Morphological POS Tagger for Hindi Language

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

Master of Engineering
in
Computer Science & Engineering

Thapar University, Patiala

By:
Rajeev Rathor
(80832018)

Under the supervision of:
Mr. Parteek Bhatia
Senior Lecturer, CSED

JUNE 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

Certificate

I hereby certify that the work which is being presented in the thesis report entitled, **"Morphological Part of Speech Tagging "**,submitted by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Parteek Bhatia* and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(***Rajeev Rathor)***

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mr. Parteek Bhatia)
Senior Lecturer
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. SEEMA BAWA)
Professor & Head
Computer Science & Engineering. Department
Thapar University
Patiala

(Dr. R.K.SHARMA)
Dean(Academic Affairs)
Thapar University,
Patiala.

Acknowledgment

I express my sincere and deep gratitude to my guide Mr. Parteek Bhatia, Senior Lecturer in Computer Science & Engineering Department, for the invaluable guidance, support and encouragement. He provided me all resource and guidance throughout thesis work.

I am thankful to Dr. (Mrs.) Seema Bawa, Head of Computer Science & Engineering department Thapar University Patiala, for providing us adequate environment, facility for carrying thesis work.

I would like to thank to all staff members who were always there at the need of hour and provided with all the help and facilities, which I required for the completion of my thesis.

I would also like to express my appreciation to my co-worker and my great friends Mohit, Arun, Neeraj, Sandeep, Rohan, Teena, Ashish Jain for motivation and providing interesting work environment. It was great pleasure in working with them during this thesis work.

At last but not the least I would like to thank God and mine parents for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Rajeev Rathor

(80632018)

.

Abstract

Hindi is a national language of India, spoken by 500 million people and ranking 4th by majority spoken in the world. But still Language is barrier to take benefits of Information Technology revolution in India. So there is need for computers to perform natural language processing so that computer based system can be interacted by users through natural language like Hindi and handled ,operated by users who have knowledge of regional language. So language Translator is important tool to resolve this problem. POS tagger is one of the important tools that are used to develop language translator and information extraction so that computer based be compatible for natural language processing.

Part-of-speech tagging is the process of assigning a part-of-speech like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence. There are different types POS tagger are exist, are based on probabilistic approach and some based on morphological approaches. But Hindi language is morphologically rich language. It has well defined morphological structure and well defined grammar. Morphology is root of all challenges that are arise in POS tagging as well as this aspect of language is also proved as boon to resolve all problems arise in POS tagging. So this thesis work, concentrating on morphological structure of language to develop better Morphology based POS tagger. In aspect of implementation of morphological based POS tagger for Hindi language, we have follow algorithm completely based on morphological structure of Hindi language and morphology based database. For implementation of Hindi POS tagger, some intermediate tools like Morph Analyzer and Stemmer Analysis and word sense disambiguation tool, are used. Stemming is process to extract original word (root) and longest suffixes from input word. Morph analyzer return correct grammatical information by conducting morphological rules. For strong morphological analysis we have proposed some grammatical rules and morphological structure based rules for determining correct tags from complex sentences. When same word has more than one sense, 'Word Sense Disambiguation' is used to find correct sense. But the limitation of system is that some types of ambiguities are still there. Handling the unknown word and proper noun are still a problem.

Hindi is a free order language and morphological rich, so ambiguity resolution and fixed order word group extraction for right tagging, is challenge for correct Hindi word tagging. In this Thesis work, make an attempt to resolve the ambiguity and assign to correct tag to each word of sentences. Accuracy of POS tagger is dependent on strength of morph analysis. Accurate POS tagger is dominating in accurate language translation and other natural language processing based application. So our efforts in this thesis, to improve the word level as well as sentence level accuracy using morphological and grammatical structure of language. Word level accuracy significantly better but sentence level accuracy still low. So some future amendment, are required to improve the sentence level accuracy at best level.

Contents

List of Table

List of Figures

Chapter 1

Natural Language Processing and POS Tagger

1.1 Meaning of Natural Language Processing

A Natural Language (NL) is any of the languages naturally used by humans, *i.e.* not an artificial or man-made language such as a programming language. Natural Language Processing (NLP) is a convenient description for all attempts to use computers to process natural language. NLP is an area of artificial intelligence research that attempts to reproduce the human interpretation of language for computer system processing. The ultimate goal of NLP is to determine a system of language, words, relations, and conceptual information that can be used by computer logic to implement artificial language interpretation [16]. A complete natural-language processor extracts meaning from language on at least seven levels. However, we'll focus on some of important levels which are as follows:

Morphological: A morpheme is the smallest part of a word that can carry a discrete meaning. Morphological analysis works with words at this level. Typically, a natural-language processor knows how to understand multiple forms of a word *i.e.* its plural and singular, for example.

Syntactic: At this level, natural-language processors focus on structural information and relationships.

Semantic: Natural-language processors derive an absolute (dictionary definition) meaning from context.

Pragmatic: Natural-language processors derive knowledge from external commonsense information.

Correct Stemmer Extraction: NLP should extract correct suffixes and root word from input words using Suffix Replacement Rules.

Retain Meaning of sentence: NLP should retain original meaning of the sentence after processing.

1.2 Application of Natural Language Processing

NLP can play a vital role in the following areas:

- Automatic summarization
- Part-of-speech tagging
- Information extraction
- Information retrieval
- Machine translation
- Named entity recognition
- Natural language generation
- Optical Character Recognition
- Question answering
- Speech recognition
- Spoken dialogue system
- Text simplification
- Text to speech
- Text-proofing

1.3 Limitations of Natural Language Processing

One of the major limitations of modern NLP is that most linguists approach NLP at the pragmatic level by gathering huge amounts of information into large knowledge bases that describe the world in its entirety. These academic knowledge repositories are defined in ontology that take on a life of their own and never end up in practical, widespread use. Even natural-language modules that perform specific, limited, linguistic service are not financially feasible for use by the average developer. In general, NLP faces the following challenges [20]:

- **Physical limitations:** The greatest challenge to NLP is representing a sentence or group of concepts with absolute precision. The realities of computer software and hardware limitation make this challenge nearly insurmountable. The realistic amount of data necessary to perform NLP at the human level requires a memory

space and processing capacity that is beyond even the most powerful computer processors.

- **No unifying ontology:** NLP suffers from the lack of a unifying ontology that addresses semantic as well as syntactic representation. The various competing ontology serve only to slow the advancement of knowledge management.
- **No unifying semantic repository:** NLP lacks an accessible and complete knowledge base that describes the world in the detail necessary for practical use. The most successful commercial knowledge bases are limited to licensed use and have little chance of wide adoption
- **Semantic overloading of information retrieval systems:** The performance of most of the current information retrieval systems is affected by semantic overload. Web crawlers, limited by their method of indexing, more often than not return incorrect matches as a result of ambiguous interpretation.
- **Sense Ambiguity:** A striking feature of NLs is that many words and sentences have more than one meaning (*i.e.* are semantically ambiguous), and which meaning is correct depends on the context.

1.4 **Part-of-Speech (**POS) Tagger

As discussed earlier, POS tagging is an important application of NLP. Our thesis is an attempt to design a POS tagger of Hindi Language.

Part-of-speech tagging is the process of assigning a part-of-speech like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence., means which word is noun, adjective, verb and also evaluate tense, gender, number, actor and other aspects of Hindi grammar. Morphological Analysis provides all grammatical information about the words, on the basis of sentence structure and morpheme properties [2]. So, in development of morphology based Hindi POS tagger, Morphological analysis play crucial role in correct tagging. Degree of morphological analysis reflects the accuracy of POS tagger. If degree of analysis is very high, means accuracy of tagger is high.

Morphological analysis means to evaluate the structure of sentences and order of appearance of words in sentences. This analysis also includes the association of word

from root word, their suffix, prefix and stemmer of words. Morphological analysis will be as follows:

Input sentence: अध्यापक ने लड़कों को पुरानी पाठमालाओ□को पढ़ाया था।

Input word = पढ़ाया

Category = Verb

Root = पढ़ना

Suffix = या

Input word = लड़कों

Category = Noun

Root = लड़का

Suffix = ओ□

Person = 3rd person

Tense = Past

Gender = Male, Plural

It is a tedious work rather than just having a list of words with tag in dictionary or word database and by applying any search techniques or query. We assign tag to each part of speech. Some words may have more than one tag. This problem is commonly found as huge numbers of words-forms are ambiguous and exist in more then one.

For example: "सोना को टूटा हुए पलाश पर सोना पसद है।".

Here, 'सोना' word appears two times and has two different senses and it returns three grammatical tags. The word 'सोना' may act as Proper Noun (Name), Noun (Gold) or Verb. So, there are ambiguities in assignment of tag to part of speech. Building a POS-Tagger for Hindi becomes complicated; as the language is agglutinative *i.e.* the postpositions (suffix) are joined to the root word to form a single word and this word look quite different with different sense tag.

For Example: सोना + या = सोया

Here, 'सोना', 'सोया' are both verbs. 'सोया' is technically quite different from 'सोना' where 'सोना' is root word of 'सोना' and 'सोया' words. Here, both words are used in different sense *i.e.* noun, verb or adjective.

1.5 Need of Part of Speech Tagging for Hindi Language

Part of Speech tagging is dominating technique in Natural Language Processing. It is used in several Natural Languages processing based software implementation. Language Translation is one of them. Through POS, we are evaluating correct grammatical information corresponding to each word and by applying grammatical rules of destination language; we can determine correct word order and word group order.

By parsing a natural language we can chunk or recognizing structural units of the language that allow us to identify the meaning of sentence. Such Chunking may be achieved using different methods. Among Indo-Aryan language, Sanskrit is purely free-order language but some language like Hindi or Bengali have partially free order language. Means Hindi language is free order at word group level but in context word group internal structure, it is fixed order. So, in internal structure of word group, words appear in fixed order. This aspect of language yields some problem in Machine learning tools implementation. But this is also boon to proceed toward problem solution. English is fixed order language and also reveal the lack of morphologically richness. In case English, there are no big challenges in POS tagger implementation as in Hindi.

Let us consider sentence for Hindi:

उपदेश पुराने महल को देखकर □र जाता है।

This sentence can be written in many forms as:

पुराने महल को देखकर उपदेश □र जाता है।

उपदेश □र जाता है पुराने महल को देखकर।

It is clear that Hindi language is fixed order at word group internal structure but free order at word level.

1.6 Previous Work on Hindi POS Tagging

There have been many implementations approaches of POS tagger using machine learning techniques, mainly for morphological and corpus-rich languages like Hindi. These techniques can be based on probabilistic, morphological or rule based. There is some amount of work done on morphology-based Hindi POS tagging. Some other techniques also followed for development of Hind Pos tagger like HMM model, maximum entropy model, neural network based model for improving accuracy of tagger.

C-DAC and TDIL IIT Bombay give admirable contribution on this research project 'POS tagger for Hindi'. The computational Paninian parser developed by Bharti [12] based on a technique where POS tagging is implicit and is merged with the parsing phase. Ray proposed an algorithm that identifies Hindi word groups on the basis of the lexical tags of the individual words. Their partial POS tagger reduces the number of possible tags for a given sentence by imposing some constraints on the sequence of lexical categories that are possible in a Hindi sentence. UPENN also has an online Hindi morphological tagger1 but there exists no literature discussing the performance of the tagger [16].

Some example of Pos tagger Such as, transformation-based error-driven learning based tagger and maximum entropy Markov model based tagger. A statistical POS tagger based on Markov models with a smoothing technique and methods to handle unknown words .Another approach for POS tagging is based on incorporating a set of linguistic rules in the tagger [11].

There has been some previous work towards building a Hindi POS tagger, such as the partial POS tagger discussed by Ray and Shrivastava, propose harnessing morphological characteristics of Hindi for POS tagging [9]. This was further enhanced in, which suggests a methodology that makes use of detailed morphological analysis Stemmer analysis and lexicon lookup for tagging. The results are further improved by applying disambiguation rules for Hindi. One more approach come in existence for POS tagger, is morphological analysis based on Maximum Entropy model. But all tagger have been developed, retain their accuracy and performance problem still. So Another Hybrid approach based on both morphological approach and probabilistic approach, under processing to improve performance [13].

1.7 Issues Involved in POS Tagging For Hindi

Many issues have to be considered for better morphological analysis so that good POS tagger can be implemented with high accuracy and good performance. These issues are as follows:

- High strength of Morphological analysis as far as possible because accuracy and performance of system is depends on it.
- Stemmer Analysis of inflected words for grammatical information extraction.
- Generation of Suffix replacement rules for extracting root word and associated suffix.
- Generating Morphological rules in Sentence formation (word sequence in sentences).
- Handle the ambiguity associate with words in sentence; meaning and tag of word depend on context of statement.
- Algorithm design and implementation of system on the basis of morphological structure Rules for grammatical tag evaluation and ambiguity resolution.
- To Prepare the Strategy to handle unknown word and Proper Noun (Name).
- To periodically update the database or dictionary.
- Generation of Tagged output of Hindi sentence.

Chapter 2

Basics of POS Tagger

A Natural Language (or ordinary language) is a language that is spoken, written, or signed by humans for general-purpose communication such as Hindi, Tamil, Bengali and English *etc.* NL is often contrasted with artificial or constructed languages such as C language, Java, Perl *etc.* The understanding of natural languages reveals much about not only how language works in terms of syntax, semantics, phonetics, phonology *etc.* but also about how the human mind and the human brain process language. In linguistic terms, 'Natural Language' only applies to a language that has evolved naturally, and NL primarily preferred by native (first language) speakers [22].

2.1 Grammar & Grammatical Attributes

Grammar is a set of the rules governing the Natural Language. The first systematic grammars originate in Iron Age India, with Panini (4th BC). Hindi grammar provides rules to specify how to construct sentences and order of words and how meanings are created in native language sentence. Hindi grammatical rules also specify syntax and word association with suffix and prefixes *etc*. The theory of universal grammar proposes that all natural languages have certain underlying rules which constrain the structure of the specific grammar for any given language.

2.1.1 Grammatical Attributes for Hindi Language: There are following grammatical attributes as given below.

1-Noun

2-Proper Noun (Name) 'राम', 'राजीव', 'ताज' 'महल', 'आगरा'

3-Gender

MALE: 'लड़का', 'बालक', 'शेर', 'आदमी'

FEMALE: 'लड़की', 'औरत', 'गाय'

4-Person

1st person: 'मैं', 'हम'

2nd person: 'तुम', 'आप'

3rd person: 'वह', 'उसने', 'वे', 'ये', Proper Noun

5- Pro-Noun: 'मैं', 'हम', 'तुम', 'आप', 'उसने', 'वे', 'ये', 'अपना', 'हमारा'

6-Adjective: 'अच्छा', 'काला', 'पुराना'

7-Verb: 'खाना', 'पिया', 'देगा'

8-Adverb: 'निश्चित', 'मुश्किल से',

9-Number: Singular: 'लड़का', 'आगरा', 'मानव'

Plural: 'लड़के', 'शेरों', 'गायों'

10-Case Information: Nominative, Possessive, Accusative.

11-Tense: Present, Past, Future

2.2 Meaning of Morphology

Morphology is the field of linguistics that studies the internal structure of words. While words are generally accepted as being the smallest units of syntax, it is clear that in most (if not all) languages, words can be related to other words by some grammatical rules. The rules understood by the speaker reflect specific patterns (or regularities) in the way words are formed from smaller units and how those smaller units interact in speech.

In this way, "morphology is the branch of linguistics that studies patterns of word-formation within and across languages, and attempts to formulate rules that model the knowledge of the speakers of those languages".

2.3 Models of morphology

There are three principal approaches to morphology, which each try to capture the distinctions above in different ways. These are:

- **Morpheme-based morphology:** Morpheme-based morphology which makes use of an Item-and-Arrangement approach.
- **Lexeme-based morphology:** Lexeme-based morphology which normally makes use of an Item-and-Process approach.
- **Word-based morphology**: Word-based morphology which normally makes use of a Word-and-Paradigm approach.

2.3.1 Morpheme-based morphology

In morpheme-based morphology, word-forms are analyzed as arrangements of morphemes. A morpheme is defined as the minimal meaningful unit of a language. In a word like 'लड़कों', we say that 'लड़का' is the root, and that 'कों' is an inflectional morpheme. This way of analyzing word-forms as if they were made of morphemes put after each other like beads on a string, is called Item-and-Arrangement. The fundamental idea of morphology is that the words of a language are related to each other by different kinds of rules. Analyzing words as sequences of morphemes is a way of describing these relations, but is not the only way. In actual academic linguistics, morpheme-based morphology certainly has many adherents, but is by no means the dominant approach [22].

2.3.2 Lexeme-based morphology

Lexeme-based morphology is an Item-and-Process approach. Instead of analyzing a word-form as a set of morphemes arranged in sequence, a word-form is said to be the result of applying rules that alter a word-form or stem in order to produce a new one. An inflectional rule takes a stem, changes it as is required by the rule, and outputs a word-form. A derivational rule takes a stem, changes it as per its own requirements, and outputs a derived stem; a compounding rule takes word-forms, and similarly outputs a compound stem.

2.3.3 Word-based morphology

Word-based morphology is a Word-and-Paradigm approach. This theory takes paradigms as a central notion. Instead of stating rules to combine morphemes into word-forms or to generate word-forms from stems, word-based morphology states generalizations that hold between the forms of inflectional paradigms. The major point behind this approach is that many such generalizations are hard to state with either of the other approaches. A

morpheme-based theory would call an inflectional morpheme, corresponds to a combination of grammatical categories, for example, "third person plural." Morpheme-based theories usually have no problems with this situation, since one just says that a given morpheme has two categories. Item-and-Process theories, on the other hand, often break down in cases like these. Because they often assume that there will be two separate rules here, one for third person, and the other for plural.

Word-and-Paradigm approaches treat these as whole words that are related to each other by analogical rules. Words can be categorized based on the pattern in which they fit. This applies both to existing words and to new ones. Application of a pattern different than the one that has been used historically can give rise to a new word.

2.4 Lexeme

A lexeme is an abstract unit that roughly corresponds to a set of forms, taken by a single word. Lexemes are often composed of smaller units with individual meaning called morphemes.

2.5 Morpheme

A morpheme is the smallest linguistic unit that has semantic meaning or in other word minimal meaningful language unit. It can not be further divided into smaller meaningful units. Morpheme further categorized into types morphemes as follows.

2.5.1 Bound Morphemes: Bound Morphemes can not occur on their own (root form), *e.g.* *लड़का + ए= लड़के.*

2.5.2 Free Morphemes: Free Morphemes can occur as separate words, *e.g.* 'मानवता', 'भगवान'.

In a morphologically complex word, a word composed of several morphemes but one constituent may be considered as the basic one, the core of the form, with the others treated as being added on. The basic or core morpheme in such cases is referred to as the stem, root, or base, while the add-ons are affixes**.**

2.6 Inflection

In grammar, inflection or inflexion is modification of a word (or more precisely lexeme) to reflect grammatical information, such as gender, tense, number, case, person. The concept of a "word" independent of the different inflections is called a lexeme. The form of a word that is considered to have no or minimal inflection is called a lemma. Some morphological rules relate different forms of the same lexeme are called inflectional rules.

In Hindi noun words are inflected for number, gender information. This grammatical information is hidden in last suffix that associated with word. Using Stemmer techniques, we can determine grammatical information.

For examples: Let us consider the word 'लड़के' is inflected word and its original word (root word) is 'लड़का'. Here inflected word: 'लड़के' conceal grammatical information like given word is Gender: Male, Number: plural.

Main verbs are inflected and concealed several grammatical information like tense, Gender, Number, Case.

Let us consider sentence 'एक लड़की लड़के को किताब देगी।'

Here inflected verb: 'देगी'

Root verb (word): 'देना'

Suffix: 'ऐगी'

Grammatical Information: Gender: Female, Number: Singular, Tense: Future, Case: Direct

2.7 Inflection vs. word-formation

It is possible to distinguish two kinds of morphological rules. Some morphological rules relate different forms of the same lexeme; while other rules relate two different lexemes. Rules of the first kind are called inflectional rules, while those of the second kind are called word-formation. The Hindi plural, as illustrated by 'लड़का' and 'लड़के', is an inflectional rule. But compound words like 'किताबी-कीड़ा' provide an example of a word-

formation rule. Informally, word-formation rules form new words (that is, new lexemes), while inflection rules yield variant forms of the same word (lexeme).

There is a further distinction between two kinds of word-formation: derivation and compounding. Compounding is a process of word-formation that involves combining complete word-forms into a single compound form; 'किताबी-कीड़ा' is therefore a compound, because both 'किताबी' and 'कीड़ा' are complete word-forms in their own right before the compounding process has been applied, and are subsequently treated as one form. But there are many examples where linguists fail to agree whether a given rule is inflection or word-formation.

2.8 Classification of Tagger

Tagger can be characterized as rule-based and statistical based.

- Rule-based
- Stochastic based tagger

2.8.1 Rule-based POS Tagger

Rule-based tagger use linguistic rules to resolving the tag ambiguity. Since Indian languages are morphologically rich in nature. There is a need to develop linguistic rules based on grammatical structure and morphological structure of language. In Rule based tagger we evaluating or resolving the correct tag on the basis of word order in sentence and word order in word group. Along with we applying stemmer analysis to extracting the correct root (original) word and corresponding associated suffixes. Stemmer analysis plays vital role in evaluation of grammatical information number, gender, tense *etc* from the inflected words. Further on the basis of stemmer analysis, we can identify the new word or unknown word.

2.8.2 Stochastic Based POS Tagger

Stochastic POS tagger is based on the probabilities of occurrences of words for a particular tag. Any model, which incorporates with frequency or probability of word, *i.e.* statistics, may be properly labeled stochastic. The simplest stochastic taggers disambiguate words based solely on the probability that a word occurs with a particular

tag. In other words, the tag encountered most frequently in the training set, is assigned to an ambiguous instance of that word.

In word frequency approach, occurrence of word is to calculate the probability of a given sequence of tags occurring from given training set or tagged corpus. This is referred as the *n-gram* approach, referring to the fact that the best tag for a given word is determined by the probability that it occurs with the *n-1* previous tags. Probabilistic approach, are better for those language, which are fixed order language. But in case of free order language, Stochastic Approach does not return better result. Along with there is another problem, word ambiguity *i.e.* one word has more then one meaning with more than tag and meaning of word are depend on context of sentence. So in case of Hindi, stochastic POS tagger is to fail to give better result.

There are four approaches to implement Stochastic based POS tagger:

- Conditional Random Fields
- Maximum Entropy Model
- Hidden Markov Model
- Memory Based Learning

Chapter 3

Challenges of POS Tagging

3.1 Problems with POS Tagging in Hindi

Hindi is morphological rich language so one word exist in many form and by adding suffix or prefix with the word, generate new word which looking quite different and sometimes with different meaning to the origin. Such a word has multiple entries in dictionary (table) or the lexicon (one for each category). So, it is clear, a word or a morpheme displays POS ambiguity and sometimes same word with different meaning in different context in same sentence. So it is critical problem in POS tagging for Hindi [2].
The complexity of the task can be understood looking at the following Hindi sentence where the word ` पी ' falls into two different POS categories-

1-राजीव पी हुई पुस्तकें देखकर परेशान हो जाता है ।

2-राम ने पुस्तकें पी ।

In most cases word 'पी'is a verb but in this sentence it is actually an adjective.

1. **POS**: Adjective, **Number**: plural

राम ने पुस्तकें *पी ।*

'पी' word in this sentence is verb

2. **POS**: Verb, **Number**: singular

One of the difficult tasks here is to choose the appropriate tag based on the morphology of the words, word order based on grammatical rules and the context in which words are used. There is one more problem; new words appear all the time in the vocabulary of language. Thus, a method for determining the tag of a new word is needed when it is not present in the dictionary. This is done using context information and the information coded in the affixes or suffix. Affixes and suffixes in Hindi (especially in nouns and verbs) are strong indicators of a word's POS category.

The complexity further Increases when it comes to tagging for free-word order language like Hindi where almost all the permutations of words order combination in a clause are possible. This aspect of language makes the task of a stochastic tagger difficult.

3.2 Challenges in POS Tagging for Indian Languages

Correct POS tagging is critical task for any NLP application implementation. There are many challenges that should be handled to achieve good accuracy and good performance, are given below.

1. POS ambiguity
2. Free word ordered structure
3. Complex morphology of Indian Languages
4. Word Inflection (Stammering)
5. Word sense ambiguity
6. Problem to handle Unknown word and newly create word
7. Problem to handle Proper Noun (Name)

3.3 POS Tagger Ambiguity

Hindi is partial free order language and morphological rich languages so several types of ambiguities are exist. Ambiguity means one thing (word, sentence) are exist in many form. For example in word level ambiguity, one word is used in many senses (meaning) and with different tag. So it is very critical task to decide what should be exactly there.
POS ambiguity categorized into two categories.
1-Inter-POS Ambiguity
2-Intra-POS Ambiguity

3.3.1 The inter-POS ambiguity: The inter-POS ambiguity surfaces when a word or a morpheme displays an ambiguity across POS category. Such a word has multiple entries in the lexicon or database (one for each category). After stemming, the word would be assigned all possible POS tags based on the number of entries it has in the lexicon. The complexity of the task can be understood looking at the following Hindi sentence.

सोने लाल सोने जैसे आभूषणों को छिपाकर घर मे सोने गया ।

Where the word सोना falls into three different POS categories and also have three different meaning –

'सोने' – Name of person - Proper Noun

'सोने' – Gold - Adjective

'सोने'- Sleeping -Verb

सोने लाल सोने जैसे आभूषणों को छिपाकर घर मे सोने गया ।

So it is clear what is problem in tagging. It is very crucial and complex task to evaluate correct tag and correct sense. The complexity further increases when it comes to tagging a free-word order language like Hindi where many the combinations of words order in a clause are possible.

For Example

A- सोने लाल सोने जैसे आभूषणों को छिपाकर घर मे सोने गया ।

B-सोने जैसे आभूषणों को छिपाकर सोने लाल घर मे सोने गया ।

3.3.2 Intra-POS ambiguity: Intra-POS ambiguity arises when a word has one POS with different feature values, *e.g.*, the word `लड़के' (लड़का/लड़के) in Hindi is a noun but can be analyzed in two ways in terms of its feature values:

एक लड़की ने लड़के को किताब दी ।

Input Word: 'लड़के'.

POS: *Noun*, **Number**: Singular **Case**: *Oblique*

इस घर के लड़के थापर में पढ़ते है ।

Input Word: 'लड़के'.

2. **POS**: *Noun*, **Number**: Plural, **Case**: *Direct*

One of the difficult works, is to choose the appropriate tag based on the morphology of the word and the context in which they are used. Also, new words appear all the time in the vocabulary. Thus, a method for determining the tag of a new word is needed when it is not present in the dictionary or lexicon. This is done using context information and the information coded in the affixes, as affixes in Hindi (especially in nouns and verbs) are strong indicators of a word's POS category. For example, it is possible to determine that the word ` जायेगा' *(will go)* is a verb, based on the environment in which it appears and the knowledge that it carries the inflectional suffix 'एगा' that attaches to the base verb 'जा'.

But this problem can be resolve without using any neural network or any artificial intelligent system. We just follow morphological structure of language and we express already that Hindi is Fix order language at word level. So we must generate grammatical and morphological rules of sentence formation on concentrating over word group structure. These rules are converted into programming logics and implementation of ambiguity resolution on the basis of these rules, are discuss in next chapter.

3.4 Word Order

"Word order typology defined as sentence of any languages can be arranged in different way according to their constituents (words) but obtained sentences, relative to each other, and there are semantically correspondences between these arrangements." But sense (meaning) of sentence should be retained as original. There are two categories.

1-Free order

2- Fixed order

3.4.1 Free order

"The distribution of words in a sentence and distribution of sentence, are independent to each other, is called free order language". But the meaning of the statement (order free) should be retain the original sense and properly understood, *i.e.* it is not absolutely free. In these languages there must be a significant amount of morphological marking (richness) to disambiguate the roles of constituents. Sanskrit is purely 'Free order' Language. Some more Examples are Hindi, Bengali, Hungarian.

For Example: पुरानी किताबों को पढ़कर राधा अपने बिस्तर पर सो गयी ।

This sentence further rearrange in two more form as.

1- राधा अपने बिस्तर पर पुरानी किताबों को पढ़कर सो गयी ।

2- राधा पुरानी किताबों को पढ़कर अपने बिस्तर पर सो गयी ।

Here both sentence are grammatically and semantically are correct and retain its original sense.

Free order language creates complexity and ambiguity in POS tagging. Sense identification is determined by human intelligence. So correct POS tagging for free order language, is tedious and critical work.

3.4.2 Fixed order

"The distribution of words in a sentence and distribution of sentence, are dependent to each other, is refer as fixed order language" *i.e.* constituent of language (words) are appear in fixed order in formation of sentence. If we make attempt to rearrange them then sentence lose its original sense or grammatical error are arise. So, it is clear that POS tagging is easy and with less ambiguity because we just create grammatical rules according to according to words order in sentence after that easily convert it into computer logics. English categorized in fixed order language.

3.5 Problem in POS tagging due to Word order For Hindi

Some indo-Aryan languages like Hindi, Bengali are partial Fee-order language which means language is fixed order at word group level but free at word level. Internal structure of Word group is follow 'fixed order' but order of 'Word Group' is free to appear in sentence. We can elaborate, as noun always come after Adjective. Word group can be arbitrarily arranged in sentence formation without affecting original sense.

This aspect of language create problem in POS tagging. We can not create absolute, perfect rules of sentence formation for POS tagging system design. So there is need some artificial intelligence system for evaluating correct tags.

But there is one aspect 'Fixed order at word Group level' is proved boon for POS system implementation. So to resolve complexity and ambiguity in POS tagging, we have to

design rules at word group level. According to word order in word group rules are created and apply in computer logics implementation. Consider Hindi Sentence:

पुरानी किताबों को पढ़कर राधा अपने बिस्तर पर सो गयी थी ।

This sentence can be written two more form which are grammatically and semantically correct and retain original sense.

(i) राधा पुरानी किताबों को पढ़कर अपने बिस्तर पर सो गयी थी ।

(ii) राधा अपने बिस्तर पर पुरानी किताबों को पढ़कर सो गयी थी ।

So there is some complexity in tag assignment. But if illustrating on some word group (word group) which are always appear together.

So there are word group

1- "पुरानी किताबों को पढ़कर"

2- "अपने बिस्तर पर"

3- "सो गयी थी"

So it is clear, we should form rules on consideration of 'word group' not on whole sentence structure and observe rules. Some are given below.

Rules:

1-Adjective +Noun

2- Main verb +Helping Verb

3- Relative Noun+ rule 1

3.6 Problem in Word Inflection (Stammering)

Words that constitute the statement or sentence, are not present in their original form. These words are associated with some suffixes and prefixes. These types are called inflected word. If we are search these words in database, show word are not found. These inflected words reveal many grammatical information.

So there are various challenges as.

1- Identify that words are inflected or presented in original form. This is critical again a crucial job in POS tagging system. *i.e.* to differentiate inflected and original word in sentence is tough work.

   For example लड़कियों ने लड़के को किताब दी। 2nd sentence: सोने लाल को सोना पसद्व है।It is clear to determine inflected and original word in sentence.

2- After evaluating inflected word, we have to determine associated suffix with the word. But length of suffix is not fix. For same root word suffixes are different and many. There is another problem length of suffixes is different in some case it is only last latter mantra, some cases it is one character, some case it is two or three characters long.

3- Some that are looking inflected but not inflected and are used in original word's sense. For example 'सोने लाल को सोना पसद्व है' here word 'सोने लाल' are presented in original form but it is treated as inflected corresponding to verb 'सोना'.

3.7 Problem of Word sense ambiguity

When a word, term, notation, sign, symbol, phrase, sentence, or any other form used for communication, are called ambiguous if it can be interpreted in more than one way. So correct identification of word's meaning or sense in sentence, is very critical job. Let consider sentences.

1- राजू कलम से कागज पर लिखता है।

2- राजू ने बाग में कलम लगाई।

In these sentences word 'कलम' has two meaning.

In first sentence 'कलम' means pen. But in Second sentence 'कलम' means branch used for plantation. This problem can resolve by 'Word sense disambiguation' (WSD) tool which identified right sense of word.

3.8 Problem to handle Unknown word and newly create word

To handle unknown word in POS tagger, is challenge. New words are come in use with time and some existing words are out of use with time. Together with, suffix association with word also change by authors due to which existing words look quite different. Because we search word from dictionary but these type word are missing and create problem in grammatical information extraction. Together with, suffix association with word also change by authors due to which existing words look quite different.

3.9 Problem to handle Proper Noun (Name)

Proper Noun is the name of person or place. So, word which are used as name of some person or things and also appear with some information in sentence. So, it very difficult to say word are used as name or used as other grammatical aspect. If word which is used as name then we does not require to search it meaning and tags from the dictionary for tagging or translation. There are large number of proper noun may possible so they can not be stored in database and some word that can used as proper noun as well as verb, adjective, adverb or other form. So, Proper noun identification is toughest task of POS tagging. For example

1-सोना को सोना चाँदी छुपाकर घर मे सोना है ।

This sentence सोना contain three times but all three सोना are used in different sense. But 1st सोना is used as name so its identification is crucial task.

लाल सिह खाना खाता है।

In this sentences 'लाल सिह' is used as name but actually dictionary contain word 'लाल' as adjective (red) and 'सिह' (Lion) as noun.

So To solve this problem NER 'Name Entity Reorganization' approach is follow. The key problem is to extract some special name entities, such as person name, location, and organization, in a text. This is still under working.

Chapter 4
Morphological Analyzer and Stemmer

4.1 Morphological Analyzer

The Morphological Analyzer is an integral part of any Natural Language Processing system, especially in the context of Indian languages. For fixed word order languages and partial fixed order language, the semantics of a word are primarily governed by its absolute and relative position words inside a sentence. But in case of partial fixed order Indian languages, the semantics (part of speech and other subtleties) are heavily dependent on the surface structure of the word. The task of the morphological analyzer is to identify the structural components of a word. So, morph analyzer used for information extraction.

For Hindi, the morphological analyzer can identify the tense, aspect, modality and person of an inflected verb form. For Hindi, gender and number may be identified as well. Also, the root or 'धातु' of the verb will be identified by the analyzer.

For nouns, the task of the morphological analyzer is to determine its Vibhakti (inflection), root, grammatical tag and prefixes. The morphological analyzer process the lexical word groups corresponding to the noun and help to determine semantic rule.
We also aim to perform the decompositions for sandhi and samaasa (conjugating and compounding words) so as to have a powerful vocabulary for the system, and a generalized prefix and suffix handler.

4.2 Approaches for Morphological Analysis

The morphological analysis is the process of extracting grammatical information about the word on the basis of properties of the morpheme, suffix, prefix *etc*. Our approaches of morphological analysis for Hindi language are:

1. Phrase level Analysis
2. Word level Analysis and then its extension to phrase level

4.2.1 Phrase level Analysis:

The structure of sentence has been seen as the combination of both phrases. A sentence comprises a verb phrase and a noun phrase. We need to have paradigms/rules/patterns,

for representing a phrase, to identify such occurrence in any given input. For verbs, 622 paradigms have been identified. Each paradigm represents a unique Verb grouping.

Sentence represented: (Verb) 'रहे होंगें', 'चुके हैं', 'हुआ होगा'.

Root Word: 'खेलना', 'दौड़ना', 'भागना'

Given a sentence राम पुस्तक पढ़ता है।

Paradigm 1 will follow 'पढ़ता है'. Now we have the Verb group and paradigm under which it falls. The morphological analysis for this group is stored in TAM-GNP analysis and will be retrieved from there. The analysis of the sentence above will be Verb, aspect, stative, gender male [2].

Thus the paradigms help in identifying:

1. Verb group.
2. Main verb.
3- Auxiliary Verb

4.2.2 Word Level Morphological Analysis: To analyze an input word, the Morphological analyzer at word level needs he following information:

1- Grammatical category of the word.
2- Suffix present
3- Morphemes present in suffix.

Stemmer analysis approaches first two requirements. We discussed these given topics in detail in next chapter.

4.3 Classification of Morphological Analysis

Hindi is a partial free order language. So there are three types of morphological analysis are possible. One of them is used to determine morphological structure of word. Second Relation label morphological analysis, to evaluating the relation between the word groups so that order of words are determined with original sense. Thus, third one is grammatical morphological analysis, is used to obtain grammatical aspect of each word. Three type of morphological analysis are given below.

- 1-Attribute Label Resolution Morphological Analysis
- 2- Relation Resolution Morphological Analysis
- 3- Grammar Resolution Morphological Analysis

We explain each type of analysis in detail with their example as given below.

4.3.1-Attribute Label Resolution Morphological Analysis

Attribute label resolution deals with determining the Hindi equivalent of attribute labels in a word. Attribute label resolution may introduce new words, as is the case while referring to definitive and in-definitive articles. Attribute label resolution may also change the form of word depending on tense, number or gender of the word.

Attribute labels such as @def have the articles equivalent in the Hindi as वहाँ or वे depending on whether the word qualified as singular or plural. @indef maps to एक depending on the number of the word it qualifies.

For Example: 1- वे बाजार जायेंगी। (Here वे represents @def attribute)

2- राम एक आम खायेगा । (Here एक represents @indef attributes)

Attributes are divided into seven category of tense, aspect, mood, number, gender, person and vowel ending.

4.3.2 Relation Label Resolution Morphological Analysis

The resolution of a relation label depends on the presence or absence of certain properties. The rule-base is basically a set of conditions along with the corresponding action that is to be taken in the form of suffixing or prefixing a morpheme. In some cases, new word may be inserted represent clause information. For Example

उस राम ने मोहन को बचाया जो राधा से प्यार करता था।

The word 'उस' and 'जो' are added here.

4.3.3 Grammar Resolution Morphological Analysis

Relation label and Attribute label resolution have brought the form of sentence very close to its final form. Some morphology depends on the phonetic properties of the Hindi language word. In Grammatical Resolution morphology dealt with the grammatical

aspect of each word in the sentence which is depend on the order of word or syntactic or grammatical structure of sentence. This ambiguity resolution is depend on grammatical rules and word ordering in sentences. This include noun Morphology, Verb Morphology, Adjective Morphology *etc*.

4.4 Stemmer Analysis

Stemming is the process for reducing inflected (or sometimes derived) words to their, base or linguistic form, generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Stemming refers to root word origins. For example 'खोज', 'खोजेगा' and 'खोजा' all have 'खोजना' as the root stem. In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. So, there are many *stemming Algorithms*, or *stemmers*, have been developed, which attempt to reduce a word to its stem or root form. Thus, the key terms of a query or document are represented by stems rather than by the original words. This is not only means that different variants of a term can be conflated to a single representative form. It also reduces the dictionary size, that is, the number of distinct terms needed for representing a set of documents.

4.5 Word Sense Disambiguation

Word sense disambiguation (WSD) is the process of identifying which sense of a (having a number of distinct senses) is used in a given sentence. Because one word has more than one meaning and used more than one senses. But meanings and senses are depends on context information. This context information is obtained from Wordnet (NLP tool).

For Example: 1- राजीव कलम से पत्र लिखता है।

2- राजीव बाग मे पेड़ों को उगाने के लिए कलम लगाता है।

Both sentences have same word 'कलम' but in first one have 'कलम' with meaning (pen).

But in second have 'कलम' with meaning 'branch of tree used for Plantation '.

One problem with word sense disambiguation is deciding what the senses of word, are in sentence. In cases like the word '**कलम**' above, at least some senses are obviously different. In other cases, however, the different senses can be closely related, and in such cases division of words into senses becomes much more difficult. Different dictionaries will provide different divisions of words into senses. One solution some researchers have used is to choose a particular dictionary, and just use its set of senses. NLP tool 'WordNet', provide all sense and synonyms for a word with their all possible examples and context. So, result of WordNet is very useful tool for word sense disambiguation.

Chapter 5
Design of Hindi POS Tagger

5.1 Morphology Driven Tagger

Morphology driven tagger makes use of the affix information stored in a word and assigns a POS tag using no contextual information. Though, it does take into account the previous and the next word in a Verb Group to correctly identify the main verb and the auxiliaries, other POS categories are identified through lexicon lookup of the root form. The current lexicon has around 92,400 entries belonging to the major categories. The process does not involve learning or disambiguation of any sort and is completely driven by linguistic morphology rules. Morphological analysis are invoked at two level word level and group level. Brief discussion of them is given as follow.

5.1.1 At Word Level: A *stemmer* is used in conjunction with lexicon and Suffix Replacement Rules (SRRs) to output all possible root-suffix pairs along with POS category label for a word. There is a possibility that the input word is not found in the lexicon and does not carry any inflectional suffix. In such a case, *derivational morphology rules* are applied.

5.1.2 At Group Level: At this level a *Morphological Analyzer* (MA) uses the information encoded in the extracted suffix to add morphological information to the word. For nouns, the information provided by the suffixes is restricted only to `*Number*'. `*Case*' can be inferred later by looking at the neighboring words. For verbs, GNP values are found at the word level, while TAM values are identified during the VG Identification phase, described later. The analysis of the suffix is done in a discrete manner, *i.e.*, each component of the suffix is analyzed separately. A morpheme analysis table comprising individual morphemes with their paradigm information and analyses is used for this purpose.

MA's output for the word 'खाऊँगा' *(will eat)* looks like -

Stem: 'खा' *(eat)*

Suffix: 'ऊँगा' Category: *Verb*

Morpheme 1: 'ऊँ' Analysis: *1st Person, Singular*

Morpheme 2: 'ग' Analysis: *Future*

Morpheme 3: 'आ' Analysis: *masculine*

The architecture of the Morphology driven Tagger is shown in figure 5.1.

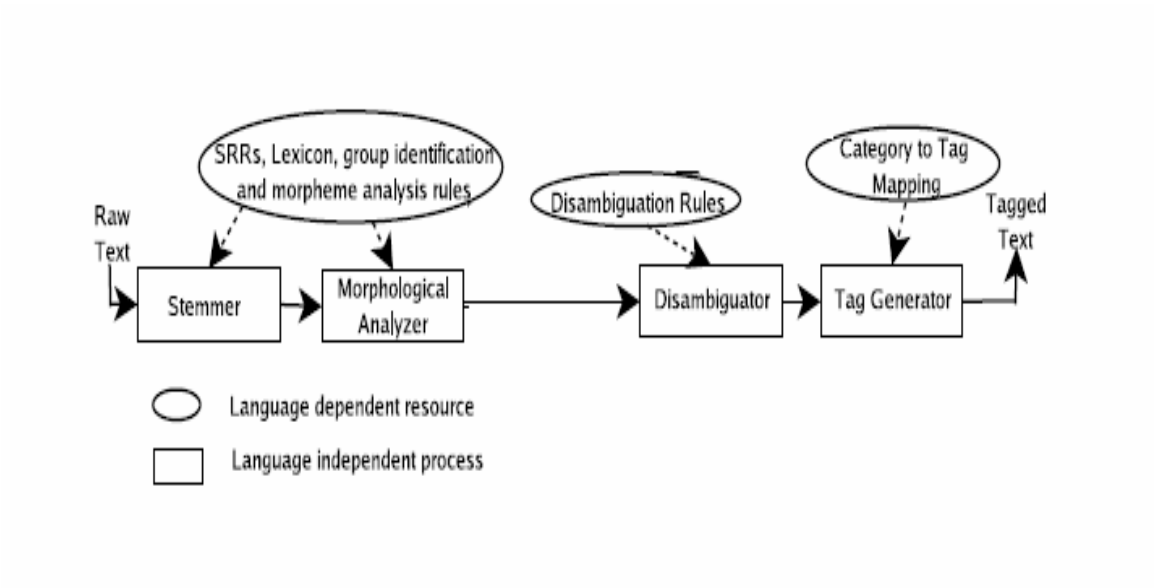


Figure 5.1 Morphological tagger Architecture

5.2 Morphological Analysis at Word Level

To analyze an input word, the Morphological analyzer at word level needs the following information:

1. Grammatical category of the word.
2. Suffix present.
3. Morphemes present in suffix.

The Stemmer satisfies the first two requirements. The resources which are used by the Morphological analyzer for evaluating suffix, root, person, tense etc grammatical information, are given below:

1. Suffix list

2. Word list/ Dictionary

3. Morpheme analysis

The first two resources are used by stemmer which is a part of the morphological analyzer. The morpheme analysis provides following feature information based on the suffix:

1. Verb: Tense, Aspect, Marker, Gender, Number and Person.

2. Noun: Gender, Case, Number.

3. Adjective: Number.

We will see category information and position information is vital while discussing the working of Suffix analyzer. Suppose the input word to stemmer system is word 'दौड़ेगा' then output of the system is given as.

Stemmer: Input:'दौड़ेगा'

Output:

Root: दौ□ना, Suffix: 'एगा' Category: verb

Suffix Analyzer: The analyzer will check for all the morphemes in the verb category. We saw in the morpheme analysis list that we store the position information of the morpheme, as the same morpheme may perform different function at different word positions. For example,

Suffix: 'एगा '

Category: Verb Analysis (with position information not stored):

'ए': 2nd Person, dual plural

'गा' : Future

'आ' :male

The duality information is given by suffix 'ए' only when it comes at the end of word like in 'पीते'. With the help of available morpheme analysis, morph analyzer searches for all possible morphemes and stores the analysis. After analyzing all the morphemes we make sure that only the analyses for the longest suffixes are returned.

The diagram for POS tagging using Morphological Analyzer at word level is shown in Figure 5.2 POS tagging using Morphological Analyzer at word level POS tagging using Morphological Analyzer at word level.

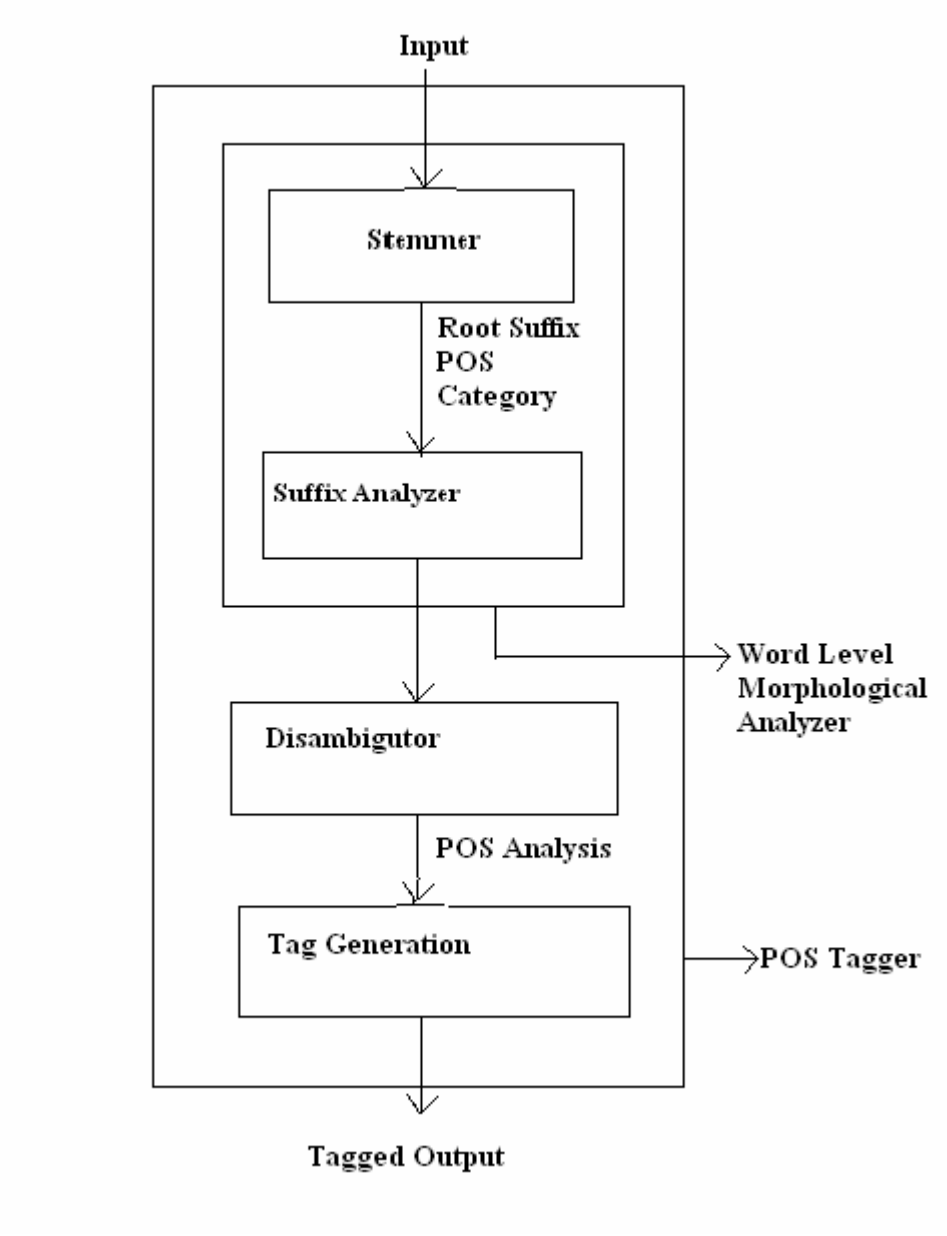


Figure 5.2 POS tagging using Morphological Analyzer at word level

5.3 Morphological based analysis for Hindi Language

The efficiency and accuracy of the system, depends on the richness of the analysis so we emphasis on morphological structure of Hindi language and Stemmer analysis. In Hindi, Nouns inflect for gender, number and case .To capture their morphological variations, they can be categorized on various paradigms based on their vowel ending, gender, number and case information *etc.*

For Example: राम ने पुस्तकों की कीमत पन्ने पर देखी है ।

Here actor ,noun or pronoun mostly come before word 'ने ' second ' का' ,'की' ,'के' use to give the characteristics of noun or pronoun. So, word after ' का' ,'की' ,'के ' *etc* mostly adjective or noun + adjective and before it noun or pronoun. Then in second step we categorized the word in inflected and original word which are appear in sentence. From the inflected cluster, we evaluate their original (root) word and their suffix with the help of stemmer.

Morphological analysis for identifying the correct POS category, are discussed in detail as given below.

5.3.1 Noun Identification

Noun's identification are depend on following rules and terms.

- Vowels Ending
- Valid suffix of word
- Last word of Sentence like ' ,हैं हु□,था, थी,थे,गा,ग़ी'
- Noun mostly appear before the word ना,ने,का,की,के,में,पर
- Noun appears just near the adjective .because adjective is characteristic of noun.
- Gender, Number , Person and Case information

There are 20000 noun classified into 20 paradigm which are developed by IIT Hyderabad and paradigm based on above parameter 1,2,6.Valid suffix is evaluating on the basis of Suffix Replacement rules of Hindi language .this analysis reducing ,a lot ambiguities in the process of right tag assignment to word. This Consideration of paradigm increases the accuracy of tagger by finding out correct root and suffix. One

thing clear that morphological analysis of suffix, root varies, depending on the mentioned paradigm.

5.3.2 Verb Identification

Hindi Verbs inflect for the following grammatical properties:

- Gender: Masculine, Feminine, Nonspecific
- Number: Singular, Plural, Non-specific
- Person: 1st, 2nd and 3rd
- Tense: Past, Present, Future
- Aspect: Perfective, Completive, Frequentative, Habitual, Durative and Inceptive.
- Modality: Imperative, Probabilities, Subjunctive, Conditional, Abilities, Permissive

A *Verb Group* (VG) primarily comprises main verb and auxiliaries. Constituents like particles, Negation markers, conjunction, *etc*. can also occur within a VG. It is important to know how much of GNPTAM feature information is stored in VG constituents individually and what is the load division in the absence or presence of auxiliaries. In a Hindi VG, when there is no auxiliary present, the complete information load falls on the main verb which carries information for GNPTAM features. In presence of auxiliaries, the load gets shared between the main verb and auxiliary, and is represented in the form of different morphemes. The morphemes attached to a verb along with their corresponding analyses help identify values for GNPTAM features for a given verb form. GNPTAM matrixes having all possible Verb groups are developed. Currently there are 622 unique paradigms in the TAM-GNP matrix. Some resource -Verb identification and linguistic resource help in correct POS tagging in following terms. Extracting Verb group the sentences by following a structured words order.

For example: राम मैदान मे खेलने जाता रहता है ।

In this sentence Verb Group 'खेलने जाता रहता' that contain 3 verbs.

Verb Group contains auxiliary verb and main verb so we should separate auxiliary and main verb. In case of Hindi auxiliary verb come at last and main verb come just before auxiliary verb.

Example In verb group 'खेलने जाता रहता', 'रहता' is auxiliary verb but 'जाता' is two main verbs.

Verbs of Hindi sentence are not present original form (root). It is inflected and its inflection reflected by noun, its number and gender.

5.3.3 Auxiliary Verbs

The identification of main verb is an important step in Verb Group morphological analysis. There are certain verbs like 'मोहन बच्चे पीटा करता है ।', which can occur as main verbs as well as auxiliary verbs. Paradigms help in resolving such ambiguities. For example: in given sentence 'पीटा' is the main verb and 'करता है' is the Auxiliary verb.

5.3.4 Adjective Analysis

Adjective refers the property of noun so it always appear near to that word, for which, it narrate the properties. But in case of ambiguity resolution, adjective always appear just before noun. So, morphological rule being like that 'Adjective + Noun'.

One more analysis for adjective identification is about 'past participle'. Past Participle is a group of more than one verb. So it is seems as verb. But, in actual 'Past Participle' express the property of noun. So past participle is always considered as Adjective not a verb.

Consider example: मोहन ने टूटा हुआ पेड़ देखा ।

Here 'टूटा हुआ 'is past participle and it is verb group of two verbs 'टूटा', 'हुआ'. But Part participle 'टूटा हुआ' explain the property of noun 'पेड़'. So part participle is adjective not a verb.

5.4 Stemmer Design

Stemming is the process of removing the suffix and producing root or stem after appropriate replacement. Morphological stemming identifies all possible root forms of an inflected, or conjugated, word. Stemming which is true to the language, and is required for real meaning extraction in text applications.stemming guided by Morphological

structure of language which is study of rules for forming admissible word. Word are made of morphemes and morphemes are minimal units which have a meaning or grammatical function. For example:

'पढ़ेगा' word formed with following morphemes पढ़,ए,ग and आ here 'पढ़ना' is root word of 'पढ़ेगा' and suffix is 'ऐगा'.

In stemming we remove only inflection morpheme. Mostly stemmer remove only longest suffix and return only the stem. we found out with the help of analysis and morphological based rules ,to return the complete root of the word. Morphological stemming are designed with great care. They are updated quarterly to reflect the latest word usages, using advanced data gathering techniques. So stemmer analyzer is widely used in 'Information Retrieval' system. There are many algorithms, which are being used for stemming. One such algorithm is "Porter's Algorithm", which has been widely used. So it was thought appropriate to use this algorithm for stemming purpose

5.4.1 Procedure of Suffix Evaluation: As previously discuss stemmer used to extract the information like root and suffixes from given inflected word. Following procedure are follows to extrct te suffix information as discuss below.

1-List of all possible suffixes along with their category information .list of possible suffixes were divided in the categories of verb, noun ,Adverb and adjective .All the suffixes and corresponding word to suffix are stored in temporary files known as rule file .

2-Maintaining a list of stop words and then removal of stop words from the given text file or database.

3-Applying the appropriate rule for stemming the words based on the Porter's Algorithm or any other approaches.

4-Getting a final list of stemmed words, which can be used in free text searching for maximum retrieval of relevant information.

5-The replacement to be made after removing of suffixes so that valid root can be formed.

6-Word list for different grammatical categories are required to in stemmer analysis. Word list is used to check whether the input word is inflected and if inflected determine

the all possible suffix, by remove them and find the root word. We also consider that addition and removal of suffix are valid or not.

5.4.2 Limitation: But there is some limitation because Stemmer is unable to produce results for a large number of words because of the following reasons:

1. Word which are not present in word-list (new Word) but they are frequently used so we need to update wordlist by adding new words.

2. Unclean data like user make spelling mistake for example 'कुनक' in place of 'कनक'. But word list have 'कनक'.

3-compound Noun and proper noun have not been handled yet. Stemmer return the root if word is inflected and present in word list. But in case of proper noun and compound noun which are not present in wordlist then how handle. For example: name of entity like 'ताजमहल','अरुन अरोरा'.

5.5 Design of Word Sense Disambiguation:

Word Sense Disambiguation (WSD) is defined as the task of finding the correct sense of a word in a specific context. This is crucial for applications like Machine Translation and Information Extraction.

The main idea is to compare the context of the word in a sentence with the contexts constructed from the Wordnet and chooses the winner. The output of the system is a particular synset number designating the sense of the word. The mentioned Wordnet contexts are built from the semantic relations and glosses, using the Application Programming Interface created around the lexical data.

We describe a statistical technique for assigning senses to words in Hindi. A word is assigned a sense with the use of (i) the context in which it has been mentioned (ii) the information in the Hindi Wordnet and (iii) the overlap between these two pieces of information. The sense with the maximum overlap is the *winner sense.*

5.6 Handling the Unknown Words:

Handling of unknown words is an important issue in POS tagging. For words which have not been seen in dictionary (database) or the training set then The estimation of such of the word, that it is presence dictionary or not. If it is absence in dictionary, it tagged with unknown word. To dealt with unknown words, first we evaluate it is inflected word or present in original form. So suffix identification is tough work in case of unknown word. So our analysis for unknown are depends on whether the word contains a particular suffix or not. For analysis we should prepare list of suffixes. At present we have 435 suffixes; many of them usually appear at the end of verb, noun and adjective words. A null suffix is also kept for those words that have none of the suffixes in the list. The probability distribution of a particular suffix with respect to specific POS tags is generated from all words in the training set that share the same suffix. Apart from suffix analysis, two other features have been included that tackle tokens of digits and symbols. For handle the unknown words following procedure are invoked as discuss.

1-Syntatic Rules (Grammatical order of sentence) rules are used to tag unknown words

2- Change the tag of an unknown word from one tag to other tag and than morphological analysis is applied.

3- If prefix or suffixes are exist than remove it by looking for prefix from table of prefixes and deleting the prefix or suffix.

Chapter 6

Algorithms and Implementation

6.1 Ambiguity Resolution Rules

Hindi is partial fixed order language, means internal structure of 'Word Group' of sentence, follow fixed order of words. But order of 'Word Group' is free order. So, there is more than one combination of 'Word group', may be possible. But in point of internal structure view of 'Word Group', there is only one possible arrangement. So, this is aspect of Hindi language is consider to evaluate correct POS tags and ambiguity resolution in POS tagging. So, we propose some algorithm and programming solution for implementation, most of them are based on 'Word group level'.

We have proposed some rules for grammatical tag evaluation are based on both 'Word Group' level and morphological structure of language. Now, we discuss the some ambiguity resolution rules that are invoked in designed POS tagging system for correct tag evaluation. These rules are following as discussing below.

6.2 Noun Identification Rules

Noun is name of person, place, or things. Some of the rules for noun identification are as follow.

Rule 1: Adjective always give explanation for noun or pronoun in 'word group' adjective should be appear before noun or pronoun. Our clue is that any if word tagged with adjective and just next word has more than one tag, then there is very high chance that word should be tagged with Noun and remaining tag should be discards. So, rule format at group level.

Rule N1: Word (i)/(Adjective) + Word(i+1)/ Noun

For Example: 1- पुरानी किताबों को पढ़कर राधा अपने बिस्तर पर सो गयी । here word group 'पुरानी किताबों' in which word 'पुरानी' as adjective and 'किताबों' as noun. So, rule format is correct as 'Adjective + Noun'

Some more examples are as following.

- 2- मोहन कि हालत <u>गलत दवाइयों</u> से बिगड़ी है।
- <u>एकान्ती लोगों</u> के लिए घर लेना मुसीबत है।
- सैनिक देश के <u>सच्चे सुभचिन्तक</u> है।
- राम <u>बहुत ईमानदारी</u> से काम करता है।

Rule 2: A relative pronoun is a pronoun that marks a relative clause within a larger sentence. It is called a relative pronoun because it relates to the word that it modifies. A relative pronoun links two clauses (statement) into a single complex clause. So, word that appears just relative noun is always noun or 'Adjective+ Noun'. So, rule format is given as.

Rule N2: Word(i)/Relative Pronoun + Word(i+1)/Noun or

Word(i) /Relative Pronoun + Word(i+1)/Adjective+ Word(i+2)/Noun

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

Examples:

- <u>नीरज</u> वह लड़का <u>जो</u> खेलने गया है ।
- ये वो <u>घर</u> है <u>जिसे</u> राजा ने बनवाया था।

Rule 3: Person Pronoun is used to reference the Noun. So, Personal pronoun ('हमारा', 'अपना', 'तुम्हारा', 'उसका', *etc.*) appear just before Noun word or 'Adjective+ Noun' word group. So, generate rules:

Rule N3:

Word(i)/ personal pronoun + Word(i+1)/Noun or

Word(i)/ personal pronoun + Word(i+1)/Adjective+ Word(i+2)/Noun or

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

For Example:

- राजीव अपनी किताब किसी को नही देता।
- हमारा विश्विद्यालय देश मे अग्रणी है।
- उसका पूरा सँभालना मुस्किल है।

Rule 4: Noun always placed after directional word or positional words. Positional word and directional words are 'में', 'पर', 'इधर', 'उधर', 'ऊपर', 'नीचे' *etc*. So, Proposed Rules given as.

Rule N4: Word (i)/Noun + Directional or Positional word +Noun or 'Noun +Adjective' or Verb *

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

For Example:

- राम ने पानी में पत्थर फेका।
- चिड़ियाँ पेड पर बेठती है ।
- पहाड़ियॉ के नीचे नदी बहती है।

Rule 5: Actor of statement and noun are always appearing just before the word 'ने'. Actor is always Noun and pronoun. So, noun always appear just before word 'ने'. So proposed rule format is given as.

Rule N5: Word(i)/ Noun/Pronoun + 'ने'.

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

For Example:

- कुत्तों ने अजनबी को देखकर भोंकना शुरू कर दिया।
- मोहन ने खाना खाया।

- गाय ने दूध दिया।

Rule 6: Some Word that show relationship with noun. These relational words are 'का', 'की', 'के', 'के लिए', 'के साथ', 'से'. Before these words, always noun or pronoun is appear and after them noun or 'Noun + adjective 'are appear.

Rule N6: Word (i)/Noun + relational word + Word(i+2)/ Noun or [Noun+ Adjective] *

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

For Example:

- राजू की किताब चोरी हो गयी।
- गीता सोने के लिये कमरे में गयी है।
- बच्चा अपनी माँ के साथ घूमता है ।

6.3 Adjective Identification Rules

Adjective explains the characteristics or attributes of noun or pronoun. So, there are some proposed rules that are used to resolve ambiguity.

Rule 1: Adjective always show the attributes of noun so it should come just before Noun. So, proposed rule format is given as.

Rule Ad1: Word (i)/ Adjective + Word (i+1)/Noun

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

Example:

- 1-मोहन कि हालत गलत दवाइयों से बिगड़ गयी। in this statement Adjective: 'गलत'and Noun:'दवाइयों'
- पुरानी किताबों को पढ़कर राधा अपने बिस्तर पर सो गयी ।

Rule 2: Adjective or 'Noun +Adjective' are appeared just after some affinity representative word like 'का', 'की', 'के', 'से'. So rule format for this adjective identification is given as.

Rule Ad2: affinity word + Word (i)/ Adjective + Word (i+1)/Noun

Where $i = 1, 2, \ldots, n$.

n is length of sentence in term of word.

Affinitive word: 'का', 'की', 'के', 'से' *etc.*

Example:

- प्रीती <u>की साइकिल</u> पुरानी है।
- मोहित <u>का अधूरा काम</u> पूरा नही है ।
- शिकोहाबाद <u>से आगरा</u> पास है।

Rule 3: Quantitative Adjective give explanations quantity of Noun or Adjective (Quantity of Attributes). So, Noun or 'adjective +noun' always appeared after Quantitative Adjective. Rule format are as follows.

Rule Ad2: Word (i)/Quantitative adjective + Word (i+1)/ Noun or [Adjective +Noun] *

Example: रोहन <u>बहुत ज्यादा दूध</u> पीता है।

* This rules show again ambiguity but it is not actually happen. Because this ambiguity solve by applying noun Rule N1: 'Adjective +Noun'. This rule first of all call (apply) at the time of execution of POS tagger program so that this type ambiguity resolved previously. So, anywhere this type of ambiguity shown means it is solved and this type ambiguity not exist. It is seen only literally.

Rule 4: Past Participle seems as verb (past form) but actually it reveal the properties of noun so past participle always consider as adjective. But when assigned all possible tag to words then only one tag associate with past participle .So, it seems that there is no ambiguity. But assigned tag not correct. So, we should check in sentence that there is any past participle, exist in sentence. Suppose if past participle is exist, an assign adjective tag instead of verb tag.

Rule format as follows.

Rule Ad3: Word (i) / past form of Verb + Word (i+1)/ Noun + Word (i+2)/ verb with suffix 'कर'

For Example:

तुम्हें मरा हुआ साप देखकर र लगता है।

Here मरा हुआ is past participle and describe the property of noun 'साप'. So, 'मरा' हुआ is not a verb, it is an adjective. Some more examples are given below.

- हमने टूटा हुआ जहाज देखा
- अमर ने जला हुआ कागज फेंका ।

6.4 Verb Identification Rules

Verb is essential constituents of every sentence and one sentence may contain more than one verbs. So, words of sentence are tagged with 'Verb' aspect, and grouped them. This is known as 'Verb Group' this verb group has one main verb and remaining are consider auxiliary or helping verb.

Verb present in verb group mostly of them are inflected. So, words in verb group are inflected according to grammatical aspects and associate the suffixes with them.

So, Verb group provide following grammatical information as tense, aspect, gender, case, person.

These values formed according to the list Verb groups according to their TAM-GNP matrix having all possible verb groups, is developed.

The Analysis and linguistic resources help in the following activities.

1- Identify Verb Group in the given sentence: Verb Group follows a structured word order. Using some paradigms developed by morphological analysis, any verb group can be identified in given sentence. For example:

राम खाना खाता रहता है।

Verb group: 'खाता रहता है'

2- Identify main verb and Auxiliary verb: Verb group has only one main verb and remaining are marked as auxiliary or helping verb. Main verb always appear last of

sentence, but just before words 'है', 'हैं', 'था', 'थी',*etc.* After identifying main verb remaining verb of verb group are marked as auxiliary verbs.

3- Identify Suffix of main verb: Verb are inflected according to actor and tense *etc.* so, suffixes that appended in inflected verbs, contain important grammatical information. Stemmer is used to identify the suffixes and root word from inflected verbs.

For example: 'खाता रहता है।'

Main Verb: 'खाता'

Root verb: 'खाना'

Suffix: 'ता '

Grammatical information: Gender of actor: Male, Singular, 3rd person
Tense: Present

6.5 Gender Determination

Hindi language has two types gender feminine and masculine. Words are presented in inflected form in sentences *i.e.* words are associated with some suffixes. Suffix association are always depends on gender and number of subject of actor, tense of statement. So, for identification of number and gender, we follow its just reverse approach (*i.e.* backtracking). For it we should perform stemmer analysis operation which provide us morpheme or original word (root word) and correct suffix. On the basis of suffix we can evaluate the Number and gender of words and actors of sentences with 90 % correct result. But some cases this evaluation is fail to give correct result so we should follow another approach and after that match the result and correct result are produced. Other approaches include verb based, last word (Auxiliary verb based) 'है', 'था', 'थी','हैं', 'होगा', 'हुआ', 'हुए'*etc*. Final decided gender should be based on both verb and word (actor).

For example: 'लड़कियॉं', 'गायकों', 'गायों', 'गधों' it is clear with this examples that all word have same suffix 'ओं' , it is tough and inaccurate works to determine correct gender

of these words. So, Gender identification should not be based on particular word for which we evaluate gender it should be depend on main verb and last word (helping verb *i.e.* 'है', 'था', 'थी', 'हैं'). So, cross validation also required for gender identification, this approaches give all most all correct result.

6.6 Number Determination

Words are presented in sentence inflected form. Inflected form means morpheme or root words are appended with appropriate suffix at tail of words. Suffix association with word are depends on gender and number of noun or actor of sentence. So, we must follow reverse approach to find the number of noun. For which we perform stemmer analysis which provide original word and suffix that associate with words. On the basis of suffix we can evaluate correct number of noun. There four things are follow for number identification.

1- Based on Actor (Noun or Pronoun)
2- Based on Main Verb
3- Based on Last word (Helping verb 'है', 'था', 'थी', 'हैं')
4- Person of Actor also help to decide the Number of Noun or Pronoun

6.6.1 Procedure To find out 'Number' of Noun: Find the correct suffix of given words and then search the corresponding suffix in database which contain number and gender information corresponding to each suffix and last Mantra of inflected word.

Let suppose some word which are not inflected but used in plural sense then main verb based analysis for number identification should be performed. Main verb based analysis means find the suffix of main verb and the again search the result from database.

For Example:

दीसी ने <u>लडके</u> को अपना दिल दिया।

<u>लडके</u> बहुत चालाक होते है।

ये किताबें <u>हैं</u> ।

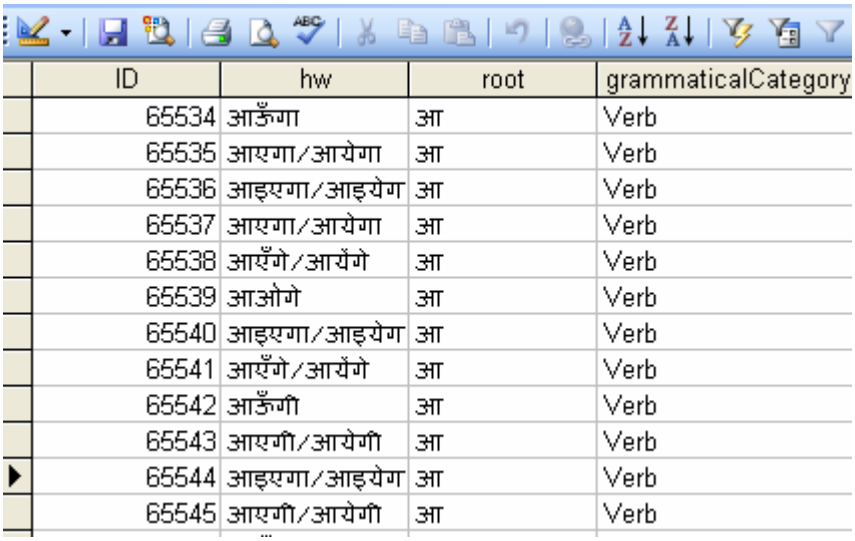
In 1st sentence 'लड़के' is used as singular. But 2nd sentence 'लड़के' used in plural sense.

6.7 Tense Determination

Tense of sentence is determined on the bases of main verb and helping verb (‘हैं’, ‘है’, ‘हूँ’, ‘था’, ‘थी’ *etc*.). Evaluation of tense depends on suffix of main verb. Sentence which does not contain helping verb then tense depends on main verb’s suffix. So, first of all find suffix of main verb and suffix looking up in corresponding stored information in database returned the result of corresponding suffix. Table Structure is shown in figure 6.2.

6.8 Database for Morph Analyzer

Pos tagging tool implementation, is required a dictionary that hold 93273 words. Dictionary contains the words with their all possible grammatical attributes and each word has all possible inflection with all possible suffixes. Table (dictionary) has four fields as following ID, hw (word), root, grammatical Category as given in figure 6.1.



ID	hw	root	grammaticalCategory
65534	आऊँगा	आ	Verb
65535	आएगा/आयेगा	आ	Verb
65536	आइएगा/आइयेग	आ	Verb
65537	आएगा/आयेगा	आ	Verb
65538	आएँगे/आयेंगे	आ	Verb
65539	आओगे	आ	Verb
65540	आइएगा/आइयेग	आ	Verb
65541	आएँगे/आयेंगे	आ	Verb
65542	आऊँगी	आ	Verb
65543	आएगी/आयेगी	आ	Verb
65544	आइएगा/आइयेग	आ	Verb
65545	आएगी/आयेगी	आ	Verb

Figure 6.1 Database for Morph Analyzer in MS-Access

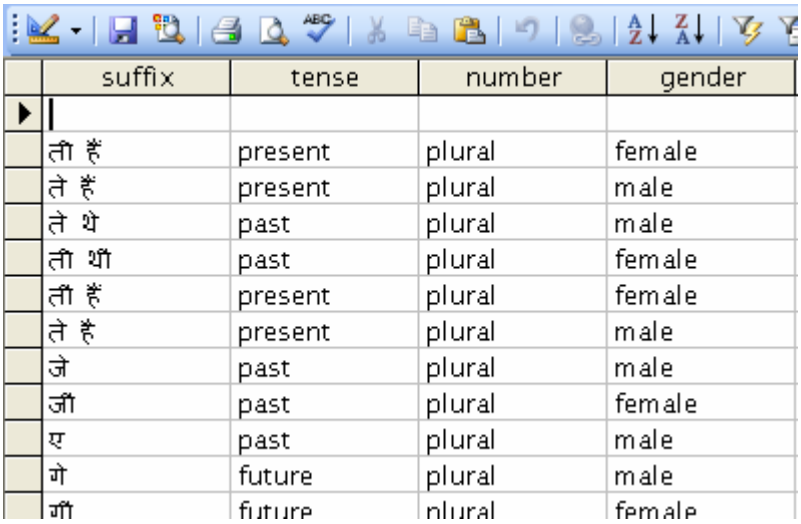
At the time of word’s tagging we search the words in dictionary and their all possible grammatical category. Corresponding root word to given word also required to store in vector or array because root word are used by stemmer analyzer. On the basis of root

word stemmer analyzer determine suffixes from inflected word and these suffixes dominating in morphological analysis and grammatical information extraction. SQL query are embed on java program to extract the information.

This query may return more than one row but have different grammatical category.

6.9 Algorithm for Number, Gender and Tense Identification

Inflected words are created by appending suffix to root (original) word. Suffix are append according to number of noun, gender of noun so its reverse approach are follow to extract these grammatical information. This is done by stemmer analyzer. For identification of gender, number, tense is based on suffix of inflected word. So, we are use a database that contains grammatical information corresponding to each suffix of words. Database structure for suffix analyzer, are given in figure 6.2.



suffix	tense	number	gender
ती हैं	present	plural	female
ते हैं	present	plural	male
ते थे	past	plural	male
ती थीं	past	plural	female
ती हैं	present	plural	female
ते है	present	plural	male
जे	past	plural	male
जीं	past	plural	female
ए	past	plural	male
गे	future	plural	male
गीं	future	plural	female

Figure 6.2 suffix Analyzer database

Algorithm based on suffix analysis *i.e.* find the last 1 to 4 letter of word. Database contains all number and gender related information corresponding to each suffixes and Tense formation are depended on last word (Sp_word category number 'है', 'था', 'थी', 'होगा' *etc.*) and main verb's suffix.

Now, we discuss about algorithm which determine number, gender, tense as given below.

gend_num_tense (n, input sentence)

1- Begin:

If Word[n]=Sp_word and Tag[n-1]=verb Then

suf1= Stemmer_Analyzier (Main verb)

{Main verb=Tag[n-1] and Sp_word= word[n] }

s1= Sp_word+" "+suf1

2- Search s1 or suf1 from database that hold tense information using SQL SELECT command

Tense = ReslutOfQuery()

End if

Else then // if Only main verb and Sp_word are absent

suf1= Suffix_Analyzer (Main verb)

Tense = ReslutOfQuery()

End Else

// For Noun's gender and Number information

3- Num_Gend (String noun)

Begin:

4- Suf2= Stemmer_Analyzer (noun word)

Search suf2 from database that hold tense information using SQL SELECT command

Gender = ReslutOfQuery(1)

Number = ReslutOfQuery()

5- Print Gender, Number

Print Tense

6- End Function

This Algorithm gives tense information along with number and gender information of Actor (कर्ता) of Sentence

6.10 Procedure to assign POS Tag

We divide our problem in two phases for determining correct grammatical tags of chunks of input sentence. In first phase we assign the all possible tag to each respective word. In second phase we resolve the ambiguity in tag *i.e.* evaluate correct tag. For developing

POS tagger, we implement some procedure and function according morphological rule format in java programming language. POS Tagger are developed in java environment using eclipse software and IDE Netbean 5.5 and for dictionary or database are build in MS-Abscess tools. The detail explanation of POS Tagger are given in two phase as follows.

6.10.1 First Phase: Before discussing procedure, describe resource and some assumptions as give below. In this assign all possible tags to each respective word.

A- One two dimensional Array Word_gram[n][n] of String type which is used to find all possible tags. Initially, it is assigned with some null values and its 1st row assigned grammatical category like Noun, Adjective, Verb, Proper Noun, Special_Word *etc*. and its 1st column assigned with all word of sentence in appearing order. But note that its very first cell (Word_gram[0,0]) should be fill with zero or some junk letter.

B- Two String type array (Vector) of same length as number of column in Word_gram[][] array. One of them store root word corresponding to each word and second array hold the all possible tag corresponding to each word respectively.

C- Two more arrays (vector) are required. One of them of integer type that hold number of tag to respective word. If tag number equal to 1 for any word, means there is number ambiguity for this word. But if more than one means there are some type of ambiguity and call ambiguity resolution function. Second vector are of Boolean type which is used for indicate, problem is handle or not.

6.10.1.1 Work Flow in First Phase Now discuss work flow of in first phase of POS Tagger as given below.

1- Provide Hindi sentence as input through GUI developed in Swing. Input sentence give Unicode format and whole sentence (all words) are consider as single string with full stop for termination condition.

2- Tokenize the Unicode (Hindi) input string into tokens *i.e.* in word. Each word of sentence are store in 1st column of Word_gram[i+1,0] and grammatical category like noun, adjective, verb *etc*. are assigned in 1st column Word_gram[0,i+1]. Where i=0, 1, 2, …..,n and n is number of word in input sentence.

3- Next take word one by one from array and search (access) all grammatical information and root word from database. For gaining the information execute following query.

SELECT root, grammatiCategory FROM Dictable WHERE hw ='word';

4- After retrieve all grammatical information, we assign 'YES' in cell for each word corresponding to its grammatical category. For example for ***ith*** word is Noun then Word_gram[i+1,2]="YES" ,where Noun is stored in Word_gram[0][1]=Noun.

Word_gram[n][m] flag[n]

0	Noun	Adjective	Verb	…..	……	……	Sp_word	0
Word1	YES	NULL	NULL	…..	…..	……	NULL	1
Word2	NULL	NULL	NULL	…..	…..	…..	YES	1
Word3	YES	YES	NULL	…..	…..	…..	NULL	2
…..	…..	…..	…..	…..	…..	…..	…..	…
….	…..	…..	…..	…..	…..	…..	…..	…
Word n	YES	YES	YES	…..	…..	…..	NULL	3

Table 6.1 2-D Array for tag assignment

Where n = length of statement in term of number of words

m = number of grammatical properties.

5- Store root information in another vector according to their order of appearance in 2-D array or sentence. Root word list is important of stemmer analysis for suffix extraction.

6- After complete fill up of 2-D array, we consider the set (YES) column value correspond to row of each word Next we set the value of flag array according count for set (YES) value of each row (word). Now we looking for grammatical

tag value for each word. So we assign all grammatical category of each word into another vector, known tag vector. So, tag vector hold all possible tag for each word respectively and flag vector hold the number of tag assigned to corresponding word.

7- If flag vector's value for each word is greater than 1 then ambiguity resolution function invoked to evaluate correct tag.

8- Print the all tag assigned with respective word along with their root word and flag values (number of tags).

Word	Root	Tag	Flag
null	Null	Null	0
Word1	Root1	Noun	1
Word2	Root2	Sp_word	1
Word3	Root3	Noun +Adjective	2
……	……	……	……
……	……	……	……
Word n	Root n	Noun +Adjective +Verb	2

NOTE: Sp_Word is abbreviation of special word. Some words which have no important grammatical category but they are very useful to identify the tag of other word which are appeared near these word. For example 'ने', 'का', 'की', 'में', 'पर', 'के लिए', 'से', 'है', 'था', 'और', 'या', ',' *etc*. They are further categorized and assigned some category number for programming solution. They are very helpful identification of correct tag of near words and responsible to make proper syntax of word order in sentence formation.

6.10.1.2 Algorithm for Evaluating Tags: Algorithm which are invoked to determine the all possible tags of each words of sentence. Algorithm is as given below.

String tagging_fun (String s1)

Begin:

1-Initialize Word_gram[][], root[], tag[] Flag[] by some initial values by NULL , NULL, NULL and 0 respectively.

gram[]={"noun","pronoun","verb","adjective","gender","number","sp_word","proper"};

2- For i =1 to 9 {because we assume nine grammatical aspects}

Begin:

Word_gram[0][i]=gram[i-1]

End For

3- Tokenize the input statement into words using function

StringTokenizer stk=new StringTokenizer(s1," ");

For i = 1 to n

Begin:

s2=Token();

Word_gram[i][0]=s2;

sent[i]=s2;

End For

4- Search word from table for retrieve all possible tag and grammatical values

While i = 1 to n {where n is length of sentence in term of words}

Begin:

execute SQL query for each word

root[I] = result (1)

s = result (2)

ch=s3.charAt(0);

switch(ch)

{

case 'N':

Word_gram[i+1][1]="YES";

```
                    break;
                    case 'P':
                    Word_gram[i+1] [2]="YES";
                    break;
                    case 'V':
                    Word_gram[i+1] [3]="YES";
                    break;
                    case 'A':
                    Word_gram[i+1] [4]="YES";
                    break;
                    case 'm'|'f':
                    Word_gram[i+1] [5]="YES";
                    break;
                    case 'p'|'s' :
                    Word_gram[i+1][6]="YES";
                    break;
                    case '5':
                    Word_gram[i+1][7]="YES";
                    break;
                    End Switch
End of while
```

5- Tag assignment and flag values

```
For I = 1 to n
Begin:
    For J = 1 to m
    Begin:
        If (Word_gram[i+1]=”YES” ) then
        Tag[I] = Tag[I]+ Word_gram[I][J]
        Flag[I]=Flag[I]+1
        End If
     End For
```

End For

6- Print tags corresponding to each word with their root word

7- Return tag[]

*where n is length of statement in term of words.

6.10.2 Second Phase: Second phase's work in POS tagging, ambiguity resolution is started from this phase. This Phase again divide into two sub-phases like ambiguity resolution at word level and ambiguity resolution at phrase level. But we explore phrase level ambiguity resolution.

6.10.2.1 Work Flow on Second Phase: Work flow of second phase is discussed in following steps

1- To check the flag values corresponding to each word if it is grater than 1 means there is ambiguity for this word if 0 means no ambiguity for this word mow forward for next word.

2- Call the ambiguity resolution function according to tags of this word for example if tags are Noun and Verb then at most two functions are called.

3- If ambiguity resolved by any function then it return Boolean value 'true' otherwise return false value means ambiguity resolution function fail so next function is called.

4- If return true means ambiguity resolve and discard all ambiguity function calls by set flag value as 'True' and flag value as 1 corresponding to current word.

5- Repeat steps 1 to 4 for all words.

6- If terminating symbol 'I' is encountered means correct tagging has finished and terminate.

7- Return correct tags corresponding to each words.

6.10.2.2 Proposed Algorithms: This algorithm use results of the previous algorithms as input data and resolve the ambiguity. Exploration of this algorithm is given as.

1- void select_tag()

Begin:

2- Call tagging_fun(String s)

Boolean flag=false;

```
3-      For I = 1 to n
        Begin:
              Flag1=false;
              If flag_v[i]>1 Then
              Tokenize the tag values for respective word
                   Temp[] =StringTokenizer (Tag[i], "+");
                   Flag1=true;
             Else
             Continue
             End if
4-           while Flag1
             Begin:
5-               For  k=0 to p
6-                     Begin:  c=temp[k].charAt(0);
                      switch(c)
                      Begin:
                            default:
                               Flag1=!( and_or(i) );
                            case 'N':
                            Flag1=!Noun_test(i);
                            Break;
                            case 'A':
                            Flag1=!Adjective(i);
                            Break;
                            case 'V':
                            Flag1=!Verb_test();
                            break;
                            case '5':
                      end  switch
             End For
         End while
```

End of for

7- Return (Tag[])

8- End of function

Noun_test(), Verb_test(), Adjective(), and_or() function are based on morphological based analysis. According to analysis some rules are formed for ambiguity resolution. In these function, we have implemented rules that described in previous section of this chapter. This resolution are based on word order or syntactic (grammatical order) of words in sentence.

Chapter 7
Results and Performance of POS Tagger

7.1 Criteria for evaluation of Performance

The performance of POS tagger is evaluated on the basis of following parameters as discuss below.

7.1.1 Complexity: POS tagging task is very complex task. In first step we tokenize the input sentence and then search all possible tags from dictionary with their grammatical information after that ambiguity resolution process are applied, according to type of ambiguity. At last stemmer analysis are done and then WSD processing also perform to evaluate correct sense of ambiguous words. So, correct tag's evaluation is very complex task.

7.1.2 Accuracy: Any system is not ideal, so 100% correctness of POS tagging system is not feasible. In POS tagging system word level accuracy satisfactory but sentence level accuracy is not better.

7.1.3 Processing performance: Processing speed of POS tagger is relatively slow because first we tokenize and then search each word for retrieving grammatical aspects from large database which has approx 93000 entries. In developed system we check ambiguity at each token (word). If we find ambiguity at any token (word), then we call ambiguity resolution procedure to remove ambiguity and determine correct POS tags of this token. So if length of sentence is more then system will take relatively more time in processing.

7.1.4 Robustness: Tagger System is robust system because all ambiguity are handle throw function and function working is based on morphological structure and grammatical rules. So, neither we are used any database for morphological structure determination nor any artificial neural network. So, word order of sentence is achieved according to grammatical rules.

7.1.5 Interoperability and Compatibility: Tagger system is developed in java environment and for native language. Unicode, UTF-8 are encoding standard which

provide support to native language like Hindi, Punjabi, Marathi *etc.* These encoding standards provide code to characters, symbols of native language. So, Java is programming language, which provides compatibility to Unicode, UTF. So, developed system is platform independent. It require only JRE enabled machine for executing developed system.

7.2 Accuracy of the Designed System

Accuracy of Morphological POS tagger is reflected by the strength (depth) of Morphological analysis. If degree of analysis is high, Accuracy of POS tagger will be definitely high. Accuracy of POS tagger is also reflected by correctness of WSD (Word Sense Disambiguation). Because it is also matter that each word has not only correct tag but also have correct meaning (sense). Accuracy of POS tagger are evaluate in three levels

- Correctness of POS tagging at word level.
- Correctness of POS tagging At Word Group level (Phrase Level) *i.e.* how successfully word group are formed and each word of phrase have correct tag with preserving sense of sentence
- Correctness of Stemmer Analysis i.e. how deep level, Stemmer analysis return correct suffix from inflected words and give correct grammatical aspects.

Accuracy further classified into two parts word level accuracy and other sentence level accuracy. Word level Accuracy of POS tagger is defined as ratio of number of tag that correctly assigned to total number of words in per sentence. Sentence level Accuracy of POS tagger is defined as ratio of number of correctly tagged sentence in a paragraph to total number of sentence in paragraph. Word level accuracy is always high but sentence level accuracy is still low and further improvement going on sentence level accuracy.

Word Accuracy = (no. of correct tag in sentence) / (total no. of word in sentence)

Sentence Accuracy = (no. of correctly tagged sentence)/ (Total no. of input sentence)

7.3 Results

POS tagger is developed in IDE NetBeans 5.5 framework and Swing Technology is used for GUI design. We are executing the system by command. Swing based GUI run and ask for Hindi input Sentence. We give the input sentence to GUI. After processing input sentence, first text area show all possible tag and second text area show correct assigned tag of word in sentence as shown in given Figure 7.1 snap shot of GUI.

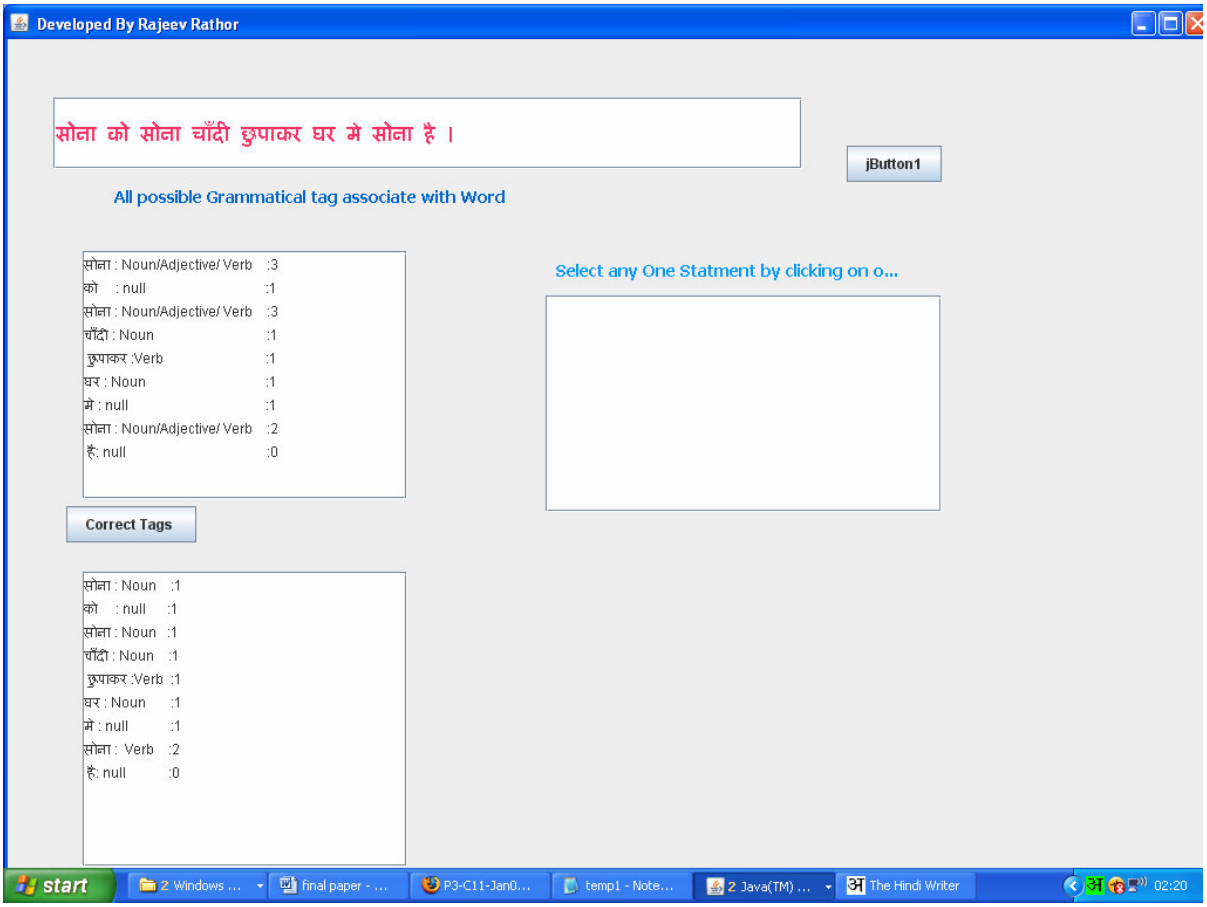


Figure 7.1 snap shot of GUI.

1- Input Sentence: एक लड़की ने लड़के को पेन दिया ।

Output:

WORD:	**TAG**			**ROOT WORD**
एक:	Article			एक
लड़की:	Noun	Gender: Feminine	Number: Singular.	लड़की
ने:	Sp_Word			ने
लड़के:	Noun	Gender: Feminine	Number: Singular	लड़का
को:	Sp_Word			को
पेन:	Noun	Gender: Feminine	Number: Singular	पेन
दिया :	Verb			देना

Tense: Past

2- Input Sentence: पुरानी किताबों को पढ़कर राधा अपने बिस्तर पर सो गयी ।

Output:

WORD:	**TAG**			**ROOT WORD**
पुरानी:	Adjective			पुराना
किताबों:	Noun	Gender: Masculine	Number: Plural	किताब
को:	Sp_Word			को
पढ़कर:	verb			पढ़ना
राधा:	Noun	Gender: Feminine	Number: Singular	राधा
अपने:	Pronoun	Gender: Feminine	Number: Singular	अपना
बिस्तर:	Noun	Gender: Feminine	Number: Singular	बिस्तर
पर:	Sp_Word			पर
सो:	Verb			सोना
गयी:	Verb			गया

Tense: Past

3- Input Sentence: राजीव को खाना खाना है।

Output:

WORD:	**TAG**			**ROOT WORD**
राजीव:	Noun	Gender: Masculine	Number: Singular	राजीव
को:	Sp_Word			को
खाना:	Noun	Gender: Masculine	Number: Singular	खाना
खाना	Verb			खाना
है	Sp_Word			है

Tense: Present

4- Input Sentence: मरा हुआ शेर देखकर बच्चे रोने लगे ।

Output:

WORD:	**TAG**			**ROOT WORD**
मरा:	Adjective			मरना
हुआ:	Adjective			होना
शेर:	Noun	Gender: Masculine	Number: Singular	शेर
देखकर:	Verb			देखना
बच्चे	Noun	Gender: Masculine	Number: Plural	बच्चा
रोने	Verb			रोना
लगे	Verb			लगना

Tense: Past

5- सोना, प्रवीण और प्रताप आम खा चुके हैं ।

Output:

WORD:	**TAG**			**ROOT WORD**
सोना	Noun	Gender: Masculine	Number: Singular	सोना
प्रवीण	Noun	Gender: Masculine	Number: Singular	प्रवीण
और	Sp_word			और
प्रताप	Noun	Gender: Masculine	Number: Singular	प्रताप
आम	Noun	Gender: Masculine	Number: Singular	आम
खा	Verb			खाना
चुके	Verb			चुकना
हैं	Sp_word			है

Chapter 8
Conclusion and Future work

8.1Conclusion

POS tagging is important tool of Natural Language processing. POS tagger is dominating part of NLP application like Language Translator and information extraction. So performance and correctness of NLP application are reflected by the accuracy, correctness and performance of POS Tagger.

In this thesis, we elaborate our work on 'Morphology Based POS Tagger for Hindi Language'. Hindi is morphologically rich Language. This aspect of language is proved as boon for tagging as well curse for tagging because Morphological structure is main reason of ambiguity creation as well as solution of ambiguity resolution. Another aspect with Hindi is also partial free order language *i.e.* internal structure of 'Word group' is fixed order. So, we explore the morphology of Hindi language and words order at word group in sentence for determining correct tag of word.

In designed POS Tagger, we will apply input sentence then proposed algorithm provide all possible tags of each word of input sentence. But some words have more than one tag which means ambiguity is there for some words. For removing ambiguity, we invoke algorithm which is based on morphological rules. This algorithm provides correct tag of each word of input sentence.

Accuracy and efficiency of Hindi morphological POS Tagger is depend on strength of Morphological analysis. So, it is also essential to perform morphological analysis and Stemmer analysis with strength and develop some more rules for defining morphological structure and grammatical structure of sentences with more precisely.

8.2 Future Work

I have explored morphological approaches to design POS Tagger. But, still designed POS Tagger is not perfect (*i.e.* not has 100% accuracy). Word level accuracy is appreciable but sentence level accuracy is not relatively good. So, designed POS Tagger has some

limitation. So, there is need of some further improvement by new phenomena and new approaches.

So, for further improvement in POS Tagger', some future work are remain to implement. Future works on this thesis are as follows.

- Designed system can be tested on tagged corpus to evaluates its accuracy and performance.
- For increase the accuracy of POS Tagger, we should increase the strength of morphological analysis. So, should propose some more morphological rules for analysis. Some proposed morphological rules should be based on syntactic structure of language
- One improvement is that used dictionary should be extendable and update dictionary by some interval of time because new word come in existence. This task will improve accuracy.
- If input sentence have more than
- Let suppose consider worst condition if there are one sentence that have 10 words and 8 words have ambiguity continuously, system is fail to resolve ambiguity. Because it is tough to decide which rules function should be follow at first to handle ambiguity because ambiguity resolution are based on neighbors words with single tag. So, there is need to assign priority to each ambiguity resolution function.
- Priority assignment to ambiguity resolution procedure is based on probabilities of ambiguity's type. Probability of ambiguity type can be evaluated from a corpus of Hindi language.
- In order to extend the accuracy and performance of POS Tagger, hybrid approaches can be adapted as discuss in previous points.

References & Bibliography

[1] Goldsmith, John, 'Unsupervised earning of the Morphology of a Natural Language' *Computational Linguistics, 27:2 (2001), Association of Computational linguistics*.

[2] Manish Srivastava, Bibhuti Mohapatra, Pushpak Bhattacharya, Notion Agrawal and Smriti Singh 'Morphology Based Natural Language Processing tools for Indian Languages.

[4] Aniket Dalal, Kumar Nagaraj, Puspak Bhatacharyya *"Building Feature rich POS tagger for Morphologically Rich Language*

[5] Bushpak Bhatacharya PPT on *"langiuage processing for Indian Processing"*

[6] M.porter "*an algorithm for suffix stripping*" Processing of SIGIR 1980.

[7] D. jurafsky and J.H. martin " speech and language processing"

[8] Pradipta, Harish V., Sudeshna Sarkar "Part of Speech Tagging and Local Word Grouping Techniques"

[9] Smriti Singh, Manish Srivastava, Kuhu Gupta 'Morphological Richness Offsets resource Demand experience in constructing POS tagger for Hindi' .

[10] D. Jurafsky and J. H. Martin, *Speech and Language Processing*.

[11] E. Brill, "A simple rule based part of speech tagger," *Proceedings of the DARPA Speech and Natural Language Workshop*, 1992.

[12] Bharati, A. and Sangal, R. *"Parsing free word order languages in the Paninian*

Framework"

[13] Ratnaparakhi. 1996. "*A Maximum Entropy Part-Of-Speech Tagger "*

[14] Bharati, V. Chaitanya, R. Sangal 1995. *Natural Language Processing: A Paninian Perspective* .Prentice Hall India.

[15] D. Manning and H. Schutze. *"Foundations of Statistical Natural Language Processing"*.

[16] R. Ray , V. Harish, A. Basu and S. Sarkar "*Part of Speech Tagging and Local Word Grouping*
Techniques for Natural Language Parsing in Hindi "

[17] Buckwalter Arabic morphological analyzer version 1.0. Technical Report LDC2002L49, Linguistic Data Consortium. available from: http://www.ldc.upenn.edu/.

[18] Automatic Language-Specific Stemming in Information Retrieval by John A. Goldsmith1, Derrick Higgins2, and Svetlana Soglasnova3

[19] Eric Brill 1992. *A simple rule-based part of speech tagger*, Proceedings of the Third Annual Conference on Applied Natural Language Processing, ACL

[20] Avinesh.PVS, Karthik G IIIT - Hyderabad "Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning"

[21] Thorsten Brants, 2000. TnT – a Statisti-cal Part-of-Speech Tagger. *Proceeding of the sixth conference on Applied Natural Language Processing (2000) pg 224-231*.

[22] Microsoft research on NLP "http://research.microsoft.com/nlp/"

Paper Published

1. Rajeev Rathor, Mr. Parteek Bhatia, "Morphology Based Part of Speech Tagging for Hindi language", Accepted in International Conference on Challenges and Development on IT (ICCDIT-2008) in PCTE Ludhiana (Punjab)
2.