Anaphora Resolution in Hindi

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computational Linguistics

by

Praveen Dakwale
201023001
`praveen.dakwale@research.iiit.ac.in`

Language Technologies Research Center (LTRC)
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2014

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Anaphora Resolution in Hindi ” by Praveen Dakwale, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Dipti Misra Sharma

Dedicated to My parents, Mother Nature and Humanity

Acknowledgments

I would like to thank all the persons in my life for their support and encouragement for my research. First of all, I would like to thank my advisor Prof. Dipti Misra Sharma for constant guidance, encouragement and support. Without guidance and motivation from her, my research would not have been possible. Her deep knowledge of linguistics, NLP or other subjects have always helped me, not only towards my research but also in other domains of life. Her dedication towards her subjects and work has inspired me to remain dedicated towards the goals for life. Next, I would like to thank the person who has helped me to change and evolve my thoughts and interest towards linguistics and science in genral: Late Prof. Laxmi Bai. Through her lectures, a new world of scientific inquiry and depth of linguistic studies was revealed to me. Her insights and teachings will always keep me informed and motivated for my future research. I would also like to thank other faculty members of LTRC family: Prof. Rajeev Sangal, Dr. Soma Paul and Dr. Radhika Mamidi whose suggestions and guidance have always directed me throughout courses and research. Their support has been incomparable throughout my stay at LTRC. Next I would like to thank all my seniors and juniors. I am specifically thankful to Himanshu, Riyaz bhai and Himani didi for all the continuous support, suggestions and criticism during my stay in IIIT. Time spent with you people have changed my thinking towards life and world. I am also grateful to all the LTRC members: Sanket Sir, Sambhav, Sandhya, Khushboo, Rafiya didi, Pratibha didi, Arpita, Urmi, Rahul, Ankhush, Kunal, Karan, Rishabh, Naman and all the other LTRC members. I would also like to thank my batch-mates: Jatin, Vikarm, Nikhil, Swagatika, Sumit, Ravi, SRK, Vasu, Pabhu and Gopi. Life without you guys would have been boring and tedious throughout these years. Special note of gratitude for Vandan who has contributed to my work with complete dedication and helped me complete my work. Finally, I would also like to thank all the staff members of LTRC family including Rambabu Sir, Srinivas Sir, Kumarswamy Sir, Priti Mam, Nandini Mam, Anita mam and other members who have made this journey easier.

Abstract

In Natural Language, Anaphora is an expression which refers to another expression in its context. This referred expression, called the antecedent provides the information for interpretation of the anaphora. In Natural Language Processing (NLP), Anaphora Resolution is the task of identifying the referent of the anaphora. Anaphora resolution is required in various NLP applications such as information extraction, summarization and Machine translation. While the task of anaphora resolution for English has been studied to a sufficiently great extent and various techniques have been proposed for it, the research for Anaphora resolution for Indian Languages has been very limited. Hence, in this thesis, we aim to develop resources and explore features and techniques for Anaphora Resolution in Hindi.
There are three important contributions of this thesis. First, we developed an anaphora annotated corpus which is required for experiments in Anaphora resolution. Towards the development of this corpus, we proposed a scheme or framework for anaphora annotation. Our goal, with this scheme, is to identify and resolve various consistency issues associated with previous schemes or framework which are used for anaphora and co-reference annotation in English and other languages. Using the proposed scheme, we annotated anaphora references over Hindi Dependency Treebank. Second, we present a hybrid approach to resolve Entity-pronoun references in Hindi. Most of the existing approaches, syntactic as well as data-driven, use phrase-structure syntax for anaphora resolution, we instead, explore the utility of dependency structures as a source of syntactic information. In our approach, dependency structures are used by a rule-based module to resolve simple anaphoric references, while supervised learning algorithms are used to resolve more ambiguous instances using grammatical and semantic features. Our results show that, use of dependency structures provides syntactic knowledge which helps to resolve some specific types of references. Semantic information such as animacy and Named Entity categories further help to improve the resolution accuracy. Finally, we also conduct some preliminary experiments in Event anaphora and co-reference resolution. Similar to Entity anaphora resolution, for event and co-reference resolution too, we explore use of dependency structures in a rule based settings. Our experiments show that even with limited data and using simple syntactic and semantic constraints, reasonable resolution accuracy can be achieved for Event anaphora and co-reference resolution.

Contents

Chapter Page

List of Tables

Table Page

Chapter 1

Introduction

Natural Language Processing aims at designing and building software that will analyze, understand, and generate human languages, so that eventually humans will be able to communicate with the machines using natural languages. Processing or understanding of human languages, especially written texts, requires analysis and implementation at different linguistic levels such as morphological-analysis and Part-of-Speech tagging at word level; Chunking and clause identification at word group level; syntactic parsing at the sentence or structural level; semantic analysis at the level of meaning and finally discourse analysis at the discourse or text level. Linguistic analysis and hence the NLP research at each level builds on the analysis obtained in the lower levels. For example, sentence level parsing requires information such as morphological properties and grammatical categories (part of speech) of the individual constituent words. Similarly, semantic analysis requires word level information as well as syntactic information. Thus the discourse level analysis benefits with the information from all the previous stages i.e. world level analysis, syntax and semantics.

It is due to this dependence on earlier stages that the performance of available tools and techniques at this level, especially for resource poor languages is very low. Discourse level analysis involves various sub-problems that deal with relationships between sentences and larger linguistic units. In this thesis, we discuss one such discourse level phenomenon known as “anaphora” and its resolution.

Anaphora is sometimes characterized as the phenomenon whereby the interpretation of an occurrence of one expression depends on the interpretation of an occurrence of another, or whereby an occurrence of an expression has its referent supplied by an occurrence of some other expression in the same or another sentence. [37]

In linguistics, anaphora is the use of an expression which refers to another expression in its context. The referring expression itself is called anaphor and referred expression is called the antecedent (referent). The antecedent provides the information for the interpretation of the anaphor[67]. Consider a simple example:

(1) Director of XYZ bank ordered an investigation against the regional manager. He claimed that the manager is involved in recently discovered loan scams.

In the above example, ‘He’ is referring to a previously mentioned entity ‘Director of XYZ bank’. The process of identification of the referent is known as ‘Anaphora Resolution’

The term used for reference to an expression occurring later than the pronoun is ‘cataphora’. However,in computational linguistics, the usage of the term ‘anaphora’ stands for the reference to any expression which may come earlier or after the anaphor.

While there has been significant research for anaphora resolution in English and other languages, NLP research including anaphora resolution for Indian languages is limited. In this thesis, we aim to study and develop resources which can be used for anaphora resolution in Hindi and explore linguistic features which are helpful in the resolution process.

To conduct experiments in Anaphora Resolution, substantial data annotated with anaphoric references is required for training and evaluation in statistical as well as rule based anaphora resolution algorithms. However, no large corpus annotated with anaphora information or any annotation framework is available for Hindi. Hence, in order to develop a corpus which is consistent and usable for anaphora resolution in Hindi, we first develop an annotation scheme to handle various consistency issues associated with anaphora annotation and later based on this scheme, we aim to annotate corpus which can be used in resolution algorithms.

In this chapter, we first discuss applications for anaphora resolution, issues and challenges associated with it and the various linguistic terminologies and categorization associated with the problem.

1.1 Why anaphora resolution is required ?

The general motivation of anaphora resolution comes from its real time applications in NLP. We discuss below some of the important applications:

1.1.1 Question Answering/Information Extraction:

Question answering task stands for searching the answer for a Natural language query from a given text[32]. Most approaches in basic question answering systems involve simple string matching or a parse analysis of the sentence.

As discussed in [63], anaphora resolution is an important sub-task for Question answering system. We explain this by following example. Consider a two sentence text as below:

(2) Nelson Mandela was born to Gadla Henry Mphakanyiswa and Nosekeni Fanny. He served as President of South Africa from 1994 to 1999.

Given the above text, and a query as *“Who served as President of South Africa from 1994 to 1999?”*, a search using string matching or parse tree analysis of the second sentence will yield the answer as *‘He’* which is not a complete answer in itself. Thus in order to obtain the complete answer, the system must have the knowledge that the pronoun *‘He’* is referring to another entity in the previous sentence i.e. *‘Nelson Mandela’*.

1.1.2 Automatic Summarization

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document[69]. Though there are different variants and methods for text summarization, the most common strategy used is to select only the important information and drop the additional information. Although summarization is used for large texts, for simplicity consider example (2) to be summarized. A possible summary of the text in example (2) may be obtained by joining the sentences as follows:

"Nelson Mandela who was born to Gadla Henry Mphakanyiswa and Nosekeni Fanny, served as the President of South Africa from 1994 to 1999".

This again requires the knowledge of the referent of the pronoun *'He'*. Thus, anaphora resolution process has important usage in Summarization.[60].

Summarization without resolving the anaphors may result in sentences which have anaphors, but no apparent referent, thus making the summary incomprehensible and incoherent.

1.1.3 Machine translation:

Machine translation stands for usage of machines and software for the translation of text from one language to another. In order to simplify the complexity of translation, majority of the machine translation research has focused on sentence level translation, i.e. translation of individual sentences considering them in isolation. However, obtaining a coherent translation of larger texts in a real time setting, requires the discourse level information. In some cases, translation of pronouns from one language to another requires knowledge of the referents of the pronouns[46]. This is because in some languages pronouns have different forms depending on the morphological properties (such as gender, number, animacy etc) of the words. Hence to obtain the correct translation of such a pronoun, knowledge of the referent of the pronoun is required which comes from anaphora resolution.

1.2 Issues and challenges in anaphora resolution:

Anaphora and co-reference resolution are considered difficult problems in NLP. Even after considerable advances in anaphora resolution in last few years, anaphora resolution algorithms have not achieved accuracy high enough to be used in higher level NLP applications [45]. This is due to various challenges associated with anaphora resolution. We discuss some of the important challenges.

- As already been discussed in the introduction, anaphora resolution being a discourse level problem, needs various lower level linguistic knowledge such as morph analysis, Part of Speech (POS) tags, syntactic and semantic structures. Due to this, anaphora resolution algorithms have to rely on pre-processing NLP tools and analysis. As a result, the low performance and errors in these pre-processing tools adds to degradation in accuracy of the anaphora resolution algorithms [45].

While some of these pre-processing tools like POS-tags have achieved fairly reliable performance, various other steps such as Parsing, Named Entity recognition require considerable improvement in order to be used for anaphora resolution. Due to its dependence on the previous linguistic information, anaphora resolution algorithms either have to rely on the performance of these pre-processing tools or assume that the input data to the algorithms have all the required information available.

- Even after assuming that all the required linguistic knowledge is available in corpus or through pre-processing tools, anaphora annotated corpus is required for training or learning in statistical approaches. Many research projects have been focusing on development of anaphora annotated corpus. Annotation tasks follow an annotation scheme or framework. This annotation framework should be consistent and simple enough so that it is able to cover wide range of linguistic phenomena and at the same time is simple enough to ensure consistent and faster annotation. There are various annotation issues which are although faced in all the languages, but are more frequent in free-word-order languages such as Hindi. However, most of the available annotation frameworks or schemes do not give emphasis to these issues and hence fail to identify some important instances pertaining to reference annotation. In chapter-2, we discuss in detail about these annotation issues.
- There is well established research available on linguistic theories such as Government and Binding [18] which give required background for resolution of structural references. Also, there has been significant research on identification and utilization of various factors that provide the background for anaphora resolution algorithms. However, there is still ambiguity on identifying the core set of factors that can be used for anaphora resolution [45]. While, most of the related works divide these factors in constraints and preferences, some [17] propose that all the factors should be considered preferences with relative weight-age. Moreover, these factors may also vary across languages and domains. Many of the features used in English may not be useful or available for other languages.

Besides the anaphora resolution challenges, there are also challenges in other tasks related to it. In the next section, we discuss some of the related terms and their differences in order to differentiate anaphora resolution from other related tasks.

1.3 Classification of References:

In computational linguistics, terms related to anaphora resolution have been used quite interchangeably. Although, a particular term may be used to include broader reference relations, the emphasis of different tasks changes with the terminology. We explain below these related terminologies and highlight their differences:

1.3.1 Co-reference Vs Anaphora:

The term co-reference is often used for all types of reference relations including anaphoric and non-anaphoric references. In most of the research, co-reference resolution task also includes anaphora resolution. However, there are subtle difference between the two terms. There is co-reference between two expression when both of them unambiguously refer to a unique physical or conceptual entity. On the other side, a reference is called anaphoric only when an expression refers to another expression called the antecedent and this antecedent is required for the correct interpretation of the anaphor. Hence, co-reference is an equivalence relation, whereas anaphora is non-transitive [28], hence not necessarily an equivalence relation. Reconsidering example (1), the reference relation between 'He' and 'Director of XYZ bank' is anaphoric, while the relation between 'regional manager' and 'manager' in second sentence is anaphoric as well as co-referential.

(3) Director of XYZ bank ordered an investigation against the regional manager. He claimed that the manager is involved in recently discovered loan scams

For simplicity, we consider all pronominal references to be anaphora and non-pronominal references to be co-referential. In this thesis, while focus of our research is anaphora (pronominal) reference resolution, we also perform some preliminary experiments for co-reference resolution.

1.3.2 Concrete (Entity) Vs Abstract (Event) reference

Reference relations can be classified on the basis of linguistic category of the expression that an anaphor is referring to. This classification includes Concrete and Abstract anaphora[26].

- Concrete (Entity) reference: In concrete reference an anaphor refers to a concrete(individual) entity like noun phrase(person,place etc), quantifiers etc. Reconsidering example (1)

  (4) Director of XYZ bank ordered an investigation against the regional manager. He claimed that the manager is involved in recently discovered loan scams

  Since, the pronoun *'He'* refers to a concrete entity *Director of XYZ bank*, this is an example of concrete reference.
- Abstract reference: Abstract reference includes the cases where an anaphor refers to an an abstract object such as event, proposition or clause. In abstract reference the referent is usually large textual expression. For example:

  (5) Many trees and poles were uprooted yesterday during the heavy rainfall. Due to this, many roads were blocked and traffic transportation became difficult.

  In above example *'this'* refers to an abstract entity *'Trees and poles being uprooted'*. Specifically the verb represents the event being referred. Thus, this is an example of Abstract reference.

In many languages same lexical forms of pronoun can be used to refer to Concrete as well as Abstract referents depending on the context and for many pronominal form, it is not possible to decide only on the basis of lexical forms whether an anaphor is referring to a Concrete or an Abstract referent. We explain about this ambiguity in detail in chapter-2.

1.4 Anaphora categories and Pronominal forms in Hindi

Hindi is an Indian language belonging to Indo-Aryan language family. It is one of the most spoken out of 22 official languages of India with around 180 million native speakers. Hindi has a rich morphology and relatively free word order. The default word order in Hindi is Subject-Object-Verb(SOV).
In this section, we discuss the categorization of anaphoric categories in Hindi, which is important for designing the rules and algorithm for anaphora resolution[24]. There are two grammatical categories in Hindi which can be anaphoric. These are *Pronouns* and *Demonstratives*. Besides these two, there are also cases of gaps. However, in the surface form in Hindi, these cases can be grammatically categorized in one of the above discussed categories.

1.4.1 Pronouns and their forms

By pronoun, we here refer to words which have a grammatical category or POS-tag as *'Pronoun'* in a give context. We consider following categorization of pronominal forms as discussed in [24]:

1.4.1.1 Reflexives

A reflexive pronoun is a pronoun that is preceded/succeeded by noun, adjective, adverb or pronoun to which it refers (its antecedent) within the same clause. In generative grammar, a reflexive pronoun is an anaphor that must be bound by its antecedent[72]. Reflexive pronouns are primarily used in three situations: when the subject and object are the same (e.g., He watched himself on TV.), as the object of a preposition when the subject and the object are the same (e.g., That man is talking to himself.), and to emphasize the subject through an emphatic pronoun (e.g., They ate all the food themselves.)

Hindi has two types of reflexives :

- **Possessive reflexives:** Possessive reflexives are only used in possession relations within the same clause and are different from third person possessive pronouns. They are not inflected with the gender and number of the possessor, but with that of the possessee.They include अपना (*apana*), अपनी (*apanii*), अपने (*apane*). Consider following example:

  (6) अभय अपने घर गया ।
  abhaya (his own) home went
  'Ram went to his (own) home'

  In above example paronoun अपने (*apane*) is reflexive which refers to the subject of the clause अभय (*'abhaya'*) and shows the possession of the घर (*home*) to the subject.

- **Non-possessive reflexives:** Non-possessive reflexives can be used in any participant position, but mostly used in object position or to shows the emphasis. They include अपने आप(apane-aap), स्वयम्(swayam), खुद(khud) representing 'self' for different persons. Consider following example:

  (7) अभय खुद ही घर की सफाई करता है ।
  abhaya himself of home cleaning do
  'abhaya himself do cleaning of house'

  In above example. reflexive pronoun खुद (*'khud'*) which refers to the subject अभय (*'abhaya'*) and is used to emphasize the subject.

1.4.1.2 Relative pronouns and Correlative:

Relative: A relative pronoun is a pronoun that marks a relative clause within a larger sentence. It is called a relative pronoun because it relates the relative (and hence subordinate) clause to the noun that it modifies. A relative pronoun links two clauses into a single complex clause. It is similar in function to a subordinating conjunction. Unlike a conjunction, however, a relative pronoun stands in place of a noun. For example, in Hindi जो, (*'which'*) and its inflected forms are used as relative pronouns.

(8) बदमाशो से दस बैग बरामद हुए जिनमे वे चोरी का सामान ले जाते थे
From thugs ten bags seized in which they looted items used to carry

'Ten bags were seized from the thugs in which they used to carry the looted items.'

In above example, the relative clause वे चोरी का सामान ले जाते थे (*'they used to carry the looted items'*) relativizes the Noun phrase बैग (*bags*) and is linked to the main clause via pronoun जिनमे (*'jinme'*) which is an inflected form of जो (*jo*). In case of relative pronoun, its referent is the Noun phrase itself that it relativizes or modifies. Thus in above example, the referent of the pronoun जिनमे (*'jinme'*) is the NP बैग (*bags*).

Correlative: Hindi generally prefers the so-called relative correlative construction, wherein a relative pronoun is used along with another corresponding pronoun. The pronouns are used in pairs which correlate two clauses. In English, a slightly awkward example is:

(9) The man who is standing there, he is my brother.

A Hindi equivalent of above example is : जो आदमी वहां खडा है, वह मेरा भाई है । While the Hindi sentence is perfectly grammatical, the English equivalent looks awkward.

1.4.1.3 Personal Pronouns :

Personal pronouns are associated with a particular gramatical person, that is, first, second or third person. It is important to note here that the term 'Personal' does not mean that it only applies in reference to *'Persons'*. Personal pronouns can refer to animals, objects and sometimes organizations or groups. Personal pronouns have different inflected forms for number and respect(honorific) and they also exhibit different forms for case marking.

- First Person : First person pronoun include मैं (I), and हम (we) and their different case forms such as मुझे (‘*to me*’), हमे (‘*hame*’) etc.
- Second Person : Hindi has three second persons, तु (‘*tu*’),तुम (‘*tum*’)and आप (‘*aap*’), all meaning (*you*), where first two are singular forms for formal and informal usage and the third one is used for plural and for respect. They are also marked for different cases in Hindi.
- Third person : Third person pronouns are difficult to resolve in Hindi, as they are also the forms of demonstrative determiners. They are also marked for number and case. We categorize third person pronouns into proximal and distal forms. Proximal pronouns and their case forms are frequently used as abstract as well as concrete references.
  - Distal : The root forms of distal pronoun are वह(he/she/it) and वे(They/Those). They have different case marked forms such as उसे (he/she: accusative),उसने (he/she: nominative),उन्हे (they: Accusative),उन्होने (they: nominative)etc
  - Proximal : The root forms of proximal pronoun are यह(he/she/it) and ये(These). The different case forms are इसे (Accusative),इसने (Nominative),इससे (Ablative) etc.

1.4.1.4 Locative pronoun :

Locative pronouns refer to location or places. They include वहां(‘there’) and यहां(‘here’). Although, generally locative pronouns are not classified separately and are considered a form of personal pronouns only, we consider ‘locatives’ as separate pronoun class. The reason for this separate classification is that in the dependency framework that we use in our work, separate labels are used to represent the locative case (the place where the action take place), thus it can help to identify the referents of the locative pronouns.

1.4.1.5 Indefinite pronouns:

An indefinite pronoun refers to something that is not definite or specific or exact.Indefinite pronouns include quantifiers (some, any, enough, several, many, much); universals (all, both, every, each); and partitives (any, anyone, anybody, either, neither, no, nobody, some, someone). Many of the indefinite pronouns can function as determiners. In Hindi, कोई (‘*someone*’) and कुछ (‘*something*’) are the root forms of indefinite pronouns. They also have inflected forms like किसे (‘*to whom*’), किससे (‘*from whom*’) etc.

1.4.2 Demonstratives:

Besides pronouns, the other grammatical category that can be anaphoric are demonstrative. Demonstratives are deictic words (they depend on an external frame of reference) that indicate which entities a speaker refers to and distinguishes those entities from others. Demonstratives do not individually refer to an entity, but they specify the entity referred by some other referring expression. In the Hindi dependency treebank, demonstrative are annotated with DEM pos-tag(as per the AnnCorra dependency guidelines) [12]. For example:

(10) श्रीलंका में आठ बजे ही भूकंप के झटके महसूस हो गये थे ।लेकिन मौसम विभाग ने
In srilanka at eight'o clock earthquake's tremors felt were. but weather department
यह सूचना प्रशासन को नहीं दी ।
this information to administration did not give

'Earthquake tremor's were felt in Srilanka at eight'o clock. But weather department did not give this information to administration'

In above example, यह (*'this'*) is a demonstrative because it does not directly refer to any entity, but specifies the Noun सूचना (*'information'*) which in actual refers to the *'information'* that is provided in the first sentence.

1.4.3 Gap

Besides the other two linguistic elements which can be termed as anaphoric are gap and ellipsis. Gaps are the instances where an element is omitted in a sentence to avoid redundancy. Sometimes gaps are anaphoric when a reference (not necessarily pronoun) is dropped. In such cases, the pronoun or the referring expression needs to be interpreted. Consider following example:

(11) चीन पाकिस्तान का तो मित्र है लेकिन भारत का शत्रु है
china paksitan's friend is. but india's enemy is

'China is Pakistan's friend. But (it) is India's enemy.'

In the Hindi example above, the possible pronoun वह (*it*) is dropped between लेकिन (*'but'*) and भारत का (*'India'*). However, dropping in English makes it ungrammatical. In case of resolving gaps, it is required to first identify the dropped pronoun or entity in order to resolve it.

As discussed above, due to the indefiniteness in identification of *'demonstratives'* and *'Gap'*, they require additional world and contextual knowledge, hence in our work, we only focus on resolution of **'Pronominal'** anaphora. Also, since there is no specific referent for *'Indefinite'* pronouns in the text, we also exclude them from the resolution.

1.5 Related terminologies:

Our discussion of anaphora resolution involves use of some linguistic terminologies. These linguistic terminologies have been discussed in detail in related literature. However, as a ready reference, we discuss here some of the relevant concepts and terminologies.

1.5.1 Part of Speech:

In grammar, a part of speech also known as word class, lexical class is a linguistic category of words, which is generally defined by the syntactic or morphological behavior of the lexical item in question. Common linguistic categories include noun and verb. Parts of speech is one of the most important information required for any NLP task or application. Part of Speech tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context. [70]. Anncora guidelines [13] define a complete list of Part of speech used to mark POS tags in Hindi dependency treebank which we use in our work. Some of the important tags are as follows:

NN : Common Nouns or in general any noun; NNP : Proper nouns; PRP : Pronouns; VM : Main or head verb; VAUX : Auxiliary verb; DEM : Demonstratives; PSP : Post positions; CC : Conjuncts

1.5.2 Chunks:

Chunks are group of closely related words, but unlike phrases they do not have syntactic behaviour. A typical chunk consists of a single content word surrounded by a constellation of function words [1]. Chunks are normally taken to be a ’correlated group of words’. Chunks are important in dependency representation and parsing because usually in free word order languages, chunks are the minimal units which do not have internal movement of its component words. Since, dependency structures do not follow order of words in a sentence, important dependency relations are realized and represented between chunks. For example [1]:

(12) ((बदमाशो से)) ((दस बैग)) ((बरामद हुए)) ((जिनमे)) ((वे)) ((चोरी का)) ((सामान)) ((ले जाते थे))
From thugs ten bags seized in which they looted items used to carry

‘Ten bags were seized from the thugs in which they used to carry the looted items.’

1.5.3 Syntax:

In linguistics, syntax is “the study of the principles and processes by which sentences are constructed in particular languages”” [19]. The term syntax is also used to refer to the rules which govern and describe the structure of Natural languages. Linguistic research in syntax is focused on analysis of languages in terms of such rules. There are different theoretical approaches and formalisms which are used to describe the syntactic structure of the sentences. These formalisms vary on their basic idea of how the complete structure and meaning of a sentences is realized from their component elements in

[1] In above example: Brackets “((” and “))” show the chunk boundaries. A detail discussion about chunk annotation can be found in [13]

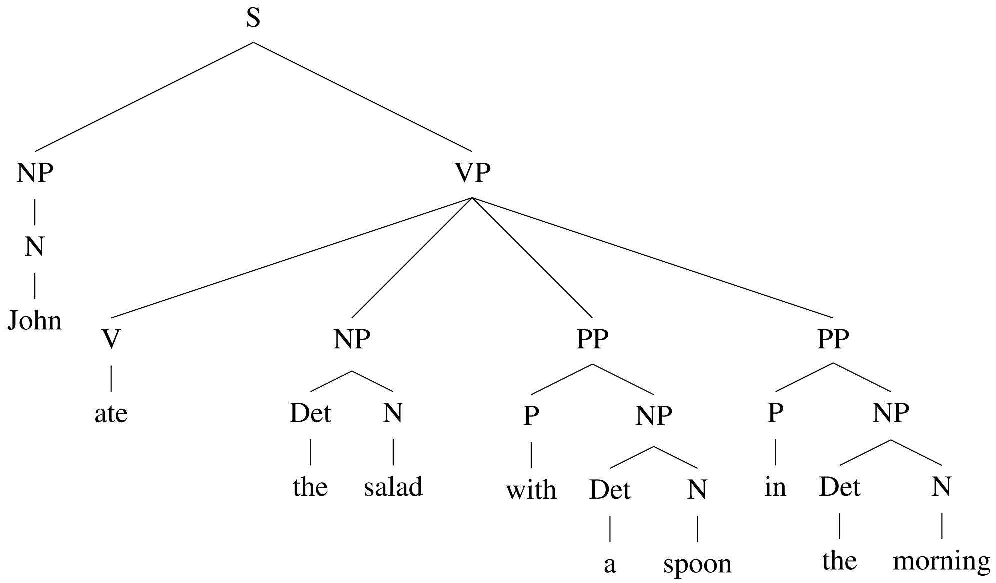
terms of the representation. We discuss here two such formalism which are most frequently used and discussed in theoretical and computational linguistics research.

1.5.3.1 Constituency grammar and Phrase structure syntax:

The idea behind phrase structure syntax is that sentences consist of groups of words called constituents or phrases which belong to certain categories based on their linguistic behavior, and in order to produce linguistically correct sentence for any particular language, these constituent can combine only in some specific ways which are govern by specific rules known as phrase structure grammar for that language. Moreover, these rules and combinations have recursive behavior, that is, phrases are combined recursively to produce larger phrases which are finally combined to produce sentences. Consider following example:

(13) John ate the salad with a fork in the morning

The following diagram shows the phrase structure of the above sentence based on given specific phrase structure grammar rules:



In a phrase structure tree, a terminal symbol representing the sentence node is the root of the tree. Each non-leaf node represents a non-terminal which is constituent of further non-terminals or terminals as are described by the rules of the phrase structure grammar. The leaf nodes of the tree represent the words in the sentence.

1.5.3.2 Dependency grammar and structure

Dependency grammar is based on the idea that the relations between words in a sentence are one to one unlike phrase structure syntax. The term dependency stands for the dependence between the words

in a sentence, that is, words in a sentence are dependent on other words forming a head-modified relationship where one word is head and some other words are its modifier. This head-modified relationship recursively result in dependency structures of larger lengths. The specific type of relationship may vary between different dependency framework but should essentially be modifier-modified relation.
Most dependency frameworks consider verb to be the root of the sentence and its arguments or participants of the action represented by the words attached under the verb. These arguments can further have modifiers. The figure below shows the dependency structure of the example (14)

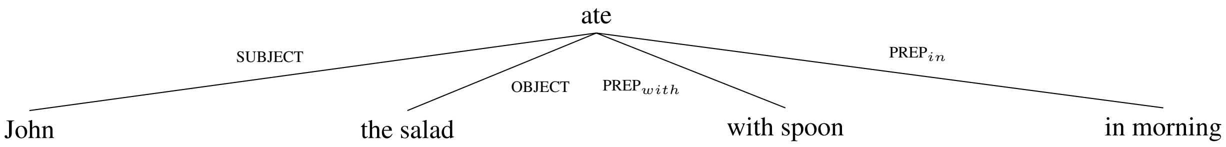


Figure 1.1 Dependency Structure Example

As seen from above figure, the head of the tree is the verb and all the arguments of the verb are represented as children of the root verb. The edges of the tree are labeled where the label represents the relationship between the words connected by the edge.

Difference between Phrase structure and Dependency structure: The conceptual difference between the phrase structure and dependency structure is that the former represents the structure of the sentence as obtained from combination of recursive constituent called phrases governed by grammar rules. As a result, the nodes in the tree can represent either words or phrases. On the other hand dependency structure represents the structure obtained by representation of one-to-one correspondence between the words in a sentence which is based on the head-modifier relationship. As a result, the nodes in the sentence necessarily represent the words or chunks.
Secondly, the phrase structure syntax necessarily shows the actual order of the words in the sentence. A left to right traversal of the leaf nodes of the tree will give the sentence in correct order, while in dependency structure, nodes need not follow any specific word order. Thirdly, edges in the dependency tree are usually labeled where the label represents the relation between the two words, while there is no need of labels in the phrase structures because the edges only represent the constituents of a phrase and not any semantic relation.

1.6 Our resource: Hindi Dependency Treebank and Shakti Standard Format

Since, we aim to annotate the anaphora relations over Hindi dependency treebank and use this data further for anaphora resolution experiments, we explain the treebank and its representation format in brief:

The ‘Hindi/Urdu Dependency Treebank’ is being developed as a part of the Multi-Representational and Multi-Layered Treebank for Hindi/Urdu [14]. Aim of this Treebank is to develop a dependency annotated corpus, but various other linguistic information such as morph, POS tag, chunk boundaries etc have also been annotated in it. The dependency annotation in this treebank is as per Computational Paninian Grammar(CPG-henceforth) based framework, as is explained in [8] and [9]. This framework is based on the notion of ‘*karaka*’ which are syntactio-semantic relations representing the participant elements in the action specified by the verb and it emphasizes the role of case endings or markers such as post-positions and verbal inflections. There are nearly 45 relations. The detailed description of these relations are given in Hindi Dependency tag-set[2]. We describe some of the relations and their brief meaning with the help of following example :

(14) अभय ने दिल्ली में रवि के भाई को अपनी कार दी ।
abhay.ERG delhi.LOC ravi.GEN brother.ACC his car gave

‘abhay told vijay that in delhi he gave his car to ravi‘s brother.’

Figure (1.2) shows the CPG based dependency structure of example(15)

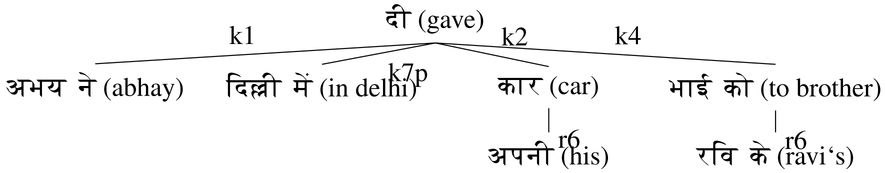


Figure 1.2 Dependency Example

Label	CPG relation	Meaning
k1	*karta*	Most independent entity which carries out the action
k2	*karma*	The entity on which action is carried out
k4/k4a	*sampradan*	Experiencer/reciever
k7p	*apaadan*	Location
r6	*sambandh*	Genitive/Possessive
rh	*hetu*	Purpose
rs	*samaanadhikaran*	Equivalance
ccof	*conjunction*	Conjunction

Table 1.1: CPG relations and equivalents

Table 1.1 shows the interpretation of the dependency relations and their approximate meaning. These dependency relations are only roughly similar to thematic relations. While grammatical roles are purely

[2] (http://ltrc.iiit.ac.in/MachineTrans/research/tb/dep- tagset.pdf)

syntactic and thematic roles are purely semantic in nature, the ‘karaka’ are syntactico-semantic. As animacy is an important information for Hindi parsing[10], a part of treebank has also been annotated with animacy information for Noun phrases. Three values have been annotated for animacy attribute : ‘human’ for human entities, ‘animate’ for non-human but animate entities and ‘rest’ for inanimate entities. Also, we used NE-Recognizer for Hindi to get the Name entity categories.
The representation format of the Treebank(SSF)[11] is Shakti Standard Format (SSF) which is a highly readable representation for storing language analysis. The information on the node is of attribute-value type, where the features are represented as values of some pre-defined attributes (e.g. name, morph, dependency relation etc). A complete description of the SSF is available at [11]. The Table 1.2 shows the layered approach of the SSF of a text with two sentences :

<sentence id = “1” >			
Offset	Token	Category	Feature structure
1	((	NP	<fs name=‘NP’ drel=‘r6:NP2’ >
1.1	अभय (abhay)	NN	<fs af=‘अभय,m,sg,3,d,,’ name=‘अभय’ >
1.2	के (ke)	PSP	<fs af=‘का,psp,m,sg,3h,d,,’ name=‘के’>
	))		
2	((	NP	<fs name=‘NP2’ drel=‘k1:VGF1’ >
2.1	पिता (pitaa)	NN	<fs af=‘पिता,m,sg,3,d,,’ name=‘पिता’ >
	))		
1	((	NP	<fs name=‘NP3’ drel=‘k1s:VGF1’ >
2.2	शिक्षक (shikhsak)	NN	<fs af=‘शिक्षक,m,sg,3,d,,’ name=‘शिक्षक’ >
	))		
3	((	VGF	<fs name=‘VGF1’ >
3.1	हैं (hain)	VM	<fs af=‘है,v,any,sg,3,,hain,hE’ name=‘हैं’ >
	))		
</sentence >			
<sentence id = “2” >			
4	((	NP	<fs name=‘NP’ drel=‘k1:VGF1’ >
4.1	वे (ve)	PRP	<fs af=‘वह,any,sg,3,d,,’ name=‘वे’ >
	))		
5	((	NP	<fs name=‘NP2’ drel=‘r6:NP3’ >
5.1	मेरे (mere)	PRP	<fs af=‘मै,any,sg,1,o,,’name=‘मेरे’ >
	))		
6	((	NP	<fs name=‘NP3’ drel=‘k1s:VGF1’ >
6.1	मित्र (mitra)	NP	<fs af=‘मित्र,m,sg,3,d,,’ name=‘मित्र’ >
	))		
7	((	VGF	<fs name=‘VGF1’ >
7.1	हैं (hain)	VM	<fs af=’है,v,any,sg,3,,hain,hE’ name=‘हैं’ >
	))		
</sentence >			

Table 1.2: SSF format

As shown in Table 1.2, sentences in a text are divided into ‘<sentence >’ nodes, with a unique id for each sentence. The SSF representation for a sentence consists of a sequence of trees. Each tree is made up of one or more related nodes. A node has properties which are given by prop-name and prop-val. For example, a node may have a word वह associated with it along with gender *‘male’*. These may be stored or accessed using prop-name TKN , and gender attribute, respectively. Every node has four system properties:

- Address : Human readable tree address, represented by first column. (1, 1.1, 2, 2.2 etc)
- Token : The actual word in the sentence or word groups, represented in second column (वे, मेरे etc)
- Category or part of speech : Represented by third column (NP, NN, PSP etc)
- Others - Used to store user-defined features which are accessed through their feature names or attribute names. Represented in the fourth column such as (<fs af=‘मैं,...... >)

Inside the sentence description, each line represents a word/token or a group except for lines with "))" which only indicate the end of a group. The line with a "((" represent the beginning of a word group or chunk and the 3rd column and 4th column for this line respectively represent the category and feature structure values for the corresponding word group.

Features for each token or word group are represented as attribute-value pairs. For example line :

```
1.1     abhaya    NN     <fs af=‘abhay,m,sg,3,d,,’ name=‘abhay’>
```

represents the token ‘अभय् (abhay)’ for which address in first column is 1.1, token is ‘अभय’, category or part of speech is ‘NN’ and feature structure is represented by

```
<fs af=‘abhaya,m,sg,3,d,,’ name=‘abhay’>
```

Similarly, first line of the following node description represents details of the node, in which second column ‘NP’ represents the category of word group as Noun phrase, while the attribute name=‘NP’ represents the attribute ‘name’ and value as ‘NP’.

```
1    ((    NP   <fs name=NP drel=r6:NP2 >
1.1  abhay NN   <fs af=abhay,m,sg,3,d,, name=abhay >
1.2   ke   PSP  <fs af=ka,psp,m,sg,3h,d,, name=ke>
      ))
```

The dependency relation in this treebank are marked between chunks or word groups using an attribute *‘drel’* and value of the form ‘label:head’ where *‘head’* represents the ‘name’ attribute of the head of this node in the dependency tree and *‘label’* represents the dependency relation of this node with its head. In above example, *drel=‘r6:NP2’* implies that in the dependency structure, the head of this node is the node with name ‘NP2’ (node with name=‘NP2’ and token as पिता in this case) and the dependency relation between the two is *‘r6’ (sambandh).*

1.7 Key contributions:

There are three main contributions of our work:

- In this work, our main contribution is the development of an approach for Entity anaphora resolution in Hindi. Towards this, we aim at exploring linguistic features that can be used for anaphora resolution, especially the possibility of using dependency structures, as a source of syntactic and semantic information. Based on the patterns in the dependency structure and the constraints, we derived rules which can be implemented in a rule based algorithm to resolve different categories of pronominal references. To resolve more ambiguous references which remain unresolved in the rule based approach, we incorporated other features such as Named entity categories and Animacy in a learning based module. Our results showed that promising performance can be achieved using only rules based on dependency structure and agreement features. This performance can be further improved by considering other semantic and salience features.
- In order to ensure consistent and efficient annotation of corpus with anaphora information which can be used in the resolution algorithm, we proposed an annotation scheme which aims to handle and resolve annotation issues with earlier annotation framework. The main feature of this scheme is division of referent span into head and modifiers with the help of dependency structures of the referent. This division ensures increased inter-annotator agreement and allows annotation of difficult references. Using this scheme, we annotated around 12000 anaphora references over 827 texts of Hindi dependency Treebank.
- Though our main focus in this work is on resolving Entity anaphora references, we also conduct preliminary experiments for Event anaphora and Co-reference resolution. Again, our results show that for resolving these problems too, dependency structures and relations can be used as a valuable source of syntactic information.

1.8 Outline

The thesis is organized into six chapters and a brief outline of the following chapters is as follows:
Chapter 2 In this chapter we explain our work on development of an anaphora annotated corpus. We discuss the major annotation issues with most commonly used previous anaphora annotation frameworks. We, subsequently propose an annotation scheme which aims at handling these issues. We also present the result of an inter-annotator agreement experiment that we conducted over our scheme and its comparison to previous framework.

Chapter 3 This chapter discusses in detail, the theoretical and computational background of anaphora resolution algorithms. Along with this, previous related work is discussed in detail. This includes core approaches in anaphora resolution, earlier work in anaphora resolution in Hindi and available previous work which explores dependency relations for anaphora resolution to some extent.

Chapter 4 In chapter 4, we discuss in detail our experiments for Entity anaphora resolution in Hindi. We discuss two important experiments in this chapter. First is to implement a hybrid approach which use deep linguistic knowledge available in the data. In the second experiment, we discuss use of shallow features for anaphora resolution in Hindi which can be extracted from limited linguistic knowledge.

Chapter 5 Though the major focus of our work is Entity anaphora resolution, we also conducted some preliminary experiments in Event anaphora and co-reference resolution, which we discuss in chapter 5.

Chapter 6 We summarize and conclude our work in chapter 6 and also discuss future work.

Chapter 2

Creating Anaphora annotated corpus

Availability of annotated corpora is an important requirement of most NLP tasks. For supervised machine learning techniques, large amount of data is required for training NLP system. Moreover, annotated corpora is required for feature extraction in unsupervised techniques and for linguistic analysis to derive rules in rule-based methods.

While a sufficient corpora with anaphora annotation for English and other languages like Spanish, Czech etc. are available, for Indian languages, such corpora are scarce. Hence in order to experiment with anaphora resolution in Hindi, an appropriate quantity of anaphora annotated corpus is required. Hence, we first aim at extending the dependency annotated (Hindi Dependency TreeBank) corpus with anaphoric relations or links. Hence we propose an anaphora annotation scheme (Dakwale et,al 2012) [23] in accordance with the representation format (SSF)[11] of the Treebank, that uses attribute-value pairs to represent linguistic information. In this scheme, we attempt to address some of the issues that are commonly faced while annotating anaphora and require efficient handling. Although the scheme is developed in accordance of the structure of the Dependency Tree-Bank, it is convertible to other formats of annotation as well.

We also conduct inter-annotator experiments to compare the reliability of scheme with MUC annotation framework which is most commonly used annotation framework for anaphora annotation.

2.1 Anaphora annotation task:

In simplest terms the annotation task is: For each anaphoric expression, a link should be annotated which connects the anaphora and its referent or shows the referent for each pronoun. However, this general definition, raises some other question and issues.

First, what should be the representation format of the link. The representation of the link should be robust in order to be able to represent different types of complex relations and should also be consistent with the representation format of the data or corpus on which the annotation is being carried out. In the subsequent sections, we discuss how earlier schemes attempted to develop consistent annotation framework, later we also discuss some issues with these schemes and how our proposed scheme attempts to improve the consistency and robustness by improving earlier schemes.

The next question is: What are the categories or expression that should be considered in annotation. This question arises because as discussed in chapter-1, there could be expression with different grammatical categories that could be referencing to another entity and hence are anaphoric. However, annotation of different categories may involve different complexities and annotation issues which may need separate handling. In this work, we only focus on pronominal annotation, that is those references which have a category or POS tag as pronoun. There are also other categories like demonstratives and Noun phrases which are anaphoric or co-referential. We discuss the annotation of co-reference in chapter-5, we do not consider annotation of demonstratives in this work.

Third, What all information should be represented in the anaphora annotation? That is, whether the links should contain information about anaphora, referent and the context ? Should there be any additional information important for the anaphora resolution that should be annotated along with the links. We explain these issues along with the description of our scheme in section 2.2.2.

2.2 Related Work

In recent years, due to increasing interest in development of statistical systems for anaphora resolution, there have been significant attempts for creation of anaphora annotated corpora and annotation schemes. The most well known among these are Message Understanding Conference (MUC-7) annotation scheme [31] and its extensions, which are used for co-reference annotation via markup tags. The MATE/GNOME project has another important scheme suitable for different types of dialogue annotations [53]. (Hajikova et, al, 2005) [39] is also a notable work towards annotating co-reference relations in a dependency TreeBank (Czech, PragueDT). Some other proposed schemes are, in Spanish and Catalan [56, 47] and in Basque [3] for 3LB corpus. The above mentioned schemes are used for anaphora annotation in English and various other languages. As per our knowledge, only known attempt for anaphora annotation in Hindi is [57] in EMILLE [44] corpus. However, they only aim at annotation of demonstrative pronouns and detailed issues of annotation have not been discussed in their work.

In subsection 2.2.1, we briefly describe the annotation framework for MUC-7 co-reference task which is most commonly used annotation scheme for annotating anaphora and co-reference relation. In section 2.2.2, we also discuss some of the annotation issues and challenges associated with MUC-7 and other schemes. In section 4 and 5, we discuss our proposed annotation scheme which aims to handle these issues consistently.

2.2.1 MUC annotation framework

The notation for co-reference in MUC framework (Hirschman et al. 1997) [31], uses SGML tagging within the text stream. The basic annotation is based on the notion of co-indexing the two co-referring entities by establishing some type of link between an explicitly marked pair of noun phrases.

For example, referring expressions and their antecedents are tagged as follows:

```
<COREF ID=``100''> Lawson Mardon Group Ltd.</COREF>
said <COREF ID=``101'' TYPE=``IDENT'' REF=``100'' >
it</COREF> ...
```

In the above example, the pronoun "*it*" is tagged as referring to the same entity as the phrase, "*Lawson Mardon Group Ltd*".

The 'ID' and 'REF' attributes are used to indicate that there is a co-reference link between two strings. The 'ID' is arbitrarily but uniquely assigned to the string during markup. The 'REF' uses that 'ID' to indicate the co-reference link. This framework further uses illustrative attributes such as "TYPE", "MIN", "STATUS" etc. Details of these attributes can be found at [31]

MUC guidelines further elaborates on identification and annotation of "Markables". Markables are the lexical expressions, acting as potential candidates which are either referred by another referring expression or can be part of a reference chain. In other words, markables are any textual elements (annotation guidelines may differ on which elements can be considered markables) which can participate in an anaphora or co-reference relation. In above example, *"Lawson Mardon Group"*, *"Lawson"*, *"Lawson Mardon"*, *"it"* are all markables.

Following is a detailed example from MUC-6 co-reference annotation task definition, as taken from MUC-6 Wall Street Journal Article.

```
 <s> <COREF ID="0">Ocean Drilling & Exploration Co.</COREF> will sell
<COREF ID="3" MIN="business"><COREF ID="2" TYPE="IDENT" REF="0">its</ COREF>
contract-drilling business</COREF>, and took a $50.9 million loss from
discontinued operations in <COREF ID="12" MIN="quarter">the third quarter
</COREF> because of the planned sale. </s>
<s> <COREF ID="9" TYPE="IDENT" REF="2" MIN="company">The New
Orleans oil and gas exploration and diving operations company
</COREF> added that <COREF ID="10" TYPE="IDENT" REF="9">it</COREF>
doesn't expect any further adverse financial impact from the
restructuring. </s>
```

2.2.2 Issues in Anaphora annotation

While MUC annotation framework has been used for annotating co-reference relations in most of the Treebanks and Shared task data, there are some challenges and issues (such as ambiguity, disagreement, and complex annotation) associated with this framework and other previously proposed schemes, which effect the consistent and efficient annotation of the anaphora/co-reference links. We discuss some of these issues with reference to MUC framework in this section. In section 2.5, we show how our proposed scheme attempts to resolve these problems.

2.2.2.1 Coordination or Multiple Non-continuous Referents

Coordination and its connection with plural reference is a difficult annotation issue for work on empirical anaphora and information structure. For example :

(15) अभय और मोहन मेरे मित्र हैं। वे मुम्बई में रहते हैं।

Abhay.NOM and mohan.NOM my friends are. They mumbai in live

'Abhay and mohan are my friends. They live in mumbai'

In above example "*they*" has a plural referent which includes "*Abhay*" and "*Mohan*". MUC framework annotates coordination by defining the markable as the complete minimal string containing both the heads as shown below:

'<COREF ID="1" MIN="*Abhay and Mohan*">Abhay and mohan </COREF>are my friends. <COREF ID="2" REF="1" TYPE="IDENT">They </COREF>live in mumbai'

However, this may cause inaccuracy in resolution because the resolution algorithms may look for a plural referent for "*they*", but the annotated coordination phrase can not capture this plurality. Moreover, it fails to annotate non-continuous plural referents which are not in coordination but are separated by intervening text. Consider following example :

(16) राम कल शाम मोहन के घर गया था।वे कई दिनों बाद

Ram.NOM yesterday evening mohan.GEN home went They many days after

एक दूसरे से मिले।

with each other met.

'Ram went to mohan's home yesterday evening. They met each other after many days.'

In the above example, again referent of "*it*" are "*Ram*" and "*Mohan*", but annotating complete phrase "*Ram went to mohan*" will be inaccurate since this way *verb-phrase* will be annotated as the referent of pronoun "*they*".

2.2.2.2 Distributed referent span

As discussed above, in coordination, multiple referent entities are either in conjunction or are separated in the text. However, there are some other cases in which span of even a single referent entity is not continuous and is separated by large distances. For example :

(17) हवा स्वच्छ है गांव की । इसमें प्रदूषण नहीं है ।

Air.NOM clean is village.GEN it.LOC pollution not is.

'Village's air is clean. There is no pollution in it'

In above example referent of इसमें (*it*) is गांव की हवा (*Village's air*) as a single phrase(or chunk). However, (in Hindi) total span of the referent is distributed and separated by other elements. While using MUC framework, the only option to annotate this referent is either to annotate only head of the phrase or to annotate complete sentence (or clause) as referent as shown below:

'<COREF ID="1" MIN=हवा">हवा</COREF >स्वच्छ है गांव की. <COREF ID="2" REF="1" TYPE="IDENT">इसमें</COREF >प्रदूषण नहीं है'
or
'<COREF ID="1" MIN=हवा">हवा स्वच्छ है गांव की</COREF>. <COREF ID="2" REF="1" TYPE="IDENT">इसमे</COREF >प्रदूषण नहीं है'

None of the above annotation captures the actual span of the referent. While such cases are rare in English, in Hindi they are quite frequent due to phenomena like scrambling and movement as a result of free word order of Hindi. This can be illustrated by another example as follows :

(18) भारत की गिरती हुई अर्थव्यवस्था के लिए केंद्र सरकार जिम्मेदार है ।हालांकि
India's falling economy.PURPOSE union-government responsible is. Though
पिछले दशक में यह काफी अच्छी स्थिति में थी ।
in-last-decade it much better condition in was.

'Union government is responsible for India's falling economy. Though in last decade it was in much better condition.'

In above example, the referent of pronoun यह(It) is भारत की अर्थव्यवस्था(India's economy), and not भारत की गिरती हुई अर्थव्यवस्था(India's falling economy). But this discontinuous referent span cannot be annotated here due to the occurrence of गिरती हुई(falling) in between.

2.2.2.3 Ambiguity in Referent span identification

One of the most difficult problem faced while annotating anaphora is that of ambiguity in identifying the actual span of the referent for larger noun-phrases and named entities. Consider following example:

(19) विजय की खोई हुई किताब मुझे मिल गयी ।विजय उसे दराज में रखकर भूल गया था
vijay‘s lost book me found vijay it in drawer after keeping forgot

'I found vijay‘s lost book. Vijay forgot it after keeping in the drawer'

In above example, the exact referent of उसे (*'it'*) must be विजय की खोई हुई किताब (*'vijay‘s lost book'*). However, there could be disagreement over the exact span of the referent as identified by different annotators depending on their comprehension. In this case, there are three possible candidate that could be identified by different annotators: (a) विजय की खोई हुई किताब (*'vijay‘s lost book'*), (b) खोइ हुई किताब (*'lost book'*), (c) किताब (*Book*)

Though MUC annotation guidelines directs the annotation of minimal phrase that constitute the actual referent of the anaphora, it does not specify on what constitute the actual referent of the pronoun.

This could also lead to increased disagreement in annotation because the length and content of the annotated span could differ depending on the comprehension by different annotators.
Also, due to the co-indexing of the references in MUC framework, a resolution algorithm will be penalized if it does not propose a referent that has an exact match with the annotated referent span. That is, in above example, if the annotated referent of उसे (*'it'*) is विजय की खोई हुई किताब (*'vijay's lost book'*) and an algorithm proposes a referent as किताब (*Book*), this will be considered incorrect. However, as it can be observed from the example, all the three candidate referents have a common head that is : किताब (*Book*). Hence, an algorithm should be given partial credit even if it correctly recognizes the referent span that has the head of the phrase(or chunk) same as that of the annotated referent span. For this purpose, there must be separate identification of the head of the referent within the annotation. Thus there is a need to introduce an option in the scheme to represent this separation.

2.2.2.4 Non sequential annotation

Anaphora in discourse usually form chains that refer to a single entity. This evokes the issue of selection of a particular entry from the multiple previous occurrences of a single entity. The linguistic aspect of this problem addresses the issue of marking a referent that is bound to the anaphora(GB Theory). e.g. in case of reflexive, if a referent-anaphora pair occurs in a construction that inherently binds the anaphora to a particular occurrence of an entity, then it is suitable to select that occurrence as the referent. However, from a computational point of view, it is more efficient to select the nearest preceding occurrence of the entity as the referent of the anaphora because it reduces number of possible candidates for the referent of an anaphora in the previous discourse. This in turn adds to computational efficiency in anaphora resolution. Consider following example :

(20) जयसिंह मेवार के राजा थे । वे एक महान शासक थे । उन्होंने जयपुर
Jayasingh mewar.GEN king was. He.NOM.HON a-great-ruler was. He.NOM jayapur
शहर की स्थापना की ।
city founded.

'Jayasingh was king of mewar. He was a great ruler. He founded Jaipur city.'

In above example the referent of pronoun वे(He) in second sentence is जयसिंह(Jayasingh) in first sentence. Similarly उन्होंने(He.HON)[1] refers to the same reference category. However, it is computationally efficient to annotate the referent of उन्होंने(He.HON) as वे(He) rather than जयसिंह(Jayasingh) since it is more nearer to उन्होंने(He.NOM), hence reducing the search space in resolution.

On the other hand consider following example :

(21) राम ने कहा कि अपनी गाडी चलाना उसे पसंद है ।
ram.ACC told that his car to drive he.ACC likes.

'Ram told that he likes to drive his car.'

[1] HON represents honorific here

Considering sequential annotation in above example, राम(Ram) would be selected as the referent of अपनी(his). However, reflexive pronoun अपनी(his) is bound to उसे(he.ACC), thus it would be linguistically justified to select उसे(he.ACC) as the referent.

MUC framework does not specify selection of referent either on the basis of linguistic justification or computational efficiency, thus any two expressions regardless of the distance or linguistic binding can be annotated as co-referring.

2.3 Motivation for new annotation scheme

Our motivation behind proposing a new scheme has two facets. Firstly, as already discussed in previous section, though MUC-7 and all the previously proposed schemes for anaphora annotation are frequently used, there are many issues and challenges associated with these schemes. In our scheme, we aim to handle these issues and propose a consistent framework which aims to improve the consistency in annotation and also be robust to be able to annotate difficult cases as discussed in previous section. Secondly, as we will discuss in subsequent chapters, we aim to explore usage of dependency as a feature for anaphora resolution. Thus, an anaphora annotated corpus along with dependency and other linguistic information is required for our experiments. We aim to use Hindi dependency Treebank(HyDT) [14] for our resolution experiments. Thus we consider annotation of anaphora references over the Hindi dependency Treebank. Thus, the anaphora annotation scheme or framework must take into account the representation format (Shakti Standard format as introduced in chapter-1) of the Dependency TreeBank. However, it is important to note that our scheme can be used to consistently annotate anaphora in any corpus that contains or is linked with these required linguistic information.

2.4 Specification and Syntax of the proposed anaphora annotation scheme

In order to annotate anaphora relations over the Hindi dependency Treebank, we propose a scheme which is consistent with the SSF representation and which also aims to resolve the annotation issues as discussed in section 2.2. Following points briefly describe the syntax and specifications of the proposed scheme.

- Chunks to be considered minimal markables, that is a markable can be a chunk or a set of chunks, but not a partial chunk. For example below:

  (22) अभय के पिता शिक्षक हैं ।वे मेरे मित्र हैं
  abhay's father teacher is. He my friend is.

  'Abhay's father is a teacher . He is my friend.'

  अभय के (*'abhaya'*), पिता (*father*), अभय के पिता (*'abhaya's father'*) etc can be markables as they are complete chunks or set of multiple chunks.

- Anaphora/co-reference links to be annotated in feature structure of the pronoun using address (chunk name/id) of the referent(s) or referent span as a value of reference attributes. For example:
  4.1 वे(ve) PRP <fs af=‘वह,any,sg,3,d,,’ name=‘वे’ **ref**=‘Chunk Id of referent’ >
- Referent span to be divided in two parts : *’head’* and *‘modifier’*, and two separate attributes *‘ref’* and *‘refmod’* to be used to represent the two parts respectively. Considering example (23), the referent of वे (*‘He’*) is अभय के पिता, thus using this scheme, *‘ref’* will contain address of the head i.e. पिता (*‘father’*) and *‘refmod’* attribute will contain address of अभय के (*‘abhay‘s’*) as follows:
  4.1 वे(ve) PRP <fs af=‘वह,any,sg,3,d,,’ name=‘वे’ **ref**=‘Chunk Id of head’ **refmod**=‘Chunk Id of head-modifier’>
- For coordination or multiple referents of plural pronouns, *‘ref’* attribute will contain chunk-id of all the referents separated by comma(,). For example:

  4.1 वे(ve) PRP <fs af=‘वह,any,sg,3,d,,’ name=‘वे’ **ref**=‘Chunk Id of referent$_1$, Chunk Id of referent$_2$’ >
- For more than one modifiers of the head of the referent span, *‘refmod’* attribute will contain chunk-id‘s of all the modifiers separated by a delimiter(/). For example:

  4.1 वे(ve) PRP <fs af=‘वह,any,sg,3,d,,’ name=‘वे’ **ref**=‘Chunk Id of head’ **refmod**=‘Chunk Id of modifier$_1$/Chunk Id of modifier$_2$’>
- References to be annotated sequentially or in chain, i.e. any anaphor should be linked to the last mention of the entity it is referring to (except in the case of linguistic binding).

In the next section, we discuss all the specifications in detail and also discuss how our scheme handles these issues

2.5 Handling annotation issues

In this section, we discuss details of how our scheme handles the complex annotation issues.

2.5.1 Markable Identification and Annotation

We consider chunk[2] to be the minimal unit of annotation. This is because , in Hindi dependency Treebank, dependency structure has chunks[2] at node level and secondly, the features of the head element in a chunk projects its properties up-to the chunk level. Chunks are already annotated with unique ids. Hence, for annotating markables, we opt to represent the markable span as a set of chunks instead of marking a continuous span.

[2] Hindi dependency treebank uses the definition of chunk as ”A minimal (non recursive) phrase(partial structure) consisting of correlated,inseparable words/entities, such that the intra-chunk dependencies are not distorted”[12]

This implies that a referent span can minimally be a chunk, thus increasing the agreement by not allowing the span to be partial chunks. These chunks can later be grouped together using multiple value property. For instance, example (20) from section 2.2.2.3 can be chunked as follows[3] :

<sentence='1' >[$_{\text{NP1}}$ विजय की] [$_{\text{VGNF1}}$ खोई हुई] [$_{\text{NP2}}$ किताब] [$_{\text{NP3}}$ मुझे] [$_{\text{VGF1}}$ मिल गयी ।]
vijay's lost book me found .

<sentence='2' >[$_{\text{NP5}}$ विजय] [$_{\text{NP6}}$ उसे] [$_{\text{NP7}}$ दराज मे] [$_{\text{VGNF2}}$ रखकर] [$_{\text{VGF2}}$ भूल गया था]
vijay it in drawer after keeping forgot

'I found vijay's lost book. Vijay forgot it after keeping in the drawer'

In above example, one of the possible markable is विजय की खोई हुई किताब(Vijay's lost book). This can be represented as a group of 3 chunks (NP1 + VGNF1 + NP2).

2.5.2 Reference Attributes

As discussed in previous section, to annotate user defined attributes, SSF uses attribute value pairs in the feature structure of words. In accordance with the SSF structure, we annotate the anaphora links as values of reference attributes, to be annotated in the feature structure of the anaphor. That is, as per our scheme, the feature structure of the anaphor will have one or more reference attributes whose value will contain the address (in case of SSF, the unique id or the name of the chunk) of the referent(s) of the anaphor.

As discussed above in Section 2.2.2.3, it is easy to identify the head of the referent span as compared to the complete span. In our scheme we propose to separately annotate the easily identifiable head part (called head-referent) of the referent span and annotate the modifiers of the head-referent as a secondary information (called referent-modifiers). This could lead to a higher agreement for head-identification. For each possible anaphora, we annotate the reference information as attribute-value pairs in the feature structure of the anaphora. Two attributes have been introduced in the feature structure namely, *'ref'* to represent the head-referent and *'refmod'* to represent the referent-modifiers. The value of these attributes specifies the unique address(es) of the above elements respectively. The addressing in current annotation is via the global address of the chunk in the document. Thus re-considering example (20) annotated with chunk information as follows :

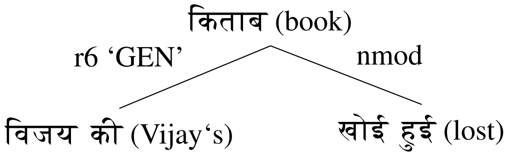
<sentence='1' >[$_{\text{NP1}}$ विजय की] [$_{\text{VGNF1}}$ खोई हुई] [$_{\text{NP2}}$ किताब] [$_{\text{NP3}}$ मुझे] [$_{\text{VGF1}}$ मिल गयी ।]
vijay's lost book me found .

<sentence='2' >[$_{\text{NP5}}$ विजय] [$_{\text{NP6}}$ उसे] [$_{\text{NP7}}$ दराज मे] [$_{\text{VGNF2}}$ रखकर] [$_{\text{VGF2}}$ भूल गया था]
vijay it in drawer after keeping forgot

'I found vijay's lost book. Vijay forgot it after keeping in the drawer'

[3] Here brackets '[' and ']' implying chunk boundaries and the values (NP, NP2, NP3...) in the subscript represent the unique id of the chunk which is equivalent to 'name' attribute in the SSF

The modifiers of the head of the span can be identified by looking at the dependency structure of the referent span. The dependency structure for the span विजय की खोई हुई किताब (Vijay's lost book) would be as follows :



If the pronoun (NP6) उसे (*'it'*) has the referent विजय की किताब (*'vijay's book'*), then since in this span किताब (*'book'*) (NP2) is the head and विजय (*'vijaya'*) (NP1) is the modifier then it will be annotated as follows with the propose scheme:

उसे <fs name='NP6' ref='NP2' refmod='NP1'>

Similarly if the pronoun (NP6) उसे (it) has the referent खोई हुई (lost book), then it will be annotated as follows :

उसे <fs name='NP6' ref='NP2' refmod='VGNF1'>

Thus, we can see that even if different annotators identify different span for the referent, a significant agreement over the head could be achieved by separating head from the modifier.

The selection criteria for the modifiers can vary depending upon the extent of information marked and the type of problem being solved. A scheme may choose to mark only those referent-modifiers that are required to uniquely identify a referent, or it may choose to mark those referent-modifiers that help in establishing co-reference relations via lexical similarity.

2.5.3 Coordination or Multiple(Plural) Referents

As described in section 2.2.2.1 (Coordination and Multiple Non continuous Referents), an anaphor can have multiple head-referents. Multiple instances can be found in the data where a part of the referent can be moved via scrambling, movement or where elements can be inserted in between. Thus it is natural to mark the referent in a way that enables maximum retrieval of information about the referent.

Chunks retain the head element feature structure and have a fixed word order internally. Hence, by considering chunk as the minimal unit for anaphora referent annotation, it can be assured that multiple referents and their respective dependencies can be handled without any information loss.

In order to annotate multiple referents, in the proposed scheme we specify chunk address/id of these multiple referents in the '*ref*' attribute separated by a delimiter(comma). Thus reconsidering the example (17) (annotated with chunk boundaries) as follows :

<sentence=‘1’ >[$_{\text{NP1}}$ राम] [$_{\text{NP2}}$ कल शाम] [$_{\text{NP3}}$ मोहन के] [$_{\text{NP4}}$ घर] [$_{\text{VGF1}}$ गया था।]
Ram.NOM yesterday evening mohan.GEN home went
<sentence=‘2’ >[$_{\text{NP5}}$ वे] [$_{\text{NP6}}$ कई दिनों बाद] [$_{\text{NP7}}$ एक दूसरे से] [$_{\text{VGF2}}$ मिले] ।
They many days after with each other met.

‘Ram went to mohan’s home yesterday evening. They met each other after many days.’

Thus, in above example, with the proposed the feature structure of pronoun NP5(वे)(They) would be as follows:

वे <fs name=‘NP5’ ref=‘NP1,NP3’ refmod=‘’>

<

ref=‘NP1,NP3’>implies that the pronoun has 2 head-referents, NP1 and NP3.

2.5.4 Multiple Referent-Modifiers

As discussed in section 2.2.2.3 (Distributed referent span), if a referent span is distributed discontiguously then it poses a problem in marking the exact span of the referent. Our scheme attempts to resolve this problem via marking the head with multiple modifiers. These modifiers are required for the correct interpretation of the pronoun. Address values of all such modifier chunks are assigned in the *‘refmod’* attribute separated by a delimiter(/). Thus reconsidering example(18) as follows :

<sentence=‘1’ >[$_{\text{NP1}}$ हवा] [$_{\text{JJP}}$ स्वच्छ] [$_{\text{VGF1}}$ है] [$_{\text{NP3}}$ गांव की ।] <sentence=‘2’ >[$_{\text{NP4}}$ इसमें]
air clean is village‘s. in it
[$_{\text{NP5}}$ प्रदूषण] [$_{\text{VGF2}}$ नहीं है ।]
pollution is not

‘Villag‘s air is clean. There is no pollution in it’

In above example the referent of इसमें(it) is गांव कि हवा(village‘s air), where हवा (air)is the head and गांव कि (village‘s) is the modifier. Hence it will be annotated as follows :

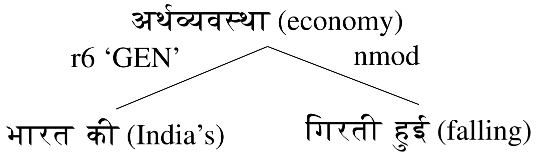
इसमें <fs name=‘NP4’ ref=‘NP1’ refmod=‘NP3’ >

Similarly reconsidering example(19) as follows :

<sentence=‘1’ >[$_{\text{NP1}}$ भारत की] [$_{\text{VGNF1}}$ गिरती हुई] [$_{\text{NP2}}$ अर्थव्यवस्था के लिए] [$_{\text{NP3}}$ केंद्र सरकार]
India’s falling economy.PURPOSE union-government
[$_{\text{VGF1}}$ जिम्मेदार है ।] <sentence=‘2’ >[$_{\text{NP4}}$ हालांकि] [$_{\text{NP5}}$ पिछले दशक में] [$_{\text{NP6}}$ यह]
is responsible. Though in-last-decade it
[$_{\text{NP7}}$ काफी अच्छी स्थिति में] [$_{\text{VGF2}}$ थी ।]
in-much-better-condition was.

‘Union government is responsible for India’s falling economy. Though in last decade it was in much better condition.’

The referent of the pronoun NP6 (यह)(It) is (भारत की अर्थव्यवस्था)(India’s economy). Head of the span NP2 (अर्थव्यवस्था)(economy) has two modifiers NP1 (भारत की) (India’s) and VGNF1 (गिरती हुई)(falling) as shown in the diagram below :



However, only NP1 is required as a modifier of NP2 for the correct interpretation of the pronoun. With the proposed scheme, we can annotate only those pronoun which are required in the referent span as shown below :

यह <fs name=‘NP6’ ref=‘NP2’ refmod=‘NP1’ >

If in some case, both the modifiers are required for the interpretation of the pronoun than both the modifiers can be included in ‘refmod’ attribute as follows:

यह <fs name=‘NP6’ ref=‘NP2’ refmod=‘NP1/VGNF1’ >

2.5.5 Sequential annotation

In view of the computational efficiency, as discussed in section 2.2.2.4 (Sequential annotation), we adopt chain marking for anaphora annotation in this scheme. That is, if an entity is referred by more than one pronouns or has repeated mentions in a discourse, then for each pronoun, we annotate the last mention of the corresponding referent-entity as the antecedent.
However, in cases where marking the nearest occurrence of the entity as referent, is not linguistically justified; the scheme allows to annotate the bound entity as the referent. Thus consider example(21) can be reconsidered as follows :

<sentence=‘1’ >[$_{\text{NP1}}$ जयसिंह] [$_{\text{NP2}}$ मेवार के] [$_{\text{NP3}}$ राजा] [$_{\text{VGF1}}$ थे ।] <sentence=‘2’ >[$_{\text{NP4}}$ वे]
Jayasinh mewar.GEN king was. He
[$_{\text{NP5}}$ एक महान शासक] [$_{\text{VGF2}}$ थे ।] <sentence=‘3’ >[$_{\text{NP6}}$ उन्होंने] [$_{\text{NP7}}$ जयपुर] [$_{\text{NP8}}$ शहर की]
a-great-ruler was. He.NOM jaipur city
[$_{\text{VGF3}}$ स्थापना की ।]
founded.

‘Jayasingh was king of mewar. He was a great ruler. He founded Jaipur city.’

The referent of pronoun NP4 (वे)(He)in second sentence is NP1 (जयसिंह)(Jayasingh) in first sentence. Similarly NP6 (उन्होंने)(He) refers to the same reference category. However, it is computationally efficient to annotate the referent of NP6 as NP4 rather than NP1 since it is more nearer to NP6, hence reducing the search space. Considering sequential annotation, we annotate the pronouns NP4 and NP6 as follows

वे <fs name='NP4' ref='NP1' refmod='' >
उन्होंने <fs name='NP6' ref='NP4' refmod='' >

On the other hand consider example (22) :

[$_{\text{NP1}}$ राम ने] कहा कि [$_{\text{NP2}}$ अपनी] गाडी चलाना [$_{\text{NP3}}$ उसे] पसंद है।
ram.ACC told that his car to drive he.ACC likes.

'Ram told that he likes to drive his car.'

Considering sequential annotation in above example, NP1(राम ने)(Ram) would be selected as the referent of NP2(अपनी)(his). However, reflexive pronoun NP2(अपनी)(his) is bound to NP3(उसे)(he.ACC), thus it would be linguistically justified to select NP3(उसे)(he.ACC) as the referent.

Hence in this example the referent of NP2(अपनी)(his) will be NP3(उसे)(he.ACC) and the referent of NP3(उसे)(he.ACC) will be NP1(राम)(Ram), with the feature structure as follows :

अपनी <fs name='NP2' ref='NP3' refmod='' >
उसे <fs name='NP3' ref='NP1' refmod='' >

2.6 Additional features and Specification

In this section we further describe the additional specifications of the scheme that can be used to handle cases of abstract anaphora, co-reference and can be used to add additional information tags like type of anaphora, reference type, direction etc.

2.6.1 Handling Abstract Anaphora

For cases in which the referent is an event or a proposition, the main verb is marked as the referent (*'ref'*). The *'refmod'* takes the participants (modifiers) of the verb as it's values. It can either take all the participants of the event as it's values, or it can choose to take only those that are required for the correct interpretation of the referent of the abstract anaphora.

(23) [$_{\text{NP1}}$ राम ने] [$_{\text{NP2}}$ मोहन को] [$_{\text{NP3}}$ पुरानी गाडी] [$_{\text{NP4}}$ ऊंचे दाम में] [$_{\text{VGF1}}$ बेची ।] [$_{\text{NP5}}$ इससे]
Ram.ERG Mohan.DAT old car high price-in sold Due-to-this
[$_{\text{NP6}}$ उसे] [$_{\text{NP7}}$ 5 लाख रुपए का] [$_{\text{NP8}}$ लाभ] [$_{\text{VGF2}}$ हुआ ।]
he.DAT 5-lakh-Rs.GEN profit be.PST

‘Ram sold an old car to Mohan at a high price. Due to this he made a profit of 5 Lakh Rs.’

In example 24, the complete referent span is NP3+NP4+VGF1 (पुरानी गाडी ऊंचे दाम में बेची) (*‘sold old car in high price’*), but the head-referent is the verb VGF1 (बेची) (*‘sold’*) and NP3(पुरानी गाडी) (*‘old car’*), NP4(ऊंचे दाम में) (*‘high price’*) are the referent-modifiers. The feature structure for pronoun NP5(इससे) (*‘due to this’*) is as follows :

इससे <fs name=‘NP5’ ref=‘VGF1’ refmod=‘NP3\NP4’>

Note that only NP3 and NP4 are considered in the *‘refmod’* attribute, because only these modifiers are required for the correct interpretation of the anaphoric relation.

2.6.2 Handling Co-reference

With the above scheme, the co-reference relations can also be annotated. In the case of co-reference, the value of the ref attribute would take the address/id(s) of the lexical items it co-refers with. However, including the addresses of all the lexical items (which may be large in number) can make the value field very lengthy. To avoid this, span marking is introduced. In span marking, the value contains the address of the starting and the ending lexical item joined by a delimiter(semicolon).

2.6.3 Additional Tags

Along with the reference attributes, additional tags could be incorporated in the feature structure which provide information about the anaphoric relation. Some of the important tags are :

- Pronoun Type : Personal, Reflexive, Relative, Co-relative, Indefinite.
- Referent Type : Concrete, Abstract.
- Direction : Cataphora, Anaphora.

2.7 Inter-Annotator Study

We conducted Inter-Annotation studies in order to verify a higher consistency of the proposed scheme, as compared to the MUC-7 annotation framework which is commonly used for Co-reference and anaphora annotation. We divide the study in two parts as follows :

2.7.1 Experiment 1

As stated in Section 2, only Concrete reference types were annotated in the first phase of the annotation. However, in Hindi same lexical pronoun can refer to Concrete as well as Abstract reference entity and many a times it becomes difficult to identify this distinction. We first establish this by conducting an experiment which involves annotating the category of a reference type as ’Concrete’,’Abstract’,or

’Other’(including the exo-phoric and indefinite reference types). Fleiss’s Kappa [29] is used to calculate the agreement, which is a commonly used measure for calculating agreement over multiple annotators. It is a measure of the degree of agreement that can be expected above chance. By using Fleiss kappa, we measure the reliability value as the degree of agreement among annotators in assigning the one of the three categories to each instance of pronoun. The final value of kappa measures the weighted average of the agreement over all anaphors in the experimental data. Given ‘n’ subjects (pronouns in our case) and ‘k’ categories in which ‘m’ number of raters (annotators) classify the subjects.
For every subject i = 1, 2, , n and evaluation categories j = 1, 2, , k. let x$_{ij}$ = the number of judges that assign category j to subject i. Thus The Fleiss’s kappa is calculated as :

$$\kappa = 1 - \frac{nm^2 - \sum_{i=1}^{n}\sum_{j=1}^{k} {x_{ij}}^2}{nm(m-1)\sum_{j=1}^{k} \bar{p}_i \bar{q}_j}, \tag{2.1}$$

$$where\ \bar{p}_i = \frac{\sum_{i=1}^{n} x_i}{nm}, \bar{q}_j = 1 - \bar{p}_j \tag{2.2}$$

The numerator of equation (2.1) gives the degree of agreement that is attainable above chance, and, denominator gives the degree of agreement actually achieved above chance.
Table 2.1 shows the method to interpret kappa values
We conducted the experiment over 29 news items from the Treebank containing 446 identified pronouns across annotations by 3 raters. Annotators were asked to assign one of the three categories, as stated above, according to the type of entity it refers to. Table 2.2 summarizes the experiment’s results.

Kappa Statistic	Strength of agreement
<0.00	Poor
0.0-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

Table 2.1: Coefficients for the agreement-rate based on [41].

No. of Annotations	Agreement	Pr(a)	Pr(e)	Kappa
446	353	0.856	0.435	0.746

Table 2.2: Kappa statistics for Category experiment

The non-perfect agreement for this experiment establishes that the type of the referent of a pronoun is ambiguous and hard to determine in many cases. Hence, to avoid inconsistencies in the distinction of Concrete, Abstract and Other types of reference; we separate out the concrete references in the above used data for the comparative study of the proposed scheme with MUC. We consider agreement over those pronouns in Experiment 2, for which all the annotators have a perfect agreement in concrete category.

2.7.2 Experiment 2

In the second experiment, the inter-annotator analysis is conducted for the concrete pronouns separated in Experiment 1. To calculate agreement over the anaphora links for each pronoun, we measure the agreement over the similarity in the co-reference chains. As suggested in [51] and [52] Krippendorff's alpha is a better metrics for calculating agreement for co-reference/anaphora annotation as compared to other metrics because it considers degrees of disagreement and in anaphora it is difficult to define discrete categories. Krippendorff's alpha [38] was then used as a statistical measure to obtain the inter-annotator agreement. Similar to [51] we consider co-reference chain as discrete categories. Consider following example in which five annotators are asked to annotate the references of all the pronouns

(24) John wanted to buy a laptop. He_1 went to the market with Bob. He_2 saw a new model of lenovo. He_3 liked it

Suppose, chains annotated by different annotators are as follows:

A1 : John, He_1, He_2, He_3
A2 : John, He_1, He_2, He_3
A3 : John, He_1, He_3, Bob, He_4
A4 : John, He_4, Bob
A5 : Bob, He_4

In above annotations, there is a total match between annotations of A1 and A2, while first chain annotated by A3 is a subset of annotation by A1. For first chain annotated by A4 and that of A1, there is only one element in common. Finally, intersection of annotations by A1 and A5 is null.

Krippendorff‘s alpha includes distance measure to calculate these partial agreement over non-discrete categories. [51] provides a detailed description of how Krippendorff‘s alpha provides a better measure to calculate agreement over anaphora/co-reference annotation. Krippendorff's alpha is defined as follows:

$$\alpha = 1 - \frac{Do}{De}, where \tag{2.3}$$

$$Do = \frac{1}{i * c(c-1)} \sum_{i \in I} \sum_{k \in K} \sum_{k' \in K'} \mathbf{n}_{ik}\mathbf{n}_{ik'}\mathbf{d}_{kk'}, \tag{2.4}$$

$$De = \frac{1}{i * c((i * c) - 1)} \sum_{k \in K} \sum_{k' \in K'} \mathbf{n}_{k}\mathbf{n}_{k'}\mathbf{d}_{kk'} \tag{2.5}$$

where

I = set of all items of annotation which in this case is set of all pronouns

K = set of categories which in this case is set of all the chains annotated by all the annotators

n$_{ik}$ = number of times item i is given the value k i.e. number of times a pronoun 'i' is assigned to a chain 'k'

n$_k$ = any number of times any item (pronoun) is assigned to a chain 'k',

i = total number of items (pronouns) to be annotated,

c = number of annotators

The distance measure d$_{kk'}$ is defined as

$$d_{kk'} = \begin{cases} 0 & \text{if k and k' are exactly same chains} \\ 0.33 & \text{if k is a subset of k' or vice versa} \\ 0.66 & \text{if there is at least one element common between k and k'} \\ 1 & \text{if intersection of k and k' is empty} \end{cases}$$

Table 2.3 shows the statistics obtained for the MUC annotation and with the proposed scheme.

Statistics	MUC-7	Proposed Scheme
No. of Annotations	239	239
alpha	0.825	0.880

Table 2.3: Inter-annotator agreement results using Krippendorff alpha

Experiment (2) also involved the same data and the same raters who carried out annotation in experiment (1).

As shown in table 2.3, there is a significant increase in the Krippendorff's alpha agreement over the proposed annotation scheme, as compared to the MUC annotation scheme. This indicates that the proposed scheme with the separation of head and modifiers in the referent span helps in achieving a consistent agreement than the continuous span annotation scheme used in MUC.

2.8 Data annotation:

Using the scheme described in this chapter, we carried out the annotation of anaphora references over the data taken from Hindi Dependency treebank. For the experiments on Anaphora resolution described in chapter-4 and 5, we annotated a part of the dependency treebank. Annotation of dependency and other linguistic information such as POS-tag, chunk boundaries, morphological properties had already been carried out on these texts. We annotated around 324 text files or articles in which we identified 4232 pronouns out of which 3206 pronouns have been identified and annotated as concrete reference

and 523 pronouns have been identified as Event or abstract reference. Rest 503 pronouns have been identified as non-anaphoric.
Besides the above described annotation for experiments, we aim to fully annotate Hindi dependency treebank with anaphora relations. Towards this, 827 texts have been annotated in which total 12361 Entity reference have been annotated till date. First phase validation has been completed for this data and second phase validation is in progress.

2.9 Summary and Discussion

In this chapter, we described development of a scheme towards annotating anaphora information as a layer in Hindi Dependency Treebank. Our main contribution is to discuss language specific issues that occur in anaphora annotation and outline a scheme that handles them efficiently. The identified issues relate to representation format, referent span identification etc. Decisions like sequential annotation and sub-tree inheritance help in reducing the computational complexity in resolution systems. The comparative inter-annotator analysis of the proposed scheme verifies that the separation of the referent span, and other features help to achieve a consistent annotation by increasing the inter-annotator agreement. The scheme can be extended for co-reference and the annotated data is convertible to other annotation formats like MUC etc.

Using this scheme a major part of the treebank has been annotated which can be used for anaphora resolution task and other NLP applications. For the scope of this work, we do not annotate anaphoric instances of gaps, ellipsis and demonstratives which are to be included in the future annotation.

Chapter 3

Background and Related Work on Anaphora Resolution

In this chapter, we discuss, the relevant theoretical background and related work for anaphora resolution research. In section (1), we discuss, the various factors or features that affect the anaphoric reference relations and that have been used by various resolution algorithms. In section 2, we discuss the related work in some of the very most important and well known anaphora resolution approaches as well as the related work for Anaphora resolution in Hindi.

3.1 Factors in Anaphora Resolution

(Mitkov, 1999) [45] discusses various factors that affect the phenomena of anaphoric reference. These are the factors mostly used while resolving the references. Of these, most frequently used are morphological agreement (gender and number), binding and c-command constraints, semantic similarity, grammatical and semantic parallelism such as verb similarity, preference, nearness or proximity etc. These factors are usually classified in two categories [6], [36] : (a) Constraints and (b) Preferences. Many resolution methods choose either of the factors or both. We explain some of the important constraints and preferences as follows:

3.1.1 Constraints:

Constraints represent linguistic restrictions that define which entities or phrases may or may not be referred by specific pronouns. Following are some important constraints:

- Gender and Number agreement: Many languages exhibit the property of agreement between anaphora (pronoun) and its referent. Thus agreement between anaphor and referent is an important feature for anaphora resolution[73]. Taking some examples from English:

  (25) John went to the shop with Mary$_i$. She$_i$ was excited to buy the new car.

  (26) John$_j$ went to the shop with Mary. He$_j$ was excited to buy the the car.

  Both of the sentences are identical except for the pronoun, since in the first sentence, a feminine pronoun is used, its referent will be Mary, while in second sentence masculine pronoun (He) will refer to John.

Similarly, number agreement too is an important constraint in anaphoric references as shown in use of pronouns in the example below:

(27) There were differences in the interests of the president$_i$ and the congress$_j$. He$_i$ vetoed their$_j$ decision.

(28) There were differences in the interests of the president$_i$ and the congress$_j$. They$_j$ vetoed his$_i$ decision.

- Binding constraints: The Government and Binding theory (reference) specifies c-command [18] constraints which restrict references of some specific pronouns within some syntactic limits. Following are some of the important constraints from c-command:

  - Referent of a reflexive (or bound) pronoun must c-command it or must be in its own governing category.

    (29) John told Bob that Jack$_i$ saw himself$_i$ in the mirror.

    Though binding theory gives details about c-command and governing category, in the above example, for simplicity, we consider the clause boundary as the governing category, thus referent of the reflexive pronoun *'himself'* must be in its own governing category or the same clause as pronoun, thus its referent is *'Bob'* which is the only entity in the same clause as pronoun and in agreement with the pronoun.

  - Referent of a non-reflexive or personal pronoun must not be an entity that c-commands it or must not be in the same governing category as the pronoun.

    (30) John scolded him harshly.

    In the above example, since *'him'* is non-reflexive, its referent can not be in the same governing category as the pronoun or in other words, it should not be in the same clause as the pronoun. Thus, since *'John'* is in the same pronoun, it could not be its referent.

- Semantic agreement: In some languages, there must be semantic agreement between anaphora and its antecedent [43]. This semantic agreement can be properties like animacy, proximity etc. Taking an example from English:

(31) Mary has a Mercedez$_i$. It$_i$ is attractive.

(32) Mary$_i$ has a Mercedez. She$_i$ is attractive.

In above example, semantics of *It* state that it should refer to an inanimate entity, while that of *She* state that it should refer to an animate entity, specifically female.

3.1.2 Preferences:

Preferences, though do not strictly rule out or enforce reference relations like constraints, but helps to select referent from possible candidates by them giving relative weights and thus giving more preference to some candidates over others, based on various properties. We explain below some of the preferences:

- Grammatical Role (Subject preference): Anaphors preferably refer to elements based on their grammatical salience. [65]

  (33) John went to the Acura dealership with Bill. He bought an Integra.

  (34) Bill went to the Acura dealership with John. He bought an Integra.

  In the above examples 'subject' of the preceding sentence (*'John'* in example 34 and Bill in example (35)) is preferably the correct referent of the pronoun as compared to entities in adverbial clause.
- Parallelism: Anaphors preferably refers to a an entity in previous discourse, which has the same semantic or grammatical role as themselves.

  (35) Mary went with Sue$_i$ to the Acura dealership. Sally went with her$_i$ to the Mazda dealership.

  In the above example, The structure of second sentence which is parallel to the first one suggest that referent of *'her'* is *Sue*.
- Verb semantics: In some cases, selection preferences or semantics decide the referent.

  (36) John$_i$ telephoned Bill. He$_i$ lost the pamphlet on Acuras.

  (37) John criticized Bill$_i$. He$_i$ lost the pamphlet on Acuras.

  In example(37), implicit cause of telephoning is subject, thus *'John'* is the referent, while in example (38) implicit cause of criticizing is object, thus *'Bill'* is the referent.
- Centering: Although constraints and preferences, as described, do help to prune the search space to identify the referent of the anaphors, in many cases even after applying these constraints, relative salience of the candidate becomes a deciding factor to resolve the references. This salience is decided on the basis of 'focus' or 'centers of discourse'. These centers may change between utterances. The Centering theory [66] is based on the concept that in general, utterance in discourse have some centers or focus points which are the entities in discussion for a local context. In coherent discourse, the transition between these centers is smooth, that is, there is only little change of focus between two consecutive utterances, while with a large change of focus, discourse becomes incoherent. Assuming that a natural discourse is coherent and hence the transition of focus is smooth, identifying the entities which when acting as focus make the transition smooth, will help to choose the correct referent from among the possible candidates. Resolution algorithms using the Centering theory are based on forward and backward looking centers and choosing those entities as centers which will make the transition smooth. We explain Centering based resolution in section 3.2.1.3

It is also evident from the above discussion on constraints and preferences, that the information or features required for anaphora resolution would come from various levels of linguistic analysis. In specific terms, categories and classes of pronouns come from lexical analysis, gender and number information for agreement detection come from morphological analysis, grammatical roles come from syntactic analysis, semantic roles, verb classes and similarities come from semantic analysis, while recency and salience come from discourse level.

3.2 Related work

Though there has been extensive research on anaphora and co-reference resolution in English and also for Multilingual co-reference resolution, most of the work can be divided into two fundamental approaches: Rule based methods and Learning based methods. However, there has also been some limited work towards using hybrid methods. Rule based method can be further categorized into Constraint based and preference based methods. Learning based methods can be categorized into Supervised and Unsupervised methods. In accordance with the relevance of our work, we divide our discussion of related work in three parts. In the first part we discuss fundamental rule based and learning based approaches which provide the core ideas about features and algorithms used in most of the subsequent anaphora resolution research. These methods include both constraint based and preference based approaches. In the second part, we briefly discuss the previous research for anaphora resolution in Hindi. Finally, we discuss the research in both the languages, which have explored or experimented with use of dependency structures for anaphora resolution.

3.2.1 Core Rule based and learning based methods:

In this section, we discuss the most fundamental rule based and learning based methods which were the earliest works to propose uses of specific features and algorithms. Various subsequent approaches have improved over these methods, however, the core idea with most of these extensions is influenced by these core approaches. We give a brief overview of these approaches. A detailed description and critical discussion about each is available in related literature.

3.2.1.1 Hobb‘s algorithm

Hobb‘s algorithm [33] is one of the earliest approaches that rely mostly on using syntactic constraints as defined in the ‘Government and Binding theory’ to identify referents of pronouns. The idea is to search for most likely candidates of a pronoun in the phrase structure parse tree of the sentence containing the pronoun and the previous sentences. The selection criteria to decide likely referents are based on c-command and binding theory. Given a a pronoun in a sentence and the parse trees of all the sentences including and preceding the one containing the pronoun, the simplified algorithm is as follows.

- Start at the pronoun node in the parse tree of the current sentence.

- Move up in the tree until an NP or S node is encountered. Let‘s call this node as ‘X’ and the path from the X to pronoun be called as ‘p’.
- Iterate or traverse through all children nodes of ‘X’ which are left to the path ‘p’. If an NP node (call NP’) is encountered which has another NP (call NP”) node lying on the path between ‘S’ or ‘X’, then propose NP’ as the referent of the pronoun.
- If the top of the tree has been reached i.e. (X=S), then do a left-to right breadth first traversal of the parse trees of previous sentences beginning with nearest one.
- Propose those NP‘s as referent which agree with the pronoun.

Consider following example containing two sentences. The task is to resolve the referent of the pronoun *‘him’*

(38) Bob lost the template. John$_i$ scolded him$_i$

Figure 3.1 shows the resolution of pronoun *‘him’* in above example using Hobb‘s algorithm.

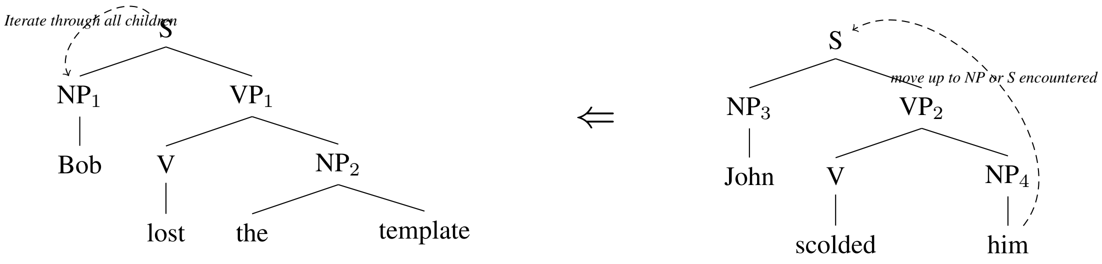


Figure 3.1

For above example, the algorithm works as follows :

- Start at pronoun node (NP$_4$) in second sentence.
- Move up in the tree up-to an ‘NP’ or ‘S’ is encountered. In this case (X=S), only child of ‘X’ left to path ‘p’ is ‘NP$_3$’, but it is rejected since there is no NP node between ‘NP$_3$’ and ‘X’ (‘S’)
- Since the ‘S’ node is reached, move to the parse tree of previous sentence and traverse in breadth first fashion starting at leftest node ‘NP$_1$’
- Since NP$_1$ is the first node, propose ‘NP$_1$’ as the referent.

Hobb‘s algorithm includes further details of agreement matching and search for the discourse (non-structural) anaphora resolution. Although, it is simple to implement and also computationally cheap, it is inadequate for some cases involving references in complex sentences. However, the main importance and motivation drawn from this algorithm is that it provides a simple method to implement constraints specified by syntactic theory for anaphora resolution.

3.2.1.2 Lappin and Leass‘s algorithm

The algorithm proposed by (Lappin and Leass, 1994) [42] uses relative salience of candidate referents to resolve the anaphora. The idea is to select highly salient candidates which are in agreement with the pronoun as the referent. It focuses on three features : recency, syntax preferences and agreement. The salience values of the candidates is to be calculated using a corpus for different languages and domains. Briefly. the algorithm can be described as follows :

- Collect the potential referents.
- Remove potential referents that do not agree in number or gender with the pronoun.
- Remove potential referents that do not pass intra-sentential syntactic co-reference constraints.
- Compute the total salience value of each mention (NP).
- Select the referent with the highest salience value(if same, select the closest reference in terms of string position).

Again, this algorithm though being simpler has limitations. For example it employs fixed values for salience based on grammatical and syntactic roles, however, other factors such as semantic roles when combined with syntactic structures may effect the reference phenomena. However, again the importance of this algorithm is that it provides a simple method to rank candidate elements based on their grammatical and syntactic roles and hence is able to resolve references even in more complex structures.

3.2.1.3 Centering based resolution

As discussed in section 3.1.2, centering theory is based on notion of ‘focus’ or centers of discourse and the way of transition between these foci for a coherent discourse. The centering based anaphora resolution approach by (Walker and Joshi, 1998) [66] has following elements:

- Utterance: Discourse is divided into individual sentences, clauses or set of sentences called utterances.
- Center: Any entity that is discussed or referred in the discourse is a center.
- Forward Looking centers: C$_f$(U$_n$) defines the set of centers that are referred in utterance U$_n$. All the centers in C$_f$(U$_n$) are ordered on the basis of salience which represents the relative prominence or preference of the centers. Usually this salience is based on the syntactic roles (Subject $>$Object $>$Indirect Object $>$Other)
- Backward Looking center : Each utterance U$_n$ has only one backward looking center C$_b$(U$_n$) which is one of the common elements in the sets of forward looking centers of U$_n$ and U$_{n-1}$. That is $C_b(U_n) \in C_f(U_n) \cap C_f(U_{n-1})$. C$_b$(U$_n$) must be the highest ranked center in C$_f$(U$_{n-1}$).
- Preferred center: Preferred center C$_p$(U$_n$) is the highest ranked center in U$_n$.

The algorithm defines following four types of transition depending on the relation between forward, backward and prefered centers of two consecutive utterances.

	$C_b(U_{n+1}) = C_b(U_n)$ or $C_b(U_n)$ = NULL	$C_b(U_{n+1}) \neq C_b(U_n)$
$C_b(U_{n+1}) = C_p U_{n+1}$	Continue	Smooth-shift
$C_b(U_{n+1}) \neq C_p U_{n+1}$	Retain	Rough-shift

According to the coherence constraint of Centering theory, following is the order of preferred transitions: Continue >Retain >Smooth Shift >Rough Shift. Consider the following example:

(39) John saw a beautiful Acura Integra at the shop (U1). He_1 showed it_1 to Bob (U2). He_2 bought it_2 (U3).

In brief, using centering based resolution, there are two choices for referent of pronoun *He_2*: *Bob* and *John*. Since *Bob* is a recently introduced entity, choosing referent of *He_2* as *Bob* will result in a **Smooth shift**, while since *John* has already been accessed (referred) in second sentence by use of pronoun *He_1*, choosing it as a referent of *He_2* will result in ***'Continue'*** transition. Since 'Continue' is preferred over 'Smooth transition' as in first scenario, *'John'* should be selected as the correct referent of *'He_2'*

Centering theory is the most important algorithm which provides a framework for including the discourse salience in the process for anaphora resolution. However, again this is limited in its scope since it consider only salience and agreement features for reference resolution

3.2.1.4 Machine learning approach

To the best of our knowledge, the earliest approach for anaphora resolution in a machine learning setting is (Connolly, et,al 1997)[20] which is a supervised learning approach. They propose anaphora resolution as two class classification problem. The idea is to represent anaphoric references as pair of anaphora and candidate referents. These pairs are represented as instances or feature vectors where each instance includes features which may effect the reference relation between the anaphor and the referent. For training, positive and negative instances are created depending on whether the candidate is actually the referent of the anaphora as in the annotated training data. The classifier trained on such training instances learns relative preference of the features important for the references. This classifier is then used to predict whether a similar instance in the test represents a possible reference relationship between the anaphora and the candidate referent terms of positive or negative output. Positively labeled instance can then be re-ranked to select the 'best' candidate.

3.2.2 Related work on Anaphora Resolution in Hindi

The earliest known work for anaphora resolution in Hindi is (Sobha and Patnaik, 2002) [58]. It makes limited use of grammatical rules and morphological markings to identify subject, object, clause etc. However, they only discuss a possible algorithm, and no actual implementation or evaluation is provided.

(Dutta et, al. 2008) [27] present an approach in which they propose adopting Hobb‘s algorithm for Hindi as a baseline algorithm. However, the algorithm is implemented over a few sentence for a limited types of pronoun and no results are reported. They state that the algorithm can be further evaluated and used once a sufficient amount of data is available, however, phrase structure data for Hindi in sufficiently large amount is yet unavailable.

The approach used by (Agarwal et, al. 2007) [4] is based on matching constraints for the grammatical attributes of different words. The important point about this approach is that along with their linguistic information, they use animate/non-animate classification of entities while resolving the reference. Though the approach claims accuracy 96% for simple sentences and 80% for complex sentences, it neither give any detail about the coverage of individual constraints or features, nor any detail about the data except that it is taken from children stories. Also it claims results on only 120 pronouns, hence reproducing or verifying the results and hence establishing the validity of the algorithm is difficult.

The most important related work for anaphora resolution in Hindi is (Prasad and Strube, 2008) [54]. This algorithm applied a discourse salience ranking to two pronoun resolution algorithms, the BFP and the S-List algorithm. This approach is mostly based on Centering theory. The S-list algorithm [61] is based on a model which consists of a single construct, called the S-list, and one operation, the insertion operation. The S-list contains some discourse entities which are realized in the current as well as the previous utterance. A ranking is imposed on the elements of the S-list, being determined by information status and/or word order, and the order among the elements provides straightforward preferences for the antecedents of pronominal expressions. For using S-List for Hindi, they propose modification to the constraints for ranking the S-list elements in Hindi. The algorithm is as follows:

- If a referring expression is encountered
  - if it is a pronoun, test the elements of the S-list in the given order until the test succeeds;
  - Update S-list; the position of the referring expression under consideration is determined by the S-list-ranking criteria which are used as an insertion algorithm.
- If the analysis of utterance U is finished, remove all discourse entities from the S-list, which are not realized in U.

(Uppalapu and Sharma, 2009) [62] extend the above algorithms by using two different list in place of a single list. They point out two limitations with the S-list algorithm of [54]. First, that in above algorithm most of the clauses in complex sentences are not divided in separate utterances except in case of coordinate clauses. Second, they point out the performance of S-List can be improved by considering two lists in place of one thus including one extra list called *‘previous-list’* to store those entities which are more than one sentence earlier to the pronoun. However, similar to most of the earlier approaches both of these method are limited in their exploration of different types of pronoun and evaluating the algorithm on a larger and diverse data set. [62] report their results only on story data.

3.2.3 Related work using dependency for anaphora resolution

To the best of our knowledge, there are only two earlier approaches which explore the use of dependency relations for anaphora resolution. First is the one discussed above, i.e. (Uppalapu and Sharma, 2009) [62] for Hindi which improves over the S-List algorithm of (Prasad and Strube, 2008) [54]. While, (Uppalapu and Sharma, 2009) [62] propose to use the traditional grammatical relations based salience i.e. (subject >object >Indirect object >others) for ranking the candidates in the list, (Uppalapu and Sharma, 2009) [62] propose to use the Paninian grammatical relations (CPG) i.e. (k1 >k2 >k3 >k4 >others). However, they only use dependency relations as a feature to rank the candidate elements in centering based approach, however, in order to consider dependency structures and relations as syntactic features and property, it is important to explore dependency relations as well as structures to identify different constraints for different pronouns that can be used for resolving the referents in the same way as phrase structure syntax is used as for resolving the referent in Hobb'' algorithm and other methods.

Besides the above approach, the only other know approach exploring dependency in anaphora/co-reference resolution is (Bjrkelund and Kuhn, 2012) [15] for English. They explore the possibility of using dependency relations as a feature for co-reference resolution in a fully learning based approach. Their results show that, a combination of dependency and phrase-structure yields better results than using phrase structures alone. This approach again is based on exploring dependency relations as an additional feature in supervised learning as an alternative to phrase structure.

Chapter 4

Entity Anaphora Resolution

As discussed in chapter-1, Entity anaphora stands for those pronominal references which refer to a Concrete Entity such as Person, place and other common nouns. Thus possible candidate referents for entity anaphora are noun phrases (NP). Consider following example:

(40) गंगा एक विशाल नदी है ।यह गंगोत्री से निकलती है ।
ganges a huge river is it from gangotri originates

‘Ganges is a huge river. It originates in gangotri’

In the above example, the referent of pronoun यह (*‘it’*) is गंगा (*Ganges*) which is a concrete Entity. Hence, this is an instance of Entity reference. The resolution of Entity anaphora involves identification of correct referent of a pronoun out of the possible noun phrases. In the above example, the possible referents of the pronoun are यह (*‘it’*) are गंगा (*Ganges*), एक विशाल नदी (*‘a long river’*), गंगोत्री (*‘gangotri’*). The task of anaphora resolution is to automatically identify the correct referent of the pronoun यह (*‘it’*) which in this example is गंगा (*Ganges*)

In this chapter we discuss our approach and experiments for Entity anaphora resolution in Hindi. Our aim is to determine up-to what extent, deep linguistic features such as dependency structures can help to correctly resolve the Entity references. Hence, in section 1, we first explore the use of dependency structures for resolving Entity references. However, an approach using deep linguistic knowledge is dependent either on the data annotated with this knowledge or on high performance pre-processing tools such as dependency parsers which may be expensive to obtain. Therefore, as discussed in section 2, we also aim to explore up-to what extent shallow features or limited linguistic knowledge can be used to resolve Entity references.

4.1 Hybrid approach using dependency

As we discussed in section 3.2.1, syntactic approaches specifically, Hobb‘s algorithm [33] are quite successful in resolving structural anaphora reference. Motivated by this, we aim to exploit syntactic structures and relations for resolving some specific categories of pronominal references such as Reflexives, Relative and Personal pronouns. However, There are two main differences of our approach from the earlier approaches.

- First, we choose a hybrid approach, rather than fully rule based or fully learning based approach.
- Second, we use dependency structure and relations instead of phrase structure as a source of syntactic information for anaphora resolution.

Most Anaphora resolution approaches which use deeper linguistic features are based on phrase structure based grammatical models. We, instead, use linguistic features from a dependency grammar model known as CPG which is discussed in chapter-1. Hence, our first aim is to develop a rule based approach by studying and analyzing patterns in the CPG based dependency structure that can be formulated as rules to resolve references. Our observation shows that some categories of pronominal references such as *Reflexives, Relatives, First and Second* person can be easily resolved by formulating rules based on dependency structures.
However, for some ambiguous references, specifically in third person pronouns, although syntactic constraint do reduce search space, they fail to uniquely identify the referent. In resolution of such references, morphological, grammatical, distance and semantic features also play an important role. Hence, in addition to the rule based approach, we also implement a supervised learning algorithm using these features, for those references for which rule based approach fails to predict any referent within the scope of its constraints. Thus. our hybrid approach includes a rule- based resolution module followed by a supervised learning module.

Figure 4.1 shows the work-flow of the resolution process.

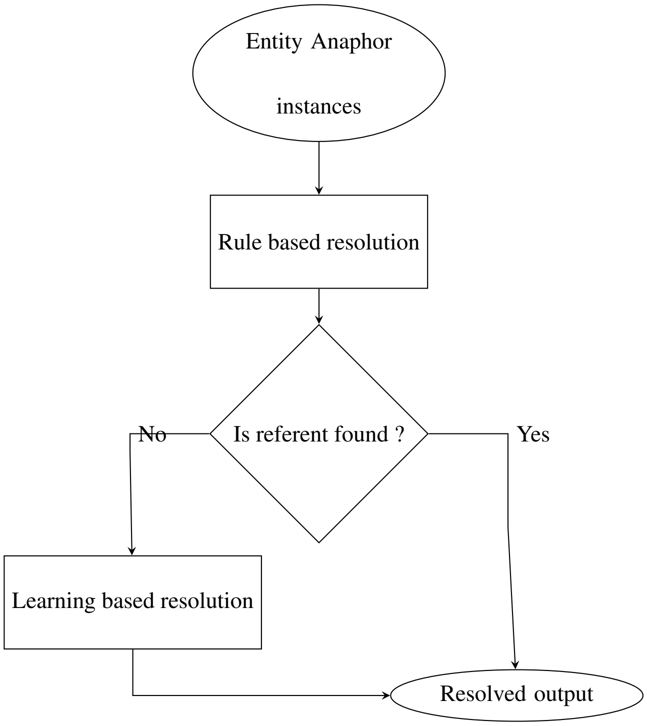


Figure 4.1 Process work-flow

We discuss different categories of pronominal forms and the structural relation between the anaphors in these categories and their referents. Simultaneously, we derive the rules which help to resolve these pronoun based on these observations.

4.1.1 Rule based resolution module

The rule based module attempts to locate the referent of the pronoun using the constraints derived from the dependency structures and relations. These constraints are defined based on the categories of the pronoun which are discussed in section 1.4.1. First, the category of the pronoun is identified using a complete list of the pronominal form as provided in Appendix A. After the category is identified, a set of rules defined for each category is applied to locate the referent. We discuss below these pronominal forms and the rules derived for their resolution:

4.1.1.1 Reflexives

As discussed in section 1.4.1, Reflexive pronouns is the anaphor which must be bound by its antecedent. Moreover, it can be derived from the discussion of c-command constraints in section 3.1.1, that the referent of the reflexive pronoun is the accessible subject in its own governing category. In Hindi most frequent types of reflexives are the possessive reflexives i.e. अपना,अपनी (*'own'*) etc. Semantically the referent of the possessive reflexives is the possessor entity which in turn is frequently the

‘SUBJECT’ of the clause or sentence. However, there is no notion of subject in CPG based dependency framework which we aim to explore for anaphora resolution. Hence, we focus on CPG based dependency relations which are described in section 1.6. We obtain two observation about referents of the reflexive pronoun from the frequency analysis of the anaphora annotated corpus. First, the referent of a reflexive pronoun is always within the same clause as the pronoun. Second, the referent is frequently the Entity which has a role of *‘karta’* (*‘k1’*) as per the CPG based dependency framework. Thus, many of the reflexive pronouns can be resolved by selecting the noun phrase with dependency label *‘k1’* within the same clause as reflexive pronoun. Consider following example:

(41) अभय ने कहा कि विजय$_i$ ने रवि को अपनी$_i$ किताब दी
abhay.ERG told that vijay.ERG ravi.DAT his.POSS.REF book gave

‘abhay told that vijay$_i$ gave his$_i$ (POSS.REF) book to ravi.’

Figure (4.2) shows the dependency structure of example (42),

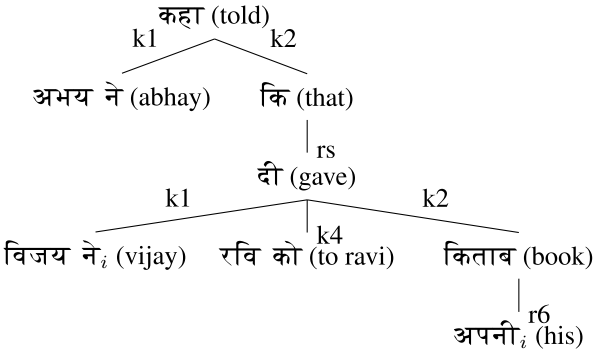


Figure 4.2

Looking at the dependency structure, the node which has the dependency label as *‘k1’* and is also a child node of the root verb of the clause should be selected as the referent of the reflexive pronoun. Figure 4.3 shows the tree traversal for searching the referent. The root verb of the clause containing possessive reflexive अपनी(*his*) is दी(*gave*) (as shown by upward traversal), It has a descendant node विजय(*vijay*) with a dependency relation *‘k1’* with the verb. Thus विजय(*vijay*) should be selected as the referent of the pronoun (as again shown by the downward traversal in the figure).

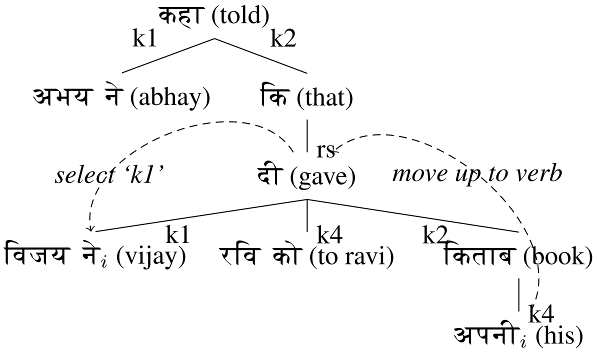


Figure 4.3

Besides the above example, there are some other cases, in which the possessor is not in the role of *‘karta’* (*‘k1’*), instead it is a modifier of the possession together with the possessive pronoun. Consider following example:

(42) सोनिया गांधी चाहती हैं कि राहुल की$_i$ अपनी$_i$ छवि बने
sonia-gandhi.NOM wants that rahul.GEN his.POSS image create.INTRANS

‘sonia gandhi wants that rahul$_i$ should have (his own)$_i$ image.’

Figure (4.4) shows the dependency structure of the example (43) where, the referent of the pronoun अपनी (*‘his own’*) is राहुल (*‘rahul’*) which is a sibling node of the pronoun and has a dependency relation *‘r6’* with its head.

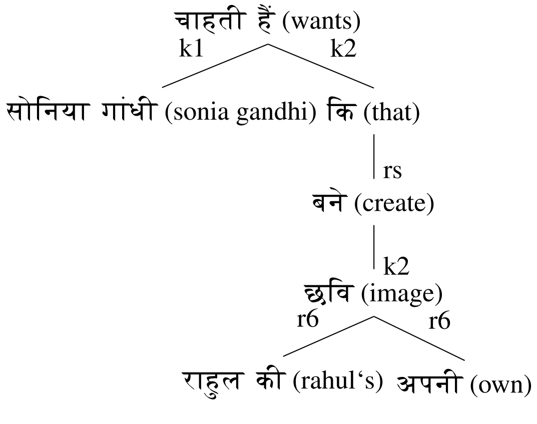


Figure 4.4

In such cases, the referent should be searched in the sibling nodes of the reflexive pronouns. Thus, if a reflexive pronoun has a sibling with dependency relation ‘r6’ with the common head, then it should be selected as the referent of the reflexive.

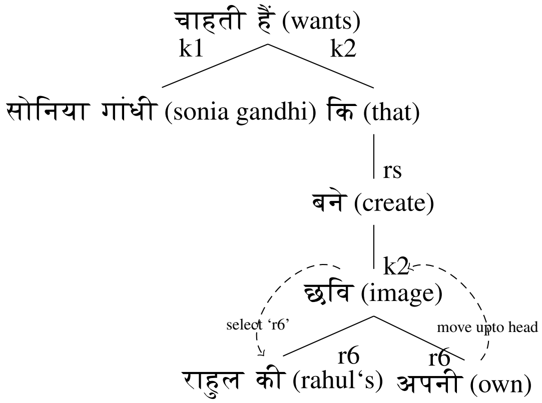


Figure 4.5

As shown in figure 4.5, the traversal is first to the head verb node and then to the sibling node with label 'r6' which is selected as the referent of the pronoun.

The rules for resolution of reflexive pronouns derived from observation in all the above examples can be consolidated in the following algorithm:

- If a node 'N' is a reflexive pronoun, then move up in the tree, until an NP node or a verb (VGF) node is encountered. Call this node 'X'.
- If 'X' is NP, then search for children nodes of 'X' other than 'N', if such a children node with dependency label 'r6' is found, propose it as referent. otherwise to go Next step.
- If 'X' is verb node, then search in the children of the node 'X' with dependency label 'k1', other than 'N', if such a node is found, propose it a referent, else traverse up in the tree.
- If 'X' is a CCP (conjunct) node with label 'k1', search in the children of 'X', propose all the NP nodes in the children set as referent.
- If 'X' is root of tree and no referent is found, then output NULL (no solution).

Although, selecting *'karta'* (*'k1'*) in the same clause or selecting the sibling node of the reflexive pronoun as discussed above identifies the correct referent in most of the cases, however, there are some cases in Hindi which are ambiguous and may refer to entities with other syntactic or semantic roles depending on the context of conversation. Consider following example:

(43) अभय ने मोहन को अपने लिये चाय बानाने के लिये कहा ।
abhay.ACC mohan.ERG self for tea make PSP asked

'Abhay asked mohan to make tea for him/himself'

In above example, the pronoun अपने (*'self'*) is ambiguous and may refer to either अभय (*'abhay'*) which is *'karta'* (*'k1'*) or मोहन (*'Mohan'*) which is *'k4a'* or *'experiencer'* of the main verb कहा (*'asked'*) and the English translation of the sentence would change depending on the referent of the reflexive pronoun. The reference in such cases can only be resolved by looking at the pragmatic context of the sentence. However, since it is not possible to incorporate context knowledge in resolution process, we only consider the above described rules for such cases too.

4.1.1.2 Spatial (Place) pronouns

As discussed in chapter-1, spatial pronouns refer to places. Similar to English, Hindi also has two spatial pronouns यहां (*‘here’*) and वहां (*‘there’*). A simple solution to resolve spatial pronoun is to identify Noun phrases representing *‘places’* and choose the most probable among them.
In CPG based framework, a label ‘k7p’ is used to annotate those entities which represent the location of the ‘action taking place’ in the sentence. Thus one solution to resolve place pronouns is to select the entities with dependency label ‘k7p’ nearest to the pronoun. Consider following example :

(44) <sent=‘1’ >[$_{\text{NER=LOC}}$ बिहार के] [$_{\text{NER=LOC}}$ सिवान में$_i$] सूखे की स्थिति गंभीर है।
bihar‘s siwaan.LOC drought‘s situation critical is.
<sent=‘2’ >आज प्रधानमंत्री वहां का$_i$ दौरा करेंगे।
today prime minister there visit

‘Situation of drought is critical in siwaan of bihar. Today Prime minister will be visiting there$_i$.’

Fig (4.6) shows the dependency structure of the two sentences in example (45).

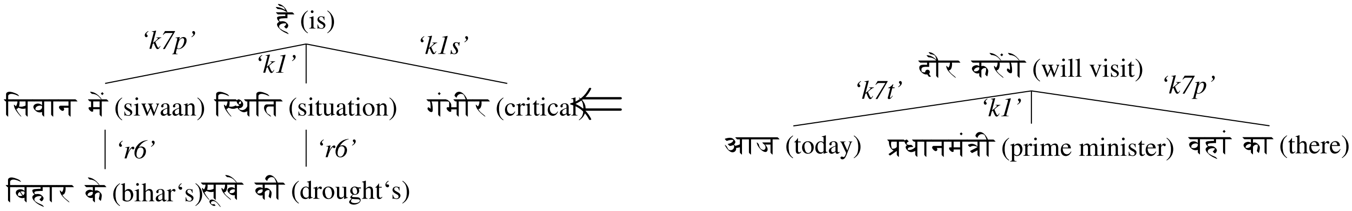


Figure 4.6

As there is no location described in the second sentence(which contains the pronoun वहां (*there*), we search for the referent in previous sentence(sentence 1). The root verb of the sentence 1 has a child node with label ‘k7p’, hence it will be selected as the referent of the pronoun as shown in figure below:

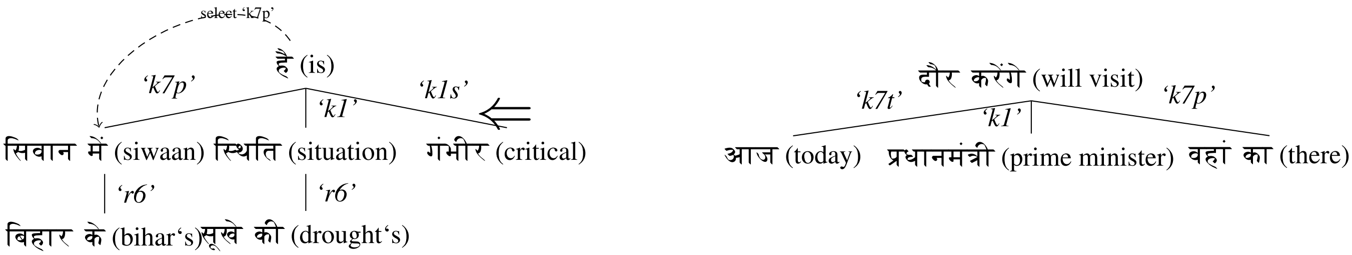


Figure 4.7

Place pronouns refer to entities which are semantically location, but these entities may or may not be in a role of location in the sentence and hence may not have a locative dependency label ‘k7p’. Therefore, although all the entities which have dependency label of *‘k7p’* are semantically *‘places’*, but not all the entities which are semantically *‘places’* are labeled as *‘k7p’*. Consider following example:

(45) <sent=‘1’ >[$_{\text{NER=LOC}}$ उज्जैन] देश के प्राचीनतम शहरों में से एक है । <sent=‘2’ >यहां
ujjain country of oldest cities out of one is. here
विश्वविख्यात महाकालेश्वर मंदिर स्थित है ।
world-famous mahakaleshwar temple situated is

‘Ujjain is one of the oldest cities of the country. World famous Mahakleshwar temple is situated here.’

In above example, the referent of the pronoun यहां (*‘here’*) is उज्जैन (*‘Ujjain’*) which is a *‘place’* entity. However, it is not in a role of location here. That is, it is not specifying the location of the action here. Hence, it is not annotated as *‘k7p’*. To identify such referents we need other features. Another alternative to identify semantic locations is to use Named Entity Category information. A named entity recognizer identifies the ’Place Entities’ and labels them with category=‘LOCATION’. Similar to the dependency label approach, place pronoun can be resolved by selecting the NP node nearest to the pronoun with a named entity category as ‘LOCATION’. Thus in above case, if the Named Entity category of the NP उज्जैन (*‘Ujjain’*) is identified as ‘LOCATION’, then it can be selected as the referent of the spatial pronoun. However, we use automatic Named entity recognizer, which may not always identify all the Named Entity categories correctly. Due to which some of the entities which are actually *‘Place’*, but not identified by NE recognizer will be missed. Therefore, to resolve maximum possible *‘Place’* referents, we use a combination of both the properties, i.e. the dependency label *‘k7p’* and Named entity category ‘LOCATION’ to identify the referents.

From the above observation, following algorithm can be derived for resolution of reflexive pronouns:

- Begin with pronoun identified as ‘spatial’ (or place pronoun).
- Search in reverse order for a noun phrase with Named Entity Category as ‘LOCATION’ and dependency label ‘k7p’
- If such a noun phrase is found, predict it as the referent.
- Else Search in reverse order only for a noun phrase with dependent label ‘k7p’ up-to 3 sentence.
- If such a noun phrase is found, predict it as the referent.
- Else Search in reverse order only for a noun phrase with Named Entity Category as ‘LOCATION’ up-to 3 sentence.
- If such a noun phrase is found, predict it as the referent.
- If no referent is found within 3 sentences, output ‘NULL’ (no solution).

4.1.1.3 Relative pronoun :

A Relative clause is a kind of subordinate clause which specifies an element, usually a Noun phrase (NP), in the main clause. Thus inherently, in terms of dependency relationship, relative clause is a modifier of this NP. Hence in CPG based dependency framework (also in other dependency frameworks), relative clause is attached under that NP node, which is relativized by the clause. Thus, given the dependency structure, the referent of the relative pronoun is essentially that Noun phrase which is head of the root verb of the relative clause. In CPG-framework relative clauses is attached to the main clause through the relativized NP and the relation is marked as *'nmod-relc'*. Consider following example

(46) बदमाशो से दस बैग बरामद हुए जिनमे वे चोरी का सामान ले जाते थे
From thugs ten bags seized in which they looted items used to carry

'Ten bags were seized from the thugs in which they used to carry the looted items.'

Figuer (4.8) shows the dependency structure of example (47), in which the relative pronoun is जिनमे (*'which'*) and the head of the relative clause is the verb ले जाते थे (*'used to carry'*) which is attached below the NP node दस बैग (*ten bags*) with a relation *'nmod-relc'*.

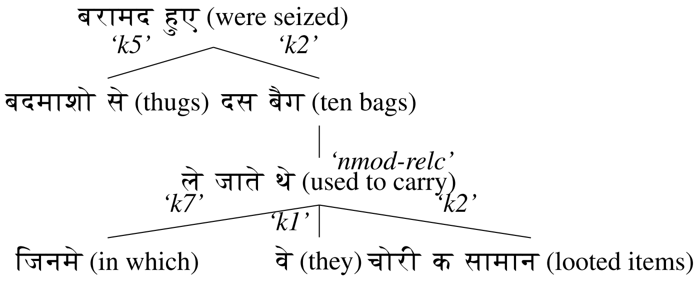


Figure 4.8

This makes the resolution of relative pronoun quite simple. The referent of the relative pronoun should be selected as that noun-phrase to which the clause containing relative pronoun is attached as shown in figure (4.9):

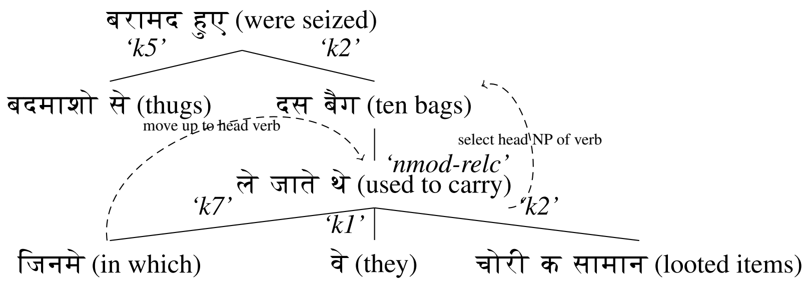


Figure 4.9

From the above observation, following algorithm can be derived for resolution of relative pronouns:

- Start with the pronoun node ‘N’, move up in the tree until a verb node (VGF or VGNF) is encountered. Call this node as ‘X’.
- If the label of attachment of the ‘X’ node with its parent node (‘H’) is ‘nmod-relc’, then select the parent node (‘H’) as the referent. Else, keep moving upwards until such a node is found.
- If the ‘X’ is the root of the tree and no referent is found, then output ‘NULL’ (no solution)

4.1.1.4 First and Second person pronouns:

First and second person pronouns usually refer to speaker and listener of a communication. First person pronouns include मैं (*‘I’*), हम (*‘we’*) and their inflected forms, and second person pronouns include तु (*‘YOU (casual)’*), तुम (*‘YOU formal’*), आप (*‘YOU (respect)’*) along with their inflected forms.
In a descriptive text such as news corpus, first and second person pronoun mostly occur in the narrative or attributional clauses. i.e. in those subordinate clauses whose main clause has an attribution root verb such as बोलना(*to tell*), कहना(*to say*), बताना(*to tell*), समझाना(*to explain*), पुछना(*to ask*), सुनना(*to listen*).
Since attributional clause is a subordinate clause, in the CPG dependency framework, the dependency structure of the attributional clause (rooted at the attributional verb) is attached under the complementizer की (*‘that’*), which in turn is attached under the root verb of the main clause as an argument, with label *‘k2’*. Consider following example containing both first and second person pronouns in a narrative clause:

(47) उमा ने आडवाणी से कहा की आप मेरा इस्तीफा स्वीकार करे
Uma.ERG advaani.DAT said that you.HONORIFIC.ACC my resignation accept

‘Uma said to advaani that you (do) accept my resignation.’

Figure (4.10) shows the dependency structure of example (48), in which मेरा (*‘my’*) is the first person pronoun and आप(*‘you’*) is second person pronoun in the attributional clause rooted at स्वीकार करे(*‘accept’*).

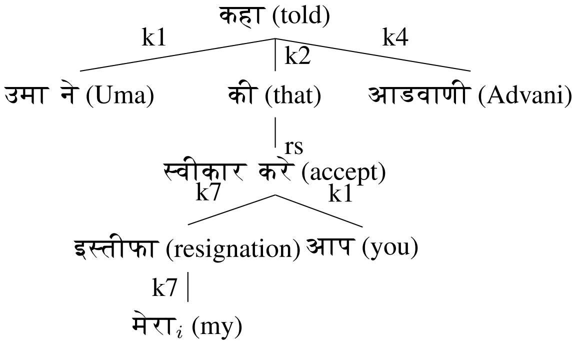


Figure 4.10

As discussed in [62], If the first person pronoun is an argument of attributional subordinate clause, then its referent is the speaker of that clause, which almost always is the *'k1'* (*'karta'*) in the main clause. Similarly the referent of a second person pronoun which is an argument of attributional subordinate clause, is mostly the *'k4'* or *'experiencer'* in the main clause. Thus first person and second person pronouns can be resolved by searching for the complementizer node up in the tree, starting from the pronoun node, then moving to its parent node i.e. root verb of the main clause and selecting that child of the main verb as the referent (for the first person pronoun) which has a label *'k1'* and that with label *'k4'* as the referent (for the second person pronoun).

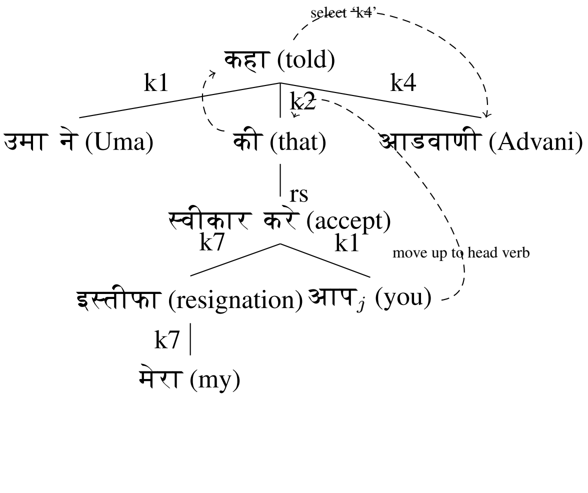


Figure 4.11

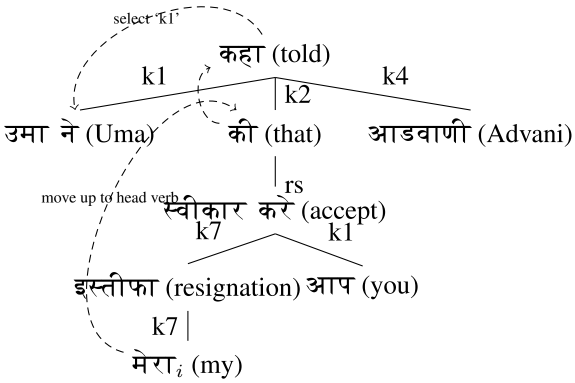


Figure 4.12

Figure (4.11) show the resolution of second person pronouns, in which आप(*'you'*) is second person pronoun in the attributional clause rooted at verb स्वीकार करे(*'accept'*), hence its referent is selected as the *'k4'* of the main clause i.e. आडवाणी (*'advani'*) and similarly Figure(4.12) shows the resolution of the first person pronoun मेरा(*'my'*), where the referent is selected as *'k1'* of the main clause i.e. उमा (*'Uma'*). In cases where the narration is broken into multiple utterances or sentences, first person or second person may refer to *'k1'* or *'k4'* of the previous sentences.

Based on the above observation following algorithms can be derived for resolution of first and second person pronouns:

- Starting at the pronoun node, move up in the tree until a complementizer node की (*'that'*) is encountered. Call this node 'C'.
- If no such node is found up-to the root, output 'no solution'
- Move up to the parent node of 'C'. If this is a verb node. call this 'V', else output 'no solution'.
- Search in the children nodes of 'V', nodes with label 'k1' and 'k4'.
- For first person pronoun, select node with 'k1' as the referent, for second person pronoun, select node with 'k4' as the referent. If no such nodes are found, output 'no solution'.

4.1.1.5 Third person pronoun:

Third person pronoun are most frequent pronoun and mostly their reference are inter-clausal or inter-sentential. References of third person pronouns are more discourse (textual) references rather than structural. Thus although intra-sentential syntactic structures do not directly effect these references, relative salience and preference of the candidate Noun Phrases have quite a significant effect on the references of third person pronouns. Hence rather than using syntactic structures for resolution of third person pronouns, we focus on dependency relations and their relative salience. For this purpose, we adopt re-ordering of the candidate elements based on the salience of dependency relations, similar to (Uppalapu and Sharma, 2009) [62]. However, there are two differences in our approach from (Uppalapu and Sharma, 2009) [62] : First, they consider the the salience ordering [*k1 >k2 >k3 >k4 >others*] to rank the candidate elements. which they adopted from the ordering of the grammatical relation [*subject >indirect object >adjunct*] as in [54], However, we adopt a slightly modified ordering of the relations [*k1 >k2 >r6 >k4 >k3 >others*] which is based on the relative frequency of the dependency relations for animate entities. Second, they only consider number as an agreement feature for pruning the candidate elements, we also use animacy along with number to prune the candidate list. As discussed in [50], animacy plays an important role in the application of agreement restrictions between pronouns and candidates, and as a result, it can improve the accuracy of anaphora resolution systems. Specifically, in Hindi, distal third personal pronominal forms वह (*'He/She/it'*) and its inflected forms such as उन्होने (*'He'*) refer animate entities, while other third pronouns may refer to animate as well as non-animate entities. Thus animacy agreement can be used to prune out the list of possible candidates.

We explain resolution of third person pronoun with following example :

(48) [$_{\text{k1,h}}$ उमा ने] [$_{\text{k7}}$ पहले] [$_{\text{k4,h}}$ शिवराज को] [$_{\text{k2,rest}}$ पत्र] लिखा ।फिर [$_{\text{k1,h}}$ उन्होने] [$_{\text{k4,h}}$ उन्हें]
Uma.ACC first shivraaj.ACC letter wrote. later she.HON.ACC him.HON
[$_{\text{k3,rest}}$ कोर्ट से] [$_{\text{k2,rest}}$ नोटिस] भिजवाया
from court sent.CAUSATIVE

'First uma wrote a letter to shivraaj. Later she sent a court notice to him'

In above example there are two pronouns in the second sentence : first is उन्होने(*she*) (gender neutral), the possible referents for this pronouns and their ordering based on salience is as follows : [उमा (*umaa*), पत्र (*letter*),शिवराज (*shivaraj*)]. Since the top element i.e. *uma* (*umaa*) agrees with the pronoun in number and animacy, it is selected as the referent for उन्होने (*he.ACC*). Thus the ordered list of candidate becomes : [पत्र (*letter*), शिवराज (*shivaraj*)]. The second pronoun is उन्हे (*him*) (gender neutral) and the top element in the list is पत्र (*letter*), however it doesn't agree with pronoun either in number or animacy. Hence the agreement is checked with next element that is शिवराज (*shivaraj*) which agrees with the pronoun in both number and animacy, hence it is selected as the referent of the pronoun.

In the rule based approach for third person pronoun, we search for referent, only up-to a fixed number of previous sentences *'n'*, i.e. if the a referent could not be resolved within the *'n'* sentences, the pronoun is passed for learning based resolution.

Based on the above observations, following algorithm can be derived for resolution of third person pronouns:

- For each pronoun, make a list of possible candidate referents by selecting all the Noun phrases (or markables) within a limit of 'n'[1] sentences previous to and including the one containing the pronoun.
- Re-rank the list of candidates based on two parameters: First by dependency relation and second by recency, that is proximity to the pronoun.

[1] we determine the optimum value of 'n' by parameter tuning over development set

- Select the first element in the list as the referent of the pronoun, which agrees with the pronoun in number, gender and animacy.
- Replace the resolved referent with the pronoun in the list. Reorder the list based on the above parameters.
- Before resolving the next pronouns, in the candidate list, resolved referents will be replaced by the corresponding pronoun and new candidate NP‘s preceding the new pronoun will be added.
- If no referent is resolved within *‘n’* sentences, output no solution.

4.1.2 Evaluation of Rule based approach:

We first evaluate the performance of the rule based resolution and based on these results, we implement the learning based approach.

4.1.2.1 Data Set

A part of anaphora annotated Hindi treebank as described in chapter 2 is taken for evaluating the performance of the resolution approach. Total data consists of 324 texts (news articles) containing 3206 annotated **concrete** pronouns. One third (1/3rd) of the data is held out (fixed) for testing. Thus the testing data contains 1071 **concrete** pronouns. The remaining two-third (2/3rd) data is used in 4-fold iteration for training and development, i.e. the remaining data is again divided randomly in 4 parts and evaluated in 4 iterations. In each iteration, one fourth of data is used as development (tuning) and the remaining three fourth (3/4th) is used for training (in supervised learning). For the rule based setting the only parameter which we tune is the number of sentences *‘n’* (that should be considered while searching for the referent).
The following table shows number of pronouns corresponding to each category in the test set.

	Total
Reflexive Pronoun	156
Relative Pronouns	80
Locative Pronouns	48
1st and 2nd person Pronouns	81
Third person Pronouns	706
Total	**1071**

Table 4.1: Pronouns by category in Test set

Note that since pronouns occur in discourse, training and testing set are divided in files and not in pronouns. Thus across all the iteration, it is the number of files that remain same but the number of pronouns may vary since texts (files) may contain different number of pronouns.

4.1.2.2 Evaluation Measures:

Precision: In general terms, precision is the fraction of retrieved instances that are relevant. In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). In terms of anaphora resolution, precision can be termed as the ratio of 'Number of correctly resolved anaphors' to 'Number of anaphors outputted by the system' [7], [49].

$$precision = \frac{Number\ of\ correctly\ resolved\ anaphors}{Number\ of\ anaphors\ outputted\ by\ the\ system} \tag{4.1}$$

Recall: Recall (also known as sensitivity) is the fraction of relevant instances that are retrieved, Recall in terms of classification is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been). In terms of anaphora resolution, recall is defined as the ratio of 'Number of correctly resolved anaphors' to 'Number of all anaphors' [7], [49].

$$recall = \frac{Number\ of\ correctly\ resolved\ anaphors}{Total\ number\ of\ all\ anaphors} \tag{4.2}$$

Pairwise F-score: F-score is calculated as the harmonic mean of the precision and recall

$$F\text{-}score = \frac{2 * precision * recall}{precision + recall} \tag{4.3}$$

4.1.2.3 Results of the rule based approach

Tables 4.2 shows the average values of precision, recall and f-score over 4-fold iteration on development data for each individual pronominal forms calculated for varying values of *'n'* (number of previous sentences considered to search referent) from 0 to 4.

	n=0			n=1			n=2			n=3			
	P	R	F	P	R	F	P	R	F	P	R	F	P
Reflexive Pronoun	.80	.80	.80	.80	.80	.80	.80	.80	.80	.80	.80	.80	.80
Relative Pronouns	.82	.82	.82	.82	.82	.82	.82	.82	.82	.82	.82	.82	.82
Locative Pronouns	.80	.62	.69	.80	.69	.74	.82	.73	.77	.82	.75	.78	.80
1st and 2nd person Pronouns	.92	.89	.90	.97	.95	.96	.97	.95	.96	.97	.95	.96	.97
Third person Pronouns	.28	.10	.14	.34	.26	.29	.42	.32	.36	.51	.49	.50	.45
Overall	.46	.33	.38	.51	.45	.47	.56	.49	.52	**.61**	**.60**	**.60**	.58

Table 4.2: Average results on development data for n=0...4; P=precision, R=recall; F=F-score

As can be seen from above table, the results for Reflexive and Relative are uniform regardless of the number of sentences considered, which is quite obvious because both of these are structural references, that is they have referent mostly in the same sentence as the pronoun. There is a slight increase in accuracy for first and second person pronouns from 'n=0' to 'n=3' which is constant later for all other values of *'n'*. Finally, the accuracy for third person shows a sharp increase from 'n=0' to 'n=3'.

Based on these observation, we consider the optimal value of 'n' equal to 3, that is we consider 3 sentences previous to the one containing the pronoun to search the referent in the rule based setting. Table 4.3 below represent the result for rule based approach on the test data.

	Precision	Recall	F-score
Reflexive Pronoun	.82	.74	.74
Relative Pronouns	.85	.79	.85
Locative Pronouns	.77	.88	.88
1st and 2nd person Pronouns	.93	.95	.95
Third person Pronouns	.48	.35	.40
Ovearll	**.62**	**.37**	**.46**

Table 4.3: Rule based approach results on test data

From the results in table 4.3, we can note that the performance of the rule based system is quite high for First and Second person pronouns similar to (Uppalapu and Sharma, 2009) [62]. This is because in a descriptive text like our data, first and second person pronouns mostly occur only in narration clause and in almost all occurrences they refer to the speaker or listener of the narration. Similarly, accuracy for Relative and Locative too, is quite promising but not perfect because the constraints used to resolve these pronouns are although quite simple, but not definite, hence not all instances of these forms could be covered by the described rules. Similarly the resolution accuracy of Reflexive pronouns is not as high as Relative and Locative. The reason behind this is that although the rules described for resolution of Reflexive pronouns are highly specific and definite, there are various instances of ambiguous reference as already discussed in section 4.1.1.1. Finally, the resolution of accuracy for third person personal pronouns is low as compared to other pronouns. Third person pronouns are not only difficult to resolve with specific rules, but are also most frequent. This motivates us to further use learning based approach. All the pronoun which are not resolved by rule-based system are passed to the learning based approach.

4.1.3 Supervised Learning approach:

The core idea behind modelling anaphora resolution in a supervised learning based setting is to rephrase anaphora resolution as a binary classification problem as follows:

Is NP....which appears in context....an antecedent of the pronoun *'PRP'* ?.

For this purpose, *NP+context+pronoun* is represented as feature vectors, Training data is labeled as:

NPs+context+pronoun+YES/NO.

The classifier learns to predict YES/NO for testing input vectors.

We apply two models for learning based classification: Single candidate model of (Soon et, al, 2001) [59] and Twin candidate model of (Yang et, al, 2008) [76].

4.1.3.1 Single Candidate Model:

Given an *'anaphora'* and possible candidates NP$_1$, NP$_2$,.....NP$_n$, the traditional supervised learning approach is to determine the preference: *"p(antecedent(NP$_k$)|anaphora,NP$_1$, NP$_2$,.....NP$_n$)"* i.e. the preference that a candidate NP$_k$ is the referent of the 'anaphor', given other possible candidates NP$_1$, NP$_2$,....,NP$_n$. The assumption with single candidate model is that the preference of NP$_k$ being the actual antecedent of 'anaphor' depends only on their mutual features and is independent of the other candidates. That is:

$$p(antecedent(NP_k)|anaphor, NP_1, NP_2....NP_n) = p(antecedent(NP_k)|anaphor, NP_k) \quad (4.4)$$

We use the approach of (Soon et, al, 2001) [59] for the training instance creation. A positive instance is created for each anaphor m$_k$ and its actual antecedent m$_j$. Multiple negative instances are created by pairing the anaphor with each possible candidate [m$_{k-1}$,m$_{k-2}$....m$_{j+1}$] between the anaphor and the actual antecedent. Each instances is represented by a feature vector extracted from the anaphor, candidate antecedent and their context. The classifier learns a model based on this training data.

For testing, unlabelled instances are created by pairing the anaphor with all the Noun-phrases in 3 previous sentences including the sentence in which the pronoun is present. All the instances thus generated are then passed to a classifier, which labels these instances as positive or negative based on the model learned in the training phase. Positively labeled instance which is nearest to the pronoun is proposed as the referent of the pronoun.

Given the following example[2]:

(49) [$_{k1,h}$ खाद्य सुरक्षा विधेयक को] [$_{k7}$ आज] [$_{k4,h}$ प्रधान मंत्री की] [$_{k2,rest}$ मंजूरी] मिल गयि ।
Food Security Bill today prime minister's approval got.
[$_{k1,h}$ उन्होंने] [$_{k4,h}$ इसके] [$_{k3,rest}$ महत्त्व पर] [$_{k2,rest}$ पत्रकारों से] बातचीत की
He.NOM it's about significance with journalists had conversation

'Food security bill today got prime minister's approval. He had a conversation with journalists about it's significance'

There are two pronouns in the above example: उन्होंने (He.NOM) and इसके (it's). Training instances for इसके (it's) whose referent in the training data is labeled as खाद्य सुरक्षा विधेयक(*'Food security bill'*), will be created as below:

- इसके (it's) ,उन्होंने(He), feature-set,NO

[2] The subscript in the brackets represent the features <Dependency relation, animacy >for the Noun phrases

- इसके (it‘s) ,मंजूरी(approval), feature-set,NO
- इसके (it‘s) ,प्रधान मंत्री की(prime minister), feature-set,NO
- इसके (it‘s) ,आज(today), feature-set,NO
- इसके (it‘s) ,खाद्य सुरक्षा विधेयक(Food security bill), feature-set,YES

Similarly instances using उन्होंने (He.NOM) could be created by pairing it with previous NP‘s.

4.1.3.2 Features

We briefly describe the features used in the classification:

- Agreement features : As discussed in section 3.1.1, in many languages anaphor and antecedent agree in morphological properties. In Hindi, ‘Number’ and ‘Gender’ agreement can be observed between anaphor and the antecedent. We consider these two properties which have following values. Number : singular, plural and honorific. Gender : Masculine, Feminine, NULL (not available). Since, even in treebank data, the annotated values of these features can be ambiguous or noisy. hence instead of directly using the agreement as a feature, we allow the classifier to learn the agreement from these properties for anaphor and antecedent. This adds 4 features to the instance: Number and Gender of the anaphor, Number and gender of the candidate referent.
- Named Entity categories: Named entities are the Noun phrases which refer to person names, geographical places, organization etc. In the data used, the NE tags are according to the categories ENAMEX , NUMEX and TIMEX with further subcategories as ‘Person’, ‘Organization’, ‘Location’, ‘Number’ Named entity categorization provides important semantic information fro anaphora resolution. Some pronouns such as वह and its inflected forms more likely refer to Phrase with Named Entity category *‘Person’* and spatial pronouns more likely refer to *‘Location’*.
- Distance feature : As already discussed in chapter-3, recency is an important factor for anaphora resolution, the more nearer the pronoun to an Entity, the more likely is the reference. We consider two types distance : Number of NP chunks between the pronoun and the candidate Noun Phrase and the no of sentence between the them. The sentence distance represents recency of the candidate referent i.e. a candidate referent is more likely to be in the sentence which are nearer to the pronoun. Chunk distance represents the position of candidate NP in relation to the pronoun as compared to other referent. This is important because the position of a NP relative to the other NP‘s give information about the probable grammatical role of the candidate NP. Also, pronouns which are at the beginning of the sentence more frequently refer to the *‘SUBJECT’* of the previous sentence.
- Animacy : Though animacy can be biological or grammatical property of entities, in linguistics the term is synonymous with a referent‘s ability to act or instigate events volitionally [35]. In Hindi, use of references, choice of postpositions greatly depends on volitionality which is governed by features like animacy. [35] describes three values for animacy : ‘human’, ‘animate’ and ‘rest’ for annotation in Hindi dependency treebank. Value *‘Human’* used for entities which are

biologically animate such as person names. *'Animate'* is used for non-human entities which are not biologically animate, but realized as 'animate' in a particular context. For ex :

(50) भारत ने सार्क सम्मेलन दिसम्बर में करवाने के पाकिस्तान के प्रस्ताव को
India saarc summit in december to organize pakistan‘s proposal
खारिज कर दिया ।
rejected

'India rejected pakistan‘s proposal of organizing Saarc summit in december.'

In the above example, though भारत (*'India'*) and पाकिस्तान (*'Pakistan'*) are in actual geographical locations, and not a human entity, they are realized as humans as they refer to nations or national governments here. The value *'rest'* is used to refer to those entities which are non-animate in a given context.

4.1.3.3 Results of the hybrid approach for single candidate model

Taking results of the rule based approach as baseline, we evaluate the improvement achieved by learning based approach over rule based approach for different combination of features. We evaluate the learning based approach using three learning algorithms: Decision tree, Support Vector Machine (SVM) and Memory based learning. Table 4.4 shows the accuracy of hybrid system over development data for best combination of feature in increasing order of F-score achieved for three different learning algorithms.

	Decision Tree			SVM			Memory based learning		
	P	R	F	P	R	F	P	R	F
Rule based system	.55	.55	.55	.55	.55	.55	.55	.55	.55
Rule based+Dist+ NEC	.60	.60	.60	.56	.56	.56	.60	.60	.60
Rule based+Dist++NEC+Ag	.61	.61	.61	.57	.57	.57	.61	.61	.61
Rule based+Dist++NEC+An	.62	.62	.62	.57	.57	.57	.61	.61	.61
Rule based+Dist++NEC+An+Ag	**.63**	**.63**	**.63**	.58	.58	.58	.62	.62	.62

Table 4.4: Average results of hybrid approach using single candidate model over 4 iterations on development set for three algorithm; P= precision, r=recall; F=F-score

As it can be seen from the above table [3], there is gradual increase in accuracies with the addition of features. Almost linear improvement can be observed for the two algorithms i.e. Decision Tree and Memory based learning in which the Decision tree performs best, while the improvement for SVM from rule based to hybrid, with all features used, is nominal. It is important to note here that since the final hybrid approach attempts to resolve all the input anaphors, it predicts referent for all the input pronouns, though they may or may not be resolved correctly, thus the 'number of pronouns output by the system'

[3] D: Distance, Ag : Agreement, An: Animacy, NEC: named entity category

is same as 'total no of pronouns input to the system'. Formally:

$$\#No\ of\ pronouns\ output\ by\ system = \#Total\ No\ of\ pronouns\ input\ (or\ in\ the\ data) \quad (4.5)$$

Given equation (5), it can be derived from (2) and (3) that

$$Fscore = precision = recall \quad (4.6)$$

Hence, as can be seen in Table 4.4, values of precision, recall and Fscore are all same.
Based on the observation on development data, we run the Decision tree algorithm setting for the Test set. Since, the training data is less. we merge the annotated development set into the training data to train the classifier for final test evaluation. Table 4.5 shows the results over the test data using the decision tree classifier.

	Precision	Recall	F-score
Rule based system	.60	.60	.60
Rule based+Distance+NECat	.64	.64	.64
Rule based+Distance+NECat+Agreement	.66	.66	.66
Rule based+Distance+NECat+Animacy	.68	,68	.68
Rule based+Distance+NECat+Animacy+Agreement	**.69**	**.69**	**.69**

Table 4.5: Results of the hybrid approach with single candidate model (using decision tree algorithm) on Test data

4.1.3.4 Twin candidate model

As described in (Yang et,al. 2008) [76], the assumption behind the single-candidate model is that the probability of a candidate being the antecedent of a given anaphor is completely independent of other candidates in context. However, for an anaphor, the determination of the antecedent is often subject to preference among the candidates. To evaluate the resolution performance while using the relative preference of the possible candidates, we also implement the twin candidate model.

As proposed in [76], the twin candidate model explicitly learns a preference classifier to determine the preference relationship between candidates. Formally, Given pronoun *'ana'* and *'n'* possible candidate antecedents C_1, C_2,......C_k,......C_n, the model considers probability that a candidate C_k is antecedent as: *"the probability that the candidate is preferred over all other competing candidates"*. Formally:

$$\begin{aligned} p(ante(C_k)|ana, C_1, C_2, ...C_n) &= p(C_k \succ C_1, ...C_{k-1}, C_{k+1}, ...C_n|ana, C_1, C_2, ..., C_n) \\ &= p(C_k \succ C_1, ...C_k \succ C_{k-1}, C_k \succ C_{k+1}, ... \\ &\quad ......C_k \succ C_n|ana, C_1, C_2, ..., C_n) \end{aligned} \quad (4.7)$$

Assuming that the preference between C_k and C_i is independent of the preference between C_k and the candidates other than C_i, we get :

$$p(ante(C_k)|ana, C_1, C_2, ...C_n) = \prod_{1<i<n, i\neq k} p(C_k \succ C_i|ana, C_k, C_i) \tag{4.8}$$

In twin candidate model, training instances are of the form : i[ana,C_i,C_j], where ‘ana’ is anaphor and C_i and C_j are antecedent candidates and C_j is closer to C_i in position. Instance is labeled “10” if C_i is preferred over C_j as the antecedent, or “01” if otherwise. A training instance for an anaphor should be composed of two candidates : one being an antecedent and the other being a non-antecedent. Thus for each anaphor, training instance are created by pairing the actual antecedent C_{ante} with all the candidates C_{nc} which are non-antecedent of the anaphor. If C_{ante} is closer to *ana* than C_{nc}, then instance i[ana, C_{nc}, C_{ante}] is labeled “01”, otherwise instance i[i,C_{ante},C_{nc}] is labeled “10”.

Using feature vectors for the training instances as described in section 1.6, a classifier can be trained. For test instance all candidates are paired in the tuple with the anaphor i[ana, C_i, C_j]. For a test instance i[ana,C_i,C_j], classifier output “10” represents C_i is preferred over C_j and “01” represents otherwise.

Antecedent are selected in a tournament fashion. Candidates are compared linearly from the beginning to the end. First candidate is compared with the second one using a classifier as to which is more preferred. Less preferred is eliminated and preferred one is compared with the third one. Process continues until all are compared. Preferred candidate in the last comparison is selected as the antecedent. Following training instances will be created for इसके (it‘s) in example (50):

- इसके (it‘s) ,उन्होंने(He), feature-set-1,खाद्य सुरक्षा विधेयक(Food security bill), feature-set-2,10
- इसके (it‘s) ,मंजूरी(approval), feature-set-1,खाद्य सुरक्षा विधेयक(Food security bill), feature-set-2.10
- इसके (it‘s) ,प्रधान मंत्री की(prime minister), feature-set-1,खाद्य सुरक्षा विधेयक(Food security bill), feature-set-2,10
- इसके (it‘s) ,आज(today), feature-set-1,खाद्य सुरक्षा विधेयक(Food security bill), feature-set-2,10

Similarly training instances will be created for the other pronoun उन्होंने (He.NOM). Feature set used for twin candidate model is same as that of single candidate model:

4.1.3.5 Results of the hybrid approach for twin candidate model

Similar to single candidate model, for twin candidate model too, taking results of the rule based approach as baseline, we evaluate the improvement achieved by learning based approach for different combination of features. Results for twin candidate model on development set are as shown in Table 4.6:

	Decision Tree			SVM			Memory based le	
	P	R	F	P	R	F	P	R
Rule based system	.55	.55	.55	.55	.55	.55	.55	.55
Rule based+Distance+NEC	.608	.608	.608	.602	.602	.602	.604	.604
Rule based+Distance+NEC+Agreement	.629	.629	.629	.612	.612	.612	.626	.626
Rule based+Distance+NEC+Animacy	.639	.639	.639	.627	.627	.627	.632	.632
Rule based+Distance+NEC+Animacy+Agreement	**.646**	**.646**	**.646**	.630	.630	.630	.642	.642

Table 4.6: Average results of hybrid approach using twin candidate model over 4 iterations on development set for three algorithm; p= precision, R=recall, F= F-score

As can be observed from above table, the performance of the twin candidate model for all three algorithm is almost equal, however, similar to the single candidate model, still the highest Fscore can be observed for the Decision Tree algorithm. Based on this observation, we evaluate the hybrid system over testing data for the best performing algorithm i.e. Decision tree.
Table 4.7 shows the results obtained for twin candidate model on test data using Decision tree algorithm for combination of different features, similar to the single candidate model.

	Precision	Recall	F-score
Rule based system	.60	.60	.60
Rule based+Distance+NECat	.64	.64	.64
Rule based+Distance+NECat+Agreement	.66	.66	.66
Rule based+Distance+NECat+Animacy	.67	.67	.67
Rule based+Distance++NECatAnimacy+Agreement	**.68**	**.68**	**.68**

Table 4.7: Results of hybrid approach with twin candidate model (using decision tree algorithm) on Test data

As can be observed from above table, that although the final results for twin candidate are comparable to that of single candidate model, the best accuracy which is achieved for combination of all the features is slightly less than that of the single candidate model.

4.1.4 Error analysis and Discussion

For overall Entity anaphora resolution , we saw that the rule based approach achieved accuracy of up-to **0.6** FScore. Moreover, given the limited amount of training data and ambiguity in third person pronouns references, the learning based approach demonstrated considerable improvement (**.09** for single candidate model and **.08** for twin candidate model) over the rules based approach performance. The assumption behind experimenting with twin candidate model was that comparing one candidate referent with other would provide the classifier ability to learn the mutual preferences. However, though the result achieved using twin candidate are comparable to that of single candidate model, they are not better than those of later. The possible reason behind failure of this model could be the selection strategy of tournament elimination adopted to choose the final referent. This may be possible that since the recency is an important preference factor, the direct selection of nearest positive candidate as in single candidate model predicts the correct referent more frequently as compared to that of giving equal preference to all possible candidate referents as in twin candidate model.

As can be seen from table 4.5 and 4.7, all individual features add to the performance of learning based module and the best performance is achieved by combination of all the features. Also, as can be seen from results on development data for both the single candidate and double candidate model, among the three machine learning algorithms, decision tree performs the best. Though, along with these, other supervised learning algorithms can also be used in similar framework, however, in this work, our aim is not to compare or determine the best among the supervised learning algorithms, but to evaluate the performance of different features in anaphora resolution for Hindi in a supervised learning setting.

Though an exact comparison with the previous works in Hindi is not possible due to the unavailability of the data used in those approaches, we provide a tentative comparison of our results[4] with one of the related approach [62] (Uppalapu et al)[5].

	Uppalapu-Short	Uppalapu-Long
Precision	.86	.64
Recall	.86	.64
F1-score	.86	.64

Table 4.8: Results for (Uppalapu et, al)‘s Long and Short story data.

	Our hybrid approach
Precision	.69
Recall	.69
F1-score	.69

Table 4.9: Best results achieved by our approach

It is important to note here that [62] (Uppalapu et al) do not report there results in terms of precision, recall or F score, they report accuracy of the system which is calculated as ratio of ‘Total No of Pronouns Resolved Correctly’ to ‘Total No of pronoun’. Assuming that the ‘total number of pronoun’ as they report is same as the ‘total number of pronouns present in the evaluation data’, the precision and recall

[4] Best hybrid system result for single candidate model

[5] Uppalapu-Long and Uppalapu-Short are the results of [62] (Uppalapu et al) for short and long story data respectively

both become equals to the accuracy, hence all three values are same for both the data used in [62] (Uppalapu et al).
(Uppalapu and Sharma. 2009) [62] presented their results for two sets of data, i.e. long and short stories. From comparison of table 4.8 and 4.9, we can infer that the overall performance of our approach is better than this system for long story data, although its low for short story data for which they report exceptionally high results. We presented our results on treebank data which contains news articles from various domain where the average size of each article is 20 sentence, hence it can be observed from above results that the our approach performs consistently even for longer texts and domain independent data, however it can be inferred from results of [62] (Uppalapu) on short story data that other information can provide better results for texts of specific genre and of smaller size.

Moreover, since (Uppalapu and Sharma. 2009) [62] report their results on very small data set, it does not give a generalized picture about their performance, as the increase in test data size can significantly affect the system performance, whereas we report our results on a test data size 4-5 times of these systems which covers not only different types of pronouns, but also texts from a variety of domains.

As observed from the results in table 4.5 and 4.7, the performance of our approach for third person pronouns is relatively low than that of other type of pronouns. As discussed in chapter-1, third person pronoun in Hindi further has two forms : proximal(root form यह (*yah*)) and distal(root from वह (*vah*)). In order to further analyze the low performance of the approach on third person pronouns, we evaluate the performance of the approach separately on these two forms. Table 4.10 shows a break-up of the distribution of third person pronouns into the two forms and their respective accuracies.

	Total	Correct	F-score
Proximal	132	43	.32
Distal	574	394	.68
Total Third person	706	437	.61

Table 4.10: Seperate results for proximal amd distal third person,

As observed from above table, while the performance of the system is significant for the Distal form, its very low for the proximal form. In Hindi, the third person proximal pronoun and its forms are difficult to resolve since they are also used for demonstratives and to refer to abstract objects such as events, which creates ambiguity for manual annotation as well as for automatic resolution.

Also, dependency information can not always resolve other types of pronouns since fine-grained dependency relation are ambiguous and sparse to learn from a small data set.
In this section, we discussed a hybrid approach for anaphora resolution which relies highly on syntactic information, which in our cases is dependency structures and relations. Certainly, the requirement to use this feature is the availability of annotated dependency treebanks or parsers which can give dependency parse for sentences. In our approach, we used the treebank data annotated with dependency structures and other linguistic information. However, in real time applications of anaphora resolution, annotated

data is rarely available. Input to most of the real time applications is either raw text or text with very primitive or limited linguistic information and features. There are two possible solutions to this problem. First is to use high performing preprocessing tools such as morph analyzer and dependency parser in order to apply the syntax-based data-dependent method similar to the one described in section 4.1, However, the performance of currently available tools for Indian Language NLP, especially dependency parsers is not as high to be used for other application [34], [48]. Certainly, this error propagates further in the anaphora resolution process. The second solution is to explore the approaches involving heuristics that can be used to extract reasonably correct syntactic, semantic and discourse features from minimum information available in the input data. In the next section, we discuss our experiments to explore these shallow features for Entity Anaphora resolution.

4.2 Experiments with shallow features for Entity Anaphora Resolution:

In this section, we discuss our participation (Dakwale and Sharma, 2011) [22] at ICON-2011 Anaphora Resolution Tool Contest in Indian Languages[40] in which the task is to resolve the Entity anaphora references given the text with limited features in the *Tool Contest* data set. We implemented two approaches, first is a fully supervised learning based approach for all the languages that were evaluated in the tool contest. Second, based on the available knowledge for Hindi, we use a hybrid approach which adds over the machine learning approach. However, this hybrid approach is different from that described in section 4.1, since for this task, very limited information was given in the data unlike the treebank data used in section 4.1 where syntactic and semantic information like dependency and animacy were available. Here, we attempt to derive reasonable approximations of these high level features from the limited information available in the tool contest data. In subsection 4.2.1, we describe, Anaphora Resolution tool contest task and explain the details of the data given in the tool contest. We discuss our approaches in subsection 4.2.2 and 4.2.3.

4.2.1 Tool contest task and Data used:

Anaphora Resolution tool contest in Indian languages was held in conjunction with International Contest on Natural Language Processing (ICON), 2011. The task of the contest was to identify the antecedents for an anaphor from a POS, NP Chunked and NE tagged corpus. Participants were provided training, development and test data in CONLL format [16] to report the efficiency of their Anaphora Resolution System. Task was evaluated for three Indian languages: Hindi, Bengali and Tamil. Efficiency of the system was measured using Precision, Recall and F score.
As described above, the task data contained documents containing set of sentences annotated with Part-of-speech (POS) tag for each token, chunking and Named entity information. The training and development data were additionally, annotated with indexes representing the anaphora links for each entity reference.

4.2.2 Learning based resolution with minimum features:

We, first attempted to implement supervised learning approach with very limited features given in the data. Since, unlike the treebank data, there is no information available about possible anaphors or referents, before implementing the resolution algorithm we need to identify which expressions or units are either probable anaphors or probable candidate referents. This process is called Mention detection which we explain below:

4.2.2.1 Mention detection:

In terms of anaphora resolution, mention detection stands for the identification of the expressions or units which are either probable anaphors or probable candidate referents. Although, given the POS tags of the words, the possible anaphors can be identified by selecting the words with POS tag as that of pronoun ('PRP'). However, not all expression annotated in the category of pronouns are anaphoric. Thus, the first step is to determine the anaphoricity of a pronoun, that is whether an expression annotated as pronoun is anaphoric or not. This is a trivial task in Hindi, This is because non-anaphoric instances of pronouns like pleonastic *'it'* are not found in Hindi. Moreover, with the exception of indefinite pronouns, all the lexical elements which have a POS tag as *'PRP'* are anaphoric in almost all the cases. Thus, the simple way to identify the anaphoric pronouns is to exclude all the indefinite pronouns. Indefinite pronouns in Hindi are closed set, hence these can be identified using a complete list (which is given in Appendix A) and excluded from resolution.

Secondly, not each expression of any span length could be a possible referent candidate. Thus, in order to reduce computational complexity and to increase precision, it is important to identify which pronouns are anaphoric and which Noun phrases are potential candidates.

To identify or choose possible candidate mentions, we rely on chunk boundaries. In the dependency treebank, Chunks define the minimal units among which dependency relations are identified. Hence, chunk themselves provide minimum required information about any entity or markable which may be referred by another expression or anaphora. However, exact specification for an entity may require combination of more than one chunk. If a markable (representing complete specification of an entity) spans beyond single chunk, then we consider only head chunk, since all the linguistic features of the mention are projected on the head. Thus following rules give the identification of anaphor and candidate mentions: Thus following rules give the identification of anaphor and candidate mentions:

- Consider all NP chunks to be candidate mentions.
- Select all Pronoun chunks (NP chunks with head as pronoun) as anaphora.
- Reject any pronoun if it belongs to the list of indefinite pronouns.

4.2.2.2 Resolution algorithm:

The supervised learning based resolution algorithm that we used for the tool contest task is similar to that described in section 1.6.1, that is the single candidate model of (Soon et, al. 2001) [59]. After the pronouns and possible candidate referent mentions are identified, a classifier is used to predict whether a particular candidate referent is a actually a referent of a given pronoun. Based on the model learned in the training, classifier labels the instances, as positive or negative, which are created by pairing each pronoun with each possible candidate referent. Instances represent the set of features extracted from pronoun, candidate NP and their context. We choose the actual referent as that candidate NP which is nearest to the pronoun out of all the positively labeled instances. Similar to the approach in section 1.6.1, we consider all NP‘s up-to three sentence previous to pronoun as candidate referents.

4.2.2.3 Features:

As discussed earlier, for the machine learning approach, we use the only features provided in the data. We discuss these features as follows:

- Pronoun: Assuming no information is available about the pronoun, the pronoun itself can be used as a feature, the pronominal form can possible help the algorithm to learn pronominal category.
- Part of Speech tag: We consider POS tag of the head of the candidate NP as a feature. This is because, the head of a common noun is more frequently annotated as ‘NN’, while that of a proper noun is annotated as ‘NNP’
- Named Entity Category of candidate referent: For some of the expressions, Named entity category is provided in the data. As already discussed in section 1.6.2, NE categories are important feature which identify Noun phrases as ‘Person’, ‘Organization’, ‘Location’ etc and can help the classifier to learn which pronominal forms more preferably refers to NP‘s of which NE category.
- Distance: Similar to the approach in section 1.6.1, we consider two distances between anaphora and candidate NP as features : Number of sentences between pronoun and candidate referent and number of NP chunks between them.

4.2.2.4 Evaluation of learning based approach:

Similar to section 1, we evaluate the performance of the learning based approach for three machine learning algorithms over development data i.e. Decision Tree (DT), Support vector machine (SVM) and Memory based learning (MBL). Though, we implemented the learning based module for all three languages, since in this thesis we only focus on Hindi anaphora resolution, we report results only for Hindi. Results for Bengali and Tamil can be found in [22]. The following tables show the performance of three algorithm in terms of precision, recall and F-score:

	Precision	Recall	F-score
SVM	.296	.309	.302
Decision Tree	.298	.361	.326
MBL	.329	.316	.322

Table 4.11: Development set results with minimum feature set for **Hindi**

We see from the above results that for the basic feature set, performance of different algorithms are almost similar. Although the computed F-score for all of them is quite low. However, the results are encouraging given the limited feature set and the limited training corpus. Also it gives important insights into further applicability of these algorithms depending on the properties of the anaphoric relations in Hindi. Hence, we further use the results of these experiments to improve the performance in Hindi by enhancing the feature set and developing a Hybrid approach by incorporating a rule-based approach in conjunction to the learning approach.

4.2.2.5 Extended feature set for Hindi:

In the supervised learning approach as discussed in previous subsection, we used only those features which were directly available in the given data. However, some more features can be derived from the given features in the data. We derive some of the features from the given data as described below:

- Morphological features of the pronoun: Although, we do not use any morph analyzer, some of the morphological information (such as Number, person and case), about the pronoun can be derived from its form only. For example a pronominal form उसने (*'usne'*) suggest that it is singular, third person pronoun used in Nominative case. All these values can be identified from the list of the pronouns given in Appendix A.
- Category of pronoun: Similar to morph features, type of category of pronoun can also be easily identified from the list given in Appendix A. We use the 4 categories described in section 4.1. i.e. 'Reflexive', 'Locative', 'Relative' and 'Personal' pronouns.
- Vibhakti (post-position) associated with the pronoun and the candidate NP: As already discussed in section 4.1, dependency relations provide important syntactic information for anaphora resolution. Though, the post-position do not have one-to-one correspondence with dependency relations especially *'karakas'*, some post-positions more frequently signal some specific dependency relations. For example: postpositions ने (*'ne'*) is more frequently used with *'karta'* (*'k1'*), similarly को (*'ko'*) is more frequently used with *'karma'*. Thus post-positions can act as an approximation of the dependency relations.
- Position of the pronoun and candidate referent in the sentences: Though, Hindi is a free word order language, the frequent word order in descriptive text especially news text is 'Subject-Object-Verb', thus the relative position of the pronouns and noun phrases give some cues about the

grammatical role of the entities. That is, any entity or pronoun at the beginning of the sentence is more likely to be a *'Subject'* or *'karta'* and that at the position immediately preceding verb is more likely to be an *'Object'* or *'karma'*, as the grammatical roles are important features for anaphora resolution, we consider three positions as an approximation of these roles, which are beginning of sentence, position preceding to verb and other positions.

4.2.2.6 Evaluation of Hindi anaphora resolution with extended feature set:

	Precision	Recall	F-score
SVM	.417	.352	.381
Decision Tree	.299	.28	.28
MBL	.33	.31	.322

Table 4.12: Development set results for **Hindi** with Extended Feature set

Though, the results in the table above show considerable improvement over the shallow features in Table 4.12, the performance is still low. Hence, as a further extension to improve the performance, we implement a rule based approach in conjunction with the machine learning approach.

4.2.3 Hybrid approach with shallow features:

The work-flow of the hybrid approach is same as the one described in section 4.1, we first try to search the referent of a pronoun based on some rules, derived from shallow features. If the rules can not locate a referent, then the pronoun is passed to classifier which attempts to predict the referent. We derive some simple rules based on this information which are described below, however it is important to note here that these rules are derived from shallow features and are not hard constraints as the one used in the dependency based resolution described in section 4.1:

- The referent of a reflexive pronoun is the *'subject'* or *'karta'* (*'k1'*) of the same sentence. Post-position ने (*'ne'*) gives an approximation of the dependency relation *'karta'*. Hence, for reflexive pronouns, we select that nearest NP as the referent, which has post-position ने (*'ne'*) attached with it.
- The referent of a relative pronoun is the NP which is modified by the relative clause, In most of the cases, relative clauses are placed just next to this NP with the relative pronoun at the beginning of the relative clause, hence for the relative clause, we select the referent as the NP immediately preceding the pronoun.
- In descriptive discourses, narration by same speaker tend to occur in continuation. This implies that, if the speaker of a narrative verb (such as *'tell'*, and *'say'*) is a pronoun, then if there is another

narrative pronoun in preceding sentence, then it is more likely that the referent of pronoun in the will be that Noun phrase which is subject of the preceding narrative verb. For example:

(51) चुनाव आयुक्त ने बताया कि विधानसभा के चुनाव शांतिपूर्ण तरीके से संपन्न हुए । उन्होने आगे कहा कि सबसे अधिक मतदान मिजोरम मे हुआ ।

‘Election commissioner told that assembly elections were completed peacefully. Further, he said that the highest voting took place in Mizoram ’

In above example, pronoun उन्होने (*‘he’*) is the subject of the narrative verb कहा (*‘said’*), whose referent is चुनाव आयुक्त (*‘Election commissioner’*) which again is the subject of narrative verb बताया (*‘told’*) in the previous sentence. Thus, we derive the rule: If the pronoun is a subject of a narrative verb, then search for the subject of another narrative verb in previous four sentences and if found select it as the referent of the pronoun.

- As already discussed earlier, locative pronouns यहां (*‘here’*) and वहां (*‘there’*) in Hindi refer to entities which are names of places, ‘Locations’ can be identified by their Named Entity category label if available. Thus we derive the rule: For a locative pronoun, search for a mention, with Named entity category as ‘Location’, if found select it as the referent of the locative pronoun.

4.2.3.1 Evaluation of the Hybrid approach:

The table below shows the results of the Hybrid approach for Hindi on development data:

	Precision	Recall	F-score
SVM	.480	.452	.465
Decision Tree	.455	.468	.461
MBL	.469	.488	.478

Table 4.13: Development set results for Hybrid approach for **Hindi**

As can be seen from above table, the results for the hybrid approach for all three algorithms is mostly similar. The probable reason behind this is that in when machine learning algorithms are used in conjunction with the rule based approach, majority of the reference are resolved by the rule based system and the difference achieved by machine learning algorithms is nominal and is similar for all the algorithms. However, the highest results are achieved with Memory based learning (MBL). Hence, we executed the hybrid approach with Memory learning based algorithm on test set and submitted the output for Tool contest evaluation. The results obtained for test set are as follows:

Precision	Recall	F-score
.523	.521	.522

Table 4.14: Test set results for Hybrid approach for **Hindi**

4.2.4 Discussion:

Of all the systems participated in ICON-2011 tool contest, our approach reported best results for Hindi Anaphora Resolution [40]. As can be seen from table 4.14, the results obtained for the approach using shallow features are less than the results of experiments described in section 4.1 (Results in Table 5). This is because, the approach described in section 4.1, uses higher level knowledge such as dependency and animacy annotated in the treebank, while the approach described in this section used shallow features and rules extracted from the limited information given in data. Certainly, these features are only approximation of the syntactic and other grammatical features which are otherwise available mostly noise-free in the treebank data. Nevertheless, this shallow approach has achieved considerable performance given the limitations of available information. However, further analysis and experiments are required to improve the extraction of shallow features from limited information.

4.3 Summary

In this chapter, we discussed our approaches and experiments for Entity anaphora resolution in Hindi. We discussed two approaches. In section 1, we discussed a hybrid approach in which the focus in on using deep linguistic features such as dependency structures, morphological properties, animacy and other agreement features. This approach achieved accuracy up-to 69% which gives encouragement towards further exploration of high level features and more advance and detailed algorithms for anaphora resolution in Hindi. In section-2, we discussed our experiments for anaphora resolution using shallow features derived from limited linguistic knowledge. For these experiments shallow features were extracted as an approximation of the deep linguistic features. With these experiments too, we achieved an accuracy of up-to 52% which implies that while these deep features do play an important role for anaphora resolution, they can also be extracted from limited features and the words of the discourse itself.

Chapter 5

Experiments in Event Anaphora and Co-reference Resolution

Though, the focus of our work is on Entity-anaphora resolution, we also did some preliminary work on Event anaphora and Co-reference resolution. Similar to Entity anaphora resolution, for event and co-reference too , we explore use of dependency structures. In section 1, we discuss our experiments with event anaphora resolution and in section 2, we discuss co-reference resolution.

5.1 Experiments in Event anaphora resolution

The reason for having separate resolution process for Event and Entity anaphora is obvious. In concrete or Entity reference, an anaphora is referring to a concrete entity, thus possible referents are Noun phrases, while in Abstract or event reference, an anaphor refers to an abstract object, thus the possible referents are verbs, clauses and propositions. Moreover, features and linguistic properties of candidates in both cases are different. Thus it is efficient to consider separate resolution processes for both types of references.
As discussed in chapter 1 and already established in chapter 2, in Hindi (and in many other languages), for some pronouns, it is difficult to determine whether they refer to an entity or event, only on the basis of their lexical form, hence in order to consider resolution of both the types separately, a classification module is required apriori which classifies ambiguous anaphors into Entity and Event references. However, in this work, we only focus on resolution of Event anaphora assuming that anaphors referring to Events are already identified. We explain the event references by a simple example:

(52) मंत्री जी ने   चुनाव के बाद अपना वादा   नहीं निभाया ।यह उनके समर्थकों कि
minister.NOM after election his   promise did not fulfill. this his supporters
नाराजगी का कारण बना ।
anger‘s   reason became

‘The Minister did not fulfil his promises after wining the election. This became the reason for his supporter‘s anger.’

In above example: the pronoun यह (this) is referring to the event: [वादा नहीं निभाया (*‘did not fulfil promise’*)]. That is, an event of ‘not fulfiling the promise’. Here, the main event or the head of the referent is the verb : [निभाना (*'to fulfil'*)] and its modifiers are negation [नहीं (*'did not'*)] and [वादा (*'promise'*)], both of which are participants of the event. This could also be observed from following dependency tree.

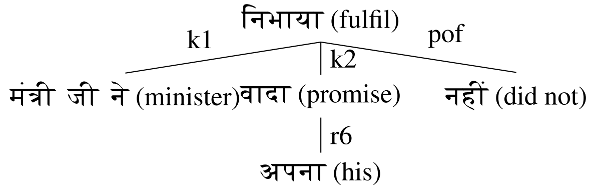


Figure 5.1

For event anaphora resolution, first it is important to identify the head of the referent that is the finite or non-finite verb. Participant elements of the event will always be the modifiers of the root verb. Identification of these participants requires knowledge of the verb semantics and the discourse structure. However, in this work, we only focus on resolving the head of the referent of event anaphora that is the head verb of the event.

Though similar to Entity anaphora, for Event anaphora resolution too, we use constraints based on dependency structures, however, due to less data available for Event anaphora, we implemented only a rule based approach unlike the hybrid approach for Entity anaphora resolution.

5.1.1 Data used:

For experiments in Event anaphora resolution, we annotated the same 324 text data as used in Entity anaphora resolution using dependency structures (section 4.1), In this data, 523 pronouns are identified and annotated as 'Event reference'. Since, for Event anaphora, we apply only a rule based approach, no data is required in training, hence to get the resolution performance on a larger set of pronouns, we use complete data for evaluation. Thus the total 324 text data containing 523 event pronouns is considered as test data.

5.1.2 Rule based resolution for Event anaphora resolution

We identify some pronominal forms which refer to events and derive rules to locate the referent in the dependency structure or the discourse context of the pronoun.
In Hindi, only pronominal form which is ambiguous in its reference type is यह (*'yaha'*) and its inflected forms such as [इसके (*'of this'*)], [इससे (*'from this'*)][1] which represent the different forms inflected for number, case etc. These can refer to both Entity or Event. There can be two translations of यह (*'yaha'*) in English : *'it' (proximal)* and *'this'*.

[1] A complete list of pronominal forms is provided in Appendix A

Following are some observation about these pronouns and correspondingly the rules are derived for resolution.

5.1.2.1 Resolving यह *'yaha'*

In many cases, यह (*'yaha'*) in its inflected form refers to a proposition which is expressed or narrated in the same sentence. We discuss some of these cases below and finally derive an algorithm for resolving structural references in these cases:

- In some sentences involving a complementary clause, pronoun यह (*'yaha'*) act as an argument of the main verb and simultaneously refers to the event in the complementary clause. As per the CPG based dependency annotation framework, the complementary clause in these cases, is in a *'samanaadhikaran'* relationship with the pronoun, that is the complementary clause is equivalent to the pronoun, hence there are two ways of annotating it in the dependency structure of the sentence.

  - The first way is to consider the complementary clause as the modifier of the pronoun and attach its dependency structure (rooted at the verb) under the complementizer की (*'that'*), The complementizer is in turn attached under the pronoun (with a label *'rs'* representing *'samanaadhikaran'*) in the dependency tree of the complete sentence. Thus in these cases, reference can be resolved by selecting the root verb of the subtree attached under the complementizer. Consider following example:

    (53) यह अलग बात है की मन्त्री जी ने चुनावों के पहले ही अपनी हार
    it a different thing is that minister.NOM elections even before his defeat
    मान ली
    accepted
    'It's a different thing that minister accepted his defeat even before elections.'

    Figure 5.2 shows the dependency structure of the example (54).

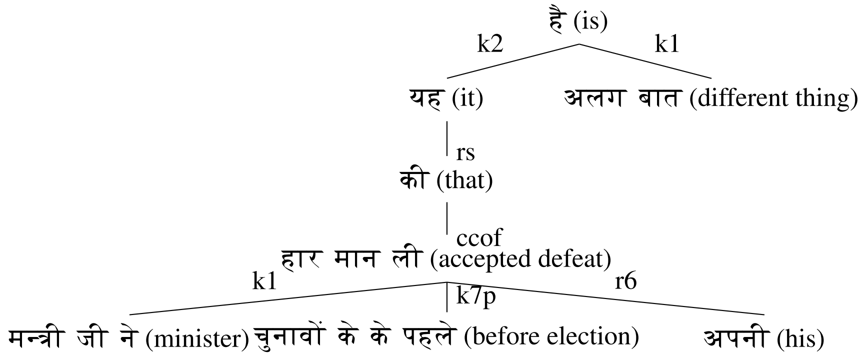

है (is)
k2
k1
यह (it)
अलग बात (different thing)
rs
की (that)
ccof
हार मान ली (accepted defeat)
k1
k7p
r6
मन्त्री जी ने (minister)
चुनावों के के पहले (before election)
अपनी (his)

Figure 5.2

In above example, the root verb of the complementary clause [(मन्त्री जी ने चुनावों के पहले ही अपनी हार मान ली) ((minister accepted his defeat even before elections))] is attached under the complementizer की (*'that'*) which in turn is attached under the pronoun यह (*'this'*). This same subordinate clause is the referent of the pronoun. Thus, referent can be selected by moving downwards from the pronoun up-to the root verb of the subordinate clause and propose it as the referent as shown in figure below:

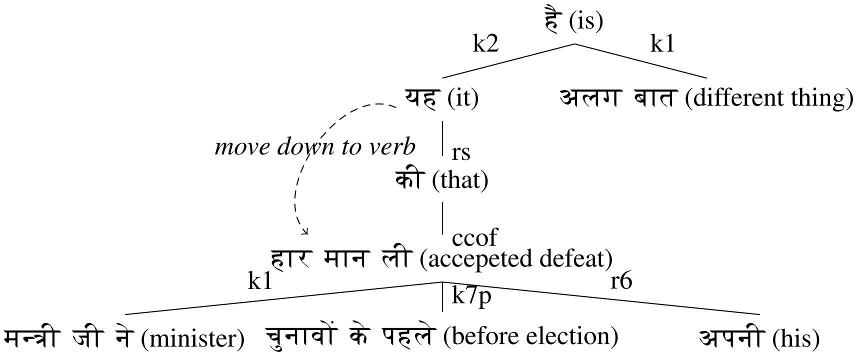


Figure 5.3 Resoution of pronoun यह with complementizer

– The second way to annotate the complementary clause, (especially in case of a non-finite attribution verb) according to CPG framework is to attach the complementizer as an argument of the attribution verb of the main clause, that is as a sibling of the pronoun and a relation equivalent to *'karaka'* of the pronoun such as *'k1s'*, *'k2s'*. Consider following example:

(54) यह पूछने पर कि क्या कांग्रेस आम आदमी पार्टी को समर्थन
this on asking that will (question marker) congress to aam aadmi party support
देगी , उन्होंने कोई जवाब नहीं दिया ।
give she any answer didn't give
'On asking that will Congress support Aam aadmi party, she didn't give any answer.'

Fig below shows the dependency structure of the example (55):

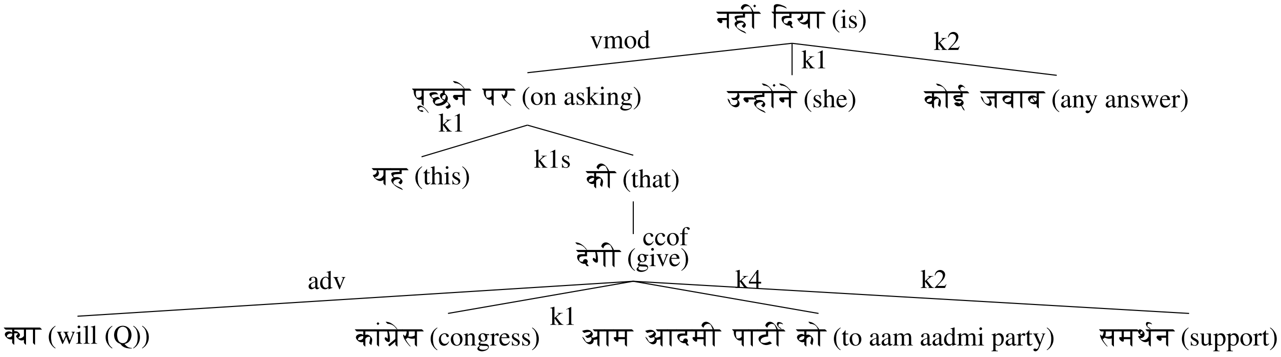


Figure 5.4 Dependency structure of example (13)

In above example, the complementary clause क्या कांग्रेस आम आदमी पार्टी को समर्थन देगी (*'Will congress support Aam Aadmi party ?'*) which is the referent of the pronoun यह (*'this'*) is attached under the complementizer की (*'that'*) which in turn is attached under the main verb with the dependency label *'k2s'* as a sibling of the pronoun. In such cases, the reference can be resolved by first moving to the sibling node of the pronoun and the selecting the head verb of the attribution clause by to its child node as shown in figure below:

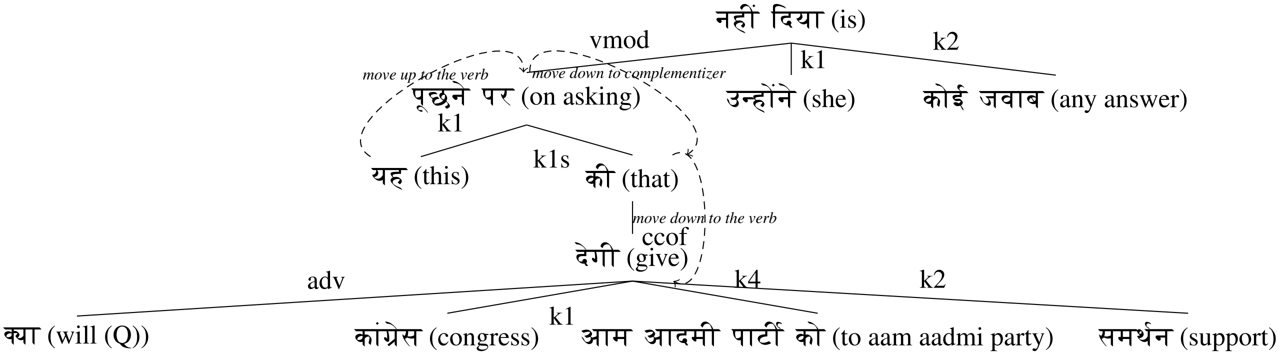


Figure 5.5 Resoution of यह with complementizer

- There are some other cases, in which there is no complementizer and the attributional or propositional clause is moved in beginning, in such cases, as per the CPG based framework, the root verb of the propositional or narration clause is considered and annotated as the root of the dependency tree of the sentence and the main clause containing pronoun is attached as its modifier or argument. In such cases, there is no subtree under the pronoun, however the referent of the pronoun is the root verb of the propositional clause which is represented as the root of the main clause. Consider following example:

(55) बूटा सिंह वास्तव में क्या करेंगे यह तो कांग्रेसी ही जानते हैं ।
buta singh in reality what will do this congressmen only know
‘Only congressmen know (this) what buta singh will do.’

Fig below shows the dependency structure of example (56)

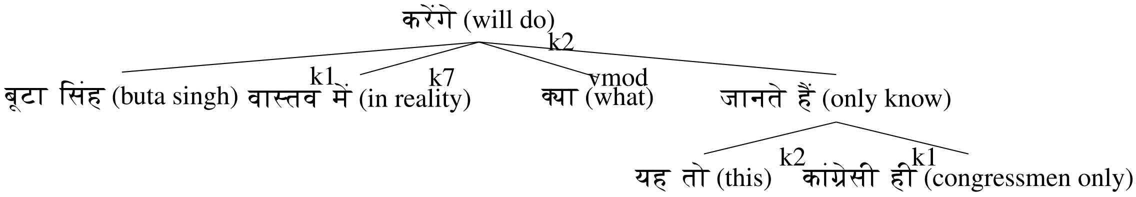


Figure 5.6

In above example, the pronoun यह *‘yaha’* is an argument of the verb जानते हैं (*‘know’*). This verb is again attached as an argument of the root of the propositional clause करेंगे, In such cases, the referent can be selected by moving upward in the tree from the pronoun up-to the root of its main verb. Thus, in above case the referent of the pronoun which is करेंगे *will do* can be achieved by first moving to the root verb of the pronoun and then moving to the head of this verb which is the root verb of the propositional clause as shown in figure below.

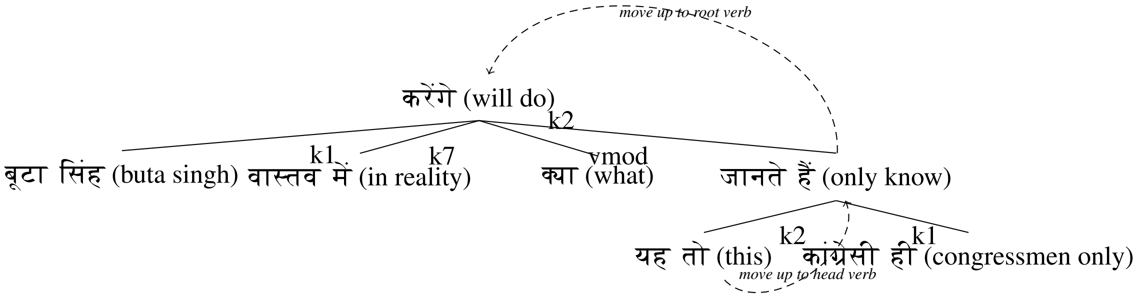


Figure 5.7 Resolution of यह

Based on the above observation, following rule can be derived for resolution of यह in cases where it has a structural or inter-sentential reference:

- If the event pronoun is यह *‘yaha’*
    - Starting from the pronoun,
    - if there is a subtree attached under the pronoun,

        - ∗ Move downwards in the tree
        - ∗ If the first node downwards is complementizer की (*‘that’*), move further downwards, else return *‘NULL’* (no referent).
        - ∗ If the next node downwards is a finite or non-finite verb, propose it as the referent and stop, else return *‘NULL’* (no referent).
    - – Else, move upwards to the head verb node of the pronoun.
    - – Check, all other children nodes (sibling nodes of the pronoun) of the verb node.
    - – If there is a verb node with dependency relation *‘k1s’* or *‘k2s’*, then propose it as a referent.
    - – Else, move further upwards from the verb node, to its head node.
    - – If this head node is a finite verb, propose it as a referent, else return no referent

5.1.2.2 Resolving इसलिए (*‘due to this’* or *‘therefore’*)

In Hindi, इसलिए (*‘isalie’*) as a pronoun is always used as a connective and refers to an event. Though, it represents causal relationship, there are multiple possible structures for sentences containing इसलिए (*‘isliye’*) depending on the other connectives with which it appears in pair:

- In some cases, where इसलिए (*‘isaliye’*) represents a *‘reason’* sense between two clauses, it appears with three possible connectives: चूंकि (*‘chunki’*), क्योंकि (*‘kyonki’*), ताकि (*‘taaki’*), all meaning *‘because’*. In these cases, the clause containing इसलिए (*‘isliye’*) is the *‘result’* and the clause containing the other connective is the *‘reason’*. Hence, the pronominal connective इसलिए (*‘isliye’*) is equivalent and referring to the *‘reason’* clause. In such cases, as per the CPG framework, the ‘reason’ clause rooted at the verb is attached under the pronoun इसलिए (*‘isliye’*) through the other connective. Hence the referent can be achieved by moving downwards in the tree from the pronoun and selecting the root verb of the ‘reason’ clause. Consider following example:

(56) दिल्ली चूंकि मिडिया का केंद्र है इसलिए यहां पब्लिसिटी कि गारंटी
delhi because media‘s center is therefore (due to this) here of publicity guarantee
होती है ।
is

‘Because delhi is the center of the media, (therefore) there is guarantee of publicity here’

Figure below shows the dependency structure of example (57)

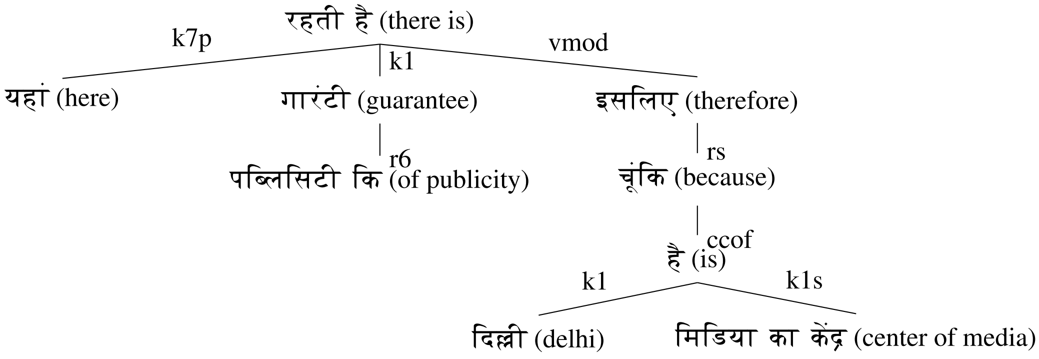


Figure 5.8

In above example, the ‘reason’ clause “दिल्ली चूंकि मिडिया का केंद्र है” (*since delhi is the center of media*) which is also the referent of the pronoun इसलिए (*‘therefore’*) is attached under the other connective चूंकि (*‘because’*) which in turn is attached under the pronoun इसलिए (*‘therefore’*). Thus, referent can be reached by moving downwards from the pronoun and selecting the root verb है (*‘is’*) attached under the other connective चूंकि (*‘because’*) as shown below:

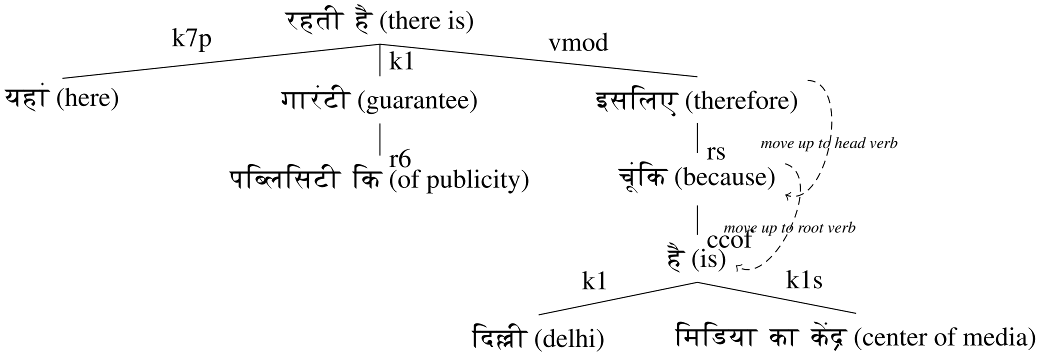
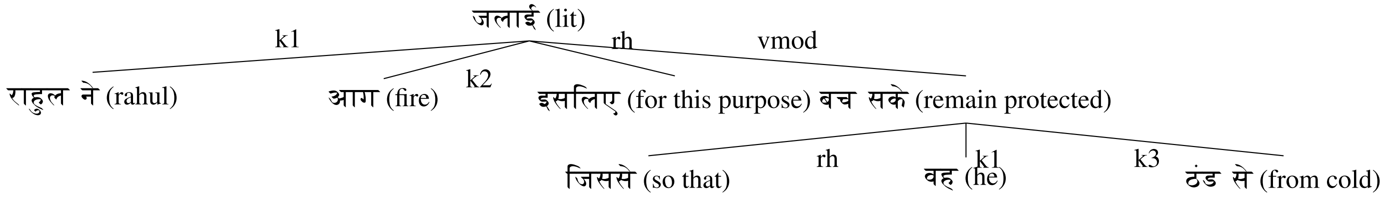


Figure 5.9

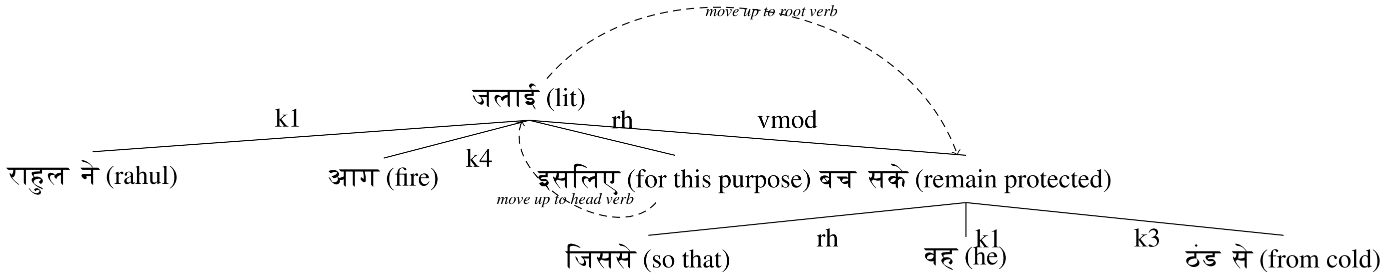
- In some cases, इसलिए (*‘isliye’*) is used to refer an event which shows the *‘goal’* of another event with which the pronoun इसलिए (*‘isliye’*) is related. However, the sense of the pronoun is redundant and it can even be dropped. In such cases, the *‘goal’* clause is attached as an argument of the *‘reason’* clause. Consider following example:

(57) राहुल ने आग इसलिए जलाई जिससे वह ठंड से बच सके
rahul the fire for this purpose lit so that he from cold remain protected
‘Rahul lit the fire (for this purpose) so that he can remain protected from the cold.’

Figure below shows the dependency structure of the example (58):



In these cases, the referent of the pronoun इसलिए (*'isaliye'*) can be reached by traversing to that verb node which is the sibling node of the pronoun and has a dependency label as *'vmod'*. In above example, the referent of the pronoun इसलिए (*'isaliye'*) is the verb node बच सके (remain protected) which is reached by moving to the sibling node of the pronoun as shown in figure below:



Based on the above observations, following algorithm can be derived for resolution of इसलिए *isliye* when it has a structural reference to an event:

- Start at the pronoun node इसलिए (*'isaliye'*).
- If there is a node attached under the pronoun with dependency relation *'rs'*
  - Move down to the child node. If there is a verb node attached under this node, select the child node as the referent, else return *'NULL'* (no referent)
- Else, move up to the head verb node 'V' of the pronoun
- Search in the children nodes of the verb node (that is sibling nodes of the pronoun)
- If there is verb node 'V_1' with dependency relation *'vmod'*, move down to this node, else return NULL.
  - Search in the children nodes of the verb node 'V_1'
  - If there is a node with dependency label *'rh'*, propose it as the referent, else return NULL.

5.1.2.3 Resolving inter-sentential references:

Since, event anaphors refer to abstract objects in preceding discourse, they also behave as discourse connectives, thus establishing a discourse relation between the sentence containing the pronoun and some previous sentence or set of previous sentences. Since in most cases, the discourse relation is between two consecutive sentences, hence in such cases, the pronoun refers to the root verb of the previous sentence. This can also be observed by a frequency analysis of the data. Thus, to resolve other pronouns, we select the head verb of the preceding sentence as the referent of the event anaphor.

5.1.2.4 Result of Rule based approach for Event anaphora resolution:

Table below shows the precision, recall and F-score achieved by the set of constraints as explained in 5.1.1 and also the overall accuracy of the rule based approach for event anaphora. Note that the first two rows of the table shows the precision, recall and F-score over the particular pronominal forms यह (*‘yaha’*) and इसलिए (*isliye’*). That is, the first column in these two rows refer to ‘total number of pronouns’ of this form and the second column ‘Total output by system’ refers to the total number of cases for these pronouns for which a referent is located. The unresolved pronoun (for which a referent can not be located using the constraints) for these two forms, are resolved by the ‘nearest verb selection rule’. Thus the third row shows the total number of pronouns which are resolved by this rule. Since this rule attempts to resolve all the event pronouns including the cases unresolved by the other two set of constraints, the total number of pronoun attempted and outputted by the system are same. The last row shows the overall accuracy of the approach.

	Total no of pronouns	Total output by system	Correct	Precision	Recall	F-score
यह *‘yaha’*	211	107	102	.95	.48	.637
इसलिए *isliye’*	59	22	21	.95	.37	.536
Resolution by nearest verb rule	253	253	138	54	.54	.54
Total event references	523	382	261	.68	.49	.58

Table 5.1: Results for Rule based Event anaphora Resolution

5.1.3 Discussion:

As can be seen from above table, the precision for both यह (*‘yaha’*) and इसलिए (*isliye’*) is very high, that is the system is able to correctly identify the actual referent for almost all the event pronouns for which it predicts the referent, using rules as discussed above. However, there are many instances for which the approach can not locate any referent, hence the low recall. These references are mostly inter-sentential or non-structural references. The referent for such instances is selected using the ‘nearest verb selection’ rule. As shown in third row of the table, the ‘nearest verb selection’ heuristics identifies nearly half the references correctly. Though the overall results for Event anaphora resolution are not very high, however the accuracy of the approach is encouraging given the limited data and using very simple heuristics based on dependency structures. Using the dependency based constraints as baseline, the performance of the approach can further be improved by using other syntactic and semantic features. Also, in this work, we have only focused on identification of the head of the event referent i.e. the root verb of the referent span. However, the actual referent span also includes the participants or arguments of the verb. In most of the cases, this span can be identified by selecting all the participants or arguments of the verb. However, in some complex cases, it may require discourse or pragmatic context of the sentences. Since, with the given data and features it is not possible to include this information in the resolution process, we have only focus on identification of the head of the referent span.

5.2 Co-reference Resolution

Next, we discuss our experiments in co-reference resolution. As discussed in chapter-1, although in many related works, co-reference and anaphora are treated as a single problem, there is basic difference between them. While co-reference is an equivalence relation, anaphora is not even transitive and hence it is non-equivalence. In other words, co-reference between two or more entities or expression implies that all these expression refer to a single real word entity. However, in anaphora, one expression refers to another expression called the antecedent and the interpretation or semantics of the anaphora comes from this antecedent[28]. Anaphora implies context-sensitivity of interpretation but this is not true for co-reference. For example, a name बिहार के मुख्यमन्त्री (*'Chief minister of Bihar'*) and नीतीश कुमार (*'Nitish kumar'*) can be co-referent without any of the two depending on the other for its interpretation. Of course, anaphoric references can coincide with co-reference, but not all co-referential relations are anaphoric, nor all anaphoric relation are co-referential.

Moreover, though there are some similarities between the approaches and features used for co-reference resolution and anaphora resolution, there are also many difference in the patterns that could be observed in analyzing and resolving the two phenomenon. Hence, we consider separate process for co-reference resolution. As we will discuss in the subsequent sections, similar to anaphora resolution, for co-reference resolution too, there is an important scope for using dependency structures and relations.

It is important to note here that for co-reference resolution, we only conduct preliminary experiments based on the observation of the patterns and features that effect the co-reference relations. We attempt to resolve simple and easily identifiable co-reference relations. However, even with the basic algorithm, we are able to achieve encouraging results.

5.2.1 Interpretation of co-reference relations

Before discussing co-reference classification and resolution, it is important to look at the information that is required for the interpretation and resolution of coreference relations. We discuss below two such variations which are relevant to our work:

5.2.1.1 Interpretation by Linguistic Vs World knowledge:

We discussed in previous chapter that interpretation and resolution of anaphora requires either no or very little world knowledge, most of the anaphoric references can be resolved by syntactic, semantic or discourse information. However, for co-reference, in some cases world-knowledge becomes crucial for resolution. This include not only situational but also temporal world knowledge. Consider following example:

(58) भारत के प्रधानमंत्री आज तीन दिन के दौरे पर मास्को जा रहे हैं ।श्री सिंह वहां
india‘s prime minister today three day on visit moscow going. mr. singh there
G20 सम्मेलन में हिस्सा लेंगे ।
in G20 summit will participate

‘Prime minister of India is going today on a three day visit to Moscow. Mr. Singh will be participating in G20 summit there.’

As it can be observed from above example, there are no linguistic features, which can identify that श्री सिंह (*‘Mr. Singh’*) or मनमोहन सिंह (*‘Manmohan Singh’*) is actually the भारत के प्रधानमंत्री (*‘Prime minister of India’*). Thus, the interpretation of coreference between the two depends on the world knowledge. Moreover, it also depends upon the temporal knowledge, because even if one knows that the two expression are co- referential in the current situation, they might not have been co-referential at the time when the given text was written or read. That is, depending on the situation, text may be read and interpreted at a time when श्री सिंह (*Mr. Singh*) is not actually भारत के प्रधानमंत्री (*‘Prime minister of India’*).

On the other hand, consider following example:

(59) मनमोहन सिंह ने केजरीवाल की आलोचना करते हुए कहा कि सबसिडी देना
manmohan singh kejriwal‘s criticism doing said that subsidy giving
अर्थव्यवस्था के लिए ठीक नहीं है।मनमोहन आज घरेलु सिलिंडर की सबसिडी 9 से
for economy is not good manmohan today domestic cylinder‘s subsidy from 9
बढाकर 12 करने के अवसर पर पत्रकारों को संबोधित कर रहे थे।
increasing 12 of doing on occasion journalists addressing doing

Criticizing Kejriwal, Manmohan Singh said thet giving subsidy is not good for economy. Manmohan was addressing journalists on the occasion of increasing the subsidy of domestic cylinder from 9 to 12.’

In above case, it is easy to identify that मनमोहन सिंह (*‘Manmohan singh’*) and मनमोहन (*‘Manmohan’*) are co-referential, without any world knowledge, just by using linguistic and lexical similarity between them. To resolve the co-reference with world knowledge, large knowledge bases are required containing as much temporal and spatial information as possible. However, even with large knowledge bases, resolution of all cases is not possible since no knowledge base can contain all the world information. Thus, though using world knowledge will certainly improve the performance, it is too expensive to implement. Moreover, there are no large knowledge bases available for Hindi. Hence, for this work, we do not use world knowledge. However, we try to resolve such cases with other syntactic and semantic information wherever possible.

5.2.1.2 Identity Vs Non-identity relation

Identity relations: are those co-reference relations in which there is equivalence of expressions and which are symmetric and transitive.

(60) रेलमंत्री लालू यादव की पत्नी और बिहार की पुर्व मुख्यमंत्री
railway minister lalu yadav‘s wife and bihar‘s former chief minister
श्रीमती राबडीदेवी ने नीतीश कुमार पर आरोप लगाया की वे राज्य के कर्मचारियों पर
mrs. rabridevi on nitish kumar accusation put that he on state‘s employees
अत्याचार कर रहे हैं।श्रीमती यादव ने कहा की लालू जी ने गत वर्ष बजट मे बिहार राज्य के
injustice doing mrs yadav said that lalu last year in budget bihar state‘s
कर्मचारियों के लिये रेल किराये मे रियायत की घोषणा की थी।
for employees in rail fare of concession announcement did

‘Railway minister Lalu yadav‘s wife and bihar‘s former chief minister Rabri Devi accused Nitish Kumar that he is doing injustics to state‘s employees. She also said that Lalu declared concession in rail fare Bihar state‘s employees in budget last year.’

In above example [रेलमंत्री (*Railway minister*), लालू यादव (*Lalu yadav*), लालू (*‘lalu’*)] is an equivalence set and each element of the set refers to a common entity.

Non Identity relations: Non identity relations are non symmetric relations. They could be set/subset, part/whole, etc which are not defined as equivalence relations. Consider following example:

(61) झारखंड के माओवादी संगठन ने निर्णय लिया है कि वे राज्य में पंचायत चुनावों का बहिष्कार करेंगे ।
jharkhand‘s maoist organization have decided that they in state village council elections will boycott
राजधानी रांची में उन्होने यह घोषणा की
in capital ranchi they this declared

‘Jharkhand‘s maoist organization have declared that they will boycott village council elections in the state.

They declared this in capital Ranchi.’

In above example, there is a non-identity relation between झारखंड (*‘jharkhand’*) and राजधानी रांची (*‘capital ranchi’*) because रांची (*‘ranchi’*) is capital of *‘Jharkhand’* in given context. However this is not a type of equivalence relation, rather a part/whole relation and need semantic as well as world knowledge for resolution.

In this work, we only focus on Identity relations as resolution of Non-identity relations would need extraordinary world knowledge.

5.2.2 Related Work:

Majority of the earlier research on the co-reference resolution focus on machine learning for co-reference resolution. This includes both supervised and unsupervised techniques. Most of the supervised techniques are based on either of the three models. First is Mention-pair models which is similar to the model of [59] that we used in our work on anaphora resolution. Mention pair model was first proposed by [5], it involves pairing of mentions and then learning a classifier to predict whether other mention pairs are co-referential. To improve, the performance of this model, various extensions have been suggested such as agreement based filtering as in [74].

Second is Entity Mention model [75] in which partial clusters are created by adding mentions to the clusters incrementally and a classifier is learned to decide whether a particular mention is co-referential with a previous clusters.

Third is mention ranking model of [25] which use rankers instead of classifiers to rank the co-reference pairs based on the relative probability. This is model is similar to the twin candidate model for anaphora resolution.

A detailed survey of supervised techniques for co-reference resolution is given in [55]. The most important and relevant work to state here is [30], which uses simple resolution approach using rich syntactic and semantic features for co-reference resolution. The importance of this work is that it shows that high resolution accuracy can be achieved for co-reference resolution even with simple linguistic constraints without using any supervised or unsupervised machine learning techniques.

To the best of our knowledge, there is no earlier work that deals with non-pronominal co-reference resolution for Hindi. Due to limited availability of co-reference data for Hindi, supervised techniques are difficult to use for our purpose. Motivated by the success of [30], we aimed to implement a rule based technique using syntactic and semantic constraints to resolve reference. However, major contribution of our approach is that instead of using phrase structure and typical grammatical constraints, we explore dependency structures and lexical similarity features for co-reference resolution.

5.2.3 Data and co-reference annotation:

For our experiments in co-reference resolution, we used a part of the Hindi treebank data which was used for anaphora resolution. The annotation of co-reference is carried out using the same scheme that was used for annotating anaphora references, however, with some differences. The scheme used for annotating anaphora involves marking of the address of the referent as a feature structure of the pronoun, thus marking the reference link between anaphora antecedent pairs. However, co-reference relations should not only be represented between pairs of mentions or expression, instead they should be represented as equivalence sets in the annotation. That is annotation of co-reference for any expression should represent the equivalence class or co-reference chain to which it belongs.

To adopt the anaphora annotation scheme for annotating co-reference chains, we include one change to the annotation scheme. Unlike anaphora annotation, for annotating co-reference, we allow non-sequential annotation, this means that a mention *'M'* can be linked to any of the earlier mentions which are co-referential with *'M'* and not necessarily the most recent one. By allowing the non-sequential annotation, the equivalence sets can be extracted by taking the transitive closure of each annotated co-reference pair. Consider following example:

(62) ((रेलमंत्री))$_{NP}$ ((लालू यादव की))$_{NP1}$ ((पत्नी))$_{NP2}$ ((और))$_{CCP}$ ((बिहार की))$_{NP3}$
railway minister lalu yadav's wife and bihar's
((पुर्व मुख्यमंत्री))$_{NP4}$ ((श्रीमती राबडीदेवी ने))$_{NP5}$ ((नीतीश कुमार पर))$_{NP6}$ ((आरोप
former chief minister mrs. rabridevi on nitish kumar accusation
लगाया))$_{VGF}$ ((की))$_{CCP2}$ ((वे))$_{NP7}$ ((राज्य के))$_{NP8}$ ((कर्मचारियों पर))$_{NP9}$ ((अत्याचार))$_{NP10}$
put that he on state's employees injustice
((कर रहे हैं ।))$_{VGF2}$ ((श्रीमती यादव ने))$_{NP11}$ ((कहा))$_{VGF3}$ ((की))$_{CCP3}$ ((लालू जी ने))$_{NP12}$
doing mrs yadav said that lalu
((गत वर्ष))$_{NP13}$ ((बजट मे))$_{NP14}$ ((बिहार राज्य के))$_{NP15}$ कर्मचारियों के लिये
last year in budget bihar state's for employees
((रेल किराये मे))$_{NP16}$ ((रियायत की))$_{NP17}$ ((घोषणा की थी ।))$_{VGF4}$
in rail fare of concession announcement did

'Railway minister Lalu yadav's wife and bihar's former chief minister Rabri Devi accused Nitish Kumar that he is doing injustice to state's employees. She also said that Lalu declared concession in rail fare Bihar state's employees in budget last year.'

In above example, the feature structure of the mentions will contain following reference attributes:

- ((लालू यादव की पत्नी))$_{NP1}$: <ref='NP'>
- ((बिहार की पुर्व मुख्यमंत्री))$_{NP4}$: <ref='NP2' refmod='NP1'>
- ((श्रीमती राबडीदेवी ने))$_{NP5}$: <ref='NP4' refmod='NP3' >
- ((श्रीमती यादव ने))$_{NP11}$: <ref='NP5' >
- ((लालू जी ने))$_{NP12}$: <ref='NP1' >

By taking a transitive closure of pairs represented by annotated co-reference links, following co-reference chains can be extracted:

- ((रेलमंत्री)), ((लालू यादव की)), ((लालू जी ने))
- ((लालू यादव की पत्नी)), ((बिहार की पुर्व मुख्यमंत्री)), ((श्रीमती राबडीदेवी ने)), ((श्रीमती यादव ने))

For the co-reference experiments, we annotated 179 news articles or text from the Hindi treebank data, it contains 2905 sentences.
In the annotation, 972 co-reference chains or equivalence sets were marked which involved 3024 markables.

5.2.4 Patterns in co-reference

Although Co-reference relations could span across sentences and discourse units and hence their interpretation would require syntactic, semantic and discourse knowledge, there are some lexical and syntactic patterns or configurations which would frequently contain co-reference pairs. Identifying these configurations would help to derive constraints or rules that can be used to resolve co-reference relations related to these patterns. We discuss some of these patterns below:

5.2.4.1 Appositives:

Appositives are the linguistic construction where two elements, usually both noun phrases, are used contiguously and one element specifies the other. i.e. one element gives the description or details about the other [68]. Consider following examples:

- भारत के राष्ट्रपति, प्रणब मुखर्जी (*'India's president, Pranab Mukherjee'*)
  In this case *'Pranab Mukherjee'* is an appositive of *'India's president'* which give the detail information about the latter, that is the name of president.
- मेरा भाई अभिषेक *'my brother abhishek'*
  In this case अभिषेक (*'Abhishek'*) is an appositive of the मेरा भाई (*'my brother'*) since it specifies the phrase *'my brother'* specifying *"which of my brother I am referering to"*.
- In English, there are also cases of Appositive genitives such as *'They City of Mumbai'* where both *'The city'* and *'Mumbai'* are co-referring to same entity, though they are in genitive relation. However, in Hindi, we could not find any case of genitive appositive.

One can observe that in most of the cases of *'apposition'*, the expressions involved are in co-reference. Thus identifying apposition can help to identify co-reference.

5.2.4.2 Predicate nominals:

Predicate nominals are the construction where the object, or in many cases other arguments of copula verb are co-referential with the subject [71]. In these cases, the value of the object is assigned to the subject. Consider following example:

(63) मार्टिन शूल्ज यूरोपीय संसद के अध्यक्ष हैं
martin shulz european parliament's president is

'Martin Shulz is the president of European parliament.'

In above example, यूरोपीय संसद के अध्यक्ष *'President of European Parliament'* is a predicate nominal which is the argument of the verb हैं (*'be'*) and is coreferential with the subject मार्टिन शूल्ज (*'Martin Schulz'*)

5.2.5 Resolution Process

As discussed earlier, since the co-reference relation are references to a single entity and hence are defined as equivalence classes, the co-reference resolution stands for identifying the equivalence classes and assignment of the equivalence class for each possible mention. Formally co-reference resolution stands for:

"Given a text, detect the possible mentions and identify the set of equivalence classes to which each of the possible mention belongs."
Reconsider the example (61):

(64) रेलमंत्री लालू यादव की पत्नी और बिहार की पुर्व मुख्यमंत्री
railway minister lalu yadav‘s wife and bihar‘s former chief minister
श्रीमती राबडीदेवी ने नीतीश कुमार पर आरोप लगाया की वे राज्य के कर्मचारियों पर
mrs. rabridevi on nitish kumar accusation put that he on state‘s employees
अत्याचार कर रहे हैं। श्रीमती यादव ने कहा की लालू जी ने गत वर्ष बजट मे बिहार राज्य के
injustice doing mrs yadav said that lalu last year in budget bihar state‘s
कर्मचारियों के लिये रेल किराये मे रियायत की घोषणा की थी।
for employees in rail fare of concession announcement did

‘Railway minister Lalu yadav‘s wife and bihar‘s former chief minister Rabri Devi accused Nitish Kumar that he is doing injustice to state‘s employees. She also said that Lalu declared concession in rail fare Bihar state‘s employees in budget last year.’

Given the above text as input, the output of the co-reference resolution algorithm would be following two equivalence sets:

- रेलमंत्री (*Railway minister*), लालू यादव (*Lalu yadav*), लालू (*Lalu*)
- लालू यादव की पत्नी (*Lalu yadav‘s wife*), बिहार की पुर्व मुख्यमंत्री (*‘Former chief minister of Bihar’*), श्रीमती यादव (*‘Mrs. Yadav’*)

The accuracy of the algorithm should be decided on the basis of how similar is the output equivalence set to the equivalence sets in the annotated data.
Thus, given a text, for identifying the reference as described above, we first have to identify which expression are the possible mentions that could be co-referential. Then, for each possible mention, we have to determine if it co-refers with any expression mentioned in the text. Then, extract the equivalence sets or co-reference chains by taking closure of all the expressions referring to same entity. We describe these three process in the following subsections. However, the main algorithm of reference resolution is described in subsection (5.2.5.2), in which we discuss different rules implemented for different patterns or configurations of co-reference. Thus our approach for co-reference is essentially a rule based approach in which we attempt to identify co-reference relations based on the syntactic constraints in the dependency structure and other semantic features.

5.2.5.1 Mention detection

As discussed earlier, since not all the expressions in a text could be co-referring with other expressions, in order to identify the co-reference among expressions, we first need to detect possible mentions which could be part of a co-reference equivalence set. We use a simple heuristic based on the POS-tag and chunk information available in the data. We define all those chunks as possible mentions whose head has a POS tag as *‘NNP’* (proper noun) or *‘NN’* (common noun). Thus given example (61) with POS tags and chunking as follows[2]:

(65) ((रेलमंत्री$_{NNP}$)) ((लालू$_{XC}$ यादव$_{NNP}$ की$_{PSP}$)) ((पत्नी$_{NN}$)) ((और$_{CC}$)) ((बिहार$_{NNP}$ की$_{PSP}$))
railway minister lalu yadav POSS wife and bihar POSS
((पुर्व$_{J}$J मुख्यमंत्री$_{N}$NP)) ((श्रीमती$_{XC}$ राबडीदेवी$_{NNP}$ ने$_{PSP}$)) ((नीतीश$_{XC}$ कुमार$_{NNP}$ पर$_{PSP}$))
former chief minister mrs. rabridevi ACC. nitish kumar on
((आरोप$_{NN}$ लगाया$_{VM}$)) ((की$_{PSP}$)) ((वे$_{PRP}$)) ((राज्य$_{NN}$ के$_{PSP}$)) ((कर्मचारियों$_{NN}$ पर$_{PSP}$))
accuasation put that he state GEN. employees on
((अत्याचार$_{NN}$ कर$_{VM}$ रहे हैं ।$_{VAUX}$)) ((श्रीमती$_{XC}$ यादव$_{NNP}$ ने$_{NNP}$)) ((कहा$_{VM}$)) ((की$_{PSP}$))
injustice do PRES.CONT mrs yadav ACC said that
((लालू$_{XC}$ जी$_{NNP}$ ने$_{PSP}$)) ((गत$_{JJ}$ वर्ष$_{NN}$)) ((बजट$_{NN}$ मे$_{PSP}$)) ((बिहार$_{NNP}$ राज्य$_{NNP}$
lalu RESP ACC last year budget in bihar state
के$_{PSP}$)) ((कर्मचारियों$_{NN}$ के$_{PSP}$ लिये$_{PSP}$)) ((रेल$_{NN}$ किराये$_{NN}$ मे$_{PSP}$)) ((रियायत$_{NN}$ की$_{PSP}$))
GEN employees for rail fare in concession GEN
((घोषणा$_{NN}$ की$_{VM}$ थी ।$_{VAUX}$))
announcement did PAST

‘Railway minister Lalu yadav‘s wife and bihar‘s former chief minister Rabri Devi accused Nitish Kumar that he is doing injustice to state‘s employees. Mrs. yadav also said that Lalu declared concession in rail fare for Bihar state‘s employees in budget last year.’

Looking at the above example, in our approach, we select all those chunks as possible mentions which have a head element with POS tag either NN or NNP. Though, deciding the head of the chunk could be a complex linguistic issue, as a simple heuristics for NP chunks, we can consider all head of the chunk to be that element which is the last word in the chunk and has a POS tag other than ‘PSP’ or ‘RP’. Thus for any chunk if its head, thus identified has a POS tag as ‘NN’ or ‘NNP’, it must be considered as possible mention. A simple algorithm as follows can be devised for mention detection:

- For each NP chnuk, calculate the head as follows:
  - Read all the elements of the chunk which do not have a POS tag as ‘PSP’ or ‘RP’
  - Select the last element from above list as the head of the chunk
- If the head element has a POS tag as ‘NN’ or ‘NNP’, select the complete chunk as a possoble mention.

Using the above algorithm, following mentions would be extracted from example (8):
रेलमंत्री (*‘Railway minister’*), लालू यादव (*‘Lalu yadav’*), पत्नी (*‘wife’*), बिहार (*‘Bihar’*), पुर्व मुख्यमंत्री

[2] In this example, the subscript shows the POS tag and the bracketing shoes the chunk boundaries

(*‘former chief minister’*), श्रीमती राबडीदेवी (*‘Mrs. Rabri devi’*), नीतीश कुमार (*‘Nitish Kumar’*), राज्य (*‘state’*), कर्मचारियों (*‘employees’*), श्रीमती यादव (*‘Mrs. yadav’*), लालू जी (*‘Lalu ji’*), गत वर्ष (*‘last year’*), बजट (*‘budget’*), कर्मचारियों (*‘employees’*), रेल किराये (*‘rail fare’*), रियायत (*‘concession’*)

5.2.5.2 Co-reference Identification:

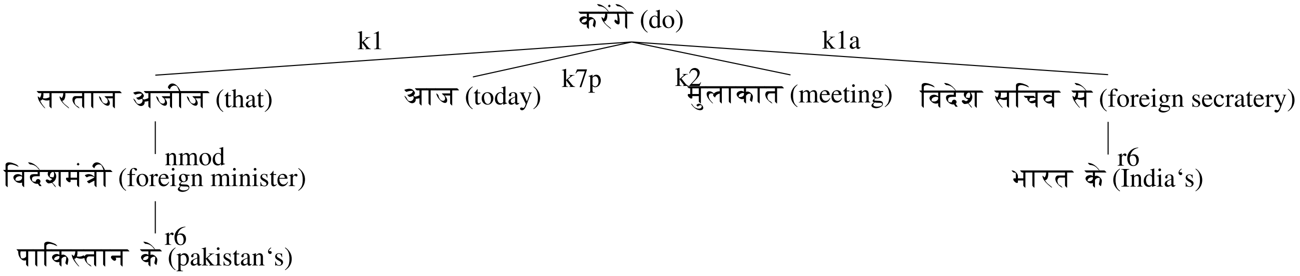
After identifying the possible mentions, the main task is to identify the co-references among the mentions. As already discussed, our approach for co-reference resolution is fully based on rules or constraints derived from the observation of different co-reference patterns. For each identified mention, we attempt to identify other expressions co-referential with this mention in previous discourse. We discuss below the rules that can be derived from syntactic structure and semantic features of the configurations discussed in section 5.2.4 and also some other simple lexical patterns.

- Appositives: As discussed in section 5.2.4, appositives are constructions where two Noun phrases occur consecutively and one modifies the other. In case of appositives, one Noun phrase is a proper noun, preferably a Named entity and the other one is a common noun phrase.
  Since one phrase is a modifier of the other, in the dependency structure, the modifying phrase is attached under the Proper noun. Moreover, a Proper Noun will have another noun as a modifier only in the case of Apposition. This make the identification of appositives and hence resolution of co-reference very easy. Using the dependency structures, appositives constructions can be identified by checking if a Proper noun has another noun phrase as its modifier and has only a modification relationship with the head. Consider following example:

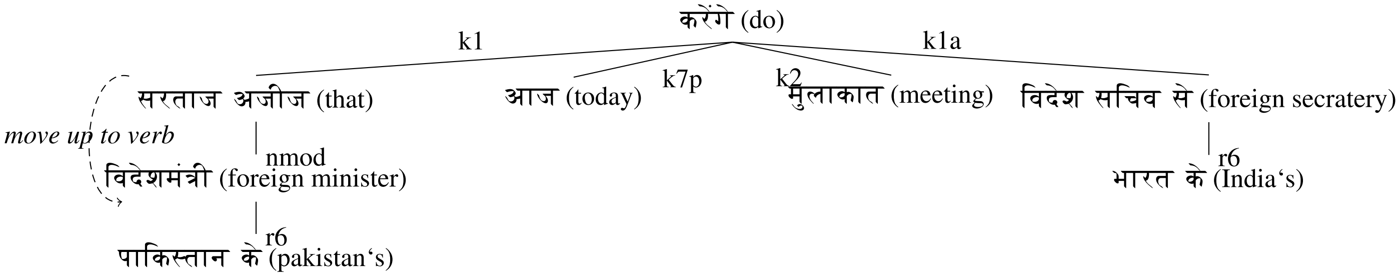
  (66) पाकिस्तान के विदेशमंत्री सरताज अजीज आज भारत के विदेश सचिव से
  pakistan‘s foreign minister sartaaj aziz today India‘s with foreign secretary
  मुलाकात करेंगे ।
  meet

  ‘Pakistan‘s foreign minister Sartaj aziz will have a meeting today with India‘s foreign secretary.’

  In above example, [पाकिस्तान के विदेशमंत्री (*‘Pakistan‘s foreign minister’*) — सरताज अजीज (*‘Sartaj aziz’*)] is an appositive construction where later one is the Proper noun-phrase and a named entity and the former one is a common-noun which specifies the other by giving additional information about it.
  The CPG based dependency structure of example (67) would be as follows:



As can be seen from above figure, Since the Phrase पाकिस्तान के विदेशमंत्री is modifier of the Proper noun phrase सरताज अजीज, hence it is attached under the later one with a dependency relation *‘nmod’*. Thus the apposition and hence the co-reference pair can be identified by selecting the subtree under the proper noun if any as shown in the figure below:



It is important to note here that the proper nouns can be detected by looking at the POS tag of the mention. If a mention has a head with POS tag as ‘NNP’, then it is a proper noun. Additionally, named entity can be identified by using the named entity recognizer.
Following rule can be derived for identification of appositives:

- If the head of the mention has a POS tag as ‘NNP’, then identify the mention as a proper noun.
- Further if the mention has another Noun phrase as modifier and the dependency label of the child node is *‘nmod’*, then select the mention pairs as appositives and identify the two expressions as co-referential.

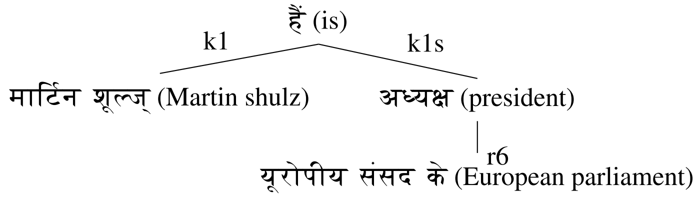
Thus in above example, पाकिस्तान के विदेशमंत्री (*‘Pakistan‘s foreign minister’*) and सरताज अजीज (*‘Sartaaj aziz’*)

- Predicate Nominals: As already discussed, predicate nominals are constructions where a Noun phrase assigns a value or attribute to another Noun phrase, usually the subject of the sentence. Thus, both expression are equivalent or co-referential to each other. Reconsider following example:

(67) मार्टिन शूल्ज् यूरोपीय संसद के अध्यक्ष हैं
martin shulz european parliament‘s president is

‘Martin Shulz is president of European parliament.’

The arguments of the copula verb हैं (*‘is’*) are मार्टिन शूल्ज् (*‘Martin Shulz’*) and यूरोपीय संसद के अध्यक्ष (*‘President of European Parliament’*). Here the value of later one is assigned to the former. Again, co-reference between the two expressions can be identified by detecting if their is predicative nominal construction involving two Noun phrases. In the dependency structure, both the participant of the predicative nominals are represented as the arguments of the verb. In CPG based dependency framework, to represent the equivalence relationship, a *‘samanadhikaran’* relation, is used in the dependency label. That is if the subject has a dependency label as *‘k1’* or *‘k2’* than the equivalent phrase will have a dependency label *‘k1s’* or *‘k2s’* correspondingly. Figure below shows the dependency structure of the example.


हैं (is)
k1
k1s
मार्टिन शूल्ज् (Martin shulz)
अध्यक्ष (president)
r6
यूरोपीय संसद के (European parliament)

Following rule can be derived for identifying the predicate nominals and hence the co-reference pairs:

- If a mention is a proper noun, check the other arguments of the head verb of the mention node.
- If the mention has a label *‘k1’*, and their is another noun phrase with the label *‘k1s’* which is an argument of the same verb, select both the Noun phrases as co-referential.
- Repeat same process as above for the mentions with label *‘k2’* and *‘k2s’*.

- Lexical Similarity: Lexical similarity of the expressions is one of the most important features for co-reference resolution. This is because in coherent text, substrings of larger expressions are used frequently to refer to a common entity in the text. For example: if एपीजे अब्दुल कलाम *‘APJ Abdul Kalam’* is an expression referring to a real word entity (a person), then frequently substrings like कलाम (*‘Kalam’*), अब्दुल कलाम (*Abdul kalam*), एपीजे अब्दुल कलाम (*‘APJ abdul kalam’*) can be used to refer to the same entity. Thus finding the substring patterns can help to identify the co-reference relations.
  However, simple substring match can not be helpful to identify all the patterns. Moreover, not every substring match will be a co-reference. We identify some configuration in which substrings or other lexical similarities can be used to identify the co-reference pairs.

- Abbreviations: Abbreviations are short forms or shortened expressions used to refer to entity with a long expression as full name. They are more frequently used for names of *'Organizations'*. Also, they are easy to identify. For any full expression, an abbreviation can be generated by taking the first letter from all its word. If a smaller expression in the text is same as the abbreviation of another larger expression, then the two expressions are co-referential. For example: राष्ट्रवादी कांग्रेस पार्टी (*'Nationalist congress party'*) and राकांपा (*'NCP'*) are co-referential because the later is an abbreviation of the former. There are also instances in Hindi, in which a popular name is used to refer to an organization which is the transliterated form of the English translation of the another expressions. For example, बीजेपी (*'BJP'*) is commonly used expression to an organization whose full name in Hindi is भारतीय जनता पार्टी (*'Bhartiya Janta Party'*). However, such cases would need word knowledge of translation and transliteration, hence such references could not be handled in our approach.
- Matching heads: In a small discourse only a few entities are invoked. Thus there is little chance that two Proper Noun expression with lexical similarities in a discourse will refer to different entities. That is, it is highly probable that if two expressions have some lexical similarities, than they will be referring to same entity and hence will be co-referential. Thus two expression which have at least one common string (which is Noun), can be co-referential as discussed above. However, a multi-word expression can contain word of different grammatical categories such as adjectives, quantifiers which may be common in expressions but do not represent reference to the same entity. We discuss three possible criteria to choose substrings which may result in better identification of coreference pairs. These are:

  - **Any substring match:** As already discussed, though selecting any substring match as co-referential will include all the correct co-referential expression, it will also include some wrong result. For example, in two expressions, बिहार सरकार (*'Bihar Government'*) and बिहार (*'Bihar'*), although there is a common word बिहार (*'Bihar'*), they are not coreferential.
  - **Head to Head match:** That is, two expressions are considered co-referential if and only if their heads are same. This means that [एपीजे अब्दुल कलाम (*'APJ abdul kalam'*), अब्दुल कलाम (*'Abdul kalam'*), कलाम (*'Kalam'*), अपीजे कलाम (*'APJ kalam'*)] will be considered co-referential, while none of the expressions [एपीजे (*APJ*) , अब्दुल (*'abdul'*) एपीजे अब्दुल] will be considered co-referential with the expressions in the former list. Though, this criteria will perform better for expressions referring to *'Persons'*, but for expressions referring to *'Organizations'* and other enities it may generate wrong results. For example, consider two expressions बिहार सरकार (*'Bihar government'*) and केंद्र सरकार (*'Union government'*). Both of these expressions have same head i.e. सरकार (*'government'*), but they refer to two different entities and hence are not coreferential.

- **Head to other words match:** Other than the above two criteria, there is also another possibility i.e. to consider those expressions to be coreferential in which the head of the following or later expression matches with any word in the former expression. It is intuitive to consider such a criteria since head of the expression represents the core semantics of an expression. Thus, if the head of an expression matches with a part of some other expression, it is highly probable that they will be coreferential.

Some of the issues discussed with above criteria could be resolved by using agreement and salience features as used in anaphora resolution. That is, some of the wrong expressions generated based on above criteria can be discarded based on agreement or salience features.Consider following example:

(68) राजद अधयक्ष लालू यादव ने पार्टी के नेता व अपनी पत्नी
RJD president lalu yadav to party leader and his wife‘s
श्रीमती राबडी देवी यादव के भाई सुभाष यादव को लोकसभा चुनाव के लिये
mrs. rabri devi brother sadhu yadav for loksabha elections
टिकिट दिया है ।हालांकि श्री यादव पहले इसके पक्ष मे नहीं थे ।
ticket gave. although mr. yadav earlier this in favour was not
‘RJD president has given ticket to party leader and his wife Rabri devi‘s brother Sadhu yadav. Although, Mr. yadav was not in favor of this earlier.’

In above example, using any of the three criterias discussed, the expression श्री यादव (*‘Subhash yadav’*) will be coreferential will three expression: (a) राजद अधयक्ष लालू यादव (*‘RJD president Lalu yadav’*), (b) श्रीमती राबडी देवी यादव (*Mrs. Rabdi Devi yadav*), (c) सुभाष यादव (*‘Subhash yadav’*)

However, in actual, it is co-referential expression with only one, in such cases the other two can be pruned out using the agreement and salience features as follows:

First consider expression: (b) श्रीमती राबडी देवी यादव (*Mrs. Rabdi Devi yadav*), since the gender of this expression is feminine while that of the expression श्री यादव (*‘Mr yadav’*) is masculine, they can not be coreferential.

Thus there are two possible expressions left: (a) राजद अधयक्ष लालू यादव (*‘RJD president Lalu yadav’*) and (b) सुभाष यादव (*‘Subhash yadav’*)

As can be recalled from chapter-4, for resolution of third person pronouns, we used a ordering based on CPG relation salience, to select most probable referent. The salience order was : (*k1 >k2 >k3 >k4 >others*). If there are multiple expressions in a sentence which can be coreferential with an expression, the above salience based ranking can be used to select the most probable coreferential expression. In above example, the dependency relation of राजद अधयक्ष लालू यादव (*‘RJD president Lalu yadav’*) is *‘k1’* while that of सुभाष यादव (*‘Subhash yadav’*) will be *‘k4’*, thus based on salience based ranking former will be selected while the later can be pruned out.

Using the above observations and criterias, we develop following constraint for Lexical similarity based coreference:

- ∗ To determine coreference for a mention *'M_k'*, based on lexical similarity, check if the head *'H_k'* of the expression is a proper noun.
- ∗ If *'H_k'* is a proper noun
  - · For all the preceding mentions M_i, check if *'H_k'* matches with any of the words in M_i
  - · If there is a match, select the expression M_i to be coreferential with M_k.
- ∗ Else if the head 'H$_k$' is a common noun
  - · For all the preceding mentions *'M_i'*'s, check if there is complete string match. That is, if all the words in *'M_k'* match with all the words in M_i
  - · If there is a match, select the expressions M_i to be co-referential with M_k.
- ∗ Remove any co-referential pairs which do not have number and gender agreement.

5.2.5.3 Equivalence class or chain extraction

Since the co-reference relations make an equivalence set, the output of co-reference resolution must be set of equivalence classes or co-reference chain. However, co-reference identification as discussed in previous subsection only identifies co-reference between two expression. Thus equivalence sets need to be extracted from these pairs. Given, the set of co-reference pairs in text or document, equivalence sets can be extracted just by taking the transitive closure of all the pairs. Consider following example:

(69) ((रेलमंत्री$_{NNP}$)) ((लालू$_{XC}$ यादव$_{NNP}$ की$_{PSP}$)) ((पत्नी$_{NN}$)) ((और$_{CC}$)) ((बिहार$_{NNP}$ की$_{PSP}$))
railway minister lalu yadav POSS wife and bihar POSS
((पुर्व$_J$J मुख्यमंत्री$_N$NP)) ((श्रीमती$_{XC}$ राबडीदेवी$_{NNP}$ ने$_{PSP}$)) ((नीतीश$_{XC}$ कुमार$_{NNP}$ पर$_{PSP}$))
former chief minister mrs. rabridevi ACC. nitish kumar on
((आरोप$_{NN}$ लगाया$_{VM}$)) ((की$_{PSP}$)) ((वे$_{PRP}$)) ((राज्य$_{NN}$ के$_{PSP}$)) ((कर्मचारियों$_{NN}$ पर$_{PSP}$))
accuasation put that he state GEN. employees on
((अत्याचार$_{NN}$ कर$_{VM}$ रहे हैं ।$_{VAUX}$)) ((श्रीमती$_{XC}$ यादव$_{NNP}$ ने$_{NNP}$)) ((कहा$_{VM}$)) ((की$_{PSP}$))
injustice do PRES.CONT mrs yadav ACC said that
((लालू$_{XC}$ जी$_{NNP}$ ने$_{PSP}$)) ((गत$_{JJ}$ वर्ष$_{NN}$)) ((बजट$_{NN}$ मे$_{PSP}$)) ((बिहार$_{NNP}$ राज्य$_{NNP}$
lalu RESP ACC last year budget in bihar state
के$_{PSP}$)) ((कर्मचारियों$_{NN}$ के$_{PSP}$ लिये$_{PSP}$)) ((रेल$_{NN}$ किराये$_{NN}$ मे$_{PSP}$)) ((रियायत$_{NN}$ की$_{PSP}$))
GEN employees for rail fare in concession GEN
((घोषणा$_{NN}$ की$_{VM}$ थी ।$_{VAUX}$)) ((लेकिन$_{CC}$)) ((नीतीश$_{NNP}$ की$_{PSP}$)) ((लापरवाही$_{NN}$ की$_{PSP}$
announcement did PAST but nitish GEN negligence GEN
वजह$_{PSP}$ से$_{PSP}$)) ((कर्मचारियों$_{NN}$ को$_{PSP}$)) ((अभी$_{NN}$ तक$_{PSP}$)) ((इसका$_{DEM}$ लाभ$_{NN}$))
because of employees to yet it's benefit
((नहीं$_{NEG}$ मिल$_{VM}$ पाया$_{VAUX}$ है ।$_{VAUX}$))
not reached has

> ‘Railway minister Lalu yadav‘s wife and bihar‘s former chief minister Rabri Devi accused Nitish Kumar that he is doing injustice to state‘s employees. Mrs. yadav also said that Lalu declared concession in rail fare Bihar state‘s employees in budget last year. But due to negligence of Nitish Kumar, its benefits has not reached to employees yet.’

Given the above example, using the process explained in section 5.2, following coreference pairs will be extracted:

- रेलमंत्री (*‘Railway minister’*), लालू यादव (*‘Lalu yadav’*)
- रेलमंत्री लालू यादव कि पत्नी (*‘Railway minister Lalu yadav‘s wife’*), श्रीमती राबडीदेवी (*‘Mrs. Rabdi devi’*)
- बिहार (*‘Railway minister Lalu yadav‘s wife’*), श्रीमती राबडीदेवी (*‘Mrs. Rabdi devi’*)
- श्रीमती राबडीदेवी (*‘Mrs. Rabdi devi’*), श्रीमती यादव (*‘Mrs. yadav’*)
- लालू यादव (*‘Lalu yadav’*), लालू जी (*‘Lalu jii’*)
- नीतीश कुमार (*‘Nitish Kumara’*), नीतीश (*‘Nitish’*)

Taking a transitive closure of above pairs, following equivalance set would be extracted:

- रेलमंत्री (*‘Railway minister’*), लालू यादव (*‘Lalu yadav’*), लालू जी (*‘Lalu jii’*)
- रेलमंत्री लालू यादव की पत्नी (*‘Railway minister Lalu yadav‘s wife’*), श्रीमती राबडीदेवी (*‘Mrs. Rabdi devi’*), श्रीमती यादव (*‘Mrs. yadav’*)
- नीतीश कुमार (*‘Nitish Kumara’*), नीतीश (*‘Nitish’*)

5.2.6 Evaluation:

5.2.6.1 Evaluation measure:

Since the coreference relation are viewed as equivalence classes, a simple pairwise F-score measure similar to that we used for evaluating anaphora resolution will not be correct for evaluation of coreference resolution. Anaphora is a non-transitive relation where the anaphora or pronoun refers to another entity, hence to evaluate anaphora resolution, it is sufficient to determine if the correct referent is predicated for an anaphor. However, for coreference, since all the co-referring expression belonging to an equivalence class refer to a single entity, evaluating the correct prediction of links is not sufficient. For coreference, it is more correct to evaluate whether the correct set of coreference chains or equivalence class is predicted. Moreover, evaluation of coreference resolution should also take into account the prediction of partially correct coreference chains. That is, even if some expression are missed in the output chains as compared to those in annotated data, partial scores should be given to the partially correct output.

Therefore, to evaluate coreference resolution, we use model theoretic MUC F-score which is defined in [64]. This measure is based on an algorithm which calculates the similarity between equivalence sets in annotated data known as *‘key’*, and those in output of the coreference resolution process which are known as *‘responses’*. Specifically, recall error is calculated by determining the minimum number of links that must be added to the response so that all entities in the equivalence set of *'key'* will be in the same equivalence class as in the *'response'*. Similarly. precision error is calculated by determining the minimum number of links that must be added to the *'key'* so that all the entities in the equivalence set of *'response'* will be in the same equivalence class as in the *'response'*. We explain this algorithm by following example given in [64]:

Assume that following coreference pairs are obtained in the annotation (*'key'*) and the output of coreference resolution (*'response'*) where the numerals 1,2,3..... represents the expression identified and the symbol "-" represents that the two expressions are coreferential:

- *'key' :* <2-3 3-4 4-5 5-7 7-8 8-10>

- *'response' :* <1-2 2-3 4-5 5-6 7-8 8-9>

By extracting the equivalence classes, we get one single class in *'key'* which is [2 3 4 5 7 8 10], and three classes in response which are [1 2 3], [4 5 6], [7 8 9]. Figure 5.10 shows the equivalence classes in *'key'* and *'responses'*. Circles show the equilavance set. *'Key'* set are represented by thick borders and *'Response'* set are represented by thin borders.

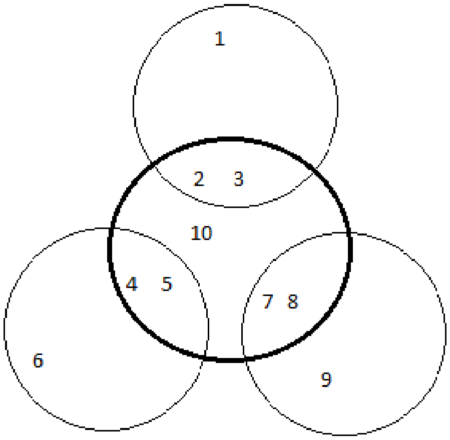


Figure 5.10 Equivalence Classes in Key and Response, Thick border = Key, Thin border = response

The MUC algorithm of [59] calculates the recall score as follows: Given an equivalence class *'S'* generated by key and let R$_1$, R$_2$, R$_3$.....R$_m$ be the equivalence generated in response, then following functions are calculated for *'S'*:
Partitions p(S) are calculated over *'S'* relative to the responses. Partitions are calculated by intersection of *'S'* with those responses which overlap *'S'*. For example, by intersecting equivalence class [2 3 4 5 7 8 10] in *'key'* with three response sets [1 2 3], [4 5 6], [7 8 9], we get four partitions as p(S) = <[2 3], [4, 5], [7 8], [10]>. Figure 5.11 explains this partitioning.

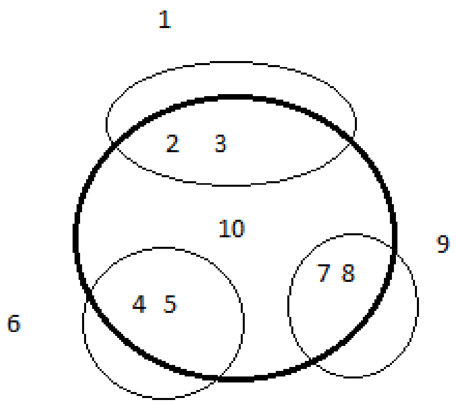


Figure 5.11 Partitions of *‘Key’* with respect to *‘Response’*, Thick border = Key, Thin border = response

Having partitioned the equivalence class in this way, Recall is calculated as

$$Recall = \frac{\sum(|S_i| - |p(S_i)|)}{\sum(|S_i| - 1)} \tag{5.1}$$

where |S$_i$| is the size of equivalence class S$_i$ and |p(S$_i$)| is the number of partitions of S$_i$. There is only one equivalence class in *‘key’* in our example S_1 with size |S$_1$| = 7, and number of partitions of S$_1$ with respect to responses is |p(S)| = 4. Thus

$$Recall = \frac{7-4}{7-1} = 0.5 \tag{5.2}$$

Similarly, precision is calculated by partitioning the equivalence classes in *‘response’* with respect to equivalence classes in *‘key’*. In our example by intersecting equivalence class S‘$_1$ = [1 2 3] in response with respect to the single equivalence class [2 3 4 5 7 8 10] in *‘key’*, we get partitions p(S’$_1$) = <[1] [2 3]>. Similarly, by partitioning response set S‘$_1$ = [4 5 6] with respect to *‘key’* we get p(S‘$_2$) = <[4 5] [6]>and by partitioning S‘$_3$ = [7 8 9] with respect to *‘key’*, we get partitions p(S‘$_3$) = <[7 8] [9] >. Figure 5.12 shows this partitioning

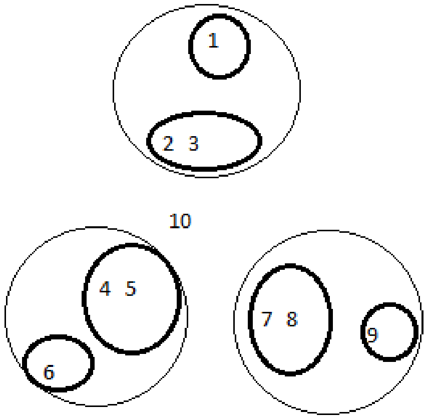


Figure 5.12 Partitions of *'Response'* with respect to *'Key'*, Thick border = Key, Thin border = response

Precision is calculated as :

$$Precision = \frac{\sum(|S_i'| - |p(S_i')|)}{\sum(|S_i'| - 1)} \tag{5.3}$$

In our example, $|S_1| = 3$, $|p(S_1)| = 2$, $|S_2| = 3$, $|p(S_2)| = 2$, $|S_3| = 3$, $|p(S_3)| = 2$. Thus,

$$Recall = \frac{(3-2)+(3-2)+(3-2)}{(3-1)+(3-1)+(3-1)} = 0.5 \tag{5.4}$$

F score is calculated as :

$$FScore = \frac{2 * precision * recall}{precision + recall} \tag{5.5}$$

In our example,

$$FScore = \frac{2 * 0.5 * 0.5}{0.5 + 0.5} = 0.5 \tag{5.6}$$

Sine, in our data, discourses are individual texts or documents, coreference resolution is performed over each document and hence the coreference score is calculated for each document. Finally, the total F-Score is calculated as an average of F-Scores of each document.

5.2.6.2 Results:

In the table below, we report the results obtained by using each individual pattern as discussed in 5.2 and also the overall result for the resolution algorithm in terms of precision, recall and F-Score.

	Appositives	Predicative Nominals	Lexical Similarity	Overall (with all patterns)
Precision	.70	.62	.54	.64
Recall	.2	.22	.49	.50
F-score	.31	.32	.50	.56

Table 5.2: Results for Coreference resolution

5.2.7 Discussion:

Table 5.2 shows the result for our coreference resolution approach for individual patterns and also for the overall algorithm. It is important to note that first three columns of the table show the precision, recall and F-score achieved when only a specific pattern is used to resolve the references, not individual coverage of that pattern in the overall system. The last column shows the result for overall process which includes all the above patterns.

As can be seen from the table, though the precision for each individual pattern is quite good, the recall is very low. This is because for each individual pattern the system outputs only those coreference pairs which are identified by that pattern. That is the system will output the coreference pair, only if resolved based on the that particular pattern. Certainly, this will result in higher precision, since the system outputs very few coreference pairs. On the other hand, using a specific pattern, not all the coreference pairs in a document can be identified. That means many of the co-referring pairs will be left out. This will result in low recall. However, as shown in the last row of the table, when including all the patterns in the resolution process, our approach has achieved substantial performance.

Also, it can be observed from the table that the resolution performance achieved by using Lexical matching is relatively quite high than the other patterns. This is because, in a coherent text or conversation, other than use of pronouns, lexically similar words such as short-names instead of full-names, titles, abbreviations are more frequently used to refer to the common entities instead of appositives. Also, the main function of appositives and nominal predicative is to assign a property to an expression or give additional information about it, hence these patterns occur very rarely.

The overall accuracy of the approach is though average, but is acceptable given the limited set of features and patterns we used. Certainly, as discussed earlier, world knowledge and deep semantic knowledge are very important for coreference resolution and having these information and features can have highly positive impact on the resolution performance. However, our results show that even in case of non-availability of these information, considerable performance can be achieved using lexical, syntactic and semantic patterns.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we discussed our work on Anaphora annotation, anaphora resolution and experiments with Co-reference resolution. Our main aim was to develop an approach for Entity anaphora resolution in Hindi.

Towards experimenting with anaphora resolution, the first requirement is the availability of annotated data. Since, no large corpus annotated with anaphora information is available for Hindi, we first aimed at developing a corpus annotated with anaphora relations. In order to develop a corpus which is consistent and usable for anaphora resolution, we first designed an annotation scheme aimed to resolve the annotation issues and later based on this scheme, we annotated a corpus which can be used in resolution algorithms. The identified issues that we handled relate to representation format, referent span identification, annotation of coordinating referents etc. Our decisions like sequential annotation could help in reducing the computational complexity in resolution systems. The comparative inter-annotator analysis of the proposed scheme verifies that the separation of the referent span, and other features help to achieve a consistent annotation by increasing the inter-annotator agreement. The scheme can be extended for co-reference and the annotated data is convertible to other annotation for- mats like MUC etc.

Our main focus in this work was to develop an approach for Entity anaphora resolution. We started with an aim to explore the possibility of using dependency and semantic information for anaphora resolution in Hindi. Towards this goal, we proposed a hybrid approach in which the rule based part uses dependency structures and relations for referent selection. The rule based system achieved a substantial F-measure of 0.6 which implies that dependency relations can provide a suitable source of syntactic information for anaphora resolution for languages like Hindi. Also, the supervised learning approach helped to achieve a significant improvement over the rule based systems which shows that the semantic feature like animacy and Named entities can be used as additional information to improve the performance.

We also explored use of shallow features for anaphora resolution, that can be extracted from limited linguistic information given in the input text. We extracted features like *'vibhakti'* (post-position), morph features, pronoun categories from the words themselves and derived rules for resolution of reflexive, relative and locative pronouns from the limited features given in the data. We implemented a hybrid approach with these rules and features. We achieved substantial results even with the shallow features.

Finally, we also conducted some preliminary experiments towards Event anaphora and co-reference resolution in Hindi. For event anaphora , we proposed a rule based approach in which structural event references are resolved using dependency structures. Though, the results obtained are low, they give insights for exploring features which may be more helpful for event anaphora resolution. For co-reference resolution, we identified some linguistic configurations such as Appositives and Predicate Nominals which frequently involve co-reference pairs. We attempted to identify expressions in these configurations using dependency structures and relation. We also explored constraints on co-reference between pairs with lexical similarity and devised an approach to identify these pairs. We achieved considerable performance given the limited availability of linguistic information. Similar to Entity anaphora resolution, our results show that dependency relations and other syntactic and semantic features provides important information for co-reference resolution.

6.2 Future work:

In this work, we have only focused on exploring features for anaphora resolution in Hindi. Our focus in those approaches was mainly to explore various features that could be used for anaphora resolution in Hindi, especially the dependency structure. Hence, in our discussion of approaches using dependency structures in hybrid system, we only focused on the algorithm and features that could be used to resolve the references given the already identified anaphors and their reference type. Therefore, the assumption in those approaches was that the input to the system will already have anaphors identified and their categorization in Entity and Events. However, in real time anaphora resolution, the data contains only expressions with no information about their anaphoricity or possible reference type. Hence, a full fledged resolution system should also include algorithms and processes for identifying what are the expressions which are possibly anaphors, mentions and their reference types. Hence, the main aim of our future work is to develop an End-to-End anaphora resolution system which includes preprocessing before the anaphor is passed for resolution to one of the two process for Entity and Event Anaphora resolution.

Moreover, we only conducted preliminary experiments for Event anaphora and co-reference resolution. However, the results show that there is further scope for performance improvement of these approaches with additional features and world-knowledge which we aim to explore in future.

Finally, we also aim to conduct experiments with dependency structures for other Indian languages like Telugu, Bengali etc, which similar to Hindi, have substantial availability of dependency data.

Related Publications

- **Praveen Dakwale**, Vandan Mujadia, Dipti M. Sharma.“A Hybrid Approach for Anaphora Resolution in Hindi.” Proceedings of International Joint Conference on Natural Language Processing (IJCNLP) 2013, Nagoya, Japan.
- **Praveen Dakwale**, Himanshu Sharma, Dipti M Sharma. “Anaphora Annotation in Hindi Dependency Treebank.” Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation(PACLIC), 2012.
- **Praveen Dakwale**, Himanshu Sharma. “Anaphora resolution in Indian Languages Using Hybrid Approaches.” NLP Tools Contest on Anaphora Resolution in Indian Languages, International Conference on Natural Language Processing (ICON), 2011.

Appendix A

Personal pronouns

Type	Root	Nominative	Accusative	Instrumental/Ablative	Dative	Genitive	Locative
1st person (singular)	मैं(I)	मैंने	मुझे	मुझसे	मुझको	मेरा, मेरी, मेरे	मुझमें
1st person (plural)	हम(We)	हमने	हमे	हमसे	हमको	हमारा,हमारी,हमारे	हममें
2nd person (intimate)	तु(You)	तुने	तुझे	तुझसे	तुझको	तेरा, तेरी, तेरे	तुझमें
2nd person (honorary)	आप(You)	आपने	आपको	आपसे	आपको	आपका, आपकी, आपके	आपमें
Third person (distal)(singular)	वह(He)	उसने	उसको	उससे	उसको	उसका, उसकी, उसके	उसमें
Third person (distal)(plural/honorific)	वे(He)	उन्होंने	उन्हें	उनसे	उनको	उनका, उनकी, उनके	उनमें
Third person (proximal)(singular)	यह(it/this)	इसने	इसे	इससे	इसको	इसका, इसकी, इसके	इसमें
Third person (proximal)(plural/honorific)	ये(these)	इन्होंने	इन्हें	इनसे	इनको	इनका, इनकी, इनके	इनमें

Relative Pronoun:

Type	Root	Nominative	Accusative	Instrumental/Ablative	Dative	Genitive	Locative
Singular	जो(which)	जिसने	जिसे	जिससे	जिसको	जिसका, जिसकी, जिसके	जिसमें
Plural	जो(which)	जिन्होंने	जिन्हे	जिनसे	जिनको	जिनका, जिनकी, जिनके	जिनमें

Reflexive Pronouns:

Form	Root	Direct	Oblique
Masculine(Possesive)	अपना	अपना	अपने
Feminine (Possesive)	अपनी	अपनी	अपनी
Plural Possesive	अपने	अपने	अपनों
Non-Possesive	अपने आप	अपने आप	अपने आप
Non-Possesive	स्वयं	स्वयं	स्वयं
Non-Possesive	खुद	खुद	खुद

Indefinite pronouns:

Form	Root	Direct	Oblique
Someone	कोई	कोई	-
Something	कुछ	कुछ	-
Everyone	सभी	सभी	सभी

Place pronouns: यहां (*‘here’*), वहां (*‘there’*), जहां (*‘where’*)

Bibliography

[1] S. Abney and S. P. Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

[2] S. Abney and S. P. Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

[3] I. Aduriz, K. Ceberio, E. H. Unibertsitatea, and D. de Ilarraza. Pronominal anaphora in basque: annotation of a real corpus. *Procesamiento del lenguaje natural*, pages 99–104, 2004.

[4] S. Agarwal, M. Srivastava, P. Agarwal, and R. Sanyal. Anaphora resolution in hindi documents. In *Natural Language Processing and Knowledge Engineering, 2007. NLP-KE 2007. International Conference on*, pages 452–458. IEEE, 2007.

[5] C. Aone and S. W. Bennett. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 122–129, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[6] B. Baldwin. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45. Association for Computational Linguistics, 1997.

[7] C. Barbu and R. Mitkov. Evaluation tool for rule-based anaphora resolution methods. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 34–41. Association for Computational Linguistics, 2001.

[8] R. Begum, S. Husain, A. Dhwaj, D. M. Sharma, L. Bai, and R. Sangal. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*, 2008.

[9] A. Bharati, V. Chaitanya, R. Sangal, and K. Ramakrishnamacharyulu. *Natural language processing: a Paninian perspective*. Prentice-Hall of India New Delhi, 1995.

[10] A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal. Two semantic features make all the difference in parsing accuracy. *Proc. of ICON*, 8, 2008.

[11] A. Bharati, R. Sangal, and D. M. Sharma. *SSF: Shakti Standard Format Guide*. LTRC, IIIT-Hyderabad, India, 2007.

[12] A. Bharati, D. M. Sharma, L. Bai, and R. Sangal. Anncorra : Annotating corpora guidelines for pos and chunk annotation for indian languages. Technical report, LTRC, IIIT-Hyderabad, 2006.

[13] A. Bharati, D. M. Sharma, L. Bai, and R. Sangal. Anncorra : Annotating corpora guidelines for pos and chunk annotation for indian languages. Technical report, LTRC, IIIT-Hyderabad, 2006.

[14] R. Bhatt, O. Rambow, B. Narasimhan, D. M. Sharma, M. Palmer, and F. Xia. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP*, 2009.

[15] A. Bjrkelund and J. Kuhn. Phrase structures and dependencies for end-to-end coreference resolution. In *Proceedings of COLING 2012: Posters*, pages 145–154. The COLING 2012 Organizing Committee, 2012.

[16] S. Buchholz and E. Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

[17] J. G. Carbonell and R. D. Brown. Anaphora resolution: a multi-strategy approach. In *Proceedings of the 12th conference on Computational linguistics-Volume 1*, pages 96–101. Association for Computational Linguistics, 1988.

[18] N. Chomsky. *Lectures on government and binding: The Pisa lectures*, volume 9. Walter de Gruyter, 1993.

[19] N. Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

[20] D. Connolly, J. D. Burger, and D. S. Day. A machine learning approach to anaphoric reference. In *New Methods in Language Processing*, pages 133–144, 1997.

[21] P. Dakwale, V. Mujadia, and D. M. Sharma. A hybrid approach for anaphora resolution in hindi. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 977–981, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[22] P. Dakwale and H. Sharma. Anaphora resolution in indian languages using hybrid approaches. *In Proceedings of the ICON-2011*, 2011.

[23] P. Dakwale, H. Sharma, and D. M. Sharma. Anaphora annotation in hindi dependency treebank. 2012.

[24] A. Davison. Lexical anaphors and pronouns in hindi. In *Lexical Anaphors and Pronouns in Selected South Asian Languages: A Principled Typology*, 2003.

[25] P. Denis and J. Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 660–669, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[26] S. Dipper and H. Zinsmeister. Towards a standard for annotating abstract anaphora. In *LREC 2010 Workshop on Language Resources and Language Technology Standards, Valletta, Malta*, pages 54–59, 2010.

[27] K. Dutta, N. Prakash, and S. Kaushik. Resolving pronominal anaphora in hindi using hobbs algorithm. *Web Journal of Formal Computation and Cognitive Linguistics*, 1(10), 2008.

[28] P. Elango. Coreference resolution: A survey. *University of Wisconsin, Madison*, 2005.

[29] J. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[30] A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1152–1161. Association for Computational Linguistics, 2009.

[31] L. Hirschman and N. Chinchor. Muc7 coreference task definition. In *Message Understanding Conference*, 1997.

[32] L. HIRSCHMAN and R. GAIZAUSKAS. Natural language question answering: the view from here. *Natural Language Engineering*, 7:275–300, 12 2001.

[33] J. Hobbs. Resolving pronoun references. In *Readings in natural language processing*, pages 339–352. Morgan Kaufmann Publishers Inc., 1986.

[34] S. Husain. *A Generalized Parsing Framework Based On Computational Paninian Grammar*. PhD thesis, PhD Thesis. IIIT-Hyderbad, India, 2011.

[35] I. Jena, R. A. Bhat, S. Jain, and D. M. Sharma. Animacy annotation in the hindi treebank. *LAW VII & ID*, page 159, 2013.

[36] C. Kennedy and B. Boguraev. Anaphora for everyone: pronominal anaphora resoluation without a parser. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 113–118. Association for Computational Linguistics, 1996.

[37] J. C. King. Anaphora. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2013 edition, 2013.

[38] K. Krippendorff. *Content analysis: An introduction to its methodology*. Sage Publications, Inc, 2004.

[39] L. Kucova and E. Hajicova. Coreferential relations in the prague dependency treebank. In *Proceedings of the 5th International Conference on Discourse Anaphora and Anaphor Resolution*, 2005.

[40] S. L and S. Bandhopaday. Nlp tool contest at icon 2011 on anaphora resolution in indian languages. *In Proceedings of the ICON-2011 Proceedings of ICON 2011 NLP Tool Contest Anaphora Resolution in Indian Languages*, 2011.

[41] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[42] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[43] Y.-H. Lin, T. Liang, and T. Hsinehu. Pronominal and sortal anaphora resolution for biomedical literature. In *ROCLING*, 2004.

[44] A. McEnery, P. Baker, R. Gaizauskas, and H. Cunningham. Emille: Building a corpus of south asian languages. *VIVEK-BOMBAY-*, 13(3):22–28, 2000.

[45] R. Mitkov. *Anaphora resolution: the state of the art*. Citeseer, 1999.

[46] R. Mitkov. Introduction: Special issue on anaphora resolution in machine translation and multilingual nlp. *Machine translation*, 14(3):159–161, 1999.

[47] B. Navarro, R. Izquierdo, and M. Saiz-Noeda. Exploiting semantic information for manual anaphoric annotation in cast3lb corpus. In *ACL 2004 Workshop on Discourse Annotation*, 2004.

[48] J. Nivre. Parsing indian languages with maltparser. *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pages 12–18, 2009.

[49] N. Nobre. Anaphora resolution. *Unpublished masters thesis, Instituto Superior Técnico-Universidade Técnica de Lisboa*, 2011.

[50] C. Orasan and R. Evans. Np animacy identification for anaphora resolution. *J. Artif. Intell. Res.(JAIR)*, 29:79–103, 2007.

[51] R. Passonneau. Computing reliability for coreference annotation. In *Proceedings of LREC*, volume 4, pages 1503–1506, 2004.

[52] M. Poesio and R. Artstein. Annotating (anaphoric) ambiguity. In *In Proc. of the Corpus Linguistics Conference*, 2005.

[53] M. Poesio and R. Artstein. Anaphoric annotation in the arrau corpus. In *LREC*, 2008.

[54] R. Prasad and M. Strube. Discourse salience and pronoun resolution in hindi. *U. Penn Working Papers in Linguistics*, 6:189–208, 2000.

[55] A. Rahman and V. Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 968–977. Association for Computational Linguistics, 2009.

[56] M. Recasens, M. A. Marti, and M. Taule. Where anaphora and coreference meet. annotation in the spanish cess-ece corpus. *Proceedings of RANLP*, 2007.

[57] S. Sinha. A corpus-based account of anaphor resolution in hindi. Masters thesis, University of Lancaster, UK, 2002.

[58] L. Sobha and B. Patnaik. Vasisth: An anaphora resolution system for malayalam and hindi. In *Symposium on Translation Support Systems*, 2002.

[59] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.

[60] J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Jeek. Two uses of anaphora resolution in summarization. *Inf. Process. Manage.*, 43(6):1663–1680, Nov. 2007.

[61] M. Strube. Never look back: An alternative to centering. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1251–1257. Association for Computational Linguistics, 1998.

[62] B. Uppalapu and D. M. Sharma. Pronoun resolution for hindi. In *7th Discourse Anaphora and Anaphor Resolution Colloquium*, 2009.

[63] J. L. Vicedo and A. Ferrández. Importance of pronominal anaphora resolution in question answering systems. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 555–562. Association for Computational Linguistics, 2000.

[64] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics, 1995.

[65] K. von Heusinger. Anaphora, antecedents, and accessibility. *THEORETICAL LINGUISTICS-BERLIN AND NEW YORK-*, 26(1/2):75–94, 2000.

[66] M. A. Walker, A. A. K. Joshi, and E. E. F. Prince. *Centering theory in discourse*. Oxford University Press, 1998.

[67] Wikipedia. Anaphora (linguistics)— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[68] Wikipedia. Apposition— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[69] Wikipedia. Automatic summarization— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[70] Wikipedia. Part-of-speech tagging— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[71] Wikipedia. Predicative expression— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[72] Wikipedia. Reflexive pronoun— Wikipedia, the free encyclopedia, 2014. [Online; accessed 17-January-2014].

[73] E. Woolford. More on the anaphor agreement effect. *Linguistic Inquiry*, 30(2):257–287, 1999.

[74] X. Yang and J. Su. Coreference resolution using semantic relatedness information from automatically discovered patterns.

[75] X. Yang, J. Su, J. Lang, C. L. Tan, T. Liu, and S. Li. An entity-mention model for coreference resolution with inductive logic programming. In *ACL*, pages 843–851, 2008.

[76] X. Yang, J. Su, and C. L. Tan. A twin-candidate model for learning-based anaphora resolution. *Computational Linguistics*, 34(3):327–356, 2008.