Chapter V

Algorithms for Resolving Anaphora in Malayalam and Hindi

The anaphora resolution algorithms for Malayalam and Hindi are discussed in this chapter. The grammar described in chapter II and III are used to develop the algorithms. The algorithm works on the parser described in chapter IV for implementing this algorithm. The input to the anaphora resolution component is the parsed output from the parser.

The algorithm for each anaphor is discussed separately.

Consider the pronouns in Malayalam first. The condition for a pronoun to corefer with a NP is as given below.

1. A pronoun P is coreferential with a NP if and only if the following conditions hold.
   a. P and NP have compatible agreement features.
   b. P does not precede NP
   c. If P is possessive, then NP is the subject of the clause, which contains P.
   d. If P is non possessive, then NP is the subject of the clause which does not contain P.

Let us consider the algorithm for identifying the antecedent for pronoun.

1. Create a list of words of S with NPs, clause and immediate clause identified (From the parser)
2. For each pronoun P in S:
   (a) if (P is poss) then
      (i) Select the NP which is the subject of the clause in which P occurs.
      (ii) If P precedes NP in S
           then STOP and RETURN.
   (b) else/*(case(p)!=poss)*/
      (i) identify the immediate clause(s) of the clause in which the pronoun...
occurs.

(ii) if one immediate clause is identified then
    (a) select the NP which is the subject of this clause.
    (b) if P precedes NP in S then STOP and RETURN

(iii) if more than one immediate clause then
    (a) select the NP which is the subject of the clause preceding the
        clause containing P
    else /*no immediate clause*/
        antecedent lies outside the S

3. If P and NP do not agree in number, gender, person, then STOP and RETURN
4. The NP is the antecedent of P.

There are four steps in this algorithm. In the first step the parsed structure from the parser
is collected. This parsed structure is the input to anaphora resolution. In the second step
the pronouns are classified into possessive and non-possessive. If the pronoun is
possessive then the NP, which is the subject of the clause in which the pronoun occurs is
selected. Then it will check whether the pronoun precedes the NP in the sentence. If it
does, then the termination condition terminates the process. When a non-possessive
pronoun is encountered, the immediate clause in which the pronoun occurs is identified.
If there is only one immediate clause, the NP, which is the subject of that clause is
chosen. If the pronoun precedes the NP that the termination condition terminates the
process. If more than one immediate clause is identified, the clause which precedes the
pronoun is selected and the subject of that clause is identified. When there are no
immediate clauses to identify the subject, it is assumed that the antecedent lies out side
the sentence. In the third step the NP and P are checked for number, person, gender
agreement. If there is no agreement between NP and P then the termination condition
terminates the program. If there is an agreement step four is activated, where the NP is
accepted as the antecedent of the pronoun. Consider the examples of the pronoun
algorithm results. The system produces the following results.
1. ra:man avanRe kuttiye kantu.
   raman he-poss child-acc see-pst
   (Raman saw his child.)

Input S: ra:man avanRe kuttiye kantu.

Parsed structure:

ra:man  <N>  <NOM>  <subj>  <m>  <s>  <third>
avanRe  <N>  <POSS>  <Pronoun>  <m>  <s>  <third>
kuttiye <N>  <ACC>  <obj>  <n>  <s>  <third>
kantu   <V>  <trans>  <pst>

anaphoric element: pronoun- avanRe

anaphora resolution:

The clause which contains the anaphor: ra:man avanRe kuttiye kantu.
The clause which contains the antecedent: ra:man avanRe kuttiye kantu.

Subject of the clause: ra:man

PNG agreement:
ra:man:  <m>  <s>  <third>
avanRe:  <m>  <s>  <third>

ra:man is the antecedent of avanRe.

2. ra:man avane aticcu ennu mo:han paRannu
   raman he-acc beat-pst comp mohan say-pst
   (Mohan said that Raman beat him)

Input S: ra:man avane aticcu ennu mo:han paRannu.

Parse structure:

ra:man  <N>  <NOM>  <subj>  <m>  <s>  <third>
avane  <N>  <ACC>  <Pronoun>  <m>  <s>  <third>
aticcu <V>  <trans>  <pst>
ennu   <Compl>
mo:han  <N>  <NOM>  <subj>  <m>  <s>  <third>
paRannu <V>  <trans>  <pst>
Anaphoric element: avane

Anaphora resolution:

Clause which contains the anaphor: ra:man avane aticcu ennu.
Clause which contains the antecedent: mo:han paRannu.

Subject of the clause: mo:han

PNG agreement
mo:han  <m>  <s>  <third>
avane  <m>  <s>  <third>

mo:han is the antecedent of avane.

The next module in anaphora resolution algorithm is the one pronoun. Consider the rules for one-pronoun.

2. A one-pronoun is coreferential with an NP iff
   a. Non subject NP is the immediate clause of one-pronoun.
   b. NP precedes the one-pronoun
   c. One-pronoun and NP must agree with respect to the C features.

The algorithm is as follows.

II.1. Create a list of words of S with NPs, clause and immediate clause identified (From the parser)
2. Identify the one-pronoun in the list.
   (i) Identify the inherent feature[C] of one-pronoun
3. Identify the NPs in the IC
   (i) Identify the non-subject NPS.
   (ii) For each non-subject NP identified
        (a) if one-pronoun is [+C]
            (i) if not [+count]
                mark the NP as NON-ANT
        (b) if one-pronoun is[-C]
            (i) if not [-animate]
                (ii) if not [-count]
                    mark the NP as NON-ANT.
        (c) if one-pronoun is[+/-C]
            (i) if not SUBJ
                mark the NP as ANT
4. The NP (or NPs) marked as “ANT” in (3) is the antecedent of one-pronoun.

This module contains four steps. The first step is the common step, which creates the structure that the parser gives. The next step identifies the one-pronouns and classifies it according to the inherent feature [C]. In the third step all the NPs are identified. The non-subject NPs of the immediate clause are identified and compared with the inherent feature of the one-pronoun. If the inherent feature of the one-pronoun is [+C], and the NP is not count, then the NP is not the antecedent of the one-pronoun. If the inherent feature is [-C], then the NP is checked for both count and inanimate. If it is not count and not inanimate, then the NP is not the antecedent of the one-pronoun. If the inherent feature is [+/-C] then any non-subject NP is the antecedent of the one-pronoun. In step four the NPs marked as ANT in step three are considered as the antecedent of one-pronoun.

Consider the examples given below.

3. na:n rantu pakshikale ku:ttil kantu ra:man orannatte marattil kantu
   I two bird-pl nest-loc see-pst raman one tree-loc see-pst
   (I saw two birds in the nest. Raman saw one on the tree)

Input S: na:n rantu pakshikale ku:ttil kantu ra:man orannatte marattil kantu

Parsed structure:

na:n    <N><NOM><subj><s><first><+count>
rantu    <N><NUM><quant><+count>
pakshikale <N><NOM><obj><n><pl><third><count>
ku:ttil  <N><LOC><obj><n><s><third><+count>
kantu   <V><tran><pst>
ra:man   <N><NOM><subj><m><s><third><+count>
orannatte <OneP><+C><+count>
marattil <N><LOC><obj><n><s><third><+count>
kantu   <V><tran><pst>

anaphoric element: one- pronoun: orannatte
anaphora resolution:

The clause which contains the anaphor: \textit{ra:man orannatte marattil kantu}

The clause which contains the antecedent: \textit{na:n rantu pakshikale marattil kantu}

pakshikale is the antecedent of orannatte.

4. \textit{ra:mu pa:lu va:nniccu pu:cca alpam kuticcu.}

\textit{(Ramu bought milk. The cat drank a little)}

\textbf{Input S: ra:mu pa:lu va:nniccu pu:cca alpam kuticcu.}

\textbf{Parsed structure:}

\begin{itemize}
  \item \textit{ra:mu} \quad \langle N \rangle \langle NOM \rangle \langle subj \rangle \langle m \rangle \langle s \rangle \langle third \rangle \langle +count \rangle \langle +animate \rangle \langle +human \rangle
  \item \textit{pa:lu} \quad \langle N \rangle \langle NOM \rangle \langle obj \rangle \langle n \rangle \langle -count \rangle
  \item \textit{va:nniccu} \quad \langle V \rangle \langle trans \rangle \langle pst \rangle
  \item \textit{pu:cca} \quad \langle N \rangle \langle NOM \rangle \langle subj \rangle \langle n \rangle \langle s \rangle \langle third \rangle \langle +count \rangle \langle +animate \rangle \langle -human \rangle
  \item \textit{alpam} \quad \langle OneP \rangle \langle -C \rangle
  \item \textit{kuticcu} \quad \langle V \rangle \langle trans \rangle \langle pst \rangle
\end{itemize}

anaphoric element: one pronoun – \textit{alpam}

\textbf{anaphora resolution:}

\begin{itemize}
  \item Clause which contains the anaphor: \textit{pu:cca alpam kuticcu}
  \item Clause which contains the antecedent: \textit{ra:mu pa:lu va:nniccu}
\end{itemize}

The antecedent of \textit{alpam} is \textit{pa:lu}.

Now consider the algorithm for the reflexive. The rules are as follows.

3.1 A non-emphatic reflexive \textit{R} is coreferential with an NP iff NP is the subject of the matrix clause if \textit{R} is contained in the embedded clause.

3.2 An emphatic reflexive \textit{R}_1 is coreferential with an NP iff NP is the subject of the clause in which \textit{R}_1 occurs.

3.3 A possessive reflexive \textit{R}_2 coreferential with an NP iff NP is the subject of the clause containing \textit{R}_2.

The algorithm for the above rules is as follows.
III. 1. Create a list of words of S with NPs, clause and immediate clause identified
(From the parser)
2. For each reflexive R in S
   (a) if R is a non-emphatic reflexive then
      (i) identify the immediate clause of the clause in which R occurs.
      (ii) select the NP which is the subject of the immediate clause.
   (b) if R is emphatic or possessive then
      (i) select the NP which is the subject of the clause in which R occurs.
3. The NP is the antecedent of R.

There are three steps in reflexive resolution module. The first step collects the parsed
structure from the parser. In the second step the reflexives are identified. For a non-
emphatic reflexive identified, the system tries to identify the immediate clause in which
the reflexive occurs. It identifies the subject of the immediate clause. For an emphatic or
possessive reflexive the subject of the clause in which the reflexive occurs is identified.
In the final step the NP is identified as the antecedent of the reflexive.

5. raman tanno:tu nunna paranni ennul kRisnan paranni
   raman refl-inst lies say-pst compl krisnan say-pst
   (Krishnan said that Raman said lies to self=(Krishnan).)

InputS: raman tanno:tu nunna paranni ennul kRisnan paranni

The parsed structure:

ra:man  <N><NOM><subj><m><s><third><+human><+animate>
tanno:tu  <R><INST>
nunna  <N><NOM>
paRannu  <V><intr><pst>
ennu  <compl>
krishnan  <N><NOM><subj><m><s><third><+human><+animate>
paRannu  <V><intr><pst>

anaphoric element: Reflexive-tanno:tu

anaphoric resolution:
The clause which contains the anaphor: ra:man tanno:tu nunna paRannu.
The clause which contains the antecedent: kRisnan paRannu.
The antecedent of tanno:tu is ra:man.

6. ra:man tannata:n sku:lil po:yi
   raman self school-loc go-pst
   (Raman went to the school himself)

Input S: ra:man tannata:n sku:lil po:yi

The parsed structure:

```
ra:man  <N><NOM><subj><m><s><third><+human><+animate>
tannata:n  <R1>
sku:lil  <N><LOC><n><s><third><-human><-animate>
po:yi  <V><intr><pst>
```
anaphoric element: tannata:n
anaphora resolution:

Clause which contains the anaphor: ra:man tannata:n sku:lil po:yi
Caluse which contains the antecedent: ra:man tannata:n sku:lil po:yi
The antecedent of tannata:n is ra:man.

7. ra:man svayam vi:ttil po:yi ennu kRisnan paRannu
   raman self house-loc go-pst compl krisnan say-pst
   (Krishnan said that Raman went to the house himself)

Input S: ra:man svayam vi:ttil po:yi ennu krisnan paRannu

Parsed structure:

```
ra:man  <N><NOM><subj><m><s><third><+human><+animate>
svayam  <R1>
vi:ttil  <N><LOC><n><s><third><-human><-animate>
po:yi  <V><intr><pst>
ennu  <compl>
kRisnan  <N><NOM><subj><m><s><third>
paRannu  <V><intr><pst>
```
anaphoric element: Reflexive-svayam
anaphora resolution:

clause in which the anaphor occurs: ra:man svayam vi:ttil po:yi
clause in which the antecedent occurs: ra:man svayam vi:ttil po:yi
The antecedent of svayam is ra:man.

8. si:ta svantam kuttiye aticcu ennu amma paRannu.
    sita refl child beat-pst compl mother say-pst
    (Mother said that Sita beat her child.)

Input S: si:ta svantam kuttiye aticcu ennu amma paRannu.

Parsed Structure:

si:ta <N><NOM><f><s><third>
svantam <pR>
kuttiye <N><ACC><n><s><third>
aticcu <V><trans><pst>
ennu <compl>
amma <N><NOM><f><s><third>
paRannu <V><intr><pst>

anaphoric element: svantam

anaphora resolution.

Clause which contains the anaphor: si:ta svantam kuttiye aticcu ennu
Clause in which the antecedent occurs: si:ta svantam kuttiye aticcu ennu

The antecedent of svantam is si:ta

Consider the next module the reciprocal anaphors. The rules for reciprocal anaphors are as follows.

1. NP is the subject of the clause which contains R’
2. NP is plural
3. NP precedes R’

The algorithm for the reciprocal anaphor is as follows.

IV 1. Create a list of words of S with NPs, clause and immediate clause identified (From the parser)
    2. For each reciprocal anaphor R’ in S
       (i) select the NP which is the subject of the clause in which R’ occurs.
       (ii) if R’ precedes the NP
then STOP and RETURN
(iii) if NP is not plural
then STOP AND RETURN.

3. The NP is the antecedent of R’

In the second step the reciprocals are identified from the input structure. After identifying the reciprocals the system identifies the NP which is the subject of the clause in which it occurs. The R’ is checked with the NP for precedence. If it precedes then the termination condition terminates the process. If it does not precede, then the NP is checked whether it is a plural or not. If the NP is not plural, then the system stops the process. But if it is plural, it will go to step three. In step three the NP is identified as the antecedent of R’.

   this children each other beat-neg
   (These children do not beat each other.)

Input S: i: kuttikal ora:l ora:le atikkilla

Parsed structure:

i: <demon>
kuttikal <N><NOM><subj><n><pl><third>
ora:l-ora:le <D>
atikkilla <V><tran><pst><neg>

anaphoric element:ora:l-ora:le

anaphora resolution:

Clause which contains the anaphor: i: kuttikal ora:l-ora:le atikkilla
Clause which contains the antecedent: i: kuttikal ora:l-ora:le atikkilla

The antecedent of ora:l-ora:le is kuttikal.

5. The fifth module in the algorithm is for the resolution of distributive anaphors.

The rules are as follows

a. NP is the subject of the clause which contains D
b. NP is plural

c. NP precede D

The algorithm for distributive anaphors D are as follows:

1. Create a list of words of S with NPs, clause and immediate clause identified
   (From the parser)
2. For each distributive D in S
   (i) select the NP, which is the subject of the clause in which R occurs.
   (ii) if D precede NP
        then STOP and RETURN
   (iii) if NP is not plural
        then STOP and RETURN
3. The NP is the antecedent of D.

In the second step the distributives are identified from the input structure. After identifying the distributives the system identifies the NP which is the subject of the clause in which it occurs. The D is checked with the NP for precedence. If it precedes then the termination condition terminates the process. If it does not precede, then the NP is checked whether it is plural or not. If the NP is not plural then the system stops the process. But if it is, it will go to step 3. In step 3 the NP is identified as the antecedent of D.

10. ammama:r avar-avarute kuttikale rakshiccu.
    mother-pl their children save-pst
    (Mothers saved their own children)

Input S ammama:r avar-avarute kuttikale rakshiccu

Parsed structure:

ammama:r  <N><NOM><p><pl><third>
avar-avarute <i>
kuttikale  <N><NOM><n><pl><third>
rakshiccu  <V><tran><pst>

anphoric element: Reciprocal avar-avarute
anaphora resolution:

Clause in which the anaphora occurs: ammama:r avar-avarute kuttikale rakshiccu
Clause in which the antecedent occurs: ammama:r avar-avarute kuttikale rakshiccu.

ammama:r is the antecedent of avar-avarute.

6. Consider the gaps. The sixth module in anaphora resolution component is the resolution of gaps. Gaps are identified by the following rules.

Forward.
NP is the subject of Q’
NP is the object of Q’
V is in Q’

Backward.
NP is the subject of Q”
V is in Q”

VI The algorithm for the above rules are as follows.

1. Create a list of words of S with NPs, VPs and two clauses identified. (From the parser)
2. if const (Q’) < const (Q”) then /* backward */
   (i) identify the types of constituent which is in Q” but not in Q’
   (ii) if identified const. contains object STOP and RETURN
   (iii) modify Q’ by adding the identified constituents in the appropriate slots of Q’
3. if const(Q’) > const(Q”) then /*forward*/
   (i) Identify the types of constituents which are in Q’ but not in Q”.
   (ii) Modify Q” by adding the identified constituents in the appropriate slot of Q”.
4. if const(Q’) = const(Q”) then /* Not a gapping */
   STOP and RETURN
5. Q’ and Q” are parallel structures which constitute Q.

construct (Q) is a function which returns the number of constituents in Q where Q is any clause.

In the first step the parsed structures from the parser are collected. It contains NPs, VPs and the two clauses. In step two the type of gap is identified. When the constituents in Q
are fewer than the same in $Q''$, then the type of gap is called the backward gap. The constituents in $Q''$, and which are not in $Q'$ are identified. When the identified constituent contains the object, the algorithm is terminated. If object is not present, the clauses are modified by adding the identified constituents in the appropriate slots of $Q'$. Step three identifies the other type of gapping, namely, the forward gapping. The same procedure is performed with minor changes. The constituents which constitute $Q'$ and $Q''$ are identified. Also identified are the constituents which are in $Q'$ but not $Q''$. The clause $Q''$ is modified by adding the appropriate constituents to the appropriate slot of $Q''$. In the fourth step the two clauses are checked and if the constituents are the same in both the clauses, then it is not a gapping construction and the termination condition is used. Depending on the input, any one of the three steps 2, 3, or 4 is fired in accordance with the type of gapping. In step five, we consider that $Q'$ and $Q''$ are two structures of equal magnitude and the gapping is solved.

11. sya:m kuttikale sne:hiccu pakse avanRe bha:rya veRuttu
  syam children like-pst but he-poss wife hate-pst
  (Syam likes children but his wife hates.=Syam likes children, but his wife hates them.)

Input S: sya:m kuttikale sne:hiccu pakse avanRe bha:rya veRuttu

Parsed structure:

sya:m  <N><NOM><subj><m><s><third><+human><+animate><+count>
kuttikale  <N><NOM><obj><n><pl><third><+human><+animate><+count>
sne:hiccu  <V><tran><pr>
pakse  <co-ord>
avanRe  <N><PRO><POSS><m><s><third>
bha:rya  <N><NOM><obj><f><s><third>
veRuttu  <V><tran><pst>

The gapped element: kuttikale
The resolved sentence is sya:m kuttikale sne:hiccu pakshe avanRe bha:rya kuttikale veRuttu.

    raman sita-acc-coord krishnan rukmini-acc-coord love-pst
    (Raman loved Sita and Krishnan loved Rukmini)


Parsed structure:

ra:man  <N><NOM><m><s><third><subj>
si:tayeyum  <N><ACC+coord><f><s><third><obj>
kRisnan  <N><NOM><m><s><third><subj>
rukminiye yum  <N><ACC+coord><f><s><third><obj>
sne:hiccu  <V><tran><pst>

The gapped element: sne:hiccu

The resolved sentence is ra:man si:taye sne:hiccu. kRisnan rukminiye sne:hiccu.

7. Consider the ellipsis, which is the last module in the algorithm.
    The algorithm is as follows:

VII 1. Create a list of words in Q and R with NPs, VPs and clauses identified.
    (from the parser)
2. Identify the question words in the Q.
3. Identify the focus word.
4. For each W in Q.
    if the identical type does not occur in R
    if W is the subj in Q.
    Change W to W' and add to R (by subj change rules)
    else
    if W is the q-word in Q
    change W to W'.
    else add to R

The first step is to identify the structure of the question sentence (Q) and the response sentence(R) from the parser. The question words are identified in the second step. In the next step the focus for the wh words are identified. Step four identifies for each word W.
in Q the identical word in the response. If the word is not found it will check whether W is the subject word in Q and using subject change rule will change the subject and add to the response. If the W is a Q word then it is changed to W' and added to the response. If it is not the Q word then add W to response. Consider the examples.

13. ni: entu kandu?
   you what see-pst
   (What did you see?)

   pattiye kandu
   dog see-pst
   (I saw a dog.)

   The parsed structure

   ni: <N><NOM><subj><n><s><second><+human>
   entu <Wh>
   kantu <V><tran><pst>
   pattiye <N><ACC ><obj><n><s><third><-human>

   The Wh word is entu and its response is pattiye

   The W (subj) is ni: and the subject of response is na:n

   The resolved response is na:n pattiye kandu

14. ni: kalicco:?
   you eat-pst-Qmorph
   (You ate(= Did you eat?))

   kaliccu
   ate-pst
   (Ate(= I ate.))

   Parsed structure:

   ni: <N><NOM><subj><n><s><second><+human>
   kalicco: <Qword>
   kaliccu <V><tran><pst>

   The Qword is kalicco: and the response is kaliccu.
The W subj is ni: and it is changed to na:n

The response: na:n kaliccu.

Turning to algorithm for Hindi anaphora resolution.

1. Pronoun resolution rules are as follows:
   A pronoun P is coreferential with an NP iff the following hold:
   a. P does not precede the NP.
   b. If P is non-possessive, then NP is the subject of the IC in which P does not occur.
   c. If P is possessive, then NP is the subject of the IC in multiple clause constructions or the NP immediately preceding the P.

   The algorithm for the above rule is

I. 1. Create a level of words of S with NPs, VPs, Clauses identified (From parser)
   2. for each pronoun P in S
      (a) if P is poss then
      (i) if S is a complex sentence
         select the NP which is the subject of the IC
      (ii) else /*simple sentence*/
           select the NP which immediately precedes P in S
      (b) if P is non-poss then
      (i) select the NP which is the subject of the IC.
   3. If P precedes the NP then
      STOP and RETURN
   4. The NP is the antecedent of P.

There are four steps in this module. The first one creates the parsed structure from the parser. At the next step pronouns are tackled separately as possessive and non-possessive. If the pronoun is possessive then it is checked whether the pronoun is in a complex sentence. If the pronoun is in a complex sentence the immediate clause in which the pronoun occurs is identified. The subject NP of this clause is chosen. If the pronoun is non-possessive the NP which is the subject of the immediate clause is identified. In the third step NP is checked whether it precedes the pronoun. If it is preceding then the termination condition terminates the process. If it does not precede then the fourth step is
activated, where the NP is accepted as the antecedent of the pronoun. The system produces the following examples.

15. **ra:m** uski **kita:b** sya:m ko di
   
   ram he-poss book syam-acc give-pst
   
   (Ram gave his book to Syam.)

Input S: **ra:m** uski **kita:b** sya:m ko di

Parsed structure:

- **ra:m** <N><NOM><subj><m><Y<third>
- **uski** <N><POSS><pronoun><m><Y<third>
- **kita:b** <N><NOM><obj><n><Y<third>
- **syam** <N><ACC><obj><n><s><third>
- **di**  <V><trans><pst>

anaphoric element: pronoun-uski

anaphora resolution:

The clause which contains the anaphor: **ra:m** uski **kita:b** sya:m ko di.
The clause which contains the antecedent: **ra:m** uski **kita:b** sya:m ko di.

Subject of the clause: **ra:m**.

**ra:m** is the antecedent of **uski**.

2. The next is the reflexive resolution. The rules for reflexives are as follows:

   An emphatic reflexive R corefers with a NP iff
   a. NP is the subject of the clause in which R occurs
   b. If NP is possessive, then the head noun is the antecedent of R

   A non-emphatic reflexive R₁ corefers with a NP respectively
   c. NP is the subject of the clause in which R₁ occurs.
   d. If NP is possessive then the head noun is the antecedent of R₁

The algorithm for the above rules is as follows.

II 1. Create a list of words of S with NPs, VPs and the clauses identified.
   (From the parser)
II 1. Create a list of words of S with NPs, VPs and the clauses identified.
   (From the parser)
2. For each reflexive R in S
   (a) If R is non-emphatic reflexive then
      (i) identify the IC in which R occurs.
      (ii) select the NP which is the subject
      (iii) if the subject is POSS NP then
      (iv) identify the head of the POSS NP
   (b) If R is emphatic then,
      (i) identify the IC of the clause in which R occurs.
      (ii) select the NP which is the subject of IC.
      (iii) if the subject is POSS NP then
      (iv) identify the head of the POSS NP.
3. The NP is the antecedent of R.

There are three steps in reflexive resolution module. The first one creates the parsed structure from the parser. At the next step the reflexives are identified. For a non-emphatic reflexive, the system tries to identify the immediate clause in which the reflexive occurs. The subject of the immediate clause is selected. If the subject is a possessive NP, the head of the possessive NP is identified. The same procedure is followed for emphatic reflexives also. The resolved output is as follows.

16. mo:han ne apnea:p ko shishe me dekha.
    mohan erg refl-acc mirror-loc see-pst
    (Mohan saw himself in the mirror.)

Input S: mo:han ne apnea:p ko shishe me dekha

Parsed structure:

mo:hanne:  <N><NOM><m><s><third><+human><+animate>
apnea:pko:  <R><ACC>
shisheme:  <N><LOC><n><s><third>
dekha:    <V><tran><pst>

Anaphoric element: Reflexive-apnea:p

Anaphora resolution:
Clause in which the anaphor occurs: \textit{mo:han ne apnea:p ko shishe me dekha}

Clause in which the antecedent occurs: \textit{mo:han ne apnea:p ko shishe me dekha}

\textit{mo:han} is the antecedent of \textit{apnea:p}

17. \textit{mo:han ne svayam ko shishe me dekha.}

\textit{mohan erg refl-acc mirror-loc see-pst}

(Mohan saw himself in the mirror.)

Input S: \textit{mo:han ne svayam ko shishe me dekha.}

Parsed structure:

\begin{verbatim}
mo:hanne: <N><NOM><m><s><third><+human><+animate>
svayamko <R>
shiseme <N><LOC><n><s><third>
dekha <V><tran><pst>
\end{verbatim}

anaphoric element: svayamko

anaphora resolution:

Clause in which the anaphor occurs: \textit{mo:han ne svayam shise me dekha}

Clause in which the antecedent occurs: \textit{mo:han ne svayam shise me dekha}

mo:han is the antecedent of svayam

Coming to reciprocal anaphors. The rules for reciprocal anaphors R’ are as follows.

3. A reciprocal R’ co-refers with an NP iff the following holds:
   a. NP is the subject of the clause in which the reciprocal occurs.
   b. NP is plural.
   c. NP precedes the reciprocal.

The algorithm for the reciprocal anaphor

III 1. Create a list of words of S with NPs, VPs and clauses identified.
   (From the parser)

2. For each reciprocal R’ in S
   (i). select the NP, which is the subject of the clause in which R’ occurs.
   (ii). if R’ precede NP
        then STOP and RETURN.
   (iii). if NP is not plural
          then STOP and RETURN.

3. The NP is the antecedent of R’.
In this module there are three steps. The first step creates the parsed structure from the parser. In the second the reciprocals are identified. The NP, which is the subject of the clause in which the reciprocal occurs, is selected. The NP and R’ are checked for precedence. If R’ precede NP then the termination condition is used. Then the NP, is checked whether it is plural or not. If it is not plural, the termination condition terminates the process. In the third step, the antecedent is assigned to R’. The following example is the output of the algorithm.

18. ye bacce ek dusre se ba:t nahi karenge
   this children each other inst talk neg be
   (These children will not talk to each other)

Input S: ye bacce ek dusre se ba:t nahi karenge
Parsed structure:

ye <demon>
bacce <N><NOM><m><pl><third>
ek dusrese <r>
ba:t <V><tran><pst>
nahi <neg>
karenge <V><intran><fut>

anaphoric element: reciprocal-ek dusrese

anaphora resolution

Clause which contains the anaphor: ye bacce ek dusre se ba:t nahi karenge
Clause which contains the antecedent: ye bacce ek dusre se ba:t nahi karenge

The antecedent of ek dusrese is bacce.

4. Now coming to distributive anaphors, the rules of coreference for a distributive anaphor D are as follows: D corefers to an NP iff:
   a. NP is the subject of the clause in which D occurs.
   b. NP is plural.
   c. NP precedes the distributive.

The algorithm for the above rules are as follows.
VI 1. Create a list of words of S with NPs, VPs and two clauses identified.
   (From the parser)
2. For each reciprocal D in S
   (i) select the NP, which is the subject of the clause in which D occurs.
   (ii) If D precede NP
        then STOP and RETURN.
   (iii) If NP is not plural
        then STOP and RETURN.
3. The NP is the antecedent of D.

In this module there are three steps. The first step creates the parsed structure from the
parser. At the second step the distributives are identified. The NP, which is the subject
of the clause in which the distributives occur is selected. The NP and D are checked for
precedence. If D precedes NP then the termination condition is used. Then the NP is
checked for whether it is plural or not. If it is not plural the termination condition
terminates the process. At the next step, the antecedent to D is assigned. The following
example shows the output of the algorithm.

19. mo:han aur ra:m ko apne apne ghar pasand hai
   mohan and ram acc each other house like copula
   (Mohan and Ram like their respective houses.)

The parsed structure:

mo:han  <N><NOM><subj><m><s><third><+human>
aur     <coord>
ra:mko   <N><ACC><m><s><third><+human>
apne apne <D>
ghar     <N><NOM><n><s><third><-human>
pasant hai <V><tran><pst>

The anaphoric element: apna apna

The anaphora resolution:
Clause in which the anaphor occurs: mo:han aur ra:m ko apne apne ghar pasand hai
Clause in which the antecedent occurs: mo:han aur ra:m ko apne apne ghar pasand hai
The antecedent of apne apne is mo:han

5. The gaps are the sixth module in the anaphora resolution component. The gaps are identified by the following rules.

Forward gapping has the following rules.
1. if the explicit constituent is in Q'.
2. V is in Q'.

Backward gapping has the following rules.
1. if the explicit constituent is in Q''.
2. NP is the object of Q''.
3. V is in Q''.
The algorithm for the above is as given below.

V 1. Create a list of words in S with NPs, VPs and two clauses identified.
(From the parser)
2. If const (Q') < const (Q'') then /* backward*/
   (i) identify the type of constituent which is in Q'' but not in Q'.
   (ii) if identified const. contains object STOP and RETURN
   (iii) modify Q' by adding the identified constituents in the appropriate slots of Q'
3. If const(Q') > const (Q'') then forward
   (i) Identify the types of constituents, which are in Q' but not in Q''.
   (ii) Modify Q'' by adding the identified constituents in the appropriate slot of Q''.
4. If const(Q') = const (Q'') then /* not gap*/
   STOP and RETURN
5. Q' and Q'' are parallel structures which constitute Q.

const (Q) is a function, which returns the number of constituents in Q where Q is any clause.

In the first step the parsed structures from the parser are collected. It contains NPs, VPs and the two clauses. In step two, the type of gap identified. If the constituents in Q' are fewer than the constituents in Q'', then the type of gap is backward gap. The type of constituents in Q'' and which are not in Q' are identified. When the constituent identified contains object the termination condition is called for. If object is not present, the clauses
are modified by adding the identified constituents in the appropriate slots of Q'. The step three identifies the other type of gapping, namely the forward gapping. The same procedure is performed with minor changes. The type of constituents which constitute Q' and Q'' are identified. The constituents which are in Q' but not in Q'' are identified. The clause Q'' is modified by adding the appropriate constituents at the appropriate slots of Q''. In the fourth step the two clauses are checked and if the constituents are the same in both the clauses, then it is not a gapping construction and the termination condition is used. In step five, we consider that Q' and Q'' are two structures with independent clause and the gapping is solved.

20. mo:han ne si:ta se aur hari ne gita se sa:di ki
    mohan erg sita inst coord hari erg gita inst marry-pst
    (Mohan Sita and Hari married Gita= Mohan married Sita and Hari married Gita)

Parsed structure:

mo:hanne <N><NOM><subj><m><s><third><+human>
si:tase <N><INST><obj><f><s><third><+human>
aur <coord>
harine <N><NOM><subj><m><s><third><+human>
gitase <N><INST><obj><f><s><third><+human>
sa:diki <V><tran><pst>

Gapped element: sa:diki

The structure: mo:han ne si:ta se sa:di ki
              hari ne gita se sa:di ki

21. si:ta ne ro:ti kha:yi, ca:y pili
    sita erg roti eat-pst tea drink-pst
    (Sita ate roti, drank tea=Sita ate roti and drank tea)

Parsed structure:

si:tane <N><NOM><subj><f><s><third><+human>
The gapped element: si:ta ne

The structure: si:ta ne ro:ti kha:yi
              si:ta ne ca:y pili

VI. 1. Create a list of words in Q and R with NPs, VPs and clauses identified (from the parser)
2. Identify the question words in Q.
3. Identify the focus word.
4. For each W in Q.
   if the identical type does not occur in R
   if W is the subj in Q.
       change W to W' and add to R (by subj change rules)
   else
   if W is the Q word in Q
       change W to W'.
   else add to R

The first step is to identify the structure of the question sentence (Q) and the response (R) from the parser. The question words are identified in the second step. In the next step the focus in R is identified. {Focus is the material that cannot be elided from the response}
In the case of wh-const the focus is the OBJECT while in the case of q-const the focus is the verb. Step four identifies for each word W in Q the identical word in the response. If the word is not found, it will check whether W is the subject word in Q and using subject change rule will change the subject and add to the response. If the W is Q word, then it is changed to W' and added to the response. If it is not the Q word then add W to responds.

Consider the examples.
22. tum kaha gayi thi?
    you where go-pst
    (Where did you go?)

    sku:l
    school
    To school. (I went to the school.)

Parsed structure:

    tum:  <N><NOM><n><s><second><+human>
    kaha <Wh>
    gayithi <V><tran><pst>
    sku:l <N><NOM><n><s><third>

    Wh word: kaha
    The response: sku:l

The structure of the response: me sku:l gayi thi.

From the above we arrive at the following. Both Malayalam and Hindi share more similarity than differences as far as anaphora resolution is concerned. The marked difference occurs in pronoun resolution. Malayalam has a unique solution for possessive pronouns where as for Hindi possessive pronouns does not have. In Malayalam the subject of the same clause is the antecedent where as in Hindi it is the immediate NP which is the solution. Other difference is in Malayalam, at the sentence level, the subject is always the antecedent of the pronoun. In Hindi it is different. It can be the object as in the example given below.

23. ra:m ne sya:m se uski kita:b leli
    ram erg syam inst he-poss book take-pst
    (Ram took his (Syam's) book from Syam)

24. ra:m ne uski kita:b sya:m se leli
    ram erg he-poss book syam inst take-pst
    (Ram took his(Ram's) book from Syam)
Here in the first (22) case the object “syā:m” “Syam” is the antecedent of “uski” “his” where as in the second case the subject “rā:m” “Ram” is the antecedent of “uski” “his”. As far as the other pronouns are concerned both Malayalam and Hindi have the same rules.

The one-pronoun algorithm does not require any change to work for Hindi. The rules for one pronoun resolution are same in the cases of both Malayalam and Hindi.

In the case of reflexives, Malayalam has more rules than Hindi, but in both cases the antecedent is the subject of the clause in which the anaphor occurs. In the case of reciprocals and distributives both obey the same rules. The algorithm does not require any change to accommodate Hindi.

As far as the gapping is concerned, backward gapping in Hindi is different from that of Malayalam. The ellipsis resolution has very little difference. Here the difference is because Hindi does not have “Q-morph” construction, which Malayalam has. This shows that the algorithm developed for Malayalam works for Hindi with minor changes.