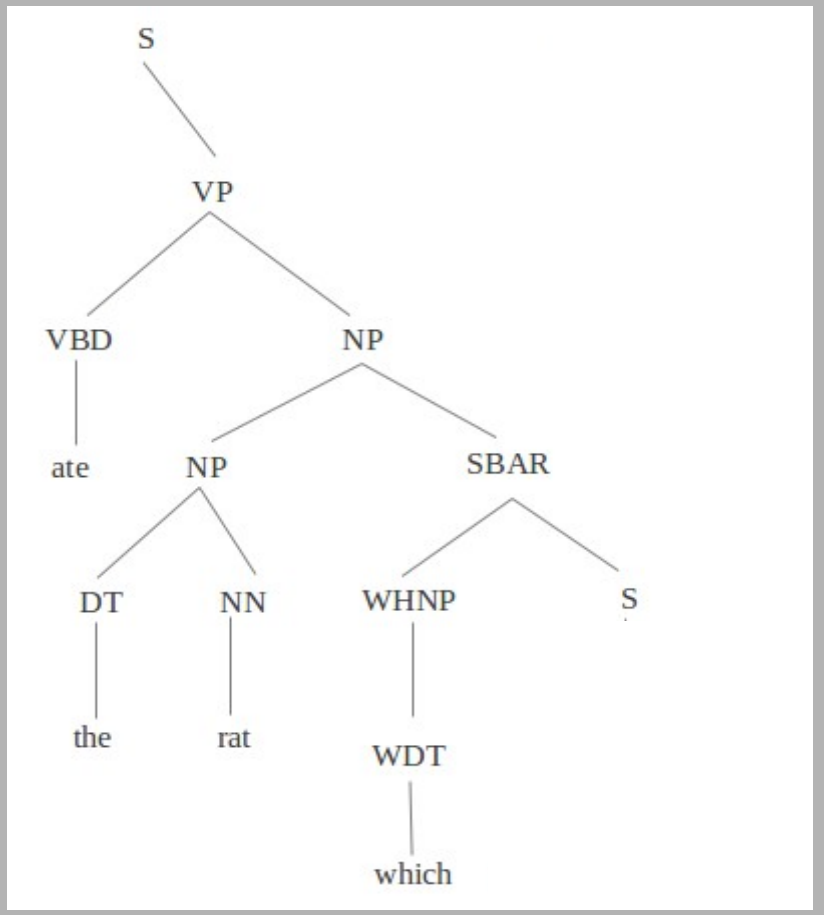


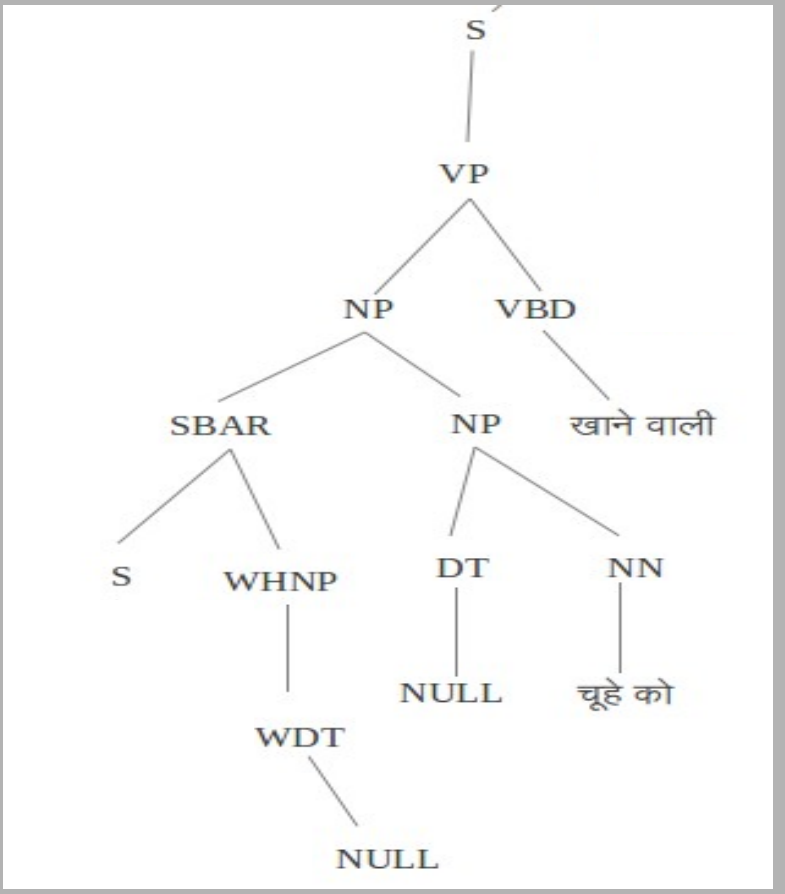
Mapping of Dependencies- Examples



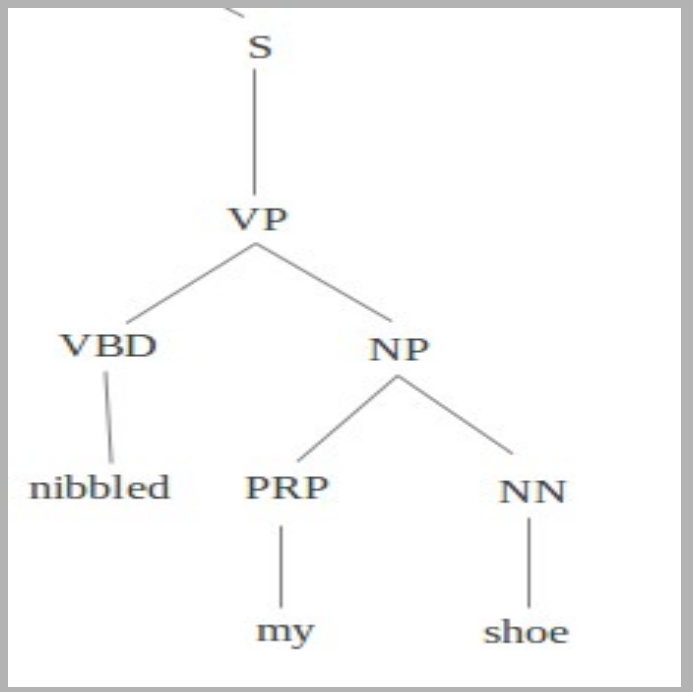


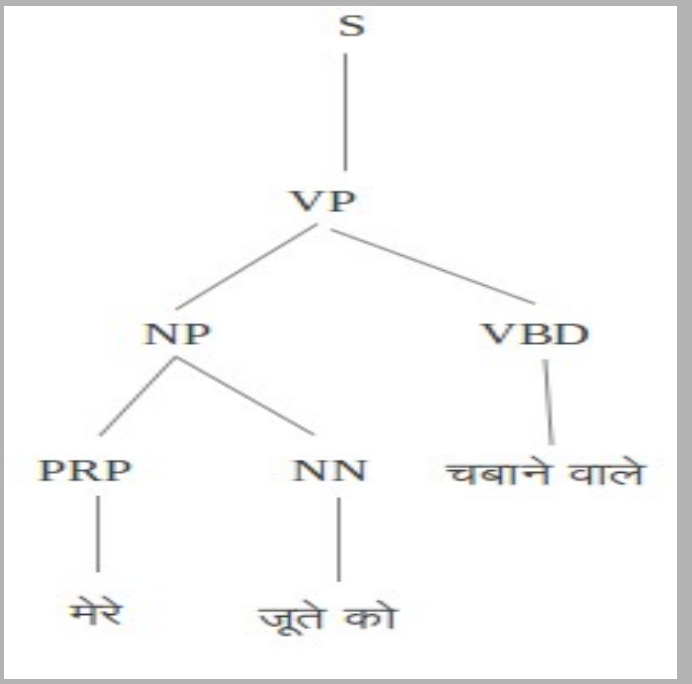
Mapping of Dependencies- Examples





Mapping of Dependencies- Examples





Part 6:

Yago Path finding

Description

- BFS traversal from the source and the target words.
- Intersection of the search spaces is performed.
- A non empty intersection set results to the end of search.
- Backtrack to find the relations in the path.

Example 1

- Given Words

  Sachin Tandulkar

  Lata Mangeshkar

Example 1

- Sachin Tendulkar:

  <wasBornIn> Mumbai

  <isLeaderOf> Suthaliya

  <hasFamilyName> Tendulkar …….....

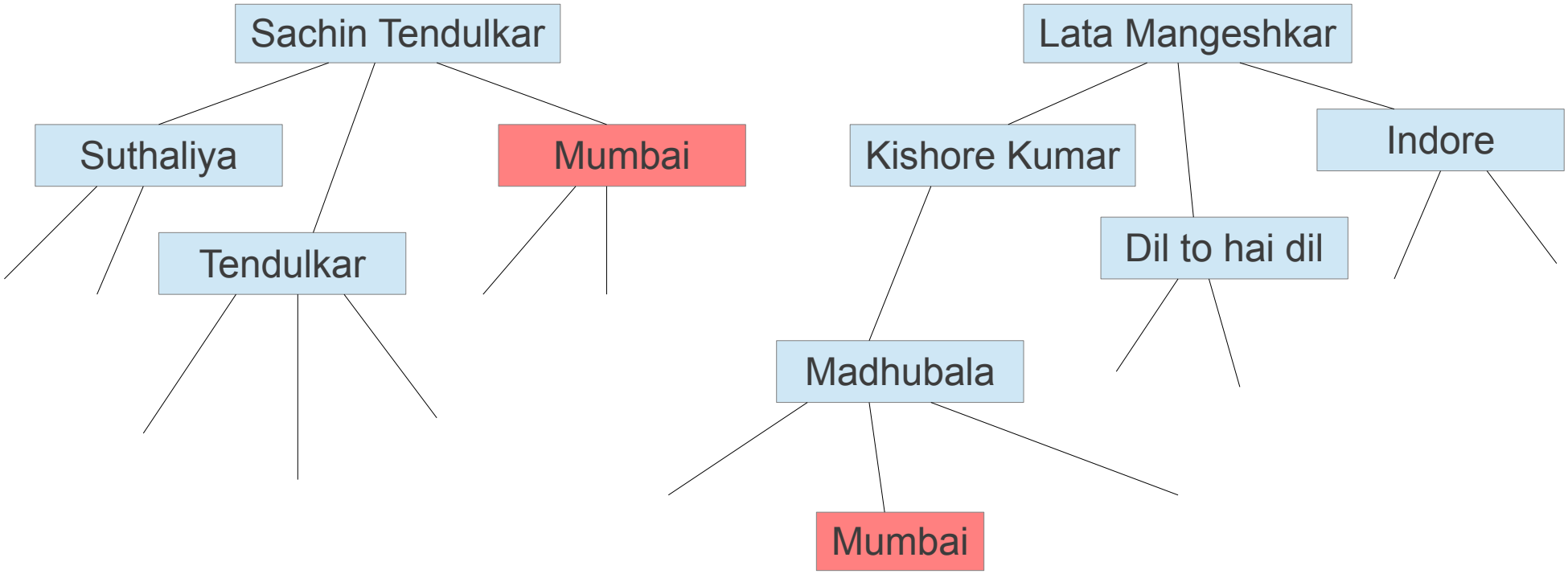
- Lata Mangeshkar:

  <linksTo> Kishore Kumar

  <wasBornIn> Indore

  <created> Dil to hai dil …….....

Example 1


Sachin Tendulkar
Suthaliya
Tendulkar
Mumbai
Lata Mangeshkar
Kishore Kumar
Madhubala
Mumbai
Dil to hai dil
Indore

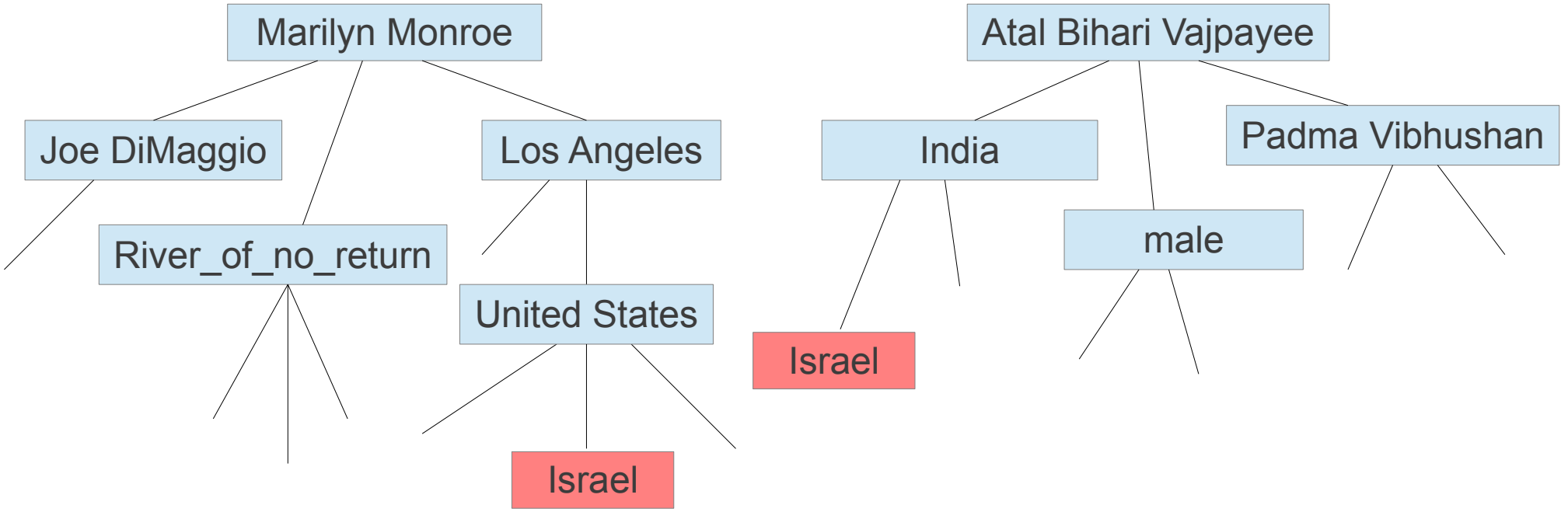
Example 2

- Given Words:
  - Marilyn Monroe
  - Atal Bihari vajpayee

Example 2

- Marilyn Monroe expands to:
  - <IsMarriedTo> Joe DiMaggio
  - <wasBornIn> Los Angeles
  - <actedIn> River_of_no_return
- Atal Bihari vajpayee expands to:
  - India <LinksTo>
  - <hasGender> Male
  - <hasWonPrize> Padma_Vibhushan

Example 2


Marilyn Monroe
Joe DiMaggio
River_of_no_return
Los Angeles
United States
Israel
Atal Bihari Vajpayee
India
Israel
male
Padma Vibhushan

Part 7:
NLTK

NLTK

- Natural Language Tool Kit
- Platform for building Python programs to work with human language data.

Category of words

- Morphological Clues
  - -ment, -ness produces Nouns

    happy → happiness, govern → government
  - present participle of a verb ends in -ing etc
- Syntactic Clues

  An adjective in English is that it can occur immediately before a noun, or immediately following the words be or very.

  a. the near window

  b. The end is (very) near.
- Semantic Clues
  - The best-known definition of a noun is semantic: "the name of a person, place or thing"
  - Semantic criteria for word classes are treated with suspicion, mainly because they are hard to formalize

Morphological Analysis

- Go away!
- He sometimes goes to the cafe.
- All the cakes have gone.
- We went on the excursion.

  Four distinct grammatical forms, all tagged as VB in some tagset.

  Brown Corpus however, tag them as VB (go), VBZ (goes), VBN (gone), VBG (going), VBD (went)

PoS Tagging

- Various tagging models employed:
  - Default Tagger
  - Regular Expression Tagger
  - Unigram Tagger
  - N-Gram Tagger
- Techniques for combining models:
  - Back-off
  - Cut-off
- Data-Structures available:
  - Dictionaries

    pos = {'furiously': 'adv', 'ideas': 'n', 'colorless': 'adj'}.

Chunking

- NLTK provides functions for creating a grammar for purpose of chunking.
- The chunking done in NLTK is bottom up that is done on POS-tagged sentences.

Thank You