Paradigm based
Hindi Morphological Analyzer

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

Master of Engineering
in
Computer Science and Engineering

By:
Vishal Kumar
(801132034)

Under the supervision of:
Ms. Rupinderdeep Kaur
Lecturer

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2013

Certificate

I hereby certify that the matter which is being presented in the thesis entitled, **" Paradigm based Hindi Morphological Analyzer"**, 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 *Ms. Rupinderdeep Kaur* and refers others researcher's works which are duly listed in the reference section.

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

(Vishal Kumar)

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

(Ms. Rupinderdeep Kaur)
Lecturer
Computer Science and Engineering Department
Thapar University, Patiala

Countersigned by

(Mr. MANINDER SINGH)
Head
Computer Science and Engineering Department,
Thapar University,
Patiala.

(Dr. S.K. MOHAPATRA)
Dean (Academic Affairs)
Thapar University,
Patiala.

i

Acknowledgement

I express my sincere and deep gratitude to my guide Ms. Rupinderdeep Kaur, Lecturer in Computer Science & Engineering Department, for the invaluable guidance, support and encouragement. She provided me all resource and guidance throughout thesis work.

I am thankful to Dr. Maninder Singh, 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 thesis work.

I would also like to express my appreciation to my friends and classmates for helping me in the hour of need and providing me all the help and support for completion of my thesis.

I am deeply indebted to my family for the inspiration and ever encouraging moral support, which enabled me to pursue my studies.

(Vishal Kumar)

801132034

Abstract

The Internet today has to face the complexity of dealing with multilingualism. People speak different languages and the number of natural languages along with their dialects is estimated to be close to 4000. Among the top 100 languages in the world, Hindi occupies the fifth position with the number of speakers being close to 200 million. The information need of this large section of humanity will place its unique demand on the web calling for knowledge processing of Hindi documents on the web. To process Hindi text, the structure of Hindi language needs to be understood. The structure of a language is defined by its morphology.

Morphology is the field of the linguistics that studies the internal structure of the words. It deals with the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes, parts of speech, intonation/stress or implied context. Morphological analyzer is an essential and basic tool for building any language processing application for a natural language. It takes as its input a word and looks up a lexicon and retrieves such information as the root of the word, gender, number, *etc*.

Table of contents

List of Figures

List of Tables

List of Algorithms

Chapter 1: Introduction

1.1 Natural Language Processing

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics. It studies the problems of automated generation and understanding of natural human languages. Natural language generation systems convert information from computer databases into human language, and natural language understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate [14]. NLP has significantly overlapped with the field of computational linguistics, and is often considered a sub-field of artificial intelligence.

A Natural Language is any of the languages naturally used by humans, *i.e.* not an artificial or man-made language such as a programming language. NLP is a convenient description for all attempts to use computers to process natural language. 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 [15]. A complete natural-language processor extracts meaning from language on at least seven levels. Some of important levels are as follows:

Morphological Analysis: 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, for example, its plural and singular forms.

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 common sense 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.

1.1.1 Application of Natural Language Processing

NLP can play a vital role in the following areas:

• Automatic summarization - process of reducing a text file to the summary; that contains the most important points of the original file, using the computer program.

• Part-of-speech tagging - process in which, a word is marked in a text (corpus) as corresponding to the particular part of speech, which is based on both its definition, as well as its context.

• Information extraction - process of automatically extracting structured information from semi-structured and structured machine-readable documents.

• Information retrieval - activity of obtaining the required information resources from a collection of information resources.

• Machine translation - is a sub-field of computational linguistics which is used to translate speech or text from one natural language to another.

• Named entity recognition - subtask of information extraction that locates and classifies basic elements in text into predefined categories like the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, *etc*.

• Natural language generation - NLP task of generating natural language from a machine representation system such as a knowledge base or a logical form.

• Optical Character Recognition - mechanical or electronic conversion of scanned images of handwritten, typed or printed text into machine-encoded text.

• Question answering - concerned with building systems that automatically answer questions posed by humans in a natural language.

• Speech recognition - translation of spoken words into text.

• Spoken dialogue system - dialog system delivered through voice. It contains two essential components: speech recognizer and text-to-speech module.

• Text simplification – modifying an existing human-readable text in such a way that the grammar and structure of the text is simplified.

• Text to speech - converts normal language text into speech.

1.2 Morphology

Morphology refers to the mental system involved in word formation or to the branch of linguistics that deals with words, their internal structure and how they are formed.

Morphology literally means the study of shape. An awareness of morphology begins in early childhood through adolescence. While younger children learn to add an "s" in order to make word plural, elder children may decipher the meaning of words by identifying their common roots with other words. The object of study in morphology is the structure of words and the ways in which their structure reflects their relation to other words. This relation can be within some larger construction such as a sentence and across the total vocabulary of the language. The importance of morphology is in the context of Machine Translation, Information Retrieval, Information Extraction and many such applications [9].

For every language in the world, whether it is written, spoken or signed, morphology is fundamentally involved in both the production of language, as well as its understanding. In order for this process to be effective, the listeners, readers and observers of language must be able to take the inflected word (actresses) and find the underlying root (actor) as well as the set of conveyed syntactic features (feminine, plural). This decoding process is called morphological analysis [22].

In linguistics, morphology is the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes, parts of speech, intonation/stress, or implied context (words in a *lexicon* are the subject matter of *lexicology*) [9].

1.2.1 History of Morphology

The term morphology comes from classical Greek (*morphe*) and it is concerned with structure and arrangement of parts of an object, and how these conform to create a whole object [8]. The objects in question can be physical objects (*e.g.* an organism, an anatomy or ecology) or mental objects (*e.g.* linguistic forms, concepts or systems of ideas).

1.2.2 Morphology in Linguistics

Morphology describes 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 rules [3]. For example, Hindi speakers recognize that the words लड़का [ladakaa] {boy}, लड़कों [ladakon] {boys} and लड़के [ladakae] {boys} are closely related. Hindi speakers recognize these relations from their tacit knowledge of the rules of word-formation in Hindi. They intuit that नगर [nagar] {town} is to नगरों [nagaron] {towns} as पत्र [patra] {letter} is to पत्रों [patrom] {letters}; similarly, नगर [nagar] {town} is to नगरवासी [nagarwaasi] {townspeople} as भारत [bhaarat] {india} is to भारतीय [bhaaratiyaa] {indian}.

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 [3].

1.2.3 Lexemes and Word Forms

The distinction between two senses of "word" is arguably the most important one in morphology. The first sense of word, the one in which नगर [nagar] {town} and नगरों [nagaron] {towns} are the same word, is called lexeme [9]. The second sense, where नगर [nagar] {m} and नगरवासी [nagarwaasi] {townpeople} refer to two different kinds of entities, is called word-form.

1.2.4 Inflectional Vs Derivational morphology

Inflectional morphology is the study of those processes of the word formation where new words with different forms but same meaning are formed from an existing stem. For example in english:

Plurals: bus –> buses, car –> cars

Past tense: die –> died, fill –> filled

Aspect: footstamp –> footstamping, microcode –> microcoding

English has a relatively simple inflectional system but Hindi is a morphologically rich language. For example, inflectional forms of noun लड़का (boy) are लड़के and लड़कों.

Derivational morphology is the study of those processes of the word formation where new words are formed from the existing stems through the addition of morphemes. The meaning of the resultant new word is different from the original word and it often belongs to a different syntactic category [16]. Like in English: go + at = goat (verb to noun)

Adjective to noun: happy –> happiness

Adjective to verb: commercial –> commercialize

Adjective to adjective: green –> greenish

Noun to adjective: success –> successful

Noun to verb: glory –> glorify

Verb to noun: compensate –> compensation

The derivational morphology of Hindi is quite complex. A very common kind of derivation in Hindi is the formation of new nouns, often from verbs or adjectives or the nouns. This process is called nominalization. For example:

महान -> महानता

प्यार -> प्यारा

1.2.5 Paradigms and Morph syntax

A paradigm is the complete set of related word-forms associated with a given lexeme. The familiar example of paradigms is the inflections of nouns. Accordingly, the word-forms of a lexeme may be arranged by classifying them according to shared inflectional categories such as tense, aspect, mood, number, gender or case. For example, the personal pronouns can be organized into the categories of person (first, second and third), number (singular *vs.* plural), gender (masculine, feminine, neuter), and case (subjective, objective, and possessive) [6].

The inflectional categories used to group word-forms into paradigms cannot be chosen arbitrarily; they must be categories that are relevant to stating the syntactic rules of the language. For example, person and number are categories that can be used to define paradigms in Hindi, because Hindi has grammatical agreement rules that require the verb in a sentence to appear in an inflectional form that matches the person and number of the subject. In other words, the syntactic rules of Hindi care about the difference between नगर [nagar] {town} and नगरों [nagaron] {towns}, because the choice between these two forms determines which form of the verb is to be used.

An important difference between inflection and word-formation is that inflected word-forms of lexemes are organized into paradigms, which are defined by the requirements of syntactic rules, whereas the rules of word-formation are not restricted by any corresponding requirements of syntax. Inflection is therefore said to be relevant to syntax, and word-formation is not [9]. The part of morphology that covers the relationship between syntax and morphology is called Morpho syntax, and it concerns itself with inflection and paradigms, but not with word-formation or compounding.

1.2.6 Need to study the Morphology of a Language

It is necessary to study the morphology of a language because of the following reasons:

- Today, Hindi is used by hundreds of millions of people. The information need of this large section of humanity will place its unique demand on the web calling for knowledge processing of Hindi documents on the web.
- The theme of the research is to justify the stand that if morphology is strong and harness-able, then lack of training corpora is not unbearable.
- The specific and primary focus is on *morphology*, and on how knowledge of morphology can be a useful step towards a more complete knowledge of a language's linguistic structure.
- Learners who understand how words are formed, by combining prefixes, suffixes, and roots, tend to have larger vocabularies and better reading comprehension. Morphology can become an instructional tool for all learners.

- Those learners who take unfamiliar words and break them down into smaller parts, or morphemes, have increased success in deciphering unfamiliar vocabulary [10].
- Morphological analysis provides a new light on a vital reading skill.
- Morphological analyzers are using lexicon/thesaurus, keep/stop lists, and indexing engines for their process.

1.2.7 Three Types of Morphology

There are three types of morphology. These are:

- **Concatenative Morphology** – Words are composed of a number of morphemes concatenated together; morphemes include stem plus prefixes and suffixes.
- **Non-concatenative Morphology** – In this type, morphemes are combined in more complex ways [11].
- **Template Morphology** – This is another type of non-concatenative morphology which is found very common in languages such as Arabic, Hebrew and other Semitic languages.

1.2.8 Models of Morphology

There are three principal models of morphology, each try to capture the distinctions described above in different ways. These are:

- Morpheme-based morphology, which makes use of an Item-and-Arrangement approach.
- Lexeme-based morphology, which normally makes use of an Item-and- Process approach.
- Word-based morphology, which normally makes use of a Word-and-Paradigm approach [9].

1.3 Hindi Word Classes

While developing a morphological analyzer, the first step is to define the word classes and grammatical information that will be required for words of these word classes natural language processing application for that language. After defining word classes for Hindi

and grammatical information that is required from words of the word classes, various paradigms for the word classes had been developed. Paradigm for root word gives information about possible word forms of it, in a particular word class and their respective grammatical information [5]. All words of word class may not follow same paradigm. It is not that all nouns will follow same inflectional pattern. So, first task was to find out various paradigms for word class and then group words of that word classes according to those paradigms. In this way, paradigms were developed for word classes which show inflection. To develop the paradigms, inflectional patterns of root words of word class were studied. And then on their basis, root words which inflect in similar way were grouped. Inflection patterns for those groups constitute group of paradigms for that word class. List of word classes is shown below along with their grammatical information being used for Hindi:

Noun: Grammatical knowledge required for Hindi nouns is: gender, number and the case. Gender can be masculine, feminine or both (some nouns can be used in this way). Number can be a singular or a plural. Case can be two types: direct and oblique [5].

Pronoun: Grammatical knowledge required is: number, case, person and gender. Gender can be a masculine, feminine or both. Number can be a singular or a plural. Person can take first, second and third person. Case can be of two types: direct and oblique.

Adjective: Grammatical knowledge required for Hindi adjectives is: gender, number and the case. Gender can be a masculine or a feminine. Number can be a singular or a plural. Case can be of two types: direct and oblique.

Verb: Grammatical knowledge required is gender, number, person and Tense Aspect Modality. Gender can be masculine or feminine. Number can be a singular or a plural. Person can take first, second or third person. Tense Aspect Modality (TAM) can take values related to tense of the verb.

Adverb: There are two classes of adverbs: inflected and uninflected. Inflected adverb behaves like noun which means no separate paradigms are required for these. Grammatical knowledge required for inflected adverbs were same as required for nouns and, for uninflected adverbs, no grammatical information is to be stored.

Sharisthi Pronoun: Grammatical knowledge required is: number, case, person, gender and parsarg. Gender can be a masculine, a feminine or both. Number can be a singular or a plural. Case can be of two types: direct and oblique. Person can take first, second, and third person. Parsarg will be shashthi.

1.3.1 Noun Classification System in Hindi

Traditional classification (the Paninian perspective) of Hindi nouns is based on the gender and stem endings. The system does not allow two nouns of the different genders or different stem endings (consonant or vowel) to be in same class. This results in large number of inflectional classes, around thirty, even if they display same inflectional behavior [17].

Many readjustment rules will be required to explain phonological changes in inflected forms of all the classes. In Table 1.1, nouns are shown with similar inflectional markers and the same gender are put into different classes because of different stem endings.

Table 1.1: Hindi Feminine Nouns taking similar inflections [17]

	रात	माता	बहु	रितु
Plural-dir	राते	माते	बहुए	रितुए
Plural-obl	रातो	माताओ	बहुओ	रितुओ

Using inflection based classification system; it was proposed that the classes should be merged into single class as they will take similar inflections. Similarly, so many classes for the masculine nouns can be merged based on the inflectional behavior. In the same way, inflectional behavior of the nouns can be captured using very small set of affixes and the readjustment rules. All nouns in class display similar inflectional behavior for all the case-number pairs.

1.3.2 Inflection in Hindi Nouns

Hindi nouns show morphological marking only for the number and case. Number can be either a singular or a plural and can be represented as binary valued feature [±pl]. Singular ([-pl]) is default value for the number which is morphologically unexpressed, while non-default value [+pl] may be phonologically realized [17]. Casex marking on the Hindi nouns is of two kinds namely, direct and oblique. Marked nouns (oblique) show cumulative exponence for the case and the number, *e.g.,* े in लड़के *(boy-Obl)* and ो in राजाओ *(kings-Obl)* for singular-oblique and plural-oblique, respectively. Gender (masculine/feminine) is not morphologically marked on the Hindi nouns. However, few nouns represent both genders, *e.g., dost* or मित्र *(friend)*. In Hindi, natural sex distinction in the humans लड़का-लड़की *(boy-girl),* बच्चा-बच्ची *(baby-boy and baby-girl),* in a few animals घोड़ा-घोड़ी *(horse-mare)* and some kinship terms दादा-दादी *(paternal grandpa grandma),* मामा-मामी *(maternal uncle-aunt)* are marked using specific stem endings, *i.e.*, feminine nouns tend to end in vowel ी while masculine nouns tend to end in ा. This is however not case with other nouns that are assigned genders arbitrarily, *e.g.,* पानी *(water)* is masculine and माला *(garland)* is feminine. All nouns thus need to be lexically specified for gender [17].

1.3.2.1 Feminine Nouns

Table 1.2 shows that Hindi feminine nouns of inflection. Type 1 choose *null* for all number-case values. Type 2 nouns take -*या* and –*यो* while Type 3 nouns take े and ो for [+pl, -oblique] and [+pl,+oblique] respectively.

Table 1.2: Type of Inflections for Hindi Feminine Nouns [17]

	Type 1		Type 2		Type 3	
	Direct	Oblique	Direct	Oblique	Direct	Oblique
Singular	*null*	*null*	*Null*	*null*	*null*	*null*
Examples	आग, प्यास	आग, प्यास	नदी, मनी	नदी, मनी	रात, बात	रात, बात
Plural	*null*	*null*	-यां	-यों	े	ो
Examples	आग, प्यास	आग, प्यास	नदियां, मनियां	नदियों, मनियों	राते, बाते	रातो, बातो

1.3.2.2 Masculine Nouns

Table 1.3 below shows inflection for masculine Hindi nouns. These nouns inflect for [+pl, -oblique] and [-pl, +oblique] and take either े or *null*. With [+pl, +oblique] they take *यो, ो* or *null*.

Table 1.3: Types of Inflections for Hindi Masculine Nouns [17]

	Type 1		Type 2		Type 3	
	Direct	Oblique	Direct	Oblique	Direct	Oblique
Singular	*null*	*null*	*Null*	े	*null*	*null*
Examples	क्रोध, प्यार	क्रोध, प्यार	लड़का, साया	लड़के, साये	आदमी, घर	आदमी, घर
Plural	*null*	*null*	े	ो	*null*	े/-यो
Examples	क्रोध, प्यार	क्रोध, प्यार	लड़के, साये	लड़को, सायो	आदमी, घर	आदमियो, घरो

1.3.3 Inflectional Classes for Hindi Nouns

Based on inflection types of masculine and feminine nouns, represented in tables earlier, Hindi nouns are categorized into the five classes: Class A, B, C, D and E. The nouns that are marked *null* for all the case-number pairs are put in Class A, which also includes Type 1 feminine and the Type 1 masculine nouns. The five classes with their inflectional behavior are shown in Table 1.4 below.

Table 1.4: Inflectional Classes and Suffixes for Hindi Nouns [17]

	Class A	Class B	Class C	Class D	Class E
Sg-dir	*null*	*null*	*null*	*null*	*null*
Sg-obl	*null*	*null*	*null*	े	*null*
Pl-dir	*null*	-यें	ें	े	*null*
Pl-obl	*null*	-यों	ों	ों	-यों/ों

- **Class A**: It includes those nouns that can take *null* for all case-numbered values such as *प्यार, क्रोध, भूख, प्यास, मिठास, etc.* Most of these nouns are abstract nouns or the uncountable nouns.
- **Class B**: It includes Type 2 feminine nouns that can take -*या* for the features [+pl, -oblique] and -*यो* for [+pl, +oblique]. These are ी or -या ending nouns such as *लड़की, मनी etc.*
- **Class C**: It includes Type 3 feminine nouns that take े with [+pl] and ो with [+pl, +oblique]. Examples of these nouns include *रात, माला, बहू, रितु etc.*
- **Class D**: It includes Type 2 masculine nouns that ends in ा or -या (except nouns borrowed directly from Sanskrit such as *राजा, पिता, युवा, देवता etc.*) such as *लड़का, धागा, लोहा etc.* A few kinship terms like बेटा, *भतिजा, भांजा, साला* are also a part of this class.

- **Class E**: It includes Type 3 masculine nouns that inflect only for features [+pl, +oblique]. The nouns in the class ends with ु, ू, ि, ी or a consonant. For example: आलू, *साधू, माली, कवी, घर, खेत, etc*. The ा ending tatsam masculine nouns like *राजा, पिता, देवता etc*. also belong to this class.

Chapter 2: Literature Review

2.1 Morphological Analysis

The morphological analysis is the process of providing grammatical information about the word on the basis of properties of the morpheme it contains [11]. It is an integral part of larger language processing projects such as text-to-speech synthesis, information extraction, syllable identification or machine translation.

2.2 Morphological Analyzer

A morphological analyzer is a program for analyzing the morphology of an input word. The analyzer includes a recognition engine, identifies suffixes and finding a stem within the input word algorithms [18]. Traditionally, morphological analyzers are composed of three parts:

- Morpheme lexicon.
- Set of rules governing the spelling and composition of morphologically complex words.
- Decision algorithm to choose from a set of possible analyses.

Morphological analyzer and morphological generator are two essential and basic tools for building any language processing application for a natural language [2]. Morphological analysis means to study the internal structure of the words of a language. A Morphological analyzer gives the morph analysis of a word *i.e.* for a given word a morphological analyzer will return its root word and word class along with other grammatical information depending upon its word class. Like for nouns it will provide gender, number, and case information and for verbs it will provide tense, phase *etc.* Morphological generator does exactly the reverse of it, *i.e.* given a root word and grammatical information it will generate the word form of that root word [1].

Google uses morphological analysis across all its products. Computational linguistic activities in India are being carried out at many institutions. Morph analyzers in Indian languages (Telugu, Hindi, Marathi, Kannada and Punjabi) have been developed. These are freely downloadable. These are developed by Akshara Bharathi Group at Indian

Institute of Technology, Kanpur, India and University of Hyderabad, Hyderabad, India. The morph analyzer for Punjabi language has been developed by Dr. Gurpreet singh Lehal, Mandeep Singh Gill, Dr. S. S.Joshi, Punjabi University, Patiala [1].

2.3 Need of Morphological Analyzer

The first question that requires to be addressed is why morphological analysis is to be performed at all. If there had been an exhaustive lexicon which listed all the word forms of all the roots and along with each word form it listed its features values then clearly there was no need of a morphological analyzer [4]. Given a word, just look it up in the lexicon and retrieve its feature values. For example, suppose an exhaustive lexicon for Hindi contains the entries, given in table 2.1, related to the roots लड़का [laDakaa] {boy} and कपड़ा [kapadaa] {cloth}:

Table 2.1: Roots लड़का [laDakaa] {boy} and कपड़ा [kapadaa] {cloth} in lexicon table [4]

Word Form	Category	Root	Gender	Number	Person	Case
लड़का	Noun	लड़का	Male	Singular	Third	Direct
लड़के	Noun	लड़का	Male	Plural	Third	Direct
लड़के	Noun	लड़का	Male	Singular	Third	Oblique
लड़को	Noun	लड़का	Male	Plural	Third	Oblique
लड़कपन	Noun	लड़कपन	Male	Singular	Third	Any
कपड़ा	Noun	कपड़ा	Male	Singular	Third	Direct
कपड़े	Noun	कपड़ा	Male	Plural	Third	Direct
कपड़े	Noun	कपड़ा	Male	Singular	Third	Oblique
कपड़ों	Noun	कपड़ा	Male	Plural	Third	Oblique
कपड़पन	Noun	कपड़पन	Male	Singular	Third	Any

Now, given a word, it can be looked up and its feature values returned. The above method has several problems. First, it is extremely wasteful of memory space. Every form of the word is listed which contributes to the large number of entries in such a lexicon. Even when two roots follow the same rule, the present system stores the same information redundantly. Second, it does not show relationships among different roots that have similar word forms. Thus, it fails to represent a linguistic generalization. This is necessary if the system is to have the capability of understanding (even guessing) an unknown word. (In fact, human beings routinely deal with word forms they have never heard before when they know the root and the affixes separately). In the generation process, the linguistic knowledge can be used if the system needs to coin a new word.

Third, some languages have a rich and productive morphology. The number of word forms might well be infinite in such a case. Clearly, the above method cannot deal with such languages.

Morphological analysis with different degrees of sophistication can be carried out. Most NLP systems use simple linguistic theories for morphological analysis. The scheme described in this project focuses on the issue of space requirement and also deal with the time issue partially.

2.4 Work done at various institutes in India

The R & D for morphological analyzers for Indian languages was spearheaded by setting up of RCILTS (Resource Center for Indian Languages Technology Solutions) by the Ministry of Communications and Information Technology (MCIT) [19]. The leading resource centers are IIIT Hyderabad, IIT Kanpur, IIT Kharagpur and IIT Mumbai.

- IIIT Hyderabad & IIT Kanpur: Morphological analyzer for Telugu, Hindi, Marathi, Kannada, Sanskrit and Punjabi languages has been developed by these centers.
- IIT Kharagpur: Morphological analyzer for Bangla language has been created at this center.
- IIT Mumbai: Work has been done for Konkani language.

Much work in the area of NLP in India has been carried out is still on at several places and in several languages. Main centers where the works is carried out are:

- National Centre for Software Technology (NCST)
- Indian Statistical Institutes (ISI)
- Thapar Institute of Engineering and Technology
- Utkal University, Anna University
- Chennai, Bhubaneshwar
- A tagged text corpus developed from using the web as source of data in Bengali at Jadavpur University
- Kolkata and University of Hyderabad,
- Indian Institute of Science, (IISc) Bangalore
- Central Electronics Engineering Research Institute (CEERI), Pilani
- Tata Institute of Fundamental Research, Mumbai
- An analyzer being developed for Manipuri
- IBM, India research lab, Microsoft India, Tata Consultancy Services, HP, HCL and Webdunia *etc*.

Work on Sanskrit informatics has been going on at following place

- C-DAC (Pune)
- Special centre for Sanskrit studies, Jawaharlal Nehru University, New Delhi
- Vanashtali Vidyapeeth, Rajasthan
- Rastriya Sanskrit Vidyapeeth Tirupathi
- Lal Bahadur Shastri Rastriya Sanskrit Vidyapeeth, New Delhi.
- Academy of Sanskrit Research, Melkote, Mysore

2.4.1 Morphological Analyzers by Akshara Bharathi Group

Morphological analyzer for Sanskrit, Telugu, Hindi, Marathi, Kannada and Punjabi has been developed by Akshara Bharathi group at Indian Institute of Technology, Kanpur, India and University of Hyderabad, Hyderabad, India (funded by Ministry of Information Technology,

India) and claim for the 95% coverage for Telugu (for arbitrary text in modern standard Telugu) and 88% coverage for Hindi [19].

2.5 Developed Morphological Analyzers for Indian Languages

A lot of morphological analyzers have been developed for Indian languages like Hindi, Punjabi, Sanskrit, Tamil, Telugu, Malayalam, *etc*. Some of the morphological analyzers are discussed below:

2.5.1 Morphological Analyzer for Punjabi

The morphological analyzer for Punjabi language has been developed by Dr. Gurpreet singh Lehal, Mandeep Singh Gill, Dr. S. S.Joshi, Punjabi University, Patiala.

A snapshot of the morph analyzer is shown in figure 2.1 below:

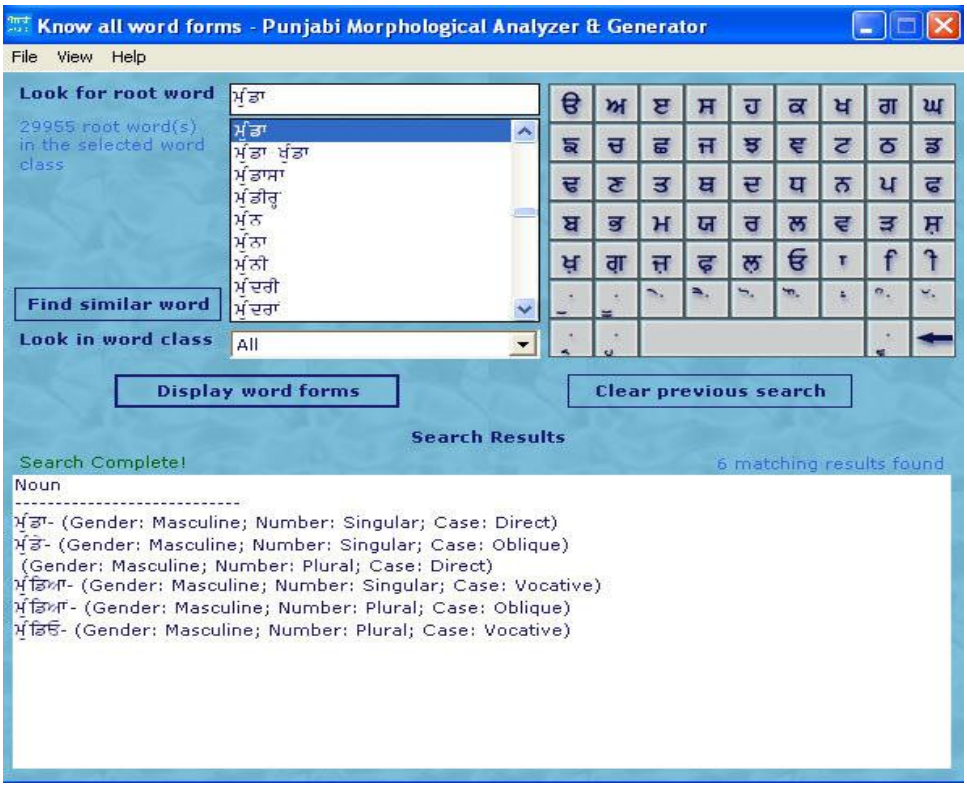


Figure 2.1: Punjabi Morph Analyzer

The database used in the software consists of more than 1.72 lakhs Punjabi words, grouped into 22 word classes such as noun, personal pronoun, reflexive pronoun, verb, inflected and uninflected adverb, inflected and uninflected adjective, conjunction, interjection *etc*. [1].

The tool displays the list of all possible word forms of all Punjabi root words, along with their grammatical information. The grammatical information will be different for different word classes, like for nouns it will contain its gender, number, and case; for verbs, it will give tense, phase, aspect, *etc*. The tool can also identify grammatical attributes of Punjabi words.

Let us give an input word: ਮੁੰਡਾ [mundaa] {boy}

Morphological analysis for this word is:

Noun

Root = ਮੁੰਡਾ

Gender: Masculine

Number: Singular

Case: Direct

2.5.2 Morphological Analyzer for Telugu

It is the rule-based morphological analyzer. In rule based approach, every rule depends on the previous rule. So if one rule fails, it will affect the entire rule that follows. Figure 2.2 shows the interface of the Telugu morphological analyzer.

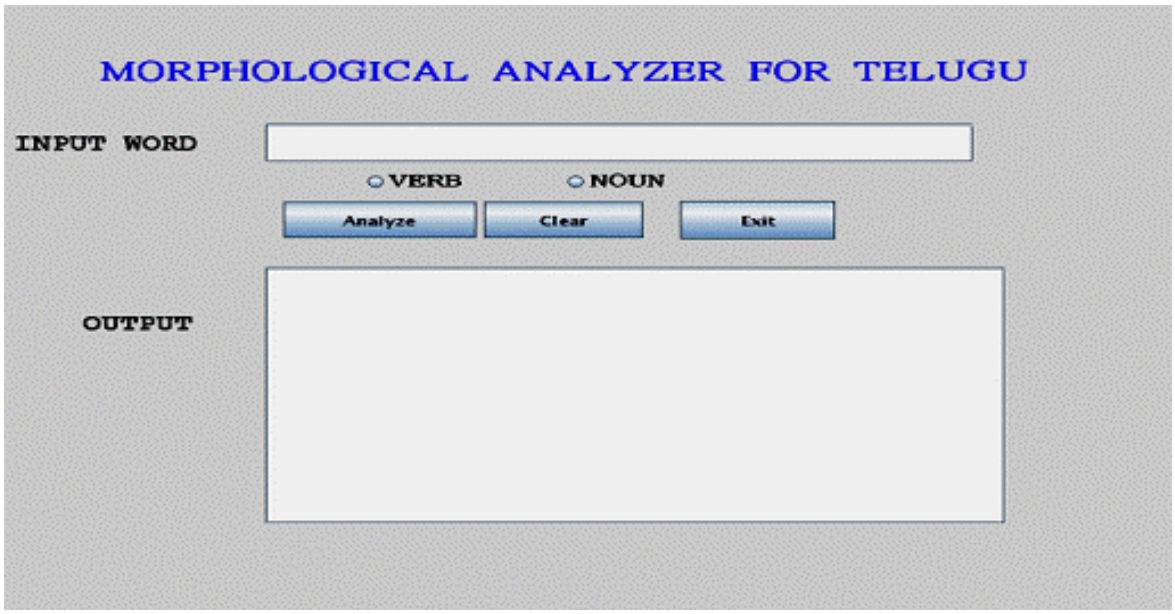


Figure 2.2: Telugu Morphological Analyzer

In Telugu, each inflected word starts with root and is having so many suffixes that point to various inflections, indicating tense, number, person and gender, negatives, imperatives [20]. These suffixes are affixed with each root word to generate word forms.

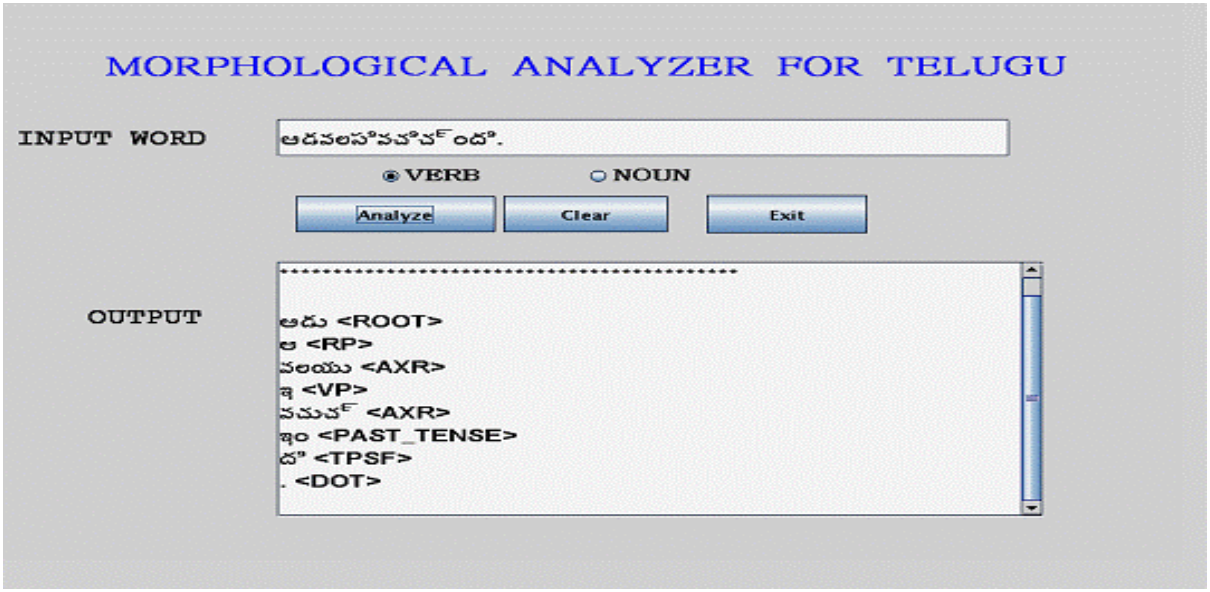


Figure 2.3: Entering a verb to see its details

When details of verb are needed, it is entered in the textbox and verb option is clicked. On pressing the 'Analyze' button, the details will be shown as in figure 2.3.

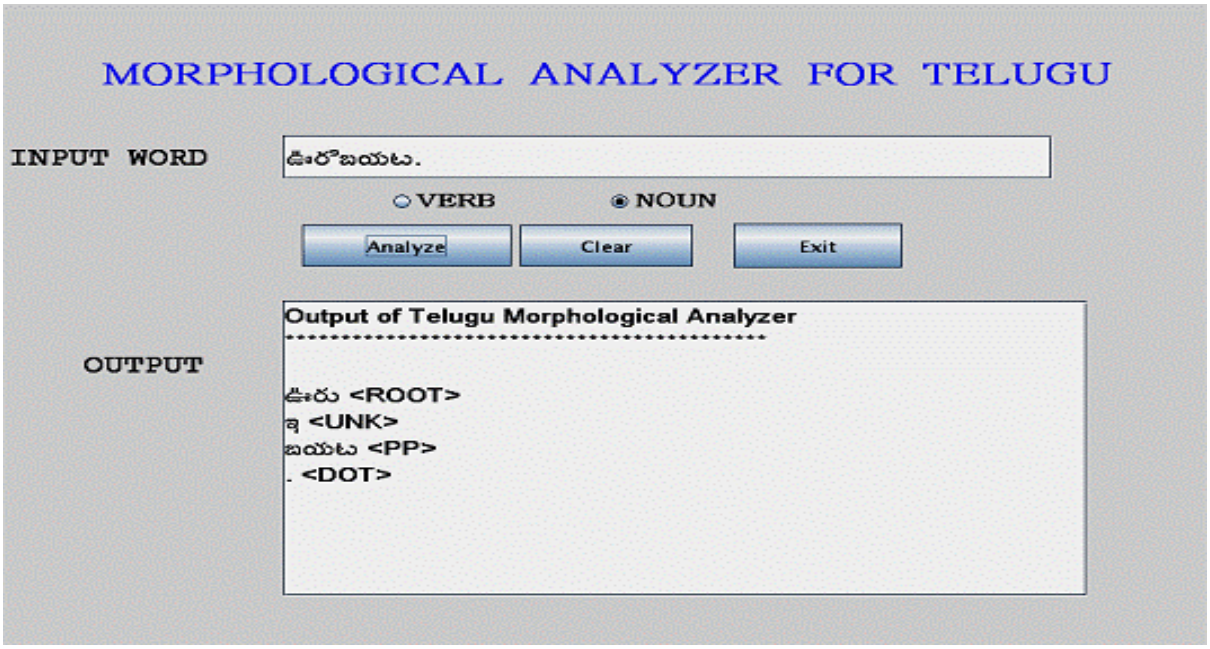


Figure 2.4: Entering a noun to see its details

When details of a noun are needed, it is entered in the textbox and noun option is clicked. On pressing the 'Analyze' button, the details will be shown as in figure 2.4

2.5.3 Morphological Analyzer for Tamil

The Morphological analyzer for Tamil Nouns and Verbs is implemented using Machine Learning approach. In machine learning approach, rules are learned automatically from data. An open source tool named SVMTool for Tamil has been used for this [21]. The morphological analyser fully depends on the automata table in the data file.

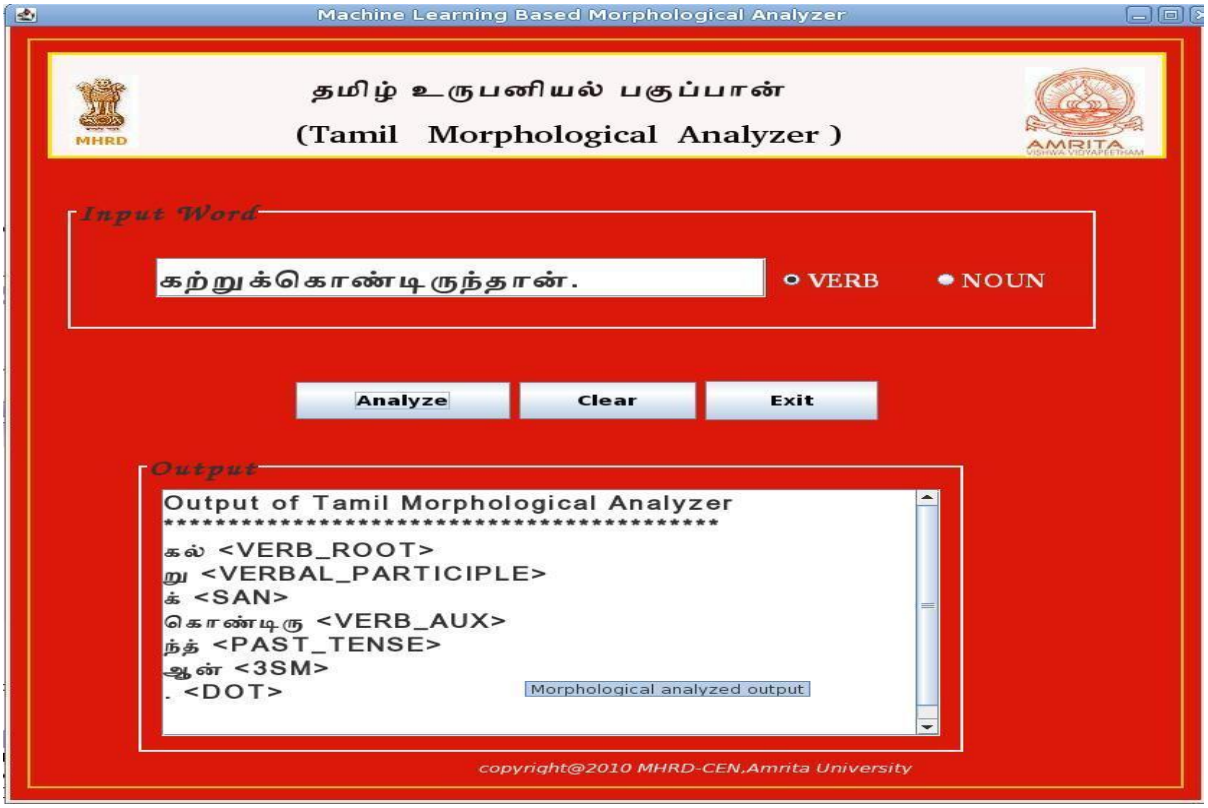


Figure 2.5: Tamil Morphological Analyzer

Figure 2.5 shows the interface of the Tamil Morphological Analyzer.

2.5.4 Morphological Analyzer for Hindi-I

It is a Rule Based Semi-Supervised Morphological Analyzer for Extending the Range of Existing System. Supervised Learning of Morphology(SLM) means that they have access to other sources of knowledge [11]. The SLM approach consists of the available

stem/affix lists with grammatical features, morph tactic rules governing their concatenation, and orthographic rules that change the shape of word constituents. New text is searched in existing morph analyzer to determine the grammatical attributes of the words. Remaining words (could not be categorized in above step) are entered in the new algorithm to know grammatical information of the words.

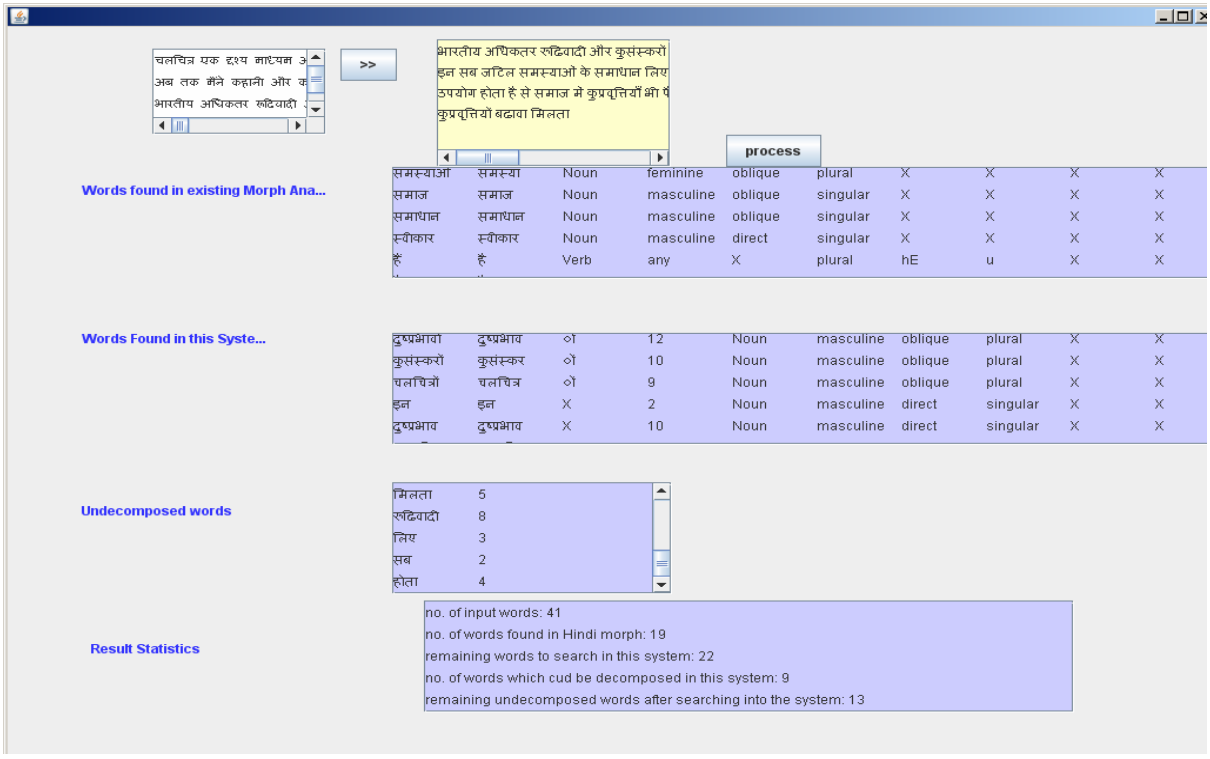


Figure 2.6: Semi-Supervised Hindi Morphological Analyzer

Figure 2.6 shows the interface of the Semi-Supervised Morphological Analyzer for the Hindi language.

2.5.5 Morphological Analyzer for Hindi-II

- It is based on the paradigm approach.
- It is developed at Language Technologies Research Centre, IIIT Hyderabad, India. The morphological analysis helps in the formation of Suffix-Replacement(S-R) rules for Hindi language [5].
- It takes 20,000 nouns which are classified in 20 paradigms.
- No interface available.

The system had been designed in Linux version. The Perl language was used to design the system and encoding was WX which is very uncommon. So, migration from Linux/Perl Based version to Windows based has been done with Graphical User Interface using ASP.NET (C# language).

2.6 Morphological Analysis for Hindi

Hindi has grammatical agreement rules that require the verb in a sentence to appear in an inflectional form that matches the person and number of the subject.

Consider a Hindi word, सुनेगा [sunega] {listen}

Morphological analysis for this word is:

Input word= सुनेगा

Category=Verb

Root= सुन

Suffix= ेगा

Person=3rd person, presence of ए

Tense=Future, presence of ग

Gender=Male, presence of ा

Root specifies the origin of the word, using which other words can also be formed. In the above example, origin of the word *i.e.* root is सुन and the suffix ेगा is used to create the word form सुनेगा.

2.6.1 Morphological Structure of Hindi

In Hindi, *Nouns* inflect for number and case. To capture their morphological variations, they can be categorized into various paradigms based on their vowel ending, gender, number and case information. A paradigm systematically arranges and identifies the uninflected forms of the words that share similar inflectional patterns. Looking at the morphological patterns of the words in a paradigm, suffix replacement rules have been

developed [11]. These rules help in separating out a valid suffix from an inflected word to output the correct stem and consequently, get the correct root. For example, लड़की inflect for feminine (Gender), direct (Case), singular (Number).

Hindi Adjectives may be inflected or uninflected.

Hindi Verbs inflect for the following grammatical properties (GNPTAM):

1. Gender: Masculine, Feminine, Nonspecific
2. Number: Singular, Plural, Non-specific
3. Person: First, Second, Third
4. Tense: Past, Present, Future
5. Aspect: Perfective, Completive, Frequentative, Habitual, Durative, Inceptive.
6. Modality: Imperative, Probabilitive, Subjunctive, Conditionalic, Abilitive.

The morphemes attached to a verb along with their corresponding analyses help identify values for GNPTAM features for a given verb form. For example, करेगी inflect for feminine (Gender), singular (Number), future (Tense), 2nd person (Person).

2.6.2 Features of Existing Morph Analyzer

Existing morph analyzer takes 20,000 nouns which are classified in 20 paradigms. It is developed at Language Technologies Research Centre, IIIT Hyderabad, India. The morphological analysis helps in the formation of Suffix-Replacement(S-R) rules for Hindi language [5]. This helps in reducing a lot of ambiguities in the process of stemming.

For example, take the word नदी [nadii] {river} and मछली [machhli] {fish}

Root: नदी

Plural direct form: नदियाँ

Suffix: इयाँ

Paradigm: नदी

Suffix-Replacement: इयाँ/ई

And,

Root: मछली

Plural direct form: मछलीयाँ

Suffix: इयाँ

Paradigm: मछली

Suffix-Replacement: ईयाँ/ई

Therefore, if a word ends with the suffix इयाँ [iiyaan] then the suffix इयाँ [iiyaan] is replaced with ई [ii] and matches with the appropriate paradigm. Thus, the analysis using the paradigms helps in increasing the accuracy by returning only the correct root. The paradigm analysis is also used by Morphological analyzer to correctly analyze suffixes.

2.6.3 Categorization of Words

Words are categorized into many categories such as adjectives, nouns and verbs. These categories can be further classified on the basis of gender, number, *etc*. The classification of adjective, noun and verb is shown below:

Adjective

Gender

- Masculine (M)
- Feminine (F)

Number

- Singular
- Plural

Noun (N)

Gender

- Masculine (M, *e.g.* लडका [ladakaa] {boy})
- Feminine (F, *e.g.* लड़की[ladakii] {girl})

Number

- Nouns that always come in the singular form and agree only with the verb with singular attribute.
- Nouns that always come in the plural form and agree only with the verb with plural attribute.

Vowel Nouns endings:

- For nouns ending with 'अ'[a]
- For nouns ending with 'आ' [aa]
- For nouns ending with small 'इ' [i]
- For nouns ending with 'ई' [ii]
- For nouns ending with 'ऊ' [oo]

Special Nouns endings:

- Nouns ending with 'ओ' [ao]
- Nouns ending with 'अँ'[ann]
- Nouns ending with 'औ' [aoo]
- Nouns ending with 'इया' [iyaa]

Verb

Gender

- Masculine (M, *e.g.*, खाना [khanaa]{to eat})
- Feminine (F, *e.g.*, सहायता करना [sahaayaataa karnaa]{to help})

Number

- Singular (*e.g.* चलेगा[chalegaa]{*e.g.* he will go with me.})
- Plural (*e.g.* चलेंगें [chalengae]{*e.g.* we all will go together.})

Verbs endings:

- For verbs ending with 'ता': [चलता [chalataa]{*e.g.* he walks}]
- For verbs ending with 'ना ': [दौडना[dorhnaa]{*e.g.* he started running.}]
- For verbs ending with small 'या ':[खाया[khaayaa]{ *e.g.* he ate food.}]
- For verbs ending with 'कर' [पढ़कर [padhkar]{ *e.g.* he slept after studying.}]
- For verbs ending with 'आ' :[भूला [bhoolaa]{ *e.g.* he forgot about his appointment.}]

2.7 Morphological Analysis using Paradigms

The important question is how the paradigms (which are specified for generation) can be used for analysis. Analysis and generation are the inverse of each other. Human experts find it easier to specify solution to the generation problem. It is the task of the computational linguist (one whose primary back ground is in computer science) to solve the indirect or the inverse problem. Solution of the indirect problem requires some amount of search. There are other instances of similar inverse problem in other domains. Humans find it easier to specify solution to one of the problems, call it the direct problem. For example, how to multiply two integers is a direct problem whose solution is neatly provided. The indirect problem, namely division, between two integers, requires some amount of search, using multiplication (the solution to the direct problem). A moment's thought would reveal that the most commonly used division algorithm for decimal numbers, actually involves a trial and error (search) at each step, to obtain a single digit which is part of the answer [4]. Other examples of inverse problem pairs are tying a knot and untying it, climbing a ladder up and climbing down, differentiation and integration, encryption and decryption, *etc*.

Now, outline a method based on search of paradigm tables for doing morphological analysis. Suppose for example, the word कपड़ों [kapdon] {clothes} is given and are asked

to find its root and feature values. Assume further that there are only two paradigm tables for लड़का [ladakaa] {boy} paradigm and भाषा [bhashaa] {language} paradigm.

The first step is to see whether कपड़ौं [kapdon] {clothes} occurs as an indeclinable word, *i.e*. the word having no inflected form. This check would be performed on a dictionary of indeclinable words, which should be available separately.

The second step is to check all the entries in all the paradigm tables having the last character 'अं' in the suffix. There is no such entry.

The third step is to check all the entries in all the tables that have 'औं' as the suffix string.

For each of the entries, add as many characters as shown from the root of the paradigm table. So, there are two roots कपड़ा [kapadaa] {cloth} and कपड़ [kapad] respectively.

These can now be checked in the dictionary of roots. The former occurs in the dictionary. It is now checked whether it has the same paradigm in whose table its suffix has matched. As the check turns out to be true, कपड़ा [kapadaa] {cloth} is identified as the root. Grammatical features associated with 'कपड़ा' [kapadaa] {cloth} in the dictionary of roots and with suffix 'औं' in the paradigm table together constitute an answer (or a lexical entry) for कपड़ौं [kapdon] {clothes}. The system continues searching for additional suffixes such as 'औं ','ड़ौं' *etc*. In case additional answers are found (none in this example) they would also be returned.

2.8 Motivation behind Migration

Following are the reasons why the migration from Linux/Perl Based version to Windows based version was done.

- There was no GUI for analyzer.
- Encoding used in the system is WX which is very uncommon.
- It is based on traditional file based system database concept.
- It follows Data Dependence Approach.

• Even the mapping of devanagri words is different for some characters *e.g.*

- w to त
- W to थ

is difficult for a layman to operate [5].

• Data Files (.p files, Ca, Ce, root) are very much dependent upon each other's format.

2.9 Comparison among the models of morphology

There are three principal models of morphology. These are:

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

2.9.1 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 नगरो [nagaron] {towns}, we say that नगर [nagar] {town} is the root, and that ओ [om] 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 [9].

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.

2.9.2 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 [9].

- 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.

2.9.3 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 [9]. Words can be categorized based on the pattern they fit into. 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, such as *older* replacing *elder* (where *older* follows the normal pattern of adjectival superlatives).

Chapter 3: Problem Statement

3.1 Objectives

In this thesis, following objectives have been proposed for the development of paradigm based Hindi morphological analyzer:

1. To study various morphological analyzers developed or proposed for Indian languages.
2. To study the existing algorithms and design a new one to create paradigm based Hindi morphological analyzer.
3. To create a tool that will prepare a dictionary of Hindi words.
4. To analyze various paradigm classes with the help of which, inflection patterns of the words can be known.
5. To design a simple and easy to use interface of the morphological analyzer for Hindi language using paradigm approach.

3.2 Methodology

To achieve the objectives discussed in 3.1, a step by step methodology has been followed. The detail of this is given below:

1. Study of various morphological analyzers has been carried out.
2. Analysis of all the Unicode characters of Hindi language has been carried out for creating the dictionary.
3. Dictionary creation tool has been created to get the list of Hindi words.
4. Analysis of Hindi nouns has been carried out to know about the classes that these follow.
5. Algorithm has been designed to create the roots using the word forms list and paradigm classes.
6. Interface has been created to perform the morphological analysis using the paradigm approach.

Chapter 4: Design and Implementation of Paradigm Based Hindi Morphological Analyzer

This morphological analyzer is a Stand-alone application designed in ASP.NET platform with Visual Studio as front end and SQL Server as the back end. This chapter discusses about the features, architecture, design and implementation of the morphological analyzer. There are three modules in this: dictionary creation, root table creation and morphological analysis. The working of the morphological analysis is explained in the last section of this chapter with all the screenshots.

4.1 Features of developed Morphological Analyzer

The main features of Morphological Analyzer are:

1. It is developed for Microsoft Windows Operating System using Visual Studio as front end and SQL Server as back end. Microsoft Visual Studio is an integrated development environment (IDE) which supports different programming languages such as C, C++, VB.NET, C#, *etc*. Language used in the developed morphological analyzer is C#. Microsoft SQL Server is a relational database management system.
2. It supports Unicode Format. Unicode is a standard used for the encoding and representing various languages.
3. It provides Graphical User Interface which is easy to operate. The controls used in the application are easily understandable.
4. There is no complex Programming required. Any programmer can easily understand the coding used.
5. It contains a dictionary of around 1 lakh words.

4.2 Architecture of Morphological Analyzer

The system is divided into three modules: tokenization, root table creation and morphological analysis.

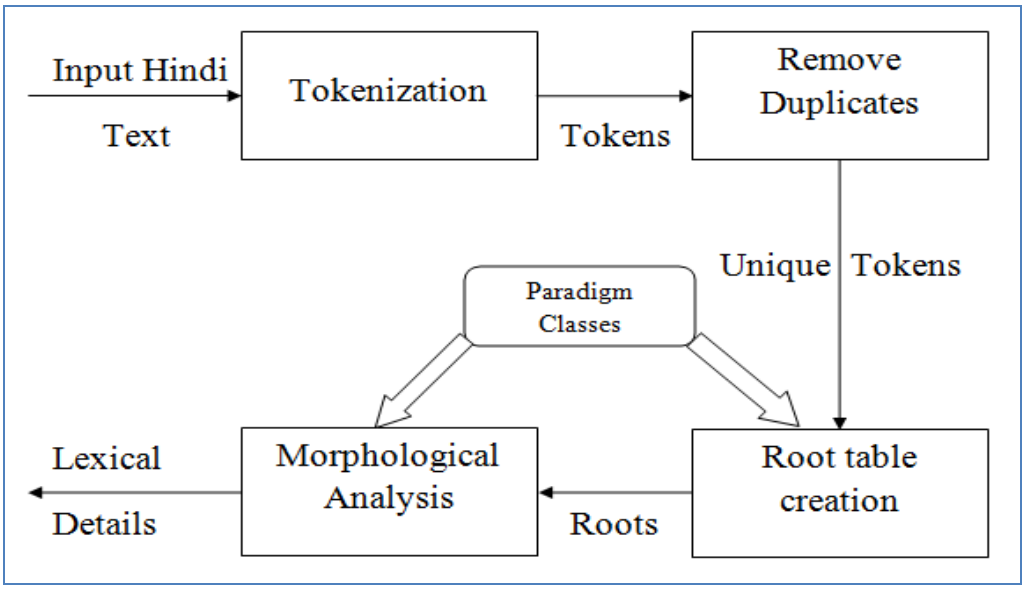


Figure 4.1: Architecture of the system

First module converts a sentence into word level tokens consisting of words without punctuation marks and other symbols. It makes the list of words and also removes the duplicate ones. It takes as input; a text file and gives as output; a file consisting of list of words. This will be the word form database to be used in the project. Second module is the creation of a table consisting of roots along with the classes that they follow. The third module analyzes each word in the input sentence and gives its morphological details as output.

Figure 4.1 shows the architecture of the Morphological analyzer. First of all, input text is tokenized by removing the punctuation marks and other symbols and then duplicate tokens are removed from the output file. These tokens help in the creation of root tables which further helps in analyzing the morphology of words.

4.3 Database Design

This Morphological Analyzer follows database driven approach. There are two tables used in the application: root table and paradigm table. Details of tables in database schema are described on the next page using tables 4.1, 4.2, 4.3, and 4.4.

- Paradigm table stored all the paradigm classes found during the noun classification.

Table 4.1: Paradigm table schema

Column Name	Data Type
par_nam	nchar(10)
sg_dir_count	int
sg_dir_nam	nchar(10)
sg_obl_count	int
sg_obl_nam	nchar(10)
pl_dir_count	int
pl_dir_nam	nchar(10)
pl_obl_count	int
pl_obl_nam	nchar(10)

Description of columns:

1. par_nam: name of paradigm
2. sg_dir_count: numbers of characters to be deleted [direct, singular].
3. sg_dir_nam: suffix to be added [direct, singular].
4. sg_obl_count: numbers of characters to be deleted [oblique, singular].
5. sg_obl_nam: suffix to be added [oblique, singular].
6. pl_dir_count: numbers of characters to be deleted [direct, plural].
7. pl_dir_nam: suffix to be added [direct, plural].
8. pl_obl_count: numbers of characters to be deleted [oblique, plural].
9. pl_obl_nam: suffix to be added [oblique, plural].

Table 4.2: Paradigm table

par_nam	sq_dir_count	sq_dir_nam	sq_obl_count	sq_obl_nam	pl_dir_count	pl_dir_nam	pl_obl_count	pl_obl_nam
प्यास	0	NULL	0	NULL	0	NULL	0	NULL
नदी	0	NULL	0	NULL	1	ियाँ	1	ियों
रात	0	NULL	0	NULL	0	ें	0	ों
लड़का	0	NULL	1	े	1	े	1	ों
आलू	0	NULL	0	NULL	0	NULL	1	ुओं
आदमी	0	NULL	0	NULL	0	NULL	1	ियों

- Root table stores the information about all the roots.

Table 4.3: Root table schema

Column Name	Data Type
root_nam	nchar(10)
par_nam	nchar(10)

Description of columns:

1. root_nam: name of the root

2. par_nam: name of the paradigm class that root follow

Table 4.4: Root table

root_nam	par_nam	root_nam	par_nam
पुस्तक	रात	बेटा	लड़का
कपड़ा	लड़का	विद्यार्थी	आदमी
बच्चा	लड़का	नदी	नदी
बकरी	नदी	साथी	आदमी
बकरा	लड़का	रात	रात
पक्षी	आदमी	कमरा	लड़का
पत्ता	लड़का	स्त्री	नदी
स्वामी	आदमी	आलू	आलू
लड़का	लड़का	दुकान	रात
आदमी	आदमी	औरत	रात

4.4 Dictionary Generation Tool

It helps to create a database for this project. Around 1 lakh words have been created using this module which will act as a database for the morphological analyzer. Input to this module is a large Hindi text.

Figure 4.2 shows the flowchart that depicts the working of the module.

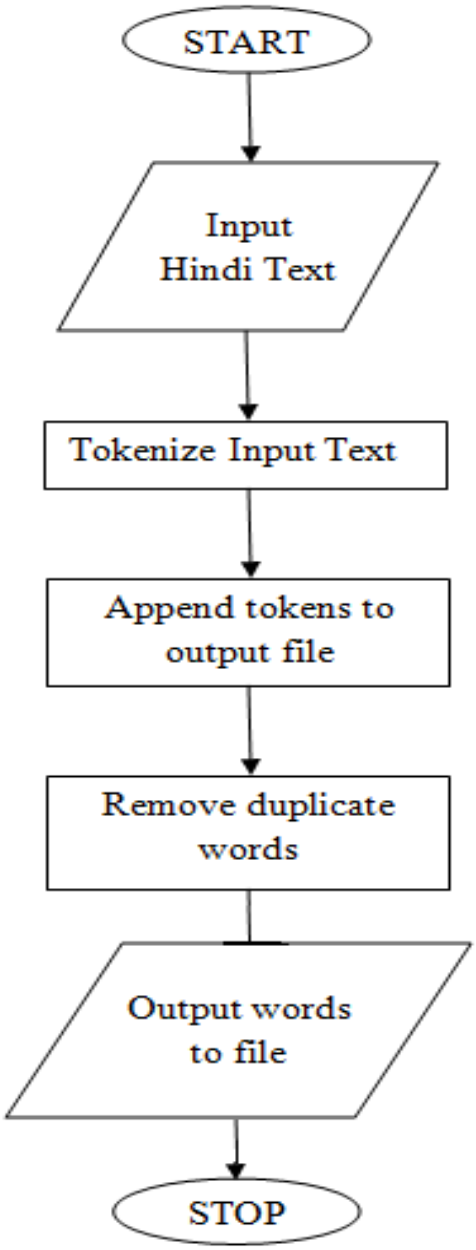


Figure 4.2: Flowchart of dictionary generation

In this, input is passed as a file containing Hindi strings. This file is tokenized and all the punctuation marks and other symbols are removed. The generated tokens are appended into the output file. Then, the duplicate tokens are removed from the output file by creating a hashset. A hashset holds a set of objects and helps in easily determine whether an object is present in the set or not. It does this by internally managing an array of words and storing the objects by assigning an index to them; which is calculated using the

hashcode of the object [13]. This results in the collection of unique words. The words are stored in the output file. In this way, dictionary generation tool works.

Figure 4.3 shows the screenshot of the application program. In this, the input text is given as a file and also the location of the output file is specified.

Figure 4.3: Snapshot of first screen of the tool

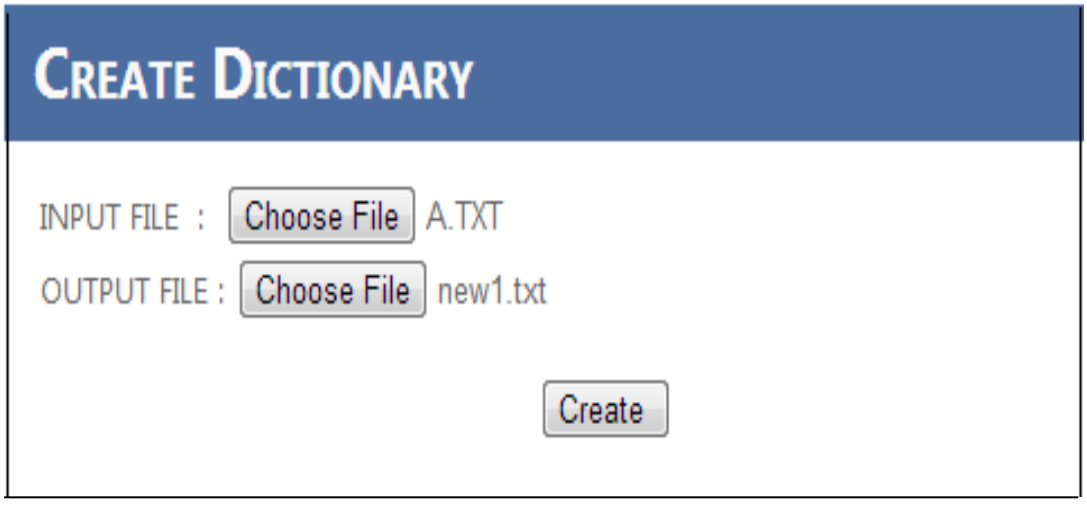


Figure 4.4: Snapshot of the tool after specifying file names

File paths are entered as shown in figure 4.4.

In this, the input file is A.TXT which contains a collection of sentences shown on the next page:

दिल्ली सरकार के उच्चपदस्थ अधिकारियों ने राइट्स के इस सर्वे की रिपोर्ट की जानकारी देते हुए यह भी कहा कि गुड़गांव और मानेसर जैसे शहर गाजियाबाद से कहीं बेहतर हैं। वजह यह है कि दिल्ली से गुड़गांव जाने वालों की तादाद ज्यादा है जबकि गाजियाबाद के मामले में वहां से दिल्ली आने वाले लोग बहुत ज्यादा हैं। बाहरी रिंग रोड पर ट्रैफिक के बढ़ते बोझ को कम करने के लिए भविष्य के विकल्प तलाशने के लिए तैयार की गई इस रिपोर्ट में कहा गया है कि यदि दक्षिण दिल्ली में सड़कों पर वाहनों की रेलमपेल मची हुई है, तो इसकी एक बड़ी वजह पड़ोस के गुड़गांव व मानेसर में हो रही अप्रत्याशित प्रगति भी है।

Figure 4.5: Input file (A.TXT)

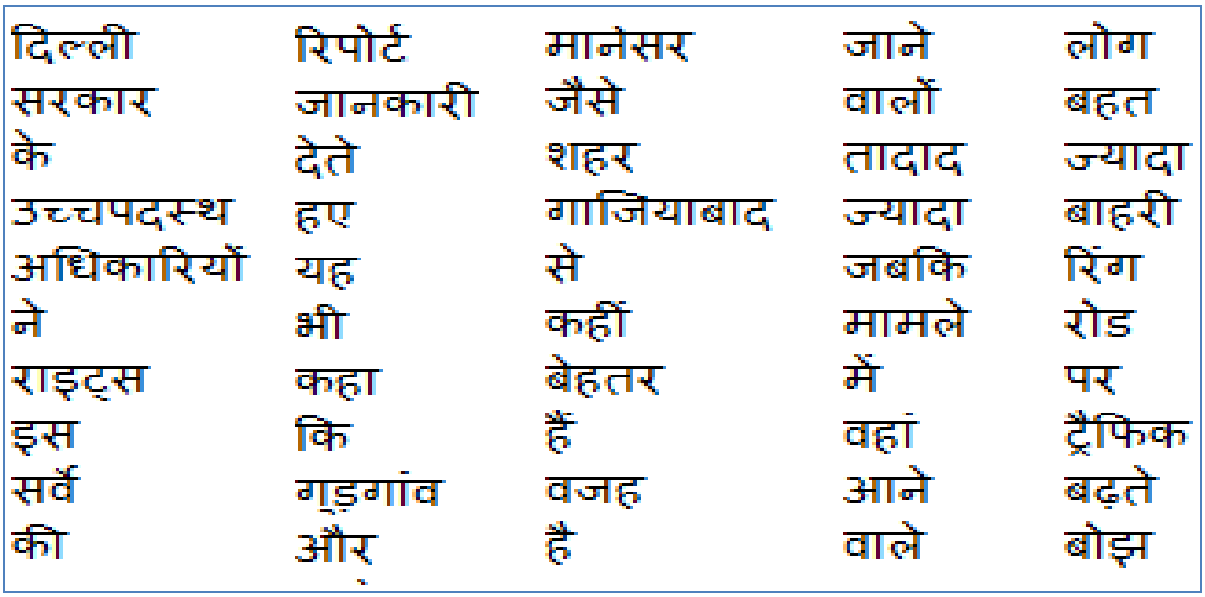
By pressing the create button, a text file (middle file) is created which contains all the hindi words in the input text as shown in figure 4.6:

दिल्ली	की	गुड़गांव	वजह	तादाद
सरकार	रिपोर्ट	और	यह	ज्यादा
के	की	मानेसर	है	है
उच्चपदस्थ	जानकारी	जैसे	कि	जबकि
अधिकारियों	देते	शहर	दिल्ली	गाजियाबाद
ने	हुए	गाजियाबाद	से	के
राइट्स	यह	से	गुड़गांव	मामले
के	भी	कहीं	जाने	में
इस	कहा	बेहतर	वालों	वहां
सर्वे	कि	हैं	की	से

Figure 4.6: Middle file

The middle file shown in figure 4.6 contains duplicate words too. Final output of dictionary generation tool is produced by removing the duplicates from the middle file. The final file contains the unique tokens of input file.

Figure 4.7 shows the output file containing the tokens generated from the input file.



दिल्ली	रिपोर्ट	मानेसर	जाने	लोग
सरकार	जानकारी	जैसे	वालों	बहत
के	देते	शहर	तादाद	ज्यादा
उच्चपदस्थ	हए	गाजियाबाद	ज्यादा	बाहरी
अधिकारियों	यह	से	जबकि	रिंग
ने	भी	कहीं	मामले	रोड
राइट्स	कहा	बेहतर	में	पर
इस	कि	है	वहां	ट्रैफिक
सर्वे	गुड़गांव	वजह	आने	बढ़ते
की	और	है	वाले	बोझ

Figure 4.7: Output file (new1.txt)

This file contains all the unique hindi words from A.TXT file.

4.5 Root table creation

It helps to create the root tables for the morphological analyzer. These tables are created with the help of word form list created in the previous module and the inflectional rules. Using these tables, we get a list of roots along with their paradigm name that they follow. For example, कपड़ा follow the लड़का paradigm which means कपड़ा has the same inflectional rules as लड़का. Once we get root tables, the morphological analyzer becomes very easy and fast.

Algorithm 4.1 show the steps required for root table creation.

In this, input is passed as a file containing Hindi tokens. For each token, the suffix is calculated with length of suffix ranging from 0 to the maximum length of the word. This suffix is matched with each of the suffixes of all the paradigm classes. Whenever a match is found with the suffix of a class, a new root is created using it. Using this root and all the four suffixes of the class, four words are created. Now, if all these words exist in the word form list, then entry in the root table is made specifying the new root with its paradigm class.

Algorithm 4.1: Algorithm for creating root table

```
1. For each word w in word form list do
2.     For i € { 0 . . . . w.len } do
3.         Take suffix s of w having length i
4.         For each paradigm class p do
5.             For every match of s with x; x € { four suffixes of p } do
6.                 set t := no. of characters to be deleted from p before adding x to it to
                                                                    make a new word
7.                 set new_root := w – suffix x + suffix of p having length t
8.                 For each y of the four suffixes of p with u being the corresponding no.
                                                                of characters to be deleted, do
9.                     set new_word:=new_root-suffix of new_root of length u+suffix y
10.                    if new_word ∉ word form list
11.                        goto step 8
12.                    END IF
13.                END FOR
14.                Make entry in the root table that new_root follows paradigm p
15.            END FOR
16.        END FOR
17.    END FOR
18. END FOR
```

Figure 4.8 shows the list of roots with their paradigm names.

Like as shown, कपड़ा follows the लड़का paradigm. It means कपड़ा will have the same inflectional rules as the word लड़का.

root_nam	par_nam	root_nam	par_nam
पुस्तक	रात	बेटा	लड़का
कपड़ा	लड़का	विद्यार्थी	आदमी
बच्चा	लड़का	नदी	नदी
बकरी	नदी	साथी	आदमी
बकरा	लड़का	रात	रात
पक्षी	आदमी	कमरा	लड़का
पत्ता	लड़का	स्त्री	नदी
स्वामी	आदमी	आलू	आलू
लड़का	लड़का	दुकान	रात
आदमी	आदमी	औरत	रात

Figure 4.8: The root table

4.6 Morphological Analysis

It gives us the morphological details of the words in the sentence entered by the user. It takes one word at a time and check which paradigm class it follow. Accordingly, it gives the lexical details as an output using the paradigm class.

Algorithm 4.2 shows the algorithm used for this purpose.

First of all, take an empty set L in which we will enter all the morphological details of the input word w. Now put the suffix of w in s of length 0,1,2,….. up to the length of w. Search this ‘s’ in every entry ‘b’ of the paradigm table P.

Algorithm 4.2: Algorithm for morphological analysis [4]

```
1. L: = empty set
2. for i: =0 to length of w do
3.     let s = suffix of length I in w
4.     for each paradigm table P
5.         for each entry b (consisting of a pair) in P do
6.             if s=suffix in entry b then
7.                 r = root of paradigm table P
8.                 j = number of characters to be deleted as shown in b
9.                 Proposed-root= (w-suffix s) + suffix of r consisting of j characters
10.                If (proposed-root is in DR) and (the root has paradigm P)
11.                    Then construct a lexical entry l by combining (a) features given
                   in DR with the proposed-root, and (b) features associated with e.
12.                    Add l to set L.
13.                END IF
14.            END IF
15.        END for every entry b in P
16.    END for every paradigm
17. END FOR
```

Whenever a match is found, fetch the root of that paradigm table as r. Find the proposed root by removing the suffix s from w and adding the suffix of r of same length as s. If this proposed root is found in the dictionary of roots then enter the entry in the set L by combining the features of w with its corresponding paradigm table. Perform these steps for every suffix of w with each paradigm table.

Figure 4.9 shows the flowchart of the whole system.

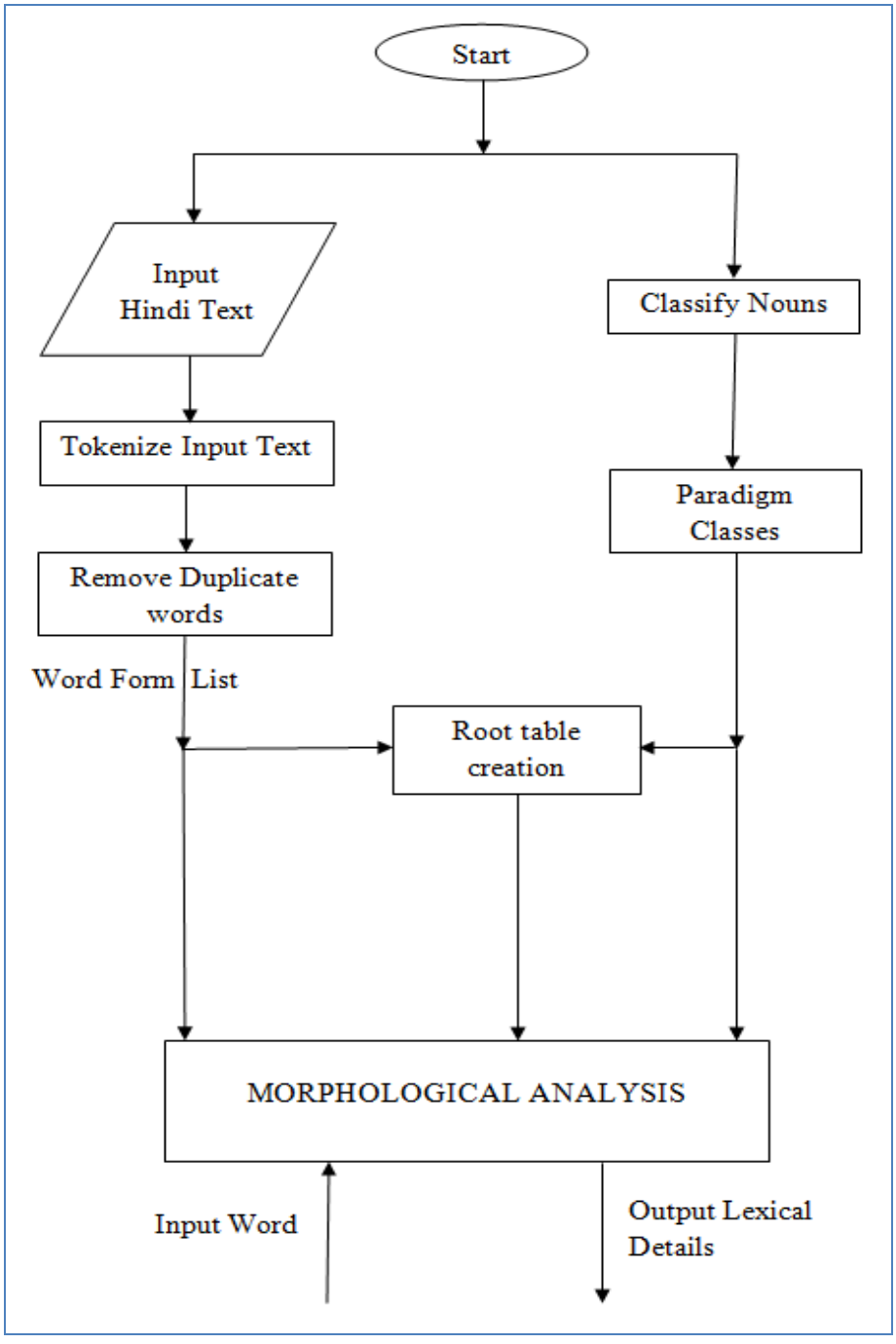


Figure 4.9: Flowchart of the whole system

Whenever the application is started, the window shown in figure 4.10 appears which contains the text area, where user can enter the input text for morphological analysis.

Figure 4.10: Paradigm based Hindi Morphological Analyzer

Figure 4.10 shows the interface of the morphological analyzer developed. It contains a textbox for giving a sentence to know the morphological details of each word in the sentence. There is a button which is used to start the morphological analysis process. When we press this button, the algorithm 4.2 gets implemented on the input text entered in textbox. Finally, we get the output in the lower pink-colored area.

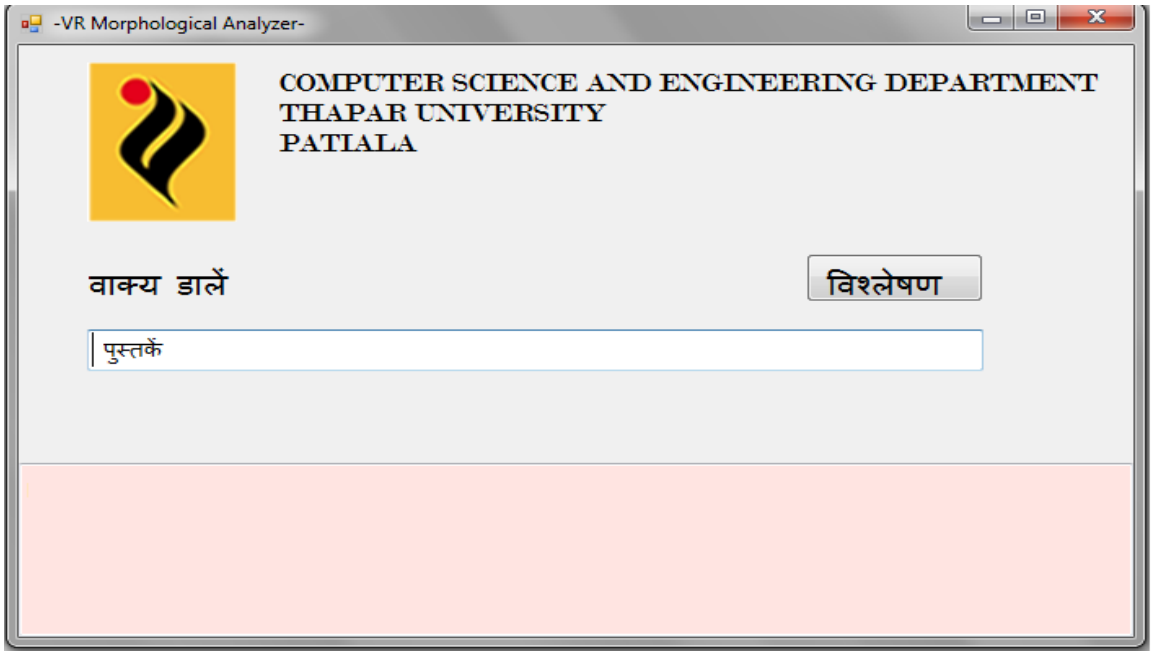


Figure 4.11: Entering the input in the developed Morphological Analyzer

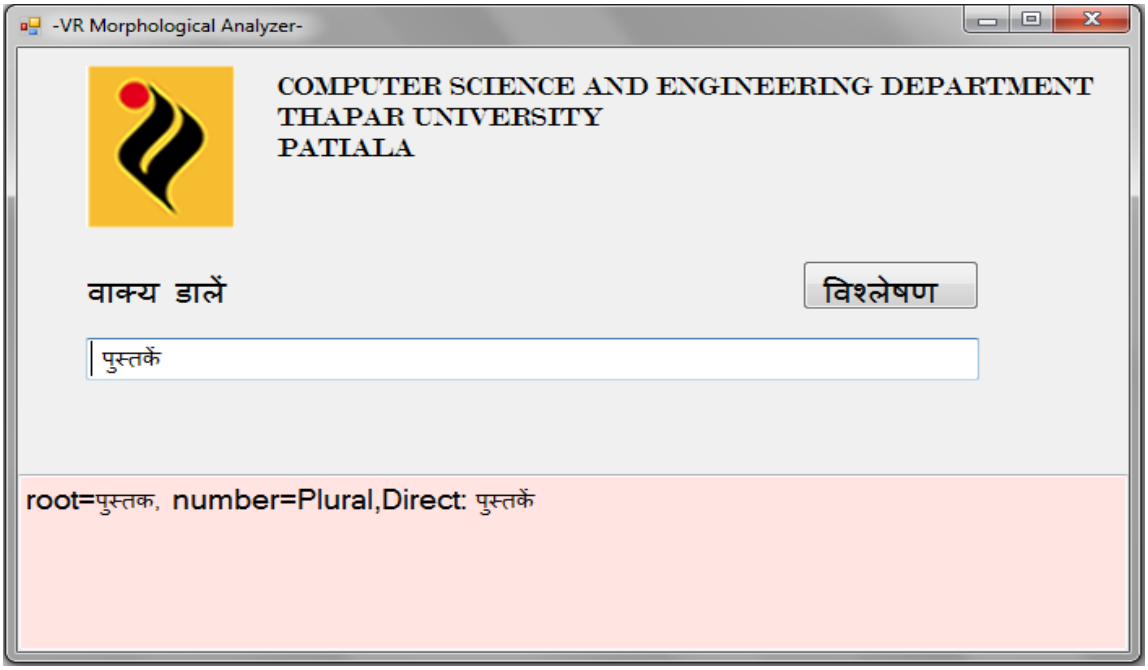


Figure 4.12: Lexical details produced on entering input

When user enters पुस्तकें as shown in figure 4.11; output is displayed showing that it is a plural and direct form of पुस्तक as shown in figure 4.12. In this way, the morphological analyzer works.

Chapter 5: Testing and results

The system is divided into three modules: dictionary creation, root table creation and morphological analysis. The testing was performed at every module individually.

5.1 Dictionary creation

Around 1 lakh words were generated from the dictionary creation tool and around 98,000 words were found to be valid. That means system is 98% correct. The 2% invalid words is the result of some wrong entries in the text taken from various newspaper websites, magazines *etc*. Figure 5.1 illustrates this data.

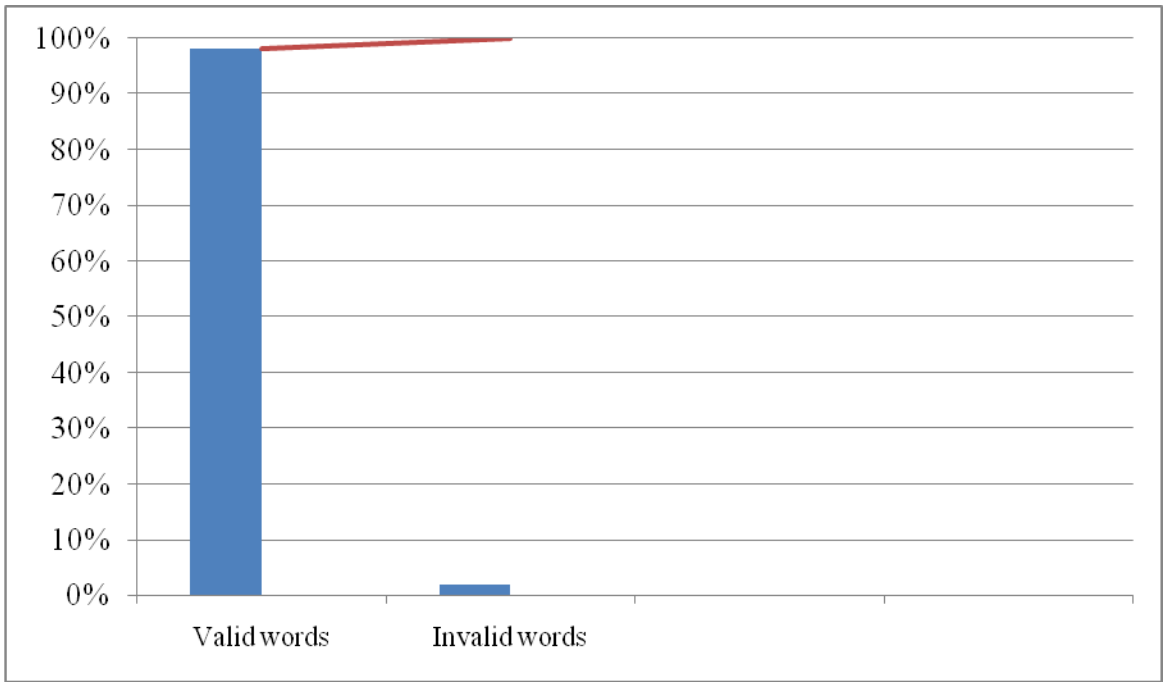


Figure 5.1: Testing of dictionary creation tool

The output of the module is the list of words which is known as word forms list.

5.2 Root table creation

Based on word forms list and the paradigm classes, 170 roots were found whose distribution is shown on the next page:

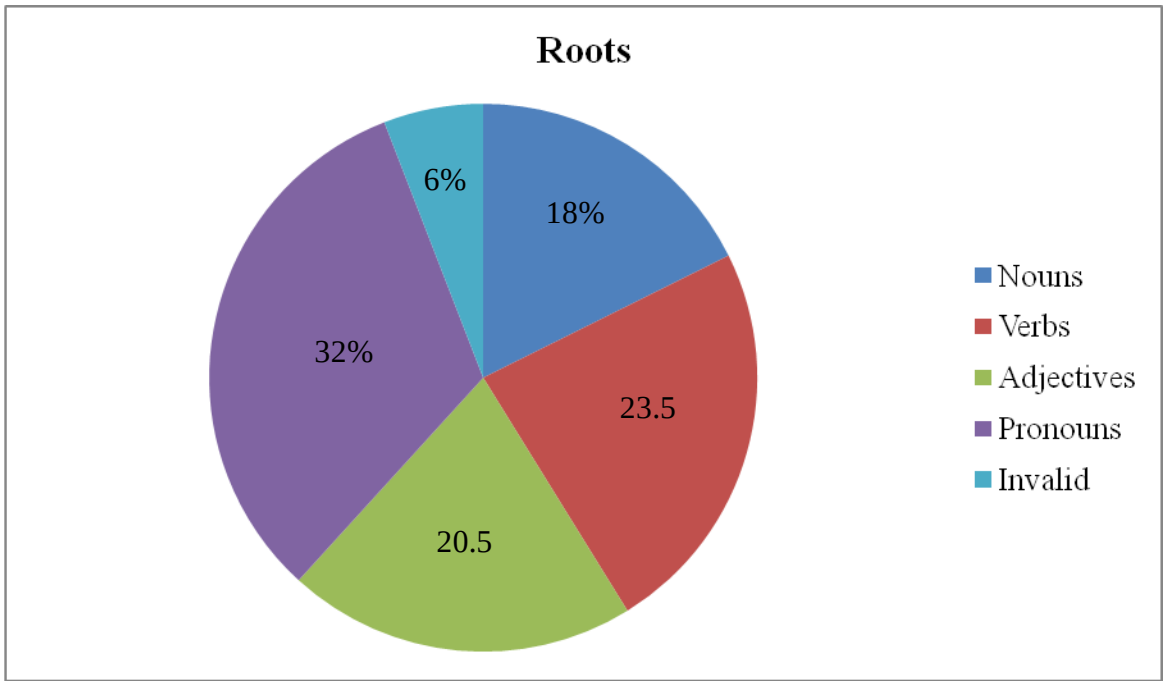


Figure 5.2: Distribution of the roots

Figure 5.2 shows a pie-chart which explains the distribution of the roots. It shows 18% roots are the nouns that are of our interest.

5.3 Morphological Analysis

Since the paradigm classes have been created for nouns only, the output corresponding to the nouns is considered to be correct. Based on this, probability of this module can be decided on the following points:

1. If a noun is entered, then it will give result if its root is in the roots table and will show nothing if the root is not in roots table.

2. Only case in which the system might give wrong output is when all the forms of input word are not there in the word form list.

Chapter 6: Conclusion and Future Scope

6.1 Conclusion

The morphological analyzer has been developed for Hindi language using the paradigm approach. A user friendly interface has been created for this purpose. It stores all the commonly used word forms for all Hindi root words in its database. This paradigm approach discussed in the thesis prefers time to memory. Non-paradigm approaches takes less memory space but greater time. These days, memory space is not a problem neither in terms of cost nor in terms of storage requirements. So, paradigm approach will work well. The search time based on the paradigm approach is very less. Another advantage of paradigm approach is that the user will always get the accurate results.

6.2 Future Scope

This system can be proved as an important tool for NLP applications.

- Word forms list needs to be extended in order to increase the database. This will also result in increase in the number of roots in the root table.
- Based on the approach discussed in the thesis, same approach can be followed to create the paradigm classes for Hindi adjectives, pronouns and verbs.
- Once all the paradigm classes are formed, the full morphological analyzer for Hindi language can be developed.

References

[1] Gurpreet Singh Lehal, "Punjabi Morphological Analyzer and Generator" [Online] Available: http://www.learnpunjabi.org/punjabi_mor_ana.asp. [Accessed: Oct. 2012]

[2] Mandeep Singh Gill, Gurpreet Singh Lehal, S. S. Joshi, "A Full-Form Lexicon based Morphological Analysis and Generation Tool for Punjabi", International Journal of Systematics, Cybernatics and Informatics, Hyderabad, pp. 38-47, 2007.

[3] Pratiksha Gawade, Deepika Madhavi, Jayshree Gaikwad, Sharvari Jadhav, Rahul Ambekar, "Morphological Analyzer for Marathi using NLP", International Journal of Engineering Research and Applications ISSN: 2248-9622 Vol. 3, Issue 2, pp.322-326, March -April 2013.

[4] Bharati, Akshar, Vineet Chaitanya and Rajeev Sangal, "Natural Language Processing: A Paninian Perspective", Prentice-Hall of India, New Delhi, 1995.

[5] Vishal Goyal, Gurpreet Singh Lehal, "Hindi Morphological Analyzer and Generator", Proceedings of First International Conference on Emerging Trends in Engineering and Technology, IEEE, pp. 1156-1159, 2008.

[6] Jisha P. Jayan', Rajeev R. R.2 and S. RnjendranJ, "Morphological Analyser for Malayalam - A Comparison of Different Approaches", IJCSIT International Journal of Computer Science and Information Technology, Vol. 2, No. 2, 2009.

[7] Shriya Sahu, Nandkishor Vasnik and Devshri Roy, "Prashnottar: A Hindi Question Answering System", International Journal of Computer Science and Information Technology (IJCSIT), Vol. 4, No. 2, pp. 149-158, 2012.

[8] "General Morphological Analysis", 2001 [Online] Available: http://www.swemorph.com/ma.html. [Accessed: Jan. 2013]

[9] Hannes Hirzel, "Morphology", Dec. 2001 [Online] Available: http://en.wikipedia.org/wiki/Morphology_(linguistics). [Accessed: Aug. 2012]

[10] Nonie Lesaux,"Morphological analysis: New light vital skill" [Online] Available: http://www.uknow.gse.harvard.edu/teaching/TC102-407.html, 2002. [Accessed: Feb. 2013]

[11] Teena bajaj, "Rule Based Semi-Supervised Morphological Analyzer for extending the Range of Existing System", M.E. Thesis, CSED, Thapar University, Patiala, Punjab, 2008.

[12] Jisha P.Jayan, Kalady Rajeev R. R. "Morphological Analyser and Morphological Generator for Malayalam - Tamil Machine Translation" International Journal of Computer Applications (0975 – 8887), Volume 13– No.8, January 2011.

[13] Nikolai Samteladze, "What is HashSet", [Online] Available: http://stackoverflow.com/questions/4558754/define-what-is-a-hashset, Nov. 2012. [Accessed: Feb. 2013]

[14] Ulf Hermjakob "Natural language processing" [Online] Available: http://en.wikipedia.org/wiki/Natural_language_processing, September 2001.

[15] Ray, P.; Harish V.; Sarkar, S.; and Basu, A.; Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi; Proceedings of the 1st International Conference on Natural Language Processing (ICON 2003), Mysore, 2003.

[16] Nikhil K V S, "Hindi Derivational Morphological Analyzer", M.S. Thesis, CSE, IIITH, Hyderabad, 2012.

[17] Smriti Singh, Vaijayanthi M Sarma "Hindi Noun Inflection and Distributed Morphology" [Online] Available: http://makino.linguist.univ-paris-diderot.fr/files/hpsg2010/file/abstracts/MFG/singh-sarma-mfg.pdf. [Accessed: Mar. 2013]

[18] Manish Shrivastava, Nitin Agrawal, Bibhuti Mohapatra Smriti Singh, Pushpak Bhattacharya, "Morphology Based Natural Language Processing tools for Indian Languages", The 4th Annual Inter Research Institute Student Seminar in Computer Science, Indian Institute of Technology, Kanpur April 1-2, 2005.

[19] Muktanand Agrawal, "Computational Morphology and Sanskrit" [Online] Available: http://sanskrit.jnu.ac.in/rstudents/mphil/muktanand/Chapter%201.pdf.

[20] "Telugu-Morph" [Online]
Available: http://nlp.amrita.edu:8080/project/mhrd/Rb-tuMorph.html
[Accessed: Dec. 2012].

[21] "Ministry of Human Resource Development" [Online]
Available: http://nlp.amrita.edu:8080/Transliteration/MorphologicalAnalyzer.html, 2010. [Accessed: Jan. 2013]

[22] Richard Wicentowski "Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework", October, 2002.