Exploring Semantic Information from Hindi Dependency Treebank for Resolving Pronominal Anaphora

Seema Mahato
Research Scholar,
Dr. C.V. Raman University,
BILASPUR (C.G.), INDIA,

Ani Thomas, PhD
Professor, Dept. of Computer Applications,
Bhilai Institute of Technology,
DURG (C.G.), INDIA,

ABSTRACT
Anaphora Resolution is exigent task in almost all NLP applications such as text summarization, machine translation, information extraction, question-answering systems, etc. A lot of work has been done for identifying and still more need to be done for finding the factors responsible for resolving the anaphoras in all languages. An attempt has been made to resolve Hindi pronominal anaphora using syntactic as well as semantic knowledge. The occurrence of particular case markers are found, which exhibit its connectivity with the pronouns leading to the anaphora resolution approach. An algorithm is designed taking into account the roles of subject, object and its impact on anaphora resolution for identifying the noun phrase antecedents of first and second person singular pronouns as well as for a third person singular pronoun and a reflexive pronoun. The algorithm applies on the inputted syntactic representation generated by a Hindi shallow parser. The authors have tested it on a text corpus containing 192 pronoun occurrences. The algorithm correctly resolved the antecedent of 145 pronouns (75.5%) of these pronoun occurrences. Experiments on pronominal anaphora help in analyzing the complexity of problems under consideration and the results of the observations are presented.

General Terms
Natural Language Processing

Keywords
Anaphora, Pronominal Resolution, Dependency Treebank, Case Marker.

1. INTRODUCTION
Machine learning that involve learning from data are also applied to many of the Natural language processing (NLP) tasks. One such core linguistic task is Anaphora Resolution (AR). This resolution includes text analysis techniques which analyze each sentence individually and the anaphora need to be changed with their corresponding nouns. An attempt is made to completely automate the resolving of anaphora in Hindi since the task of AR for English has started quite early and numerous techniques have been devised.

Hindi is a free-word-order language and the classic word order is Subject-Object-Verb (SOV). Hindi language is also characterized by a rich system of case. Pronouns in Hindi take different inflected forms depending on what case they are in and indicate their case by means of different case marker. As per Paninian grammatical model there are six basic karaka (case) relations in Hindi such as “ने”, “से”, “को”, “में”, etc. [1].

Pronominal anaphora resolution refers to the task of finding or identifying noun referents for pronouns. All the efforts are directed towards the resolving the different types of anaphora. The two major classes of pronominal anaphora include personal pronouns and reflexives pronouns. All noun phrases preceding pronouns are usually treated as potential candidates for antecedents while performing pronominal anaphora resolution. Personal pronouns includes such as मैं, हम, तुम, आप, वह, यह in its root form and their inflected form whereas reflexives pronouns includes अपना, अपनी, अपने, अपने आप, स्वयं, खुद.

2. STATE-OF-ART
In English and other European languages, ample amount of research is being done as compared to Hindi and other Indian languages. Few important areas of anaphora resolution have been explored in Hindi too, of which some are found to be noteworthy.

In Hindi and other Indian languages, AR studied is presented by Bharati et al. [2]. Authors developed tests for handling inter-sentential anaphora as well as for intra-sentential anaphora. They worked to handle reference, ellipsis and other phenomena that occur frequently in dialogues involving natural language interfaces to databases but in a restricted domain. The authors had modified Paninian parser to activate the surface level ellipsis handler when the sentence is found incomplete and handled deep level ellipsis during the intermediate representation of the question.

Prasad et al. [3] studied anaphoric reference of third person personal pronouns in Hindi. Their work was based on the principle of centering theory that the grammatical function is important for discourse salience in Hindi Language. They studied factors in Hind related to grammatical function, word order, and information status. They proposed and applied a novel method for determining the relative salience in discourse entities. They applied BFP-algorithm and S-list algorithm to resolve pronouns in Hindi texts and proposed C f-list ranking to these two algorithms. The authors used the notion of the C f-list for computing pronominal antecedents and concluded that in Hindi, C f-list ranking is crucially determined by grammatical role, and information status and word order do not have any independent effect on salience.

AR related to Hindi studied is presented by Sobha et al. [4]. VASISTH-a rule based AR system by them gave a rule based approach for the resolution of anaphora in Hindi and Malayalam as well. The system makes limited use of syntax and uses only morphological markings to identify subject, object, clause etc. It uses limited parsing: the information required from the parser is limited to parts of speech tagging, clause identification, and subject of the clauses and person-number-gender of the NPs. Initially VASISTH was developed and tested for Malayalam, and then modified for Hindi. This system can resolves all referentially dependent elements such as pronominals, non-pronominals, gaps and ellipsis. It is sensitive to ambiguity occurring in pronoun resolution but does not resolve the ambiguity. The system works with high degree of success in the case of Malayalam.

Dutta, et. al. developed a Information extraction system for Hindi texts using heuristic approach to resolve the anaphors and pronoun in descriptive texts which have limited occurrences of first and second person pronouns. The authors have used language parser HPSG (Head-Driven Phrase Structure Grammar) to add more semantic information and semantic constraints into the representation so that resolution yield more accurate results [5].

Dutta et al. [6] applied modified version of Hobbs' Naïve Algorithm for Hindi. The authors have taken into account the subject and object in Hindi sentence for resolving pronominal anaphora. They have tested the algorithm for limited set of sentences. They concluded that the limitation of proposed algorithm can be overcome by considering the semantic information.

Dutta et al. [7] highlighted the importance of anaphora resolution for machine translation application by evaluating the three existing Machine translation systems: AnglaHindi by IIT Kanpur, Matra2 by CDAC Mumbai and Google translation system. They also highlighted the significance of pronominal divergence in machine translation. Authors concluded that pronominal divergence can help in identifying anaphoric and non-anaphoric occurrences of pronoun whereas Case based divergence can be helpful in identifying the correct inflection form for the corresponding pronoun for English-Hindi Machine Translation. The authors stated that, for a successful NLP application the resolution of anaphora is essential.

Dutta et al. presented an enhanced annotation scheme with the semantic information on Emille corpus for indirect anaphora in Hindi [8]. Chatterji et. al studied data driven approach for Anaphora resolution of three Indian languages: Bengali, Hindi, and Tamil [9].

Dutta et al. [10] discussed different issues and challenges in the anaphora in Hindi. The authors have compared different methods for AR related to machine translation. They have also studied and discussed the inflection of root forms of pronouns according to cases and influence of pleonastic "it" in the resolution processes.

Dakwale et. al. proposed a scheme for anaphora annotation in Hindi Dependency Treebank and discussed issues related to anaphora annotation specific to Hindi such as distribution of markable span, sequential annotation, representation format, annotation of multiple referents [11]. The annotation has been done for a limited set of pronominal categories.

Lakhmani et. al. [12] presented a review of work done related to AR in Hindi language. The authors have discussed the issues related to syntactic and semantic structure of Hindi and influence of cases on pronouns. They have also performed manual experiment on different kinds of data set.

Dakwale et. al. presented a hybrid approach for resolving Entity-pronoun references in Hindi by using syntactic information from dependency structures [13]. The approach uses rule-based module to resolve simple anaphoric references and a decision tree classifier to resolve ambiguous instances. The approach relies on grammatical and semantic features.

Two computational models were compared based on Gazetteer method for resolving anaphora in Hindi Language. First model use the recency factor and concept of centering approach whereas second one uses the concept of Lappin Leass approach and animistic factor for resolving anaphors. The authors have analyzed the accuracy of both the model and concluded the best suitable model for Hindi Language [14].

The authors have developed a model for resolving pronominal anaphora in Hindi using Gazetteer method which uses Recency factor as the baseline factor and animistic knowledge to differentiate between animate and inanimate nouns and pronouns. They have conducted and demonstrated three experiments on different data sets consist of 10 to 30 sentences in Hindi Language [15].

At present, AR system for Hindi needs to be tested on fully parsed corpus and longer discourse and corpora.

3. HINDI DEPENDENCY TREEBANK

Dependency Treebank is formed on basis of the linguistics feature of the language. Hindi Dependency Treebank (henceforth HDT) represents a group of adjacent words which are in dependency relation with each other. Each such group is referred to as chunk. HDT includes the following information [16] [17] [18],

- Part-of-speech (POS) Information: Each word (lexical item) in a sentence is annotated with its POS tag such as NN', 'PSP', 'QC', 'NN', 'VM', etc. In grammar, a POS is generally defined by the syntactic or morphological behavior of the word.
- Chunk Information: After annotation of POS tags, each sentence is manually chunked.
- Dependency Information: After POS, and chunk annotation, dependency annotation is done. Predefined dependency tag set is used for annotating the dependency.

For each sentence, the output of HDT in Shakti Standard Format (henceforth SSF) has four columns [16] [17] [18] which are mentioned below,

- *1st Column* represents Tree address as Token id or Offset Address such as 1, 1.1, 2, 2.2 etc.
- *2nd Column* indicates Token or Chunk boundaries i.e. the actual word or word groups in the sentence. The line with a "((" represent the beginning of a word group or chunk and "))" indicate the end of a group. Every word and chunk has a name. The attribute for naming is 'name'.
- *3rd Column* specifies Category or part of speech such as NP, NN, PSP etc. For example, NP: Noun Phrase, NN: Common Nouns or in general any noun; NNP: Proper nouns; PRP: Pronouns; VM : Main or head verb; etc.
- *4th Column* represents Feature structure. It holds the feature information used to store user-defined features which are accessed through their feature names or attribute names. Morphological information, grammatical

roles, semantic information etc. are listed as features in this column.

4. FINDING EQUIVALENCE CLASS FOR RESOLUTION

For each category of similar pronouns, an individual and unique algorithm has been defined. The scope for searching potential antecedent is made for previously 2-3 sentences. Within this scope, the process considers all the NPs with case markers which are preceding an anaphor as potential candidates for antecedents.

In this methodology, the features obtained from treebank are used to develop machine learning techniques to resolve anaphora. For each type of pronoun, a unique and individual module based on a different algorithm has been developed. This module works in two major phases, preprocessing and anaphora resolution. Machine learning techniques are implemented to handle this whole phase.

The preprocessing phase captures the dependency treebank in SSF format generated by Hindi shallow parser for a sentence and save it as text file for further process. Using the machine learning procedure, the preprocessing phase analyzes each token. By carefully distinguishing these tokens, this phase sort out instances of NP, NN, PRP and VM tokens and produces a list of two dimensional dynamic arrays corresponding to instances. It eliminates rest token such as CCP, RBP, XC, JJ, JJP, QC, SYM, etc. These four types of arrays, NP, NN, PRP and VM are the building blocks for simplifying the constraints in the preprocessing phase and deriving the computational mechanisms to be used. The information stored in these arrays is shown in Table 1 and Table 2.

Constraints are then applied to this array in order to produce the "set of competing candidates" to be considered further. Constraints are implemented hardly such as, by extracting nominal and pronominal heads from NPs preceding the pronoun for potential antecedents, ignoring indefinite pronouns such as कुछ, किसी, कौन, etc., finding gender and number (henceforth GN) of all nouns which precede pronoun in case of resolving "वह" anaphora, etc.

After the preprocessing phase gets over, the post processing phase i.e. the anaphora resolution phase begins.

4.1. Data set

The authors experimented upon algorithms which identify pronominal forms of anaphora and derive rules to locate the referent. The data set for training and development contained sentences from different fields mainly related to sport news, political news, and conversation between two person, Hindi films blogs and magazine articles. The post-processing rules are devised by analyzing 19 distinct set of 5 sentences in each selected by the authors. For experiments, 95 sentences of corpus were annotated using Dependency Treebank. In these sentences, 192 pronominal pronouns were identified and out of which 145 get resolved correctly, 13 resolved incorrectly and 34 remain unresolved. A summary of sentences in dataset with other relevant information needed by resolution algorithms for first person singular pronouns (henceforth FPP), second person singular pronouns (henceforth SPP) and APNI ("अपनी") is shown in Table 1 whereas for VEH ("वह ") pronoun, it is shown in Table 2.

4.2.Algorithms

The anaphora resolution phase comprises of four major sub-modules for resolving anaphora belongs to FPP, SPP, third person pronoun VEH and reflexive pronoun APNI as shown in Figure 1.

The combination of linguistic annotation and case marker in the training data set made it possible to examine the linguistic structure of sentences. There are six basic karaka and their combinations with noun were considered as relevant term patterns for identifying antecedents. The most common term pattern [NP ((N, CM))] consisted of a noun (N) as a head and a case marker (CM). Altogether, 6 different tag combinations were considered as relevant term patterns, of which the majority were of different simple noun phrase patterns such as ((N,"ने")), ((N,"से")), ((N,"को")), ((N,"का")), ((N,"पर")) and ((N,"में")). The ongoing research till now worked on and given preference to ((N, "ने")), ((N, "से")) and ((N, "को")) patterns only and included in NP array resolving pronominal anaphora.
To resolve anaphora belongs to different class of pronouns, a unique and individual module based on a different algorithm has been developed. The algorithms for resolving anaphors are explained below. These algorithms use certain Hindi linguistic rules and are incorporated in a machine learning methods. The algorithms are an attempt to provide a domain independent anaphora resolution module. A couple of array L1 and L2 are used to store a list of words belongs to FPP and SPP respectively such as L1={'मैं',मैंने','मुझे','मुझको','मुझसे','मुझपर','मेरेपर','मुझमें'} and L2= {'तु','तुम','तुने','तुम्हे','तुमको','तुझे','तुझको', 'तुझसे','तुझपर' ,'तेरेपर','तुझमें'}.

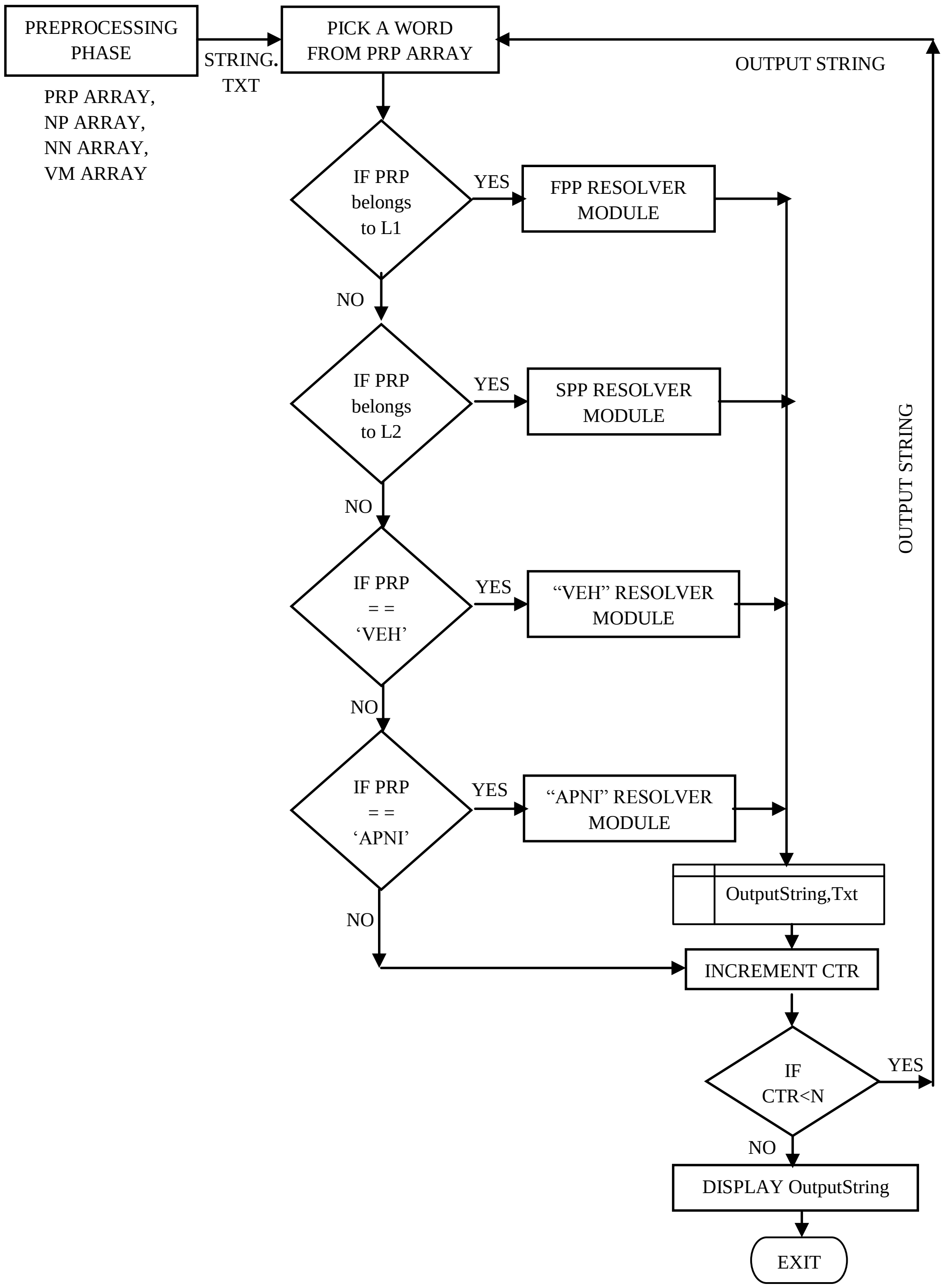

Figure 1: ANAPHORA RESOLUTION PHASE

Table 1: Summary of information extracted for algorithms related to FPP, SPP & APNI

Sentence id	**Sentences**	**PRP ARRAY**	**N**	All **NPs with CM** preceding pronouns in **((N,CM))** patterns extracted from **HDT**	**NP ARRAY** (NPs with Case Marker ने,से or को)	**R**	**RC**	**RNC**	**UR**
[1].	पाकिस्तान के निलंबित ऑफ स्पिनर सईद अजमल ने कहा कि मैं वर्ल्ड कप में खेलना और अपने देश को अच्छा करते हुए देखना चाहता हूं ।	मैं	1	((पाकिस्तान , का)), ((अजमल , ने))	((अजमल , ने))	1	1	0	0
[2].	आईएसआईएल का ट्विटर हैंडल चलाने और इस संगठन के लिए काम करने के आरोप में गिरफ़्तार मेहदी मसरूर बिस्वास ने कहा है कि पुलिस मुझको मार सकती	मुझको	1	((आईएसआईएल ,का)), ((संगठन,का)), ((आरोप,में)), ((बिस्वास , ने))	((बिस्वास , ने))	1	1	0	0

	है।								
[3].	बीबीसी रेडियो स्टोक की एक प्रस्तोता ने अपने श्रोताओं से शराब पीकर कार्यक्रम प्रस्तुत करने के लिए माफ़ी मांगी है कि मुझको माफ़ करना, मैं नशे में थी, मुझसे गलती हो गई	मुझको मैं मुझसे	3	((स्टोक ,का)), ((प्रस्तोता , ने)) , ((श्रोता ,से)), ((नशे ,में))	((प्रस्तोता ,ने)) , ((श्रोता ,से))	3	3	0	0
[4].	लॉ इंटर्न केस के आरोपी सेवानिवृत्त न्यायमूर्ति ए.के.गांगुली ने भारत के प्रधान न्यायाधीश पी सतशिवम को पत्र में पूछा है कि आरोप के दिन मैं सुप्रीम कोर्ट का हिस्सा नहीं था फिर मुझपर जांच के लिए कमेटी कैसे बनी।	मैं मुझपर	2	((केस ,का)), ((गांगुली , ने)), ((भारत , का)), ((सतशिवम ,को)), ((पत्र ,में)), ((कोर्ट ,का))	((गांगुली , ने)), ((सतशिवम ,को)),	2	2	0	0
[5].	पटना के मुख्यमंत्री जीतन राम मांझी ने कहा कि समाज में यह भ्रांति फैलाई जा रही है कि नीतीश कुमार और मुझमें मतभेद है।	मुझमें	1	((पट ,का)), ((मांझी ,ने)), ((समाज, में))	((मांझी ,ने)),	1	1	0	0
[6].	युवक ने अपनी शर्त नेता को बताई कि यदि तुम मुझे नोट दो तो मैं तुम्हे वोट दूंगा।	अपनी तुम मुझे मैं तुम्हे	5	((युवक ,ने)), ((नेता, को))	((युवक ,ने)), ((नेता, को))	5	5	0	0
[7].	ऑस्ट्रेलियाई क्रिकेटर फिलिप ह्यूज को अंतिम विदाई देने मैक्सविल पहुचे हजारो लोग की भावना यही थी अलविदा ह्यूज, तुझे न भूल पाएंगे	तुझे	1	((ह्यूज ,को)), ((लोग, का))	((ह्यूज ,को))	1	1	0	0
[8].	योगा स्वामी बाबा रामदेव ने विद्यार्थीयो से कहा कि मैं अब योग की भाषा बोलूँगा , परिवर्तन की भाषा तुम्हे बोलना है.	मैं तुम्हे	2	((रामदेव,ने)), ((विद्यार्थीयो,से)), ((योग, का))	((रामदेव,ने)), ((विद्यार्थीयो, से))	2	2	0	0
[9].	रमन ने गौतम से कहा कि मुझे तुझपर भरोसा है	मुझे तुझपर	2	((रमन ,ने)), ((गौतम, से))	((रमन ,ने)), ((गौतम, से))	2	0	0	0
[10].	मुंबई में किताब विमोचन पर पहुंचे अभिनेता आमिर खान ने अपनी जिंदगी और करियर की कई बातें साझा की। उन्होंने कहा कि मैं अपनी शर्तों पर जिंदगी जीता हूं।	अपनी मैं अपनी	3	((मुंबई,में)), ((विमोचन,पर)) , ((खान,ने)), ((करियर,का)) , ((साझा,का))	((खान,ने))	3	0	0	0
[11].	रोहन ने अपनी किताबें गौतम को दे दी।	अपनी	1	((रोहन,ने))	((रोहन,ने))	1	0	0	0
[12].	अमिताभ बच्चन को अपनी फिल्म "पा" पर गर्व हैं	अपनी	1	((अमिताभ बच्चन, को))	((अमिताभ बच्चन, को))	1	0	0	0

Where N=no. of anaphora in sentence, R= Resolved, RC= Resolved Correctly, RNC= Resolved Incorrectly, UR= Unresolved in Table 1 and Table 2.

Table 2: Summary of information extracted for VEH algorithm

Sentence id	**Sentences**	**PRP ARRAY**	**N**	All **NPs with CM** preceding pronouns in **((N,CM))** patterns extracted from **HDT**	**NP ARRAY** (NPs with Case Marker ने,से or को)	**NN ARRAY** (Extracted Noun from NP array with gender and number)	**VM ARRAY** (Extracted **VGF**, Head Verb with gender and number)	**R**	**RC**	**RNC**	**UR**
[13].	गोपाल ने आयुषी से कहा कि वह खरीदारी के लिये जा रहा है।	वह	1	((गोपाल,ने)), ((आयुषी, से))	((गोपाल,ने)), ((आयुषी, से))	((गोपाल,m.sg)) , ((आयुषी,unk,))	((ले,m,sg,या_ जा_रह+या_है))	1	1	0	0
[14].	मालती को फलो की बागवानी का शौक है और सतीश को सब्जीयों की खेती पसन्द है। वह पौधे खरीद कर लाती है।	वह	1	((मालती,को)), ((बागवानी,का)), ((सतीश, को)), ((सब्जीयों, का))	((मालती,को)), ((सतीश, को))	((मालती,f,sg)), ((सतीश,m,sg))	((लाती,f,sg))	1	1	0	0
[15].	पुलिस ने चोर का पीछा किया लेकिन वह भाग गया।	वह	1	((पुलिस,ने)) , ((चोर,का))	((पुलिस,ने))	((पुलिस,f,sg))	((गया,unk,,))	0	0	0	1

Algorithms start as,

1. Initialize N i.e. number of pronouns to resolve in a sentence and CTR=1.
2. For each remaining word in PRP array, compute antecedent from NP or NN array depending upon the category of pronoun. If a pronoun belongs to FPP, SPP or APNI then NP array is taken into account and if the pronoun is VEH, then the algorithm will search NN array to identify antecedent.
3. If an anaphora in PRP array belongs to L1, then execute the FPP resolver module.
4. If an anaphora in PRP array belongs to L2, then execute the SPP resolver module.

5. If an anaphora does not belong to either L1 or L2, then compare it with VEH or APNI and execute the respective module.
6. Represent the sentence as output string by replacing or substituting pronoun with its antecedent and consider this output string as new input for next pronoun to resolve.
7. Increment CTR on resolving each anaphora.
8. Repeat step1 to step 5 till the PRP array get reduced and none of the pronoun is remain to resolve.
9. If there is no pronoun left to resolve in PRP array, then display the final output string and exit from the procedure.

4.2.1. Algorithm for resolving anaphora belongs to first person singular pronouns (FPP)

The FPP class contains pronouns such as "मैं", "मैंने", "मुझे", "मुझको", "मुझसे", "मुझपर" ,"मुझमें". "मैं" is the root form of all the pronouns belong to this class and usually refer to speaker of a communication which is expressed or narrated in the same sentence. Except "मैं", all are inflected form of "मैं". "मैंने" refers the one who carries out the action. "मुझे" and "मुझको" usually refers 'to me'. "मुझसे" refers "from me or with me". "मुझपर" and "मुझमें" refers "on me".

The algorithm for FPP resolver module involves the information in the NP array and "ने" case marker. The algorithm proceeds by searching the "ने" case marker in NP array. On identifying the "ने", it retrieve the word before "ने" (adjacent noun) which is separated by a delimiter ',' and store the word in a temporary variable and further copy the variable value to a one dimensional array named FOUND. Then it replaces the anaphora in the input sentence by the word stored in FOUND array appended with proper endings and save the updated sentence in the output string.

4.2.2. Algorithm for resolving anaphora belongs to second person singular pronouns (SPP)

The pronouns such as "तु" ,"तुम", "तुने", "तुम्हे", "तुमको", "तुझे" ,"तुझको", "तुझसे", "तुझपर", "तुझमें" belongs to SPP class and all are formal usage of "You" refers "to you" , "for you" and "on you".

For resolving SPP, the algorithm considers the information in the NP array with "से" or "को" case marker. The algorithm begins by selecting the NP array and searching the "से" case marker in array. On recognizing the "से", it pick the word before "से" (the neighbouring noun) which is separated by a delimiter ',' and store this word in a temporary variable and further copy the variable value to a one dimensional array named FOUND. Further this pronoun is replaced by noun in FOUND array with proper suffix and after replacement the updated sentence is saved in the output string. If in case, "से" is not available in NP array then the algorithm search for को" case marker and consider its neighbouring noun as potential antecedent.

4.2.3. Algorithm for resolving third person pronoun "वह"

"वह" pronoun can be resolved by using NN and VM array. NN array is composed of noun which precedes pronouns along with their gender and number. VM array store the head verb with its gender and number (GN). The gender of pronoun can be known from its neighbouring verb only. Further, GN of this verb is compared with GN of all nouns in the NN array. Once GN of verb and noun gets matched, the algorithm replaces "वह" with this noun. If GN not get match, then few rules have been crafted for handling such cases.

4.2.4. Algorithm for resolving reflexive pronoun "अपनी"

The two criteria for resolving "अपनी" anaphora are, firstly when it occurs following FPP, SPP or वह such as मुझे अपनी, तुम अपनी, वह अपनी, likewise and secondly when it appears alone in the sentence. In former case, FPP, SPP and वह all should be resolved prior resolving "अपनी". The antecedent for "अपनी" in this case can be identified by the antecedent of corresponding FPP, SPP or "वह".

In later case, "अपनी" is resolved using three case markers "ने", "से" and "को". To resolve "अपनी" in this case, the subject of the sentence with either one of these case markers in NP array is considered as its antecedent. Generally, the initial NP in the sentence indicates the subject of sentence if the Hindi sentence is written in its default word order i.e. subject-object-verb. On discovering any one of these case markers in NP array (whichever comes first), the adjacent noun is selected as antecedent.

5. RESULT AND DISCUSSIONS

The designed algorithm relies on rules derived from syntactic and semantic knowledge to select the antecedent noun phrase (NP) of a pronoun from a list of candidates. Most of these examples are taken from different web sources of texts on which the algorithm is trained. Experiment with the dataset includes all those pronouns for which algorithms have been defined. An empirical evaluation has been conducted for pronominal anaphora resolution of FPP, SPP, "वह" and "अपनी". Table 3 shows a summary of the results that have been observed and the counts of the number of pronouns at each syntactic level are also provided. The authors have provided examples of its output for different sorts of cases.

After trying all algorithms, an accuracy of 87.01% in case of resolving FPP, SPP reports 66.07% while "VEH" reports 40% and "APNI" reports 79.5% of accuracy. If these anaphora are ranked on the basis of their resolution then FPP gain high score and lowest score is shown by VEH.

Table 3: Results – Pronominal Resolution Module: Evaluation of individual pronoun and their category wise accuracy.

Category of pronouns	Anaphora	***No. of Anaphora***	Total Resolved		Unresolved	Anaphora wise Accuracy %	Category wise Accuracy %
			Resolved Correctly	Resolved Incorrectly			
First Person Singular Pronouns (FPP)	मैं	20	18	1	1	90.00	87.01%
	मैंने	12	11	1	0	91.67	
	मुझे	15	15	0	0	100.00	
	मुझको	9	9	0	0	100.00	
	मुझपर	9	7	0	2	77.78	
	मुझमें	6	3	0	3	50.00	
	मुझसे	6	4	1	1	66.67	
Second Person Singular Pronouns (SPP)	तु	6	0	3	3	0.00	66.07%
	तुम	14	13	0	1	92.86	
	तुम्हे	7	7	0	0	100.00	
	तुझे	2	1	0	1	50.00	
	तुझको	3	0	0	3	0.00	
	तुमको	8	8	0	0	100.00	
	तुझपर	5	5	0	0	100.00	
	तुझमें	9	2	3	4	22.22	
	तुझसे	1	1	0	0	100.00	
	तुमसे	1	0	0	1	0.00	
TPP	वह	15	6	0	9	40%	
RP	अपनी	44	35	4	5	79.5%	

TPP: Third Person Singular Pronoun, RP: Reflexive Pronoun,
Anaphora wise Accuracy= (***Resolved Correctly*** / ***No. of Anaphora***) * 100,
Category wise Accuracy = (Total ***Resolved Correctly*** / Total ***No. of Anaphora***) *100, for each category of pronouns.

6. CONCLUSION

It is found that the algorithms for resolving anaphora such as मुझमें, मुझसे, तुझे, तुझमें, तुमसे, तु, तुझको, and वह still need much refinement and improvement even though a minor modification has been made for each one of them while substituting the respective antecedent and rewriting the output string after each run. The algorithms for resolving FPP and SPP are based on case markers. However, since "से" and "ने" postposition in Hindi are highly overloaded, its presence alone cannot be a decision factor always. The gender and number for resolving "वह" cannot be the only parameter. As far as "अपनी" anaphora is concerned, resolved as per its occurrence in two forms. It is essential to notice that the there are fewer instances in which cue phrases are unavailable which are employed to identify antecedents. These had lower down the accuracy of algorithms. Even in those cases, the antecedents can be identified by other relations but this need to be investigated.

An attempt has been made to resolve pronominal anaphora using syntactic and semantic knowledge. Nonetheless, this approach will prove useful in the long run, as machine learning will gradually reduce the preprocessing and resolution time that will make more compound sentence analysis more economically feasible. Further, the authors wish to extend the work for identifying both intra-sentential and inter-sentential antecedents of pronouns in text by considering more term patterns and including new ones. The algorithms are not examined for longer discourses.

7. REFERENCES

[1] Mahato, S. and Thomas, A. 2015. Machine Learning Approach For Resolving Pronominal Anaphora Using Hindi Dependency Treebank, In Proceedings of BITCON-2015 Innovations For National Development. IJAERS/Vol. IV,Issue II,Jan.-March, 2015,Pages 155-159

[2] Bharati, A., Bhargava Y. K. and Sangal, R. 1993. Reference and ellipsis in an Indian languages interface to database. Computer science and informatics, IIT Hyderabad, VOL 23; NUMBER 3, pages 60

[3] Prasad, R. and Strube, M. 2000. Discourse salience and pronoun resolution in Hindi. In Penn Working Papers in Linguistics: 6.3, 189-208

[4] Sobha, L. and Patnaik, B.N. 2000. Vasisth: An anaphora resolution system for Malayalam and Hindi. In International Conference ACIDCA'2000

[5] Dutta, K., Kaushik, S. and Prakash, N. 2004. Information extraction from Hindi texts.1911-1914, LREC

[6] Dutta, K., Prakash, N., and Kaushik, S. 2008. Resolving Pronominal Anaphora in Hindi using Hobbs algorithm. Web Journal of Formal Computation and Cognitive Linguistics, 10

[7] Dutta, K., Prakash, N., and Kaushik, S. 2009. Application of pronominal divergence and anaphora resolution in English-Hindi machine translation. Research Journal "POLIBITS" Computer Science and Computer Engineering with Applications, VOL 39, pages 55-58

[8] Dutta, K., Prakash, N., and Kaushik, S. 2011. Machine learning approach for the classification of demonstrative pronouns for indirect anaphora in Hindi news items. Prague Bulletin of Mathematical Linguistics, VOL 95, pages 33-50

[9] Chatterji, S., Dhar, A., Barik, B., Moumita PK, Sarkar, S., and Basu, A. 2011. Anaphora resolution for Bengali, Hindi and Tamil using random tree algorithm in Weka. In Proceedings of ICON2011 NLP TOOL CONTEST: 9th International Conference on Natural Language Processing

[10] Pal, T. L., Dutta, K., and Singh, P. 2012. Anaphora resolution in Hindi: Issues and challenges. International Journal of Computer Applications, VOL 42; pages 18

[11] Dakwale, P., Sharma, H., and Sharma, D. 2012. Anaphora annotation in Hindi dependency treebank. 26th Pacific Asia Conference on Language, Information and Computation, 391–400

[12] Lakhmani, P., and Singh, S. 2013. Anaphora resolution in Hindi language. International Journal of Information

and Computation Technology. ISSN 0974-2239 Volume 3, Number 7, pp. 609-616

[13] Dakwale, P., Mujadia, V., and Sharma, D.M. 2013. A hybrid approach for anaphora resolution in Hindi. International Joint Conference on Natural Language Processing, pages 977–981

[14] Singh, S., Lakhmani, P., Dr. Mathur, P. and Dr. Morwal, S. 2014. Comparative performance analysis of two anaphora resolution systems. International Journal in Foundations of Computer Science & Technology (IJFCST), Vol.4, No.2

[15] Lakhmani, P., Singh, S., Dr. Mathur, P. 2014. Gazetteer method for resolving pronominal anaphora in Hindi language. International Journal of Advances in Computer Science and Technology, Volume 3, No.3

[16] Bharati, A., Sangal, R., Sharma, D.M., and Bai, L. 2006. Anncorra: Annotating corpora guidelines for pos and chunk annotation for Indian languages. In Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad.

[17] Bharati, A, Sharma, D.M., Husain, S., Bai, L., Begum, R., and Sangal, R. 2009. Anncorra: Treebanks for indian languages, guidelines for annotating Hindi treebank (version 2.0). Retrieved from http://ltrc.iiit.ac.in/MachineTrans/research/tb/DS-guidelines / DS-guidelinesver2-28-05-09.pdf

[18] http://ltrc.iiit.ac.in/analyzer/hindi/

[19] http://ltrc.iiit.ac.in/full_analyzer/hindi/

[20] http://en.wikipedia.org/wiki/Natural_language_processing

[21] Jain, S., Jain, N., Tammewar, A., Bhat, R., A., and Sharma, D.M. 2013. Exploring Semantic Information in Hindi WordNet for Hindi Dependency Parsing. In The Sixth International Joint Conference on Natural Language Processing