Anaphora Resolution for Indian Languages

- V. A. Lalitha Kameswari 20171025

Selected Papers

1. Rule Based Anaphora Resolution in Hindi

D.Singla and P. Kumar (2017)

International Conference on Computational Intelligence in Data Science (ICCIDS)

2. Resolving Pronominal Anaphora in Hindi Using Hobbs' Algorithm

Dutta, Kamlesh & Prakash, Nupur & Kaushik, Saroj. (2019)

3. A Generic Anaphora Resolution Engine for Indian Languages

Devi, Sobha Lalitha, R. Vijay Sundar Ram and Pattabhi R. K. Rao (2014).

International Conference on Computational Linguistics (COLING)

“

***Anaphora** is a discourse level phenomenon whereby the interpretation of an occurrence of one expression depends on the interpretation of an occurrence of another.*

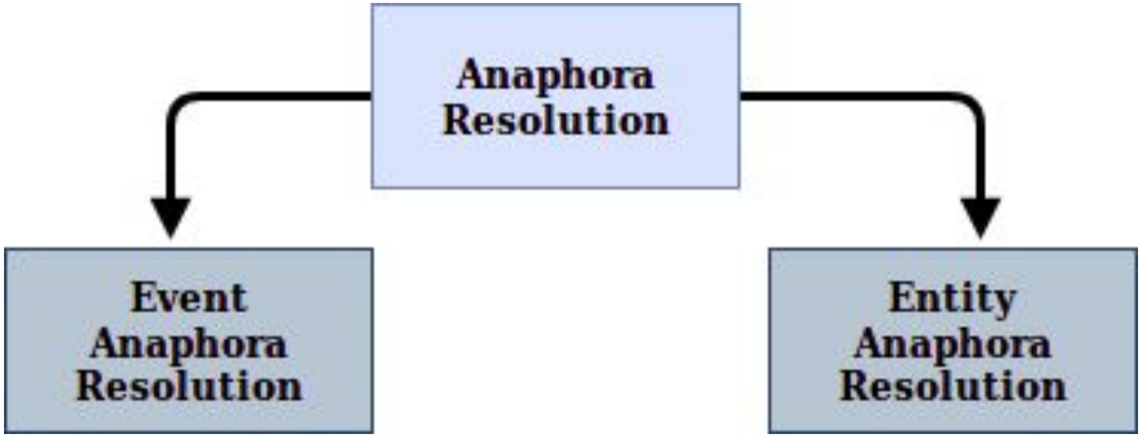
It is used to derive the “Correct interpretation” of the text.

*The referring expression is called **anaphor** and referred expression is called the **antecedent.***

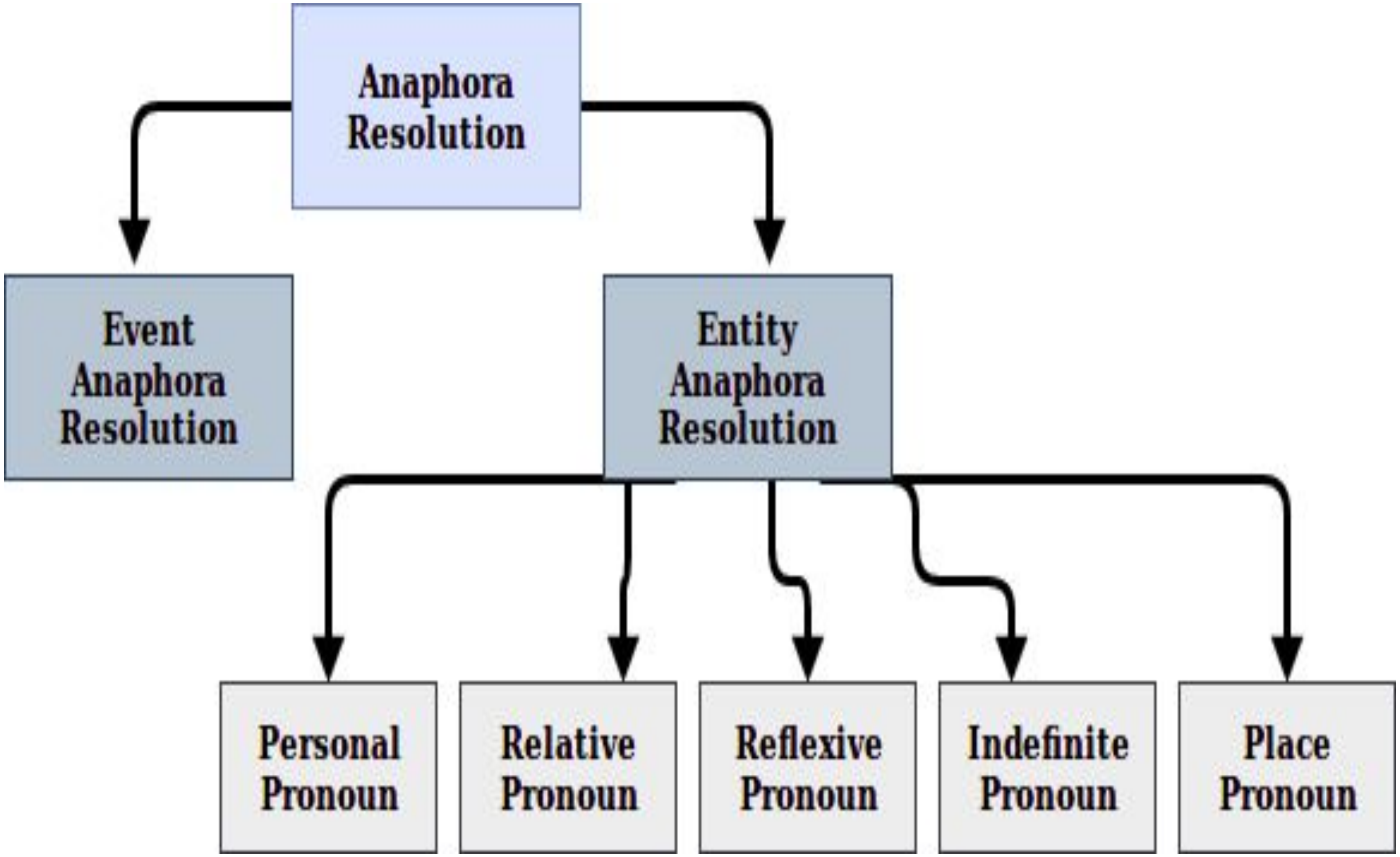
*The process of identification of referent is known as **Anaphora Resolution**.*

1.

Rule Based Anaphora Resolution in Hindi



- **Entity anaphora** stands for those pronominal references which refer to a Concrete Entity such as person, place and other common nouns.
- **Event anaphora** stands for those pronominal references which refer to an abstract object.



Anaphora Resolution

Event Anaphora Resolution

Entity Anaphora Resolution

Personal Pronoun

Relative Pronoun

Reflexive Pronoun

Indefinite Pronoun

Place Pronoun

Architecture of the system

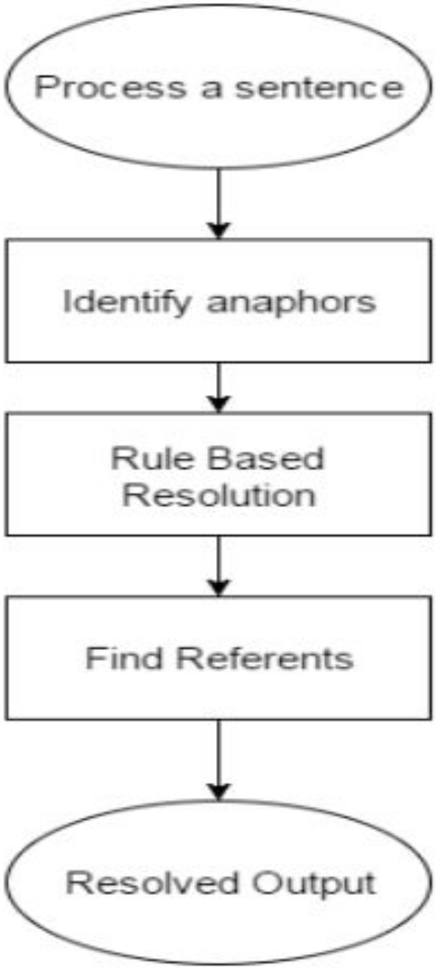


Fig.1. Process Workflow

COMPUTATIONAL PANINIAN GRAMMAR (CPG) AND DEPENDENCY

- The dependency structure is obtained by representation of one-to-one correspondence between the words in a sentence, based on the head-modifier relationship.
- A rule based approach has been used to study and analyze patterns in the CPG based dependency structure that is used to formulate rules to resolve references.

Karaka Relations used in Dependency

- Karta – agent/doer/force
  Relation label – **k1**
- Karma – object/patient
  Relation label – **k2**
- Karana – instrument
  Relation label – **k3**
- Sampradaan – beneficiary
  Relation label – **k4**
- Apaadaan – source
  Relation label – **k5**
- Adhikarana – location in place/time/other
  Relation label – **k7p/k7t/k7**

Rule 1: Reflexive Pronouns

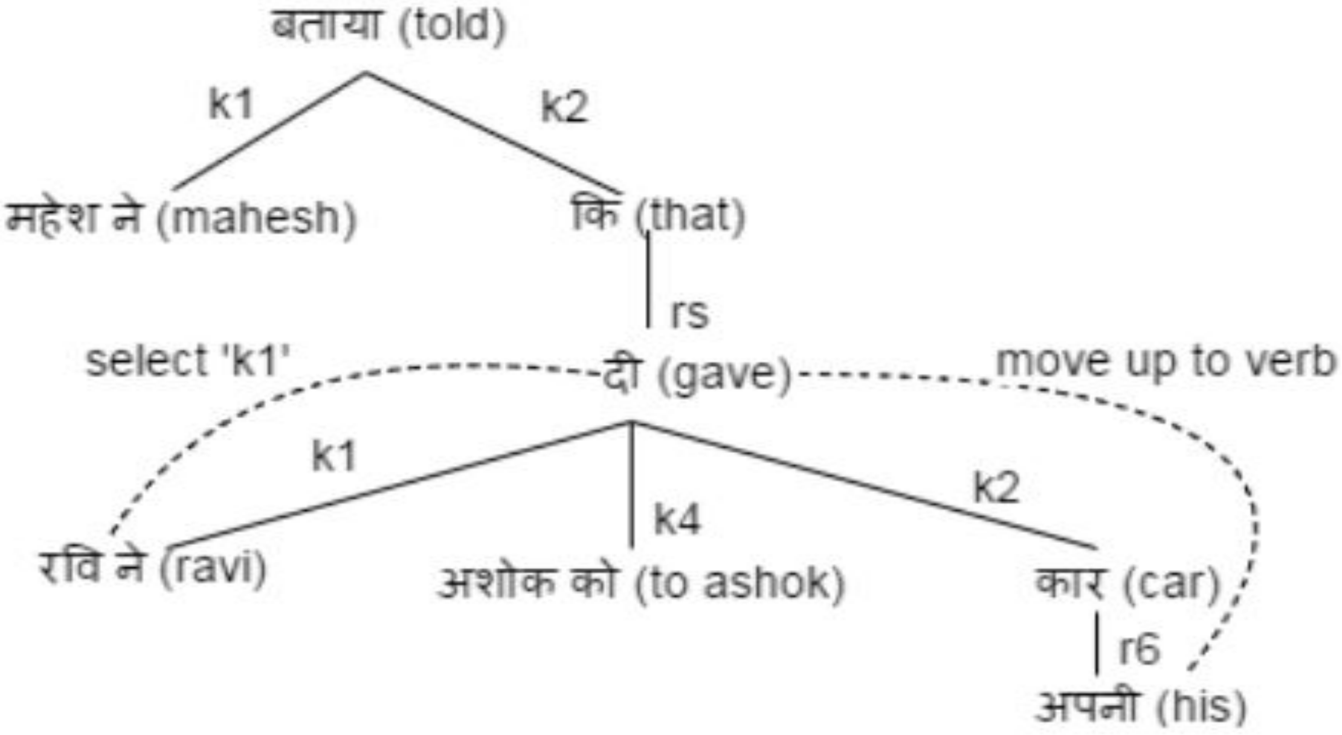
- ❑ In Hindi, frequent types of reflexives are the possessive reflexives

  i.e. apnA / apnI (‘own’) etc.

- ❑ The referent of the possessive reflexives is the possessor entity, which is the Karta of the clause or sentence.

“महेश ने बताया कि रवि ने अशोक को अपनी कार ले ली है ।“ (3)

(mahesh told that ravi gave his car to ashok.) (4)

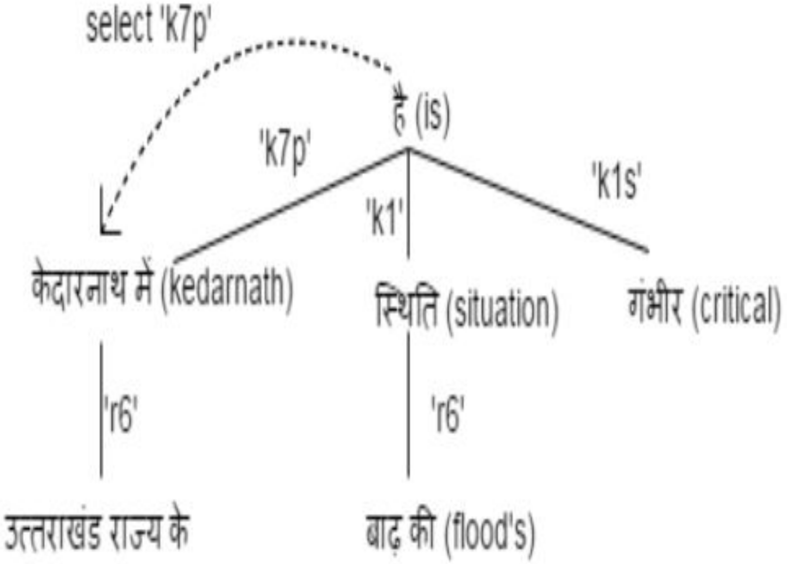


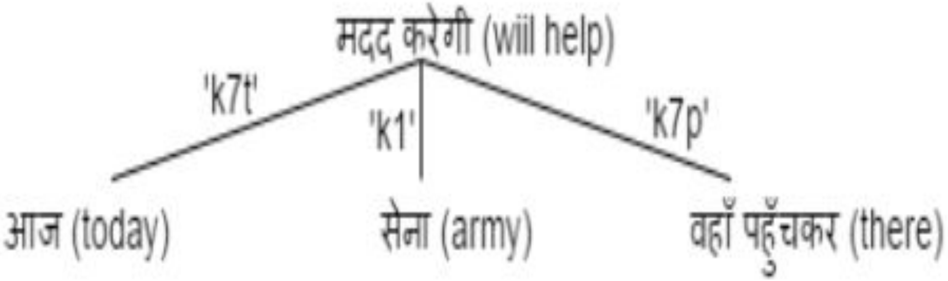
Rule 2: Spatial Pronouns

- Refer to places - e.g: yahA, wahA, etc.
- The rule is to identify the noun phrases representing ‘places’ (k7p) and choose the most probable among them.

"उत्तराखंड राज्य के केदारनाथ में बाढ़ की वजह से स्थिति गंभीर है । आज सेना वहाँ पहुँचकर मदद करेगी ।" (5)

(Situation of flood is critical in kedarnath of uttrakhand. Today army will be visiting there to help) (6)



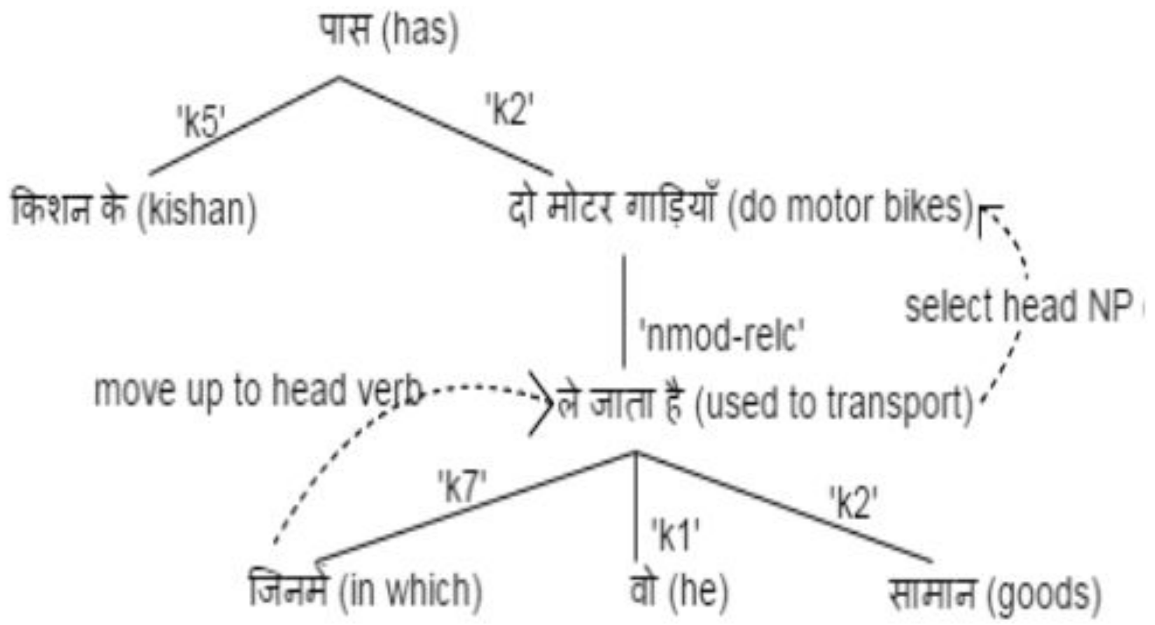


Rule 3: Relative Pronouns

- ❑ A relative clause is a kind of subordinate clause which specifies an element, usually an NP in the main clause.
- ❑ The referent of the relative pronoun is the NP which is head of the root verb of the relative clause.

“किशन के पास दो मोटर गाड़ियाँ हैं जिनमें वो समान भरकर इधर से उधर ले जाता है ।” (7)

(Kishan has two motor bikes in which he used to transport goods from one place to another.) (8)



Rule 4: First and Second Person Pronouns

- First and second pronoun usually refers directly to speaker and listener of a communication.
- First pronoun include ˜mæ (‘I’), ham˜ (‘We’) and their inflected forms.
- Second person pronoun include tU(‘You’), tum˜ (‘You’), Aap”(‘You’) along with their inflected forms.

“प्रधानमंत्री ने जनता से कहा कि आप मेरा मत्त स्वीकार करें।” (11)

(Prime Minister asked people to accept his viewpoint.) (12)

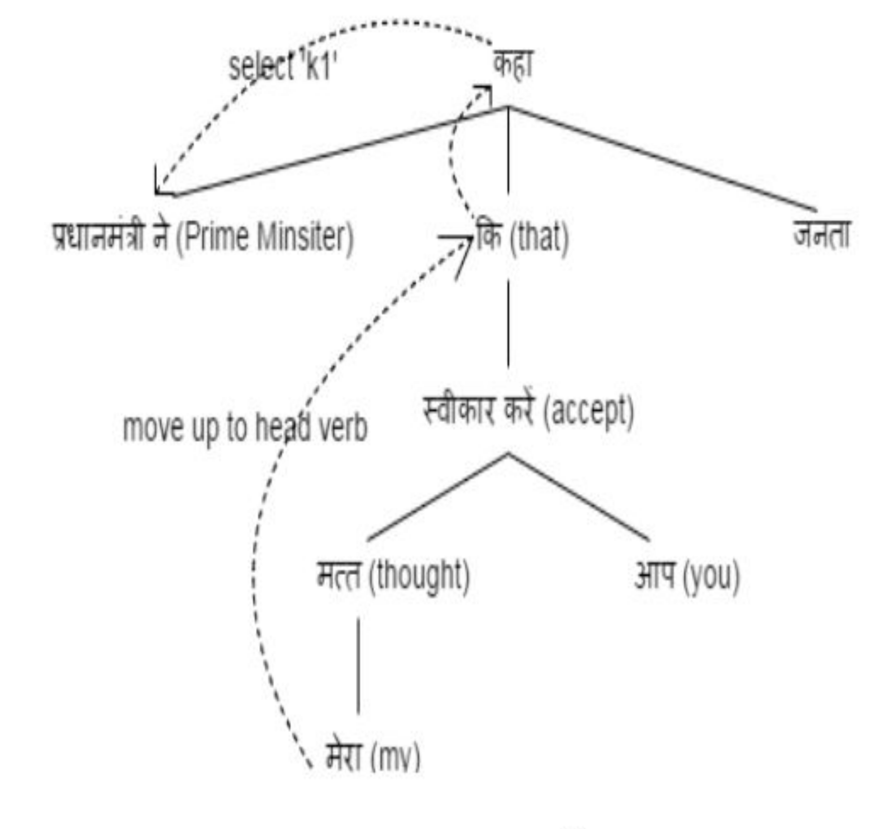


Fig. 6. Dependency structure for मेरा (‘my’)

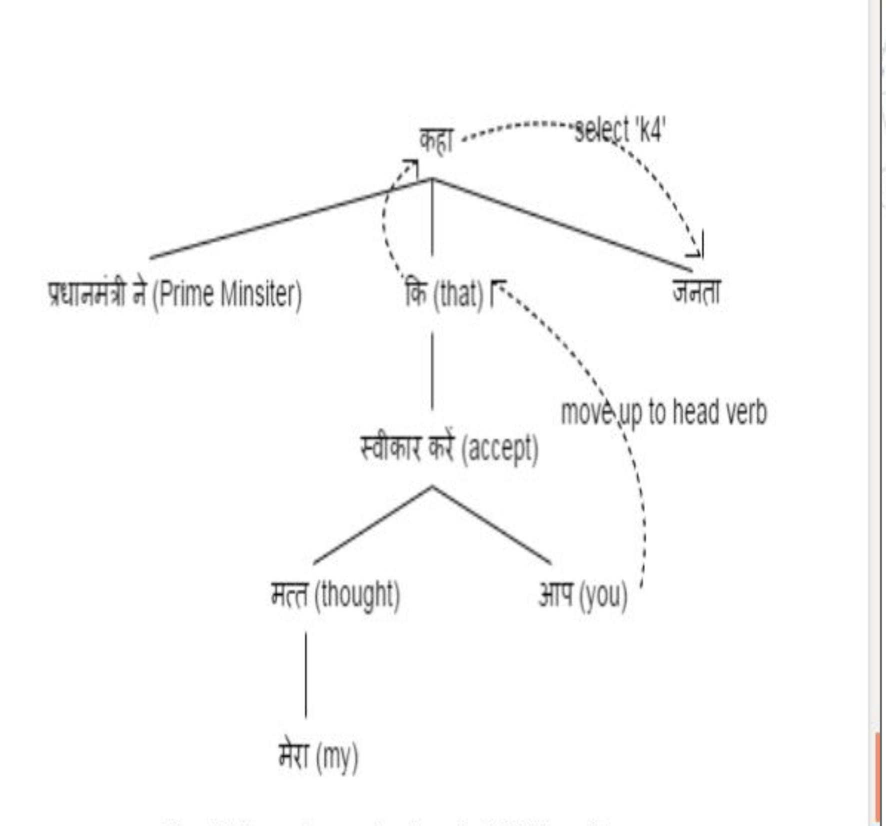


Fig. 7. Dependency structure for आप (‘you’)

2.

A Generic Anaphora Resolution Engine for Indian Languages

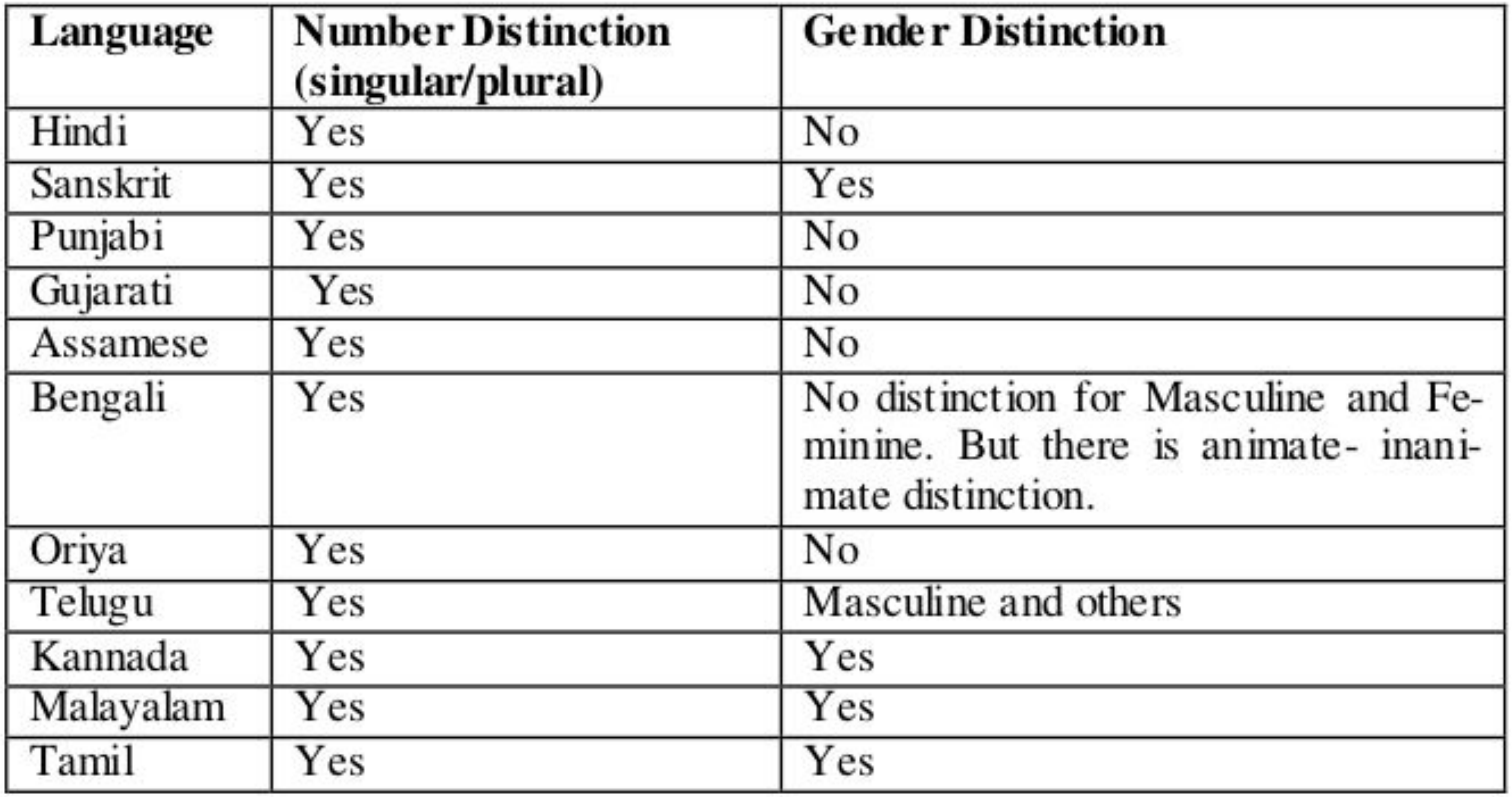
Overview

- A ***language independent*** engine, which takes shallow parsed text as input.
- The morphological richness of Indian languages is utilised for language independent resolution.
- GNP information obtained from in-depth morphological analysis.
- ***PNG agreement heuristic rules*** - capable of filtering the possible candidate antecedents for an anaphor.
- Developed and tested on Tamil, Bengali and Hindi.

Characteristics of Indian Language Anaphora

- Relatively free word order and clausal structures are more fixed order.
- *Dravidian* - Plural marker and case markers get affixed to the nouns. Tense markers and GNP markers affix with verbs.
- *Indo Aryan* - Case markers occur as postpositions following the nouns.
- Indian languages vary largely in the distinction of *Number (singular/plural)* and *Gender* in pronouns.

Variation of Pronouns with respect to Number and Gender



Language	Number Distinction (singular/plural)	Gender Distinction
Hindi	Yes	No
Sanskrit	Yes	Yes
Punjabi	Yes	No
Gujarati	Yes	No
Assamese	Yes	No
Bengali	Yes	No distinction for Masculine and Feminine. But there is animate- inanimate distinction.
Oriya	Yes	No
Telugu	Yes	Masculine and others
Kannada	Yes	Yes
Malayalam	Yes	Yes
Tamil	Yes	Yes

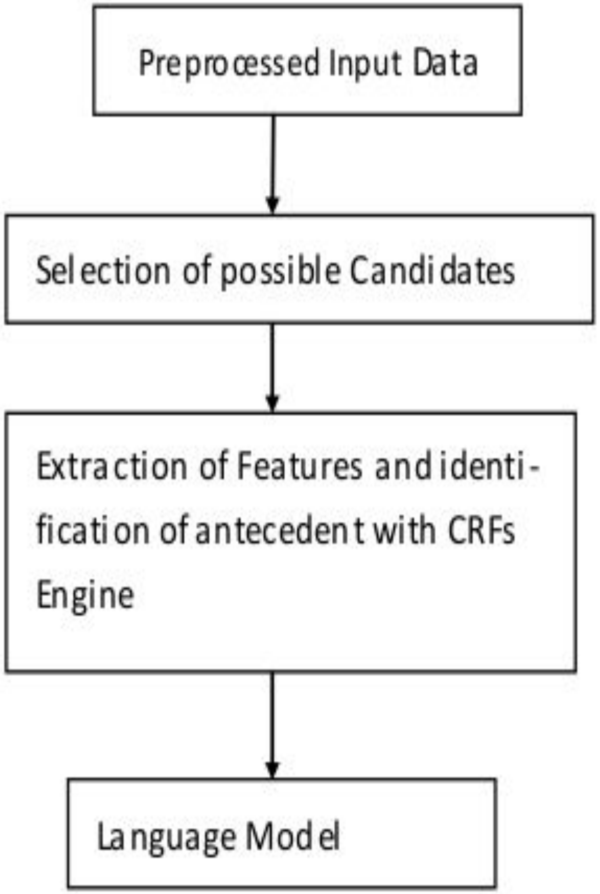
Architecture and Working

Preprocessing of Data

- Limited shallow parsing on training and testing data.
- Pre-processing with IL-ILMT tools:
  - Morph analysis
  - POS tagging
  - Chunking
  - Clause boundary identification
  - Named Entity Recognition

Detailed Morph Analysis

- Analyse both inflectional and derivational morphology.
- Following are identified-
  - Root word
  - Lexical category
  - GNP
  - Case marker (N)
  - Tense marker (V)
  - Suffixes



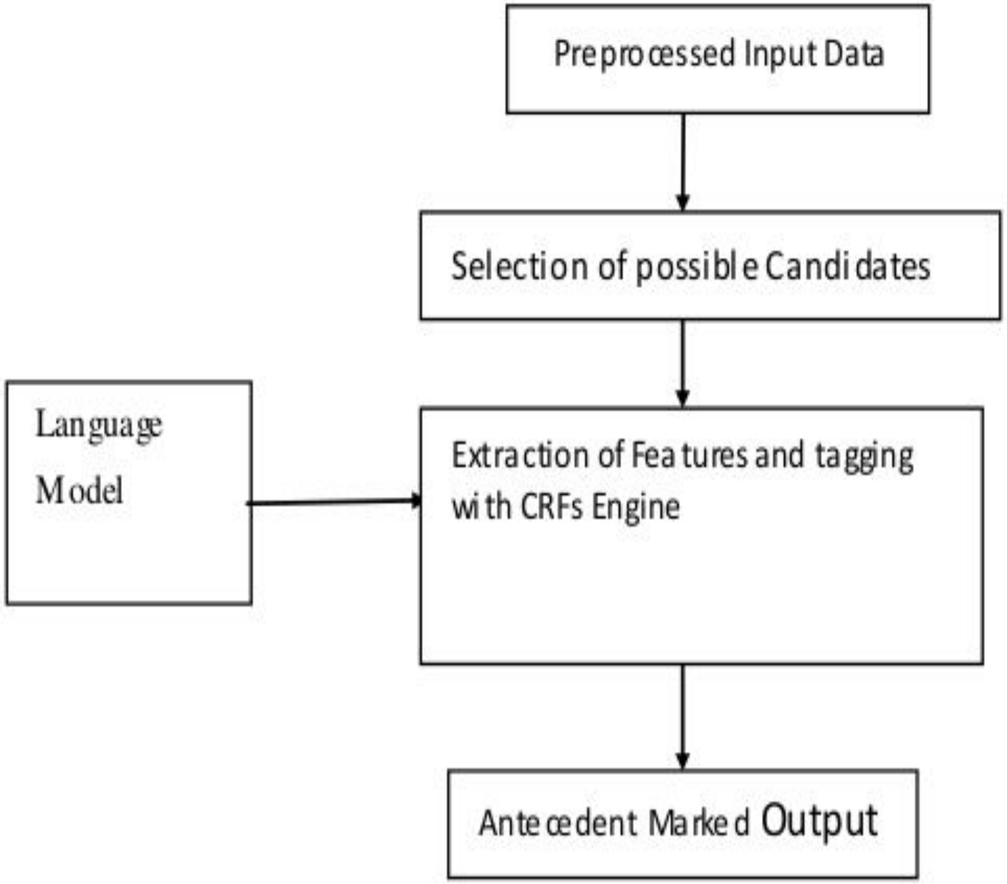
Preprocessed Input Data

Selection of possible Candidates

Extraction of Features and identi-fication of antecedent with CRFs Engine

Language Model

Figure 1: Training phase



Preprocessed Input Data

Selection of possible Candidates

Language Model

Extraction of Features and tagging with CRFs Engine

Antecedent Marked Output

Figure 2: Testing phase

Selection of Candidate Noun Phrases for Antecedent

- ***Candidate*** - NPs which agree with the pronoun in GNP
- ***Training Phase***
  - Candidate NPs occuring in between the anaphor and the antecedent are collected for each pronoun and given for training.
  - The exact anaphor and antecedent pair forms **positive pair** and other NPs and anaphor form **negative pairs** for learning.
- ***Testing Phase***
  - Candidate NPs are collected from the current sentence and four prior sentences.
  - To dynamically capture gender distinction and anaphor-antecedent agreement, set of **heuristic rules** have been presented.

Heuristic Rules

- If the gender of the pronoun is **M/F/N**, then the nouns having **masculine/feminine/neuter** gender are chosen as candidate antecedents.
- If the gender of the pronoun is **ANY**, then all the nouns are considered for **candidate antecedent set** and the nouns with gender **ANY** is given higher priority.

Once the possible candidates for antecedents for the anaphors are selected, they are given for training/testing using CRF, a linear graphical technique to learn and identify the antecedents.

Anaphora Resolution as a Binary Classification Task

- **Training phase** – system is provided with annotated data and the features for learning.
- **Testing phase** – Unseen text is given for the automatic anaphora resolution.
- **Binary classification task** – The machine has to classify whether the given candidate antecedent is the real antecedent or not based on the features of the candidate antecedents and the pronoun.
- The features for learning are extracted from the shallow parsed data, for all possible candidate antecedent and pronoun pairs.

Conclusions

- Maximum performance in **Tamil.**
- **Hindi -** pronouns such as ***vaha, us** , **unhone*** and ***khud*** etc., do not have gender distinction. PNG agreement adds more challenges, due to which the system gives more ***false positives.***
- **Bengali** – third person pronouns such as ***ami (I), túmi/tui/apni (you), se/tini (he/she), amra (we), tara/tnara (they)***, do not have gender distinction, but there is ***animacy distinction***. And also the verb has ***no gender agreement***. Hence, lesser scores than other two.

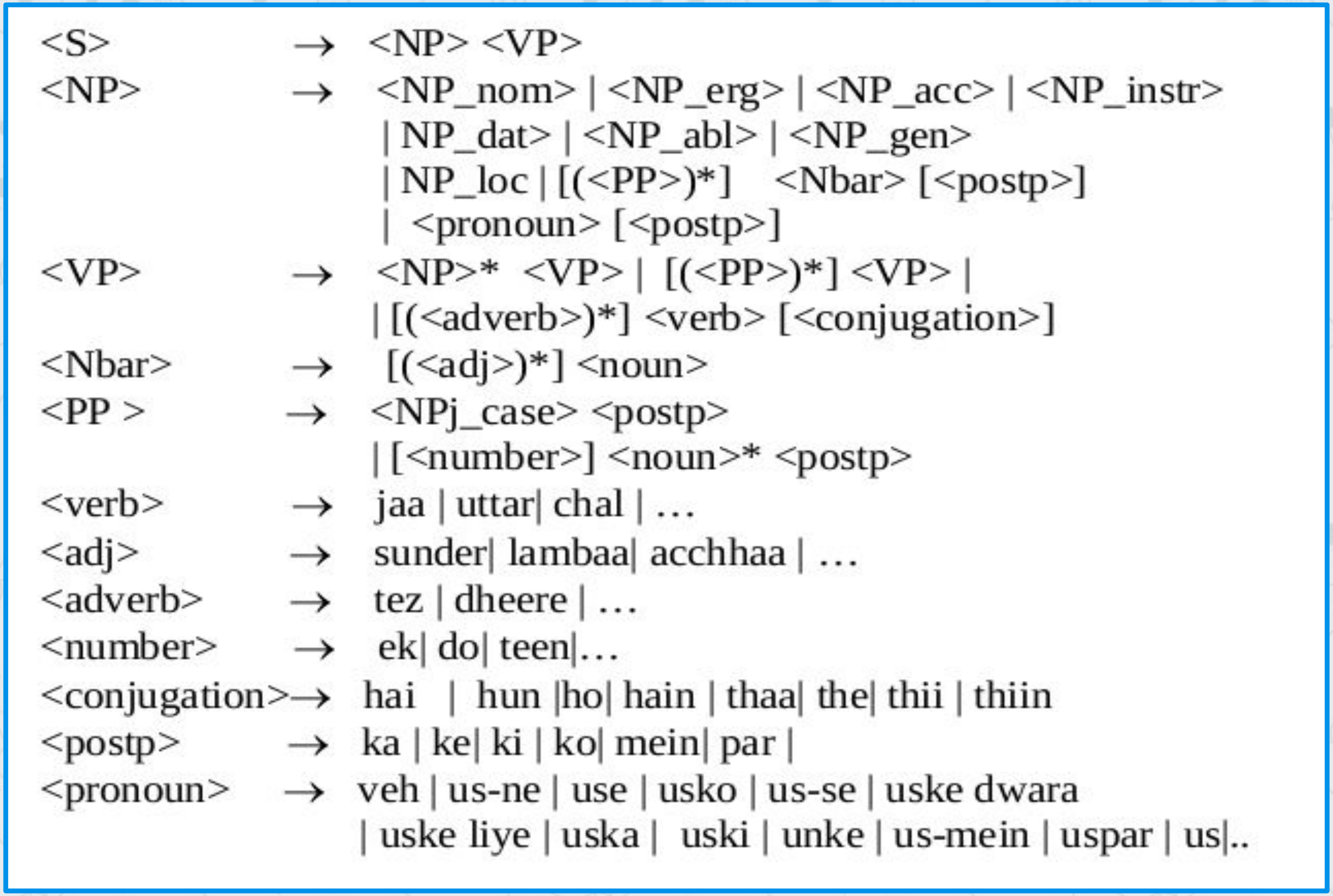
3.

Resolving Pronominal Anaphora in Hindi Using Hobbs’ Algorithm

Overview

- Hobbs’ algorithm has been improved upon for Hindi language, taking into account the impact of the subject-object roles on anaphora resolution for reflexive and possessive pronouns.
- This algorithm is computationally economical, since does not make use of semantic information and is based on syntactic information.

CFG used for surface structures



```
<S>              →  <NP> <VP>
<NP>             →  <NP_nom> | <NP_erg> | <NP_acc> | <NP_instr>
                     | NP_dat> | <NP_abl> | <NP_gen>
                     | NP_loc | [(<PP>)*]    <Nbar> [<postp>]
                     |  <pronoun> [<postp>]
<VP>             →  <NP>*  <VP> |  [(<PP>)*] <VP> |
                    | [(<adverb>)*] <verb> [<conjugation>]
<Nbar>           →  [(<adj>)*] <noun>
<PP >            →  <NPj_case> <postp>
                    | [<number>] <noun>* <postp>
<verb>           →  jaa | uttar| chal | …
<adj>            →  sunder| lambaa| acchhaa | …
<adverb>         →  tez | dheere | …
<number>         →  ek| do| teen|…
<conjugation>→  hai    |  hun |ho| hain | thaa| the| thii | thiin
<postp>          →  ka | ke| ki | ko| mein| par |
<pronoun>        →  veh | us-ne | use | usko | us-se | uske dwara
                    | uske liye | uska |  uski | unke | us-mein | uspar | us|..
```

Applying the algorithm to the surface parse tree of a sentence

- Leaves of the parse tree, in left to right order represent the ***original sentence.***
- Surface parse tree exhibits the ***grammatical structure*** of the sentence.
- **Main idea**
  - Traverse the full parse tree starting from the pronoun looking for candidate NPs, and add them to a list of candidates.
  - Left to right BFS in the subtree, subject to the constraints defined by the algorithm.

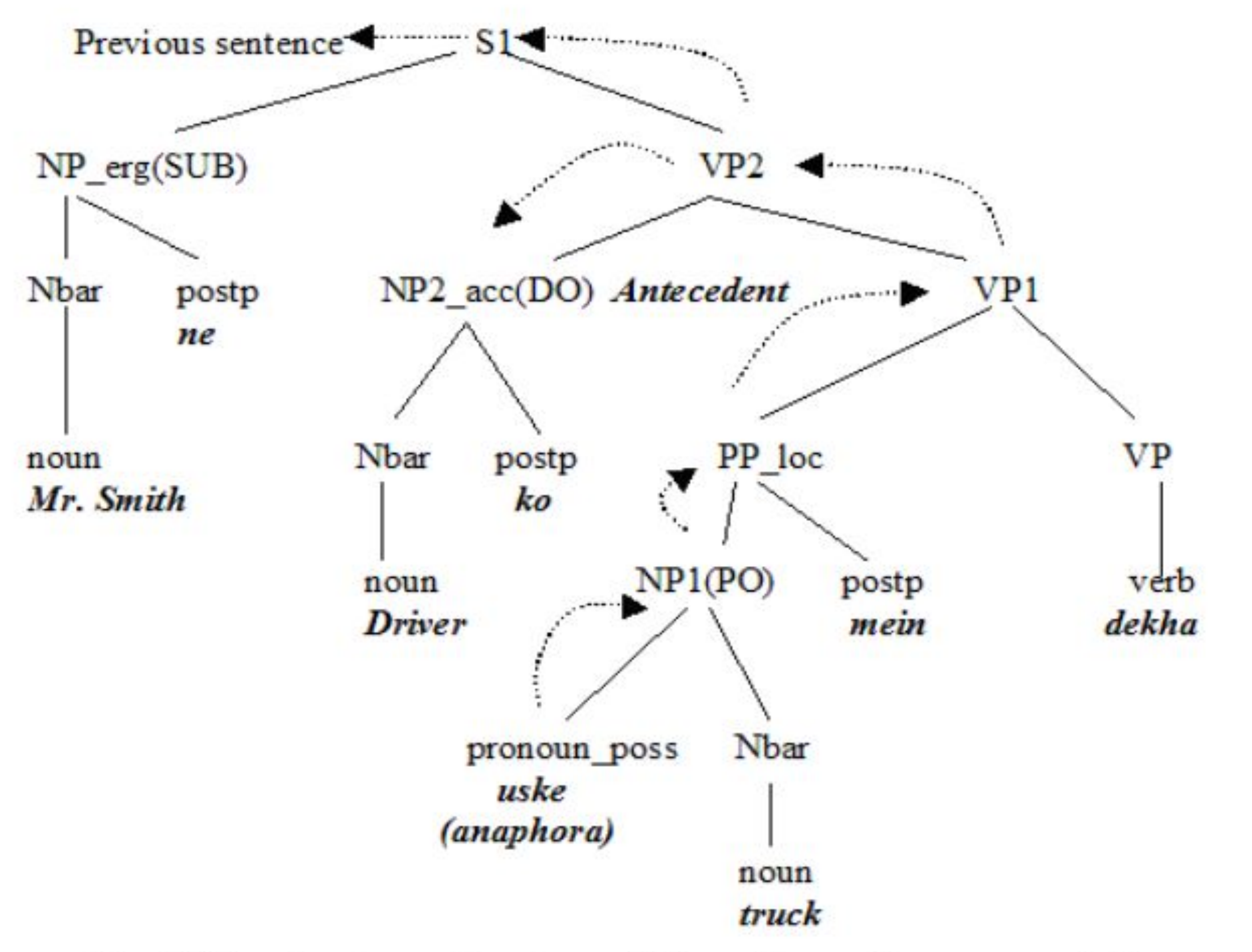
Applying the algorithm to the surface parse tree of a sentence

- For a pronoun P, antecedent is the first NP in the tree obtained by left-to-right BFS of the subtree to the left of the path T such that:
  - T is the path from the NP dominating P to the first NP or S dominating this NP
  - T contains an NP or S node N-bar which contains the NP dominating P
  - N does not contain NP. If an antecedent satisfying this condition is not found in the sentence containing P, the
  - algorithm runs recursively on preceding sentences.

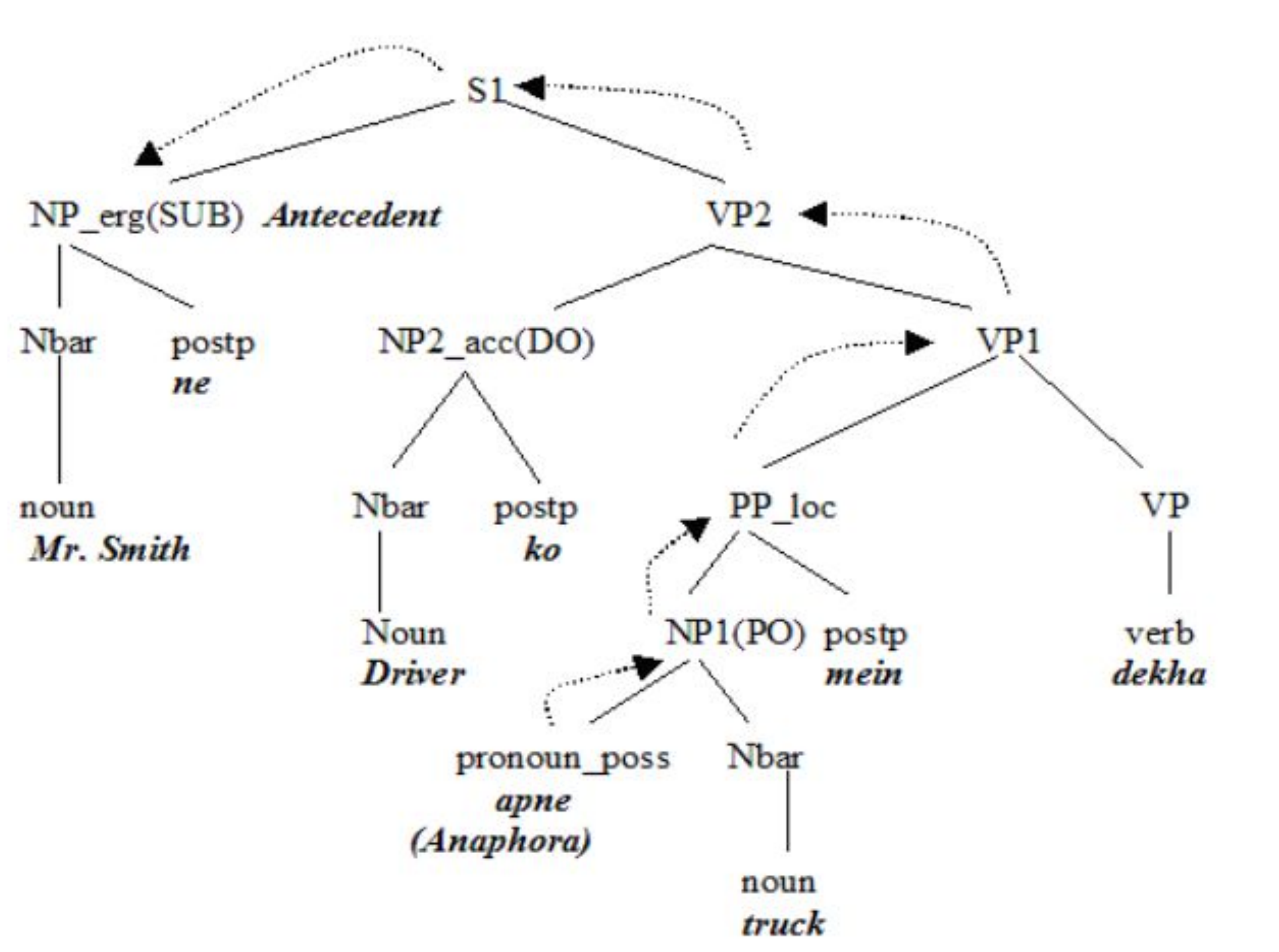
“

Mr. Smith saw a driver in his truck

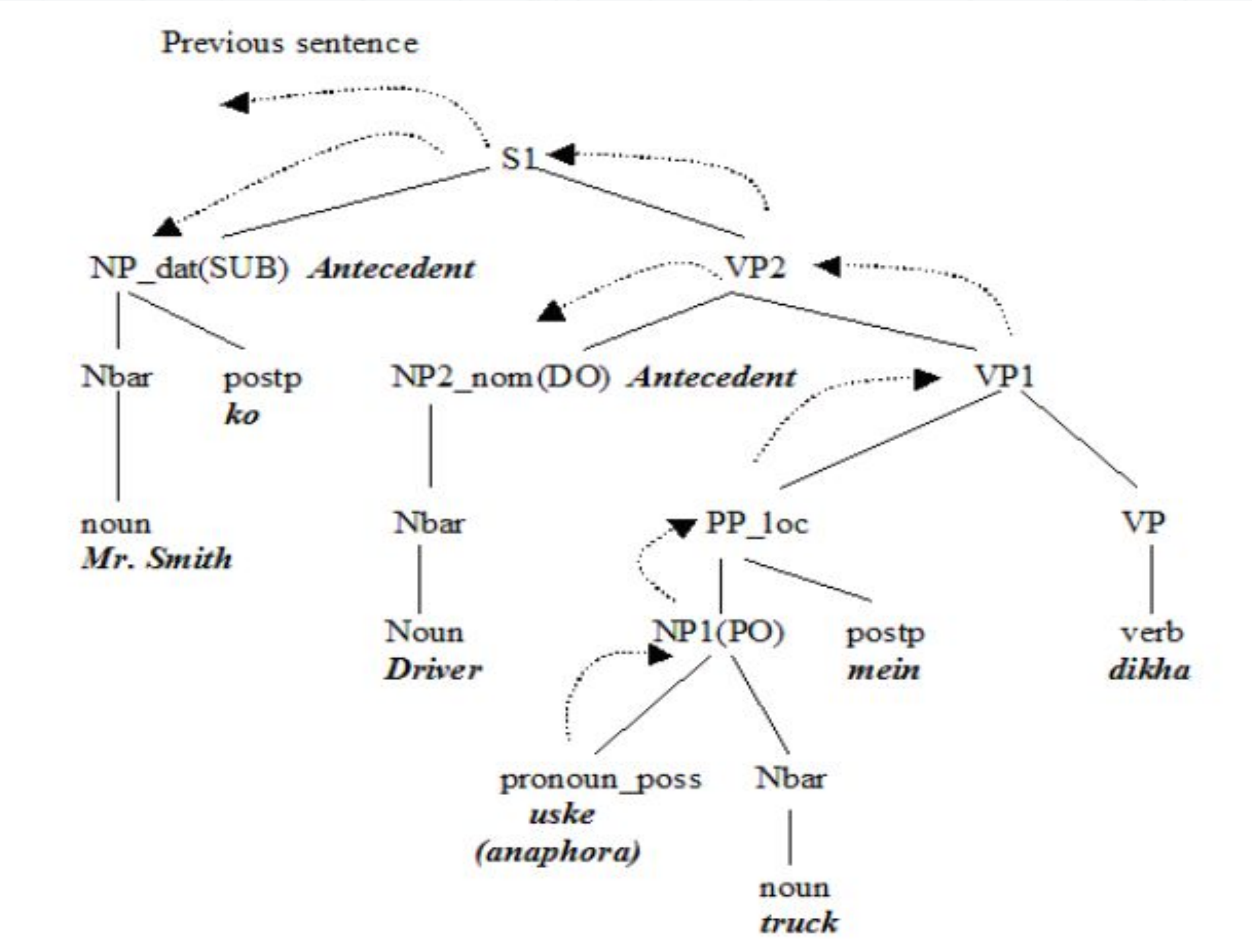
1. **Mr. Smith ne driver ko uske truck me dekha**



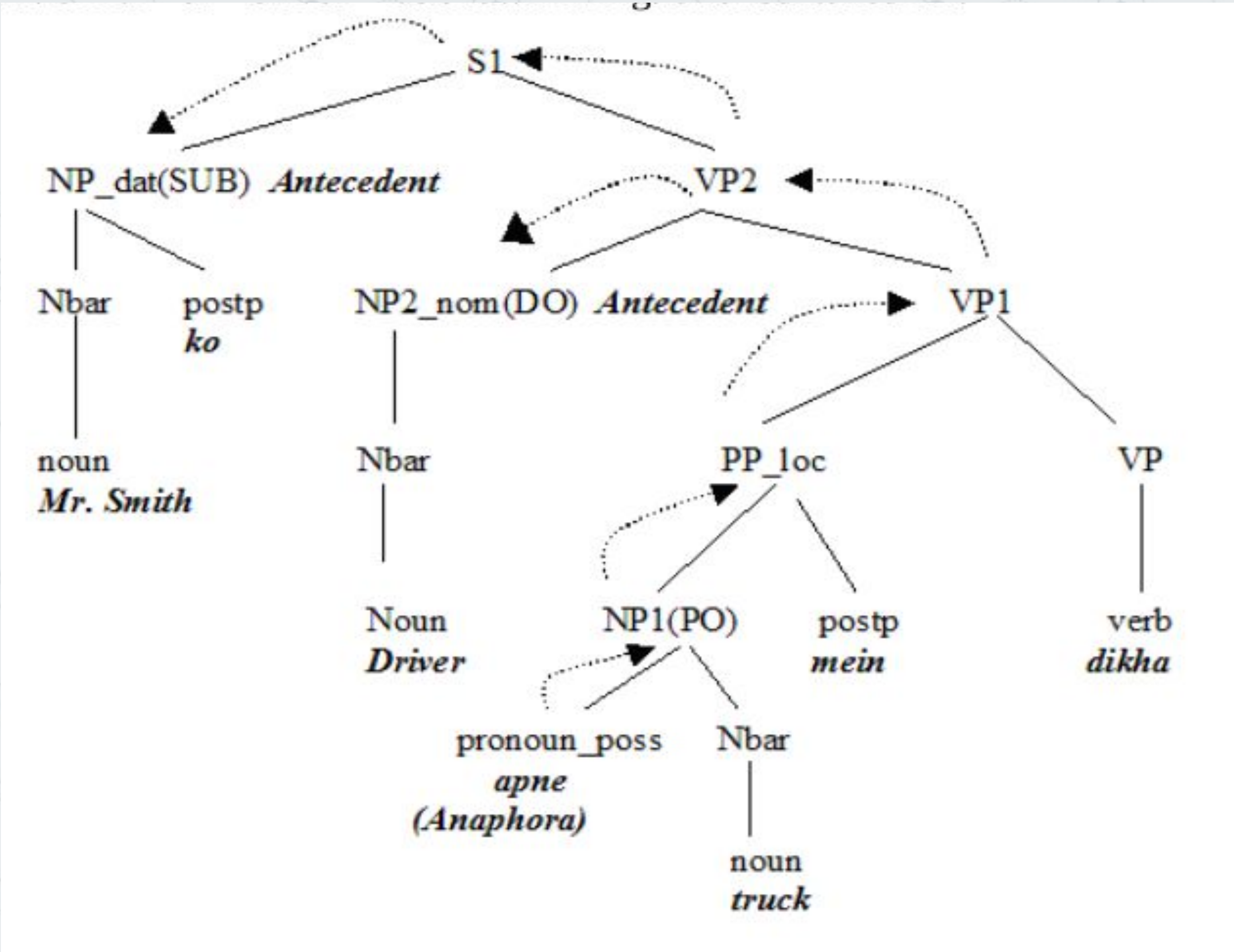
2. Mr. Smith ne driver ko apne truck me dekha



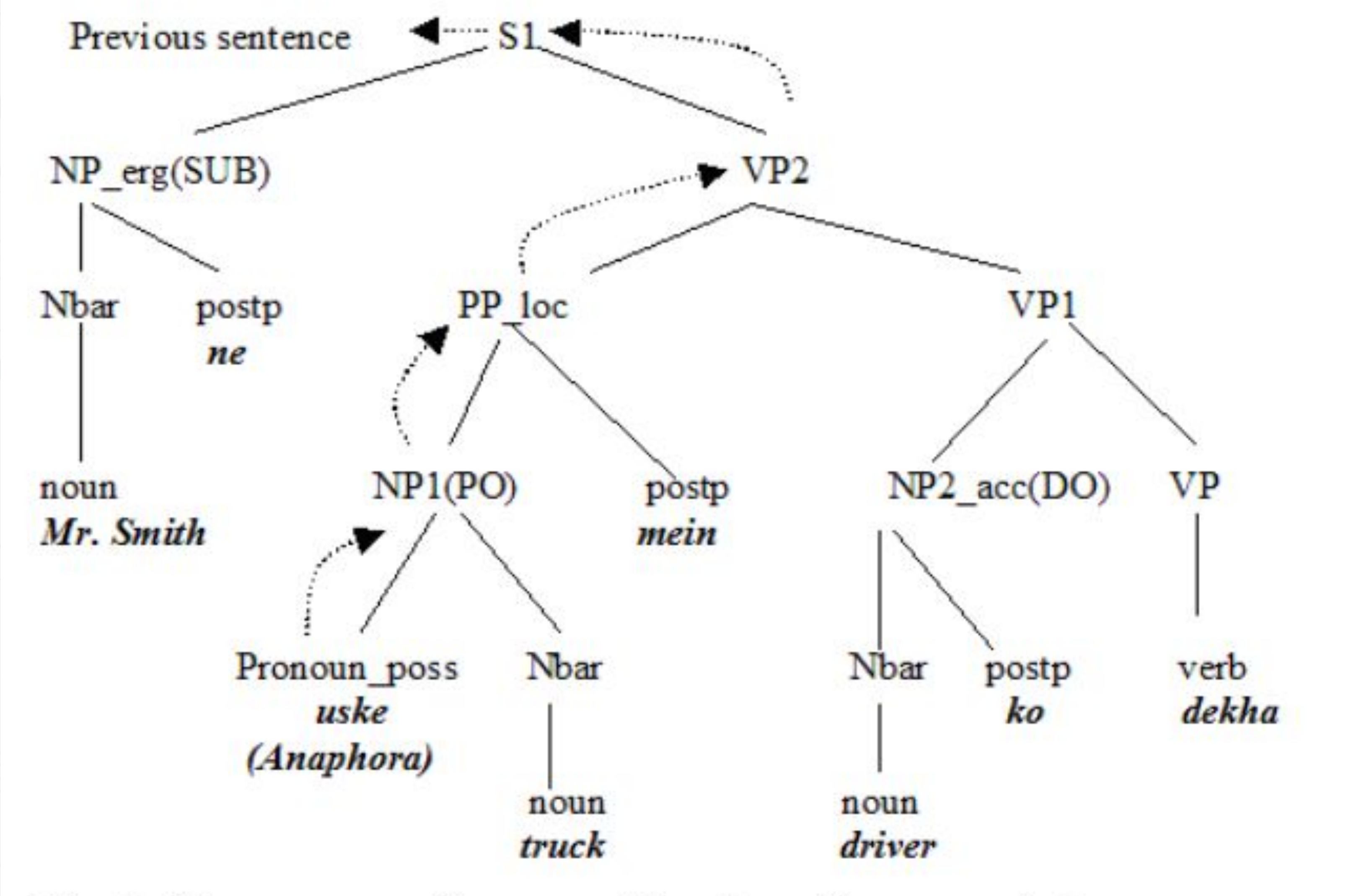
3. Mr. Smith ko driver uske truck me dikha



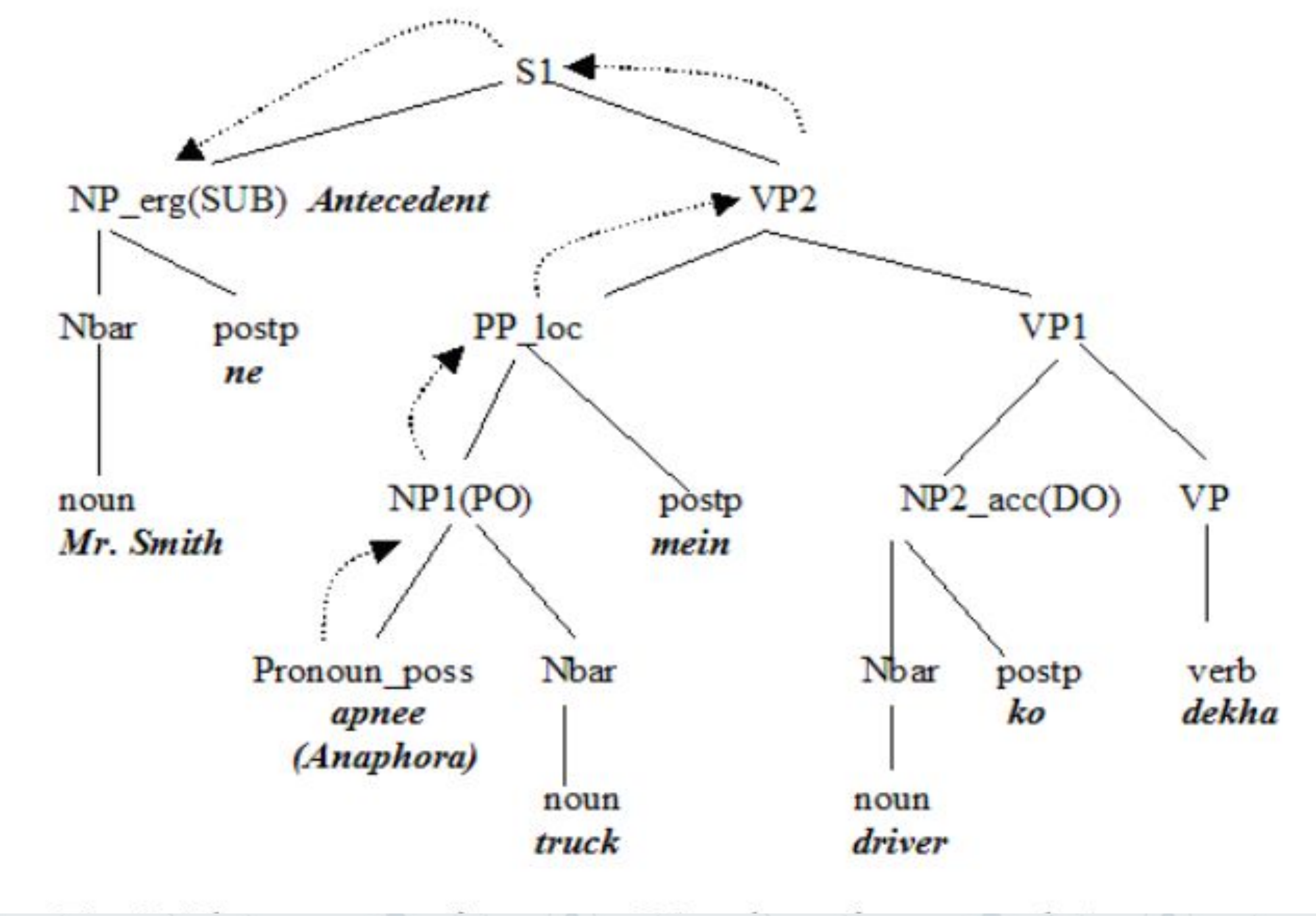
4. Mr. Smith ko driver apne truck me dikha



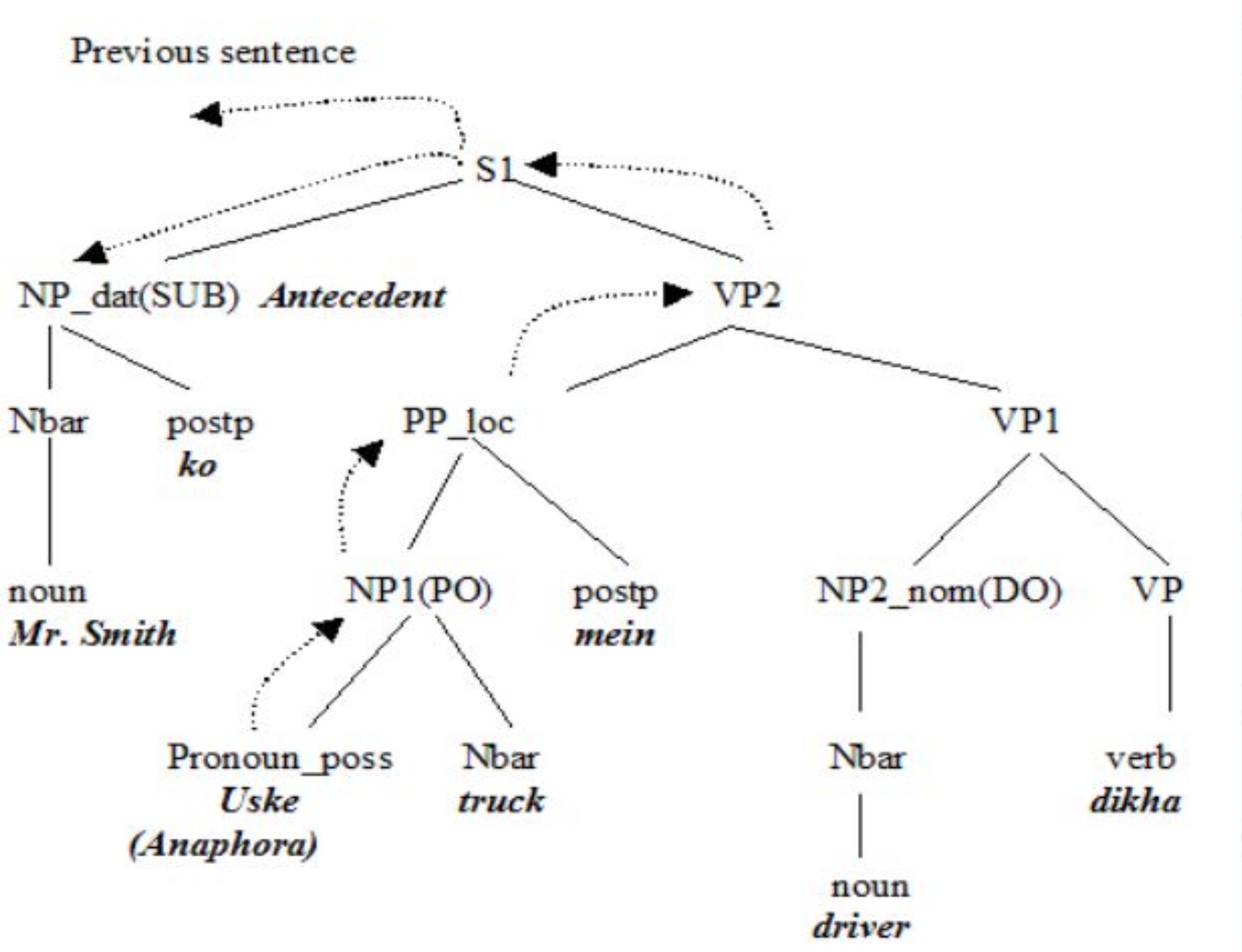
5. Mr. Smith ne uske truck me driver ko dekha



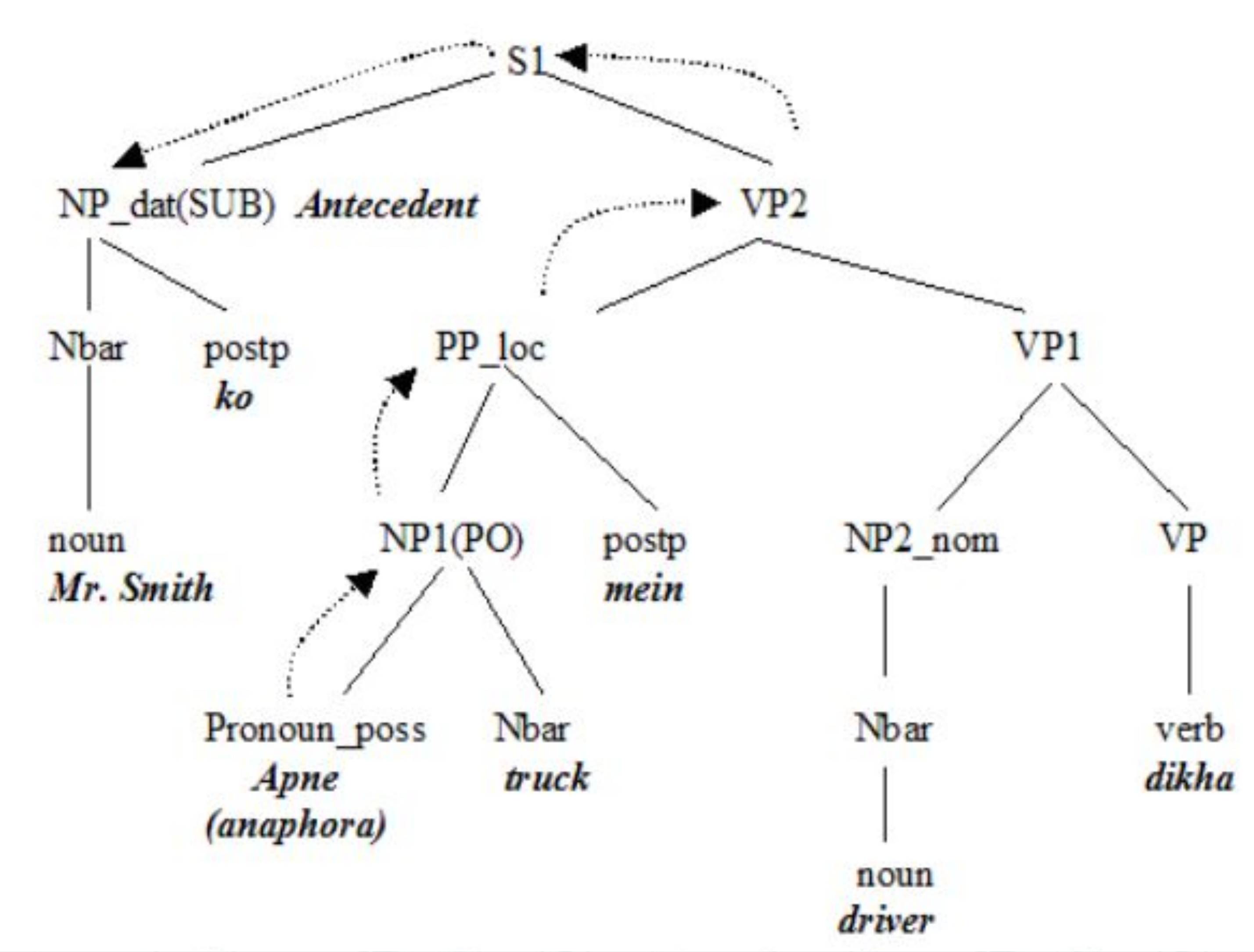
6. Mr. Smith ne apne truck me driver ko dekha



7. Mr. Smith ko uske truck me driver dikha



8. Mr. Smith ko apne truck me driver dikha



Thankyou!