International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-1 Issue-14 March-2014

Morphological Analyser for Hindi – A Rule Based Implementation

Ankita Agarwal[1], Pramila[2], Shashi Pal Singh[3], Ajai Kumar[4], Hemant Darbari[5]

Abstract

Morphological analysis is an important part of Natural Language Processing. With this, the task of Machine translation becomes very easy. Morphological analyzer can be implemented effectively for the language which is rich in morphemes. Hindi is morphologically rich language. In this paper we focus on the design of a morphological analyzer for Hindi language. The analyzer takes a Hindi sentence or a word as an input and analyzes it to generate its necessary features with its root words. The features will have categories: part of speech, gender, number, and person. The tool works on both inflectional and derivational morphemes. This works on rule based approach.

Keywords

Morphological Analysis, Inflectional, Derivational, Rule Based, Corpus, Lemmatize.

1. Introduction

In terms of linguistics, morphology refers to formation of words by focusing on their internal structure. Morphology is divided into two classes: inflectional morphology and derivational morphology. In inflectional morphology, when a word stem is combined with a morpheme it results in same class word as of the word stem while in derivational morphology, it results in a different class word other than that of the word stem. Examples of inflectional morphemes are गाड़ी(Noun)becomes गाड़ियाँ(Noun)on adding ियाँ as suffix whereas in derivational morphemes कठोर(Adj) becomes कठोरता(Noun) on adding ता as suffix.

Manuscript received January 13, 2014.
AnkitaAgarwal, Department of Computer Science, Banasthali University, Rajasthan, India.
PramilaYadav, Department of Computer Science, Banasthali University, Rajasthan, India.
Shashipal Singh, AAI, CDAC, Pune, India.
Ajai Kumar, AAI, CDAC, Pune, India.
HemantDarbari, ED, CDAC, Pune, India.

The objective is to develop a tool which works on morphemes and generate a good morphological analyzer for inflectional morphemes as well as derivational morphemes. In this paper we discuss the development of our morphological analyzer for hindi language which works on rule based approach and we also maintain a database for exceptions that does not match with the rules made. Our morphological analyzes works as follows: first we check whether a given input is a sentence or a word. If a user input is a Hindi sentence, it tokenizes it into words then for each word we check whether it is a root word or not. If it is a root word it extracts its features like 'Category', 'Gender' and 'Number' from the database. If it is not a root word then rules are applied on that given input word to extract its features. Similar process will take place if a word is given as an input.

This paper is organized as follows: first we give a brief review of the related work done (Section 2). Next we present the detailed background of Hindi language (Section 3). And last we have described our approach, its working with some instances and the results (Section 4 and 5).

2. Related Work

To our knowledge yet no successful morphological analyser based on both inflectional and derivational morphemes of Hindi is developed successfully. But a few morphological analyzers have been developed for inflectional morphemes alone and a few for derivational too. In 2012 Nikhil Kamparthi et al. [1] proposed a derivational morphological analyser for Hindi which was upgraded from its existing inflectional analyser. However, it has some drawbacks. First it has been developed using wx-encoding (wx-format) which is difficult to understand and user should have complete knowledge about this. Second, root word is not extracted successfully.

Vishal Goyal et al. [2] proposed a Hindi Morphological analyser and generator which work on paradigm approach for windows platform having GUI. They have stored all the word forms of root words but have not taken the proper nouns. FST based morphological analyser for Hindi was

proposed by Deepak kumar et al. [3]. Stuttgart FST tool has been used for generating the FST. A Literature Survey of Morphological analysis and generation done by Antony P J et al. [4] was used to understand different morphology and parser developments in Indian Language. Teena Bajaj et al. [5] explained how morph analyzer is helpful in NLP tasks if semi-supervised learning approach is followed. This is the only tool which is publically available. But still it is not possible to analyze those words accurately which are not in their database. In 2010 Niraj et al. [6] experimented with the Hindi and Gujrati languages for developing the morph analyzer which works on rule based approach but it uses dictionary and corpus for suffix replacement rules. If the word form is not present in the dictionary, it is not able to derive word's root form. Some initial work was done by Ankita Agarwal in 2013 et al. [8] which is now further executed.

In our work, we have followed the rule based approach which uses lemmatize to extract the root words properly and a corpus which stores the exceptional words which does not match with the rules made. The rules are made to incorporate almost all the word formations possible after a deep analysis and study of the dictionary and other knowledge resources available. This system is useful and provides better accuracy than the existing ones.

3. Linguistic Background

As our morphological analyzer is developed for Hindi language, we should know about the actual structure of Hindi words, how they are formed, their special characteristics etc. Hindi shares major linguistic characteristics with other Indo-Aryan languages.

Hindi morphological structure consists of various word classes in which their derivational and inflection forms are described. Word classes include nouns, verbs, adjectives, pronouns, particles, connections and interjections. Now in the coming sections, details about the word classes are provided which is presented by Omkar K. Koul et al. [7].

3.1Nouns
Nouns in Hindi are generally inflected for gender, number and case.

3.1.1Gender
There are three declensions of nouns:

Declension I have masculine nouns ending with आ/a:/
Declension II have all other masculine nouns.
Declension III have all other feminine nouns.

3.1.1.1 Most of the आ/a/ ending masculine nouns have their feminine forms ending in ई/i:/

Table 1: Word Formations ending with आ/a/

Masculine			Feminine		
प्यासा	Payasa	thirsty	प्यासी	Payasi	Thirsty
चरखा	Charkha	wheel	चरखी	Charkha	Reel

3.1.1.2 Most of the ई/i:/ ending animate masculine nouns have their feminine forms ending in -अन/-an/

Table 2: Word Formations ending with ई/i:/

Masculine			Feminine		
धोबी	Dhobbi	washer man	धोबन	dhoban	Washer woman
भोगी	Bhogi	suffering	भोगन	bhogan	Suffering

3.1.1.3 Some nouns ending in आ/a:/ form their feminine by replacing आ/a:/ with –इया/-iya:/

Table 3: Some other word Formations

Masculine			Feminine		
डब्बा	Daba:	box	डिबिया	Dibiya:	a small box
चिड़ा	Chida:	sparrow	चिड़िया	chidiya	Bird

3.1.1.4 Most of the –आ/a:/ ending nouns are masculine are replaced by - ई/-i:/ to form feminine

Table 4: Some other word Formations

Masculine			Feminine		
कटोरा	katora	a bowl	कटोरी	katori	a small bowl
चीटा	cheetaa	a large black ant	चीटी	cheetii	Ant

3.1.1.5 The suffix नी/ni:/is added to the masculine nouns to form the feminine

Table 5: Word Formations which adds suffix नी/ni:/

Masculine			Feminine		
मास्टर	master	teacher	मास्टरनी	masterni	teacher

3.1.1.6 The suffix ई/-i:/ is added to the masculine noun to form feminine

Table 6: Word Formations which adds suffix ई/-i:/

Masculine			Feminine		
सुन्दर	sundar	beautiful	सुंदरी	sundari	Beautiful woman
राजकुमार	rajkumar	king	राजकुमारी	rajkumari	Queen

3.1.2Number
These are of two types: singular and plural

3.1.2.1 Singular masculine nouns ending with आ/a: change into plural ending with ए/e/

Table 7: Word Formations ending with ए/e/

Singular		Plural	
घोडा	Horse	घोड़े	Horses
पत्ता	Leaf	पत्ते	Leaves
पिता	Father	पिता	Fathers
नेता	Leader	नेता	Leaders

3.1.2.2 All other consonants and/or other vowel-ending nouns do not change their plural forms.

Table 8: Word Formations which don't change

Singular		Plural	
फल	Flower	फल	Flowers
छेद	Hole	छेद	Holes

3.1.2.3 The feminine plurals are formed by adding the suffix एं/ẽ/ to the consonant-ending singular forms

Table 9: Word Formations which adds suffix एं/ẽ/

Singular		Plural	
खरोंच	Scratch	खरोंचे	Scratches

3.1.2.4 Feminine nouns ending with ई on adding इयाँ becomes plural.

Table 10: Word Formations which adds suffix इयाँ

Singular		Plural	
मिठाई	Sweet	मिठाइयाँ	Sweets

Here last vowel of the stem is removed.

3.1.3 Noun Derivation
Mostly nouns in Hindi are derived from nouns, adjectives and verbs by using suffixes.

3.1.3.1 Nouns from Nouns:

Commonly used suffixes aredार–da:r,गर-gar and दान –da:n

Table 11: Word Formations ending with दार–da:r, गर-gar and दान –da:n

ईमान	दार	ईमानदार	Honest
जादू	गर	जादूगर	Magician
खजाना	ची	खजानची	Cashier
कलम	दान	कलमदान	Penholder
कार	खाना	कारखाना	Factory

3.1.3.2 Nouns from Adjectives:

Mostly used suffixes for this purpose are ई-i,ता–t:a,pan, आई-a:I,इयत-iyat,आस-a:s

Table 12: Word Formations which form nouns from adjectives

Adjectives		Nouns	
खराब(kharab)	Bad	खराबी (kharabi)	Defect
सच्चा (saccha)	True	सच्चाई (sacchai)	Truth
विशेष (vishesh)	Special	विशेषता (visheshta)	speciality
गंभीर (gambhir)	Serious	गंभीरता (gambhirta)	seriousness
पागल (pagal)	Mad	पागलपन (pagalpan)	madness
असली (asli)	Real	असलियत (asliyat)	reality

मीठा (meetha)	Sweet	मिठास (meethas)	sweetness

3.1.3.3 Nouns from Verbs

Suffixes used to derive nouns from verbs are अस-as,अन-an,ई-ee,वत-vat,ना-na

Table 13: Word Formations which form nouns from verbs

Verbs		Nouns	
लिख (likh)	Write	लिखना (likhna)	writing
सींच (seench)	Irrigate	सींचना (seenchna)	irrigating
धड़क (dhadak)	Throb	धडकन (dhadkan)	throbbing
लड़ (lad)	Quarrel	लड़ाई (ladaee)	dispute
थक (thak)	be tired	थकावट (thakavat)	tiredness

3.2. Verbs

There are two types of verbs: main verb and auxiliary verb. Verbal construction is classified in the following ways:

- Intransitive verb
- Transitive verb
- Ditransitve verb
- Causative verb
- Dative verb
- Conjuct verb
- Compound verb

We have inserted only transitive and intransitive verbs in our database:

Intransitive verbs are likeआ-aa,जा-ja,उठ-uth,बैठ-baith

For eg.-वहजाताहै

He goes

Transitive verbs are derived from intransitive verbs by certain vocalic changes to the verb roots.

For eg.-

Table 14: Word Formations form adjectives

Intransitive		Transitive	
पिस	be ground	पीस	Grind
घूम	go round	घुमा	turn around

3.3. Adjectives

In Hindi these are classified as inflected and uninflected.

Table 15: Inflected Adjectives

Masculine		Feminine
singular	Plural	Singular/plural
बड़ा (bada)	बड़े (bade)	बड़ी (badi)
गोरा(gora)	गोरे (gore)	गोरी (gori)
मोटा (mota)	मोटे (mote)	मोटी (moti)

Table 16: Uninflected Adjectives:

सुन्दरलड़का/लड़की	Sundarladka/ladki	beautiful boy/girl
सुखीआदमी/औरत	Sukhiaadmi/aurat	happy man/woman

3.3.1 Derivation of Adjectives:

Adjectives are derived from nouns by adding suffixesआ-aa,ई-I, उ-u,ईला-ila,लू-lu,इक-ik, जनक-janak,दाई-daai, मय-may,वन-van,आना-ana,नाक-nau,ईन-inn,मंद-mand,दार-dar.

Table 17: Word Formations which form adjectives from nouns

Nouns		Adjectives	
भूखbhu:kh	Hunger	भूखाbhu:kha:	Hungery
पेट pet	Stomach	पेटूpetu:	Voracious
पत्थरpatthar	Stone	पथरीलाpatthari:la	Stony
कृपाkripa:	Kindness	कृपालुkripa:lu	Kind
मासma:s	Month	मासिकma:sik	Monthly
दयाdaya:	Mercy	दयावानdaya:lu:	Kind
नेक nek	Good	नेकी neki:	Goodness

4. Methodology

We developed our morphological analyser using the following methodology:

4.1 Analysing behaviour of Hindi Inflections

For successfully analysing, we first studied and identified the inflectional and derivational suffixes as described in the previous section. Different rules were made to extract features from given input words. Here, the root word is extracted by using lemmatize which is also rule based and other categories are extracted by using the rules made. For this reason some commonly used words were taken for the development of our corpus. Corpus has been designed from the raw data that we gathered from internet and books. The corpus is aligned in a proper way so that we can study each word individually without any error.

4.2 Database

A database was developed in which words were stored with root words and features like Category, Gender and Number. We have inserted mostly common nouns in our database. Our database is restricted to mostly nouns, verbs and adjectives. This database will be used just for exceptions which do not match the rules made.

The schema of our database is as follows: DATA {Word_id (Primary key), Wordname, RootWord Category, Gender, and Number}

4.3 Algorithm

The algorithm developed is as follows:

Step 1: Input Checking

First, Input is given by the user. The input may be a Hindi word or a sentence.

Step 2: Root Word Matching

2.1 If the input found is just a word.It is simply matched with the root word in the corpus and its morphological features are generated and displayed.

2.2 If the word is not found or matched with the words in the corpus,it means that it is not a root word so we use lemmatizer to generate the root word of a given input word.

Step 3: Exception Handling

In the next step the output given by a lemmatizer is matched with the words in the corpus maintained for the exceptions.

3.1 If the match is found features are displayed.

3.2 In case any word is not matched with the corpus we apply rules to generate its features.

On the other hand, if the input is found to be a sentence i.e more than one word then the sentence is tokenized into words and for each word the above process applied iteratively, after displaying the features the analyzer moves to the next token. In case any word(s) is not matched with the corpus its features are generated with rule matching process and the analyser moves to next token.

A flowchart of our algorithm is as follows:

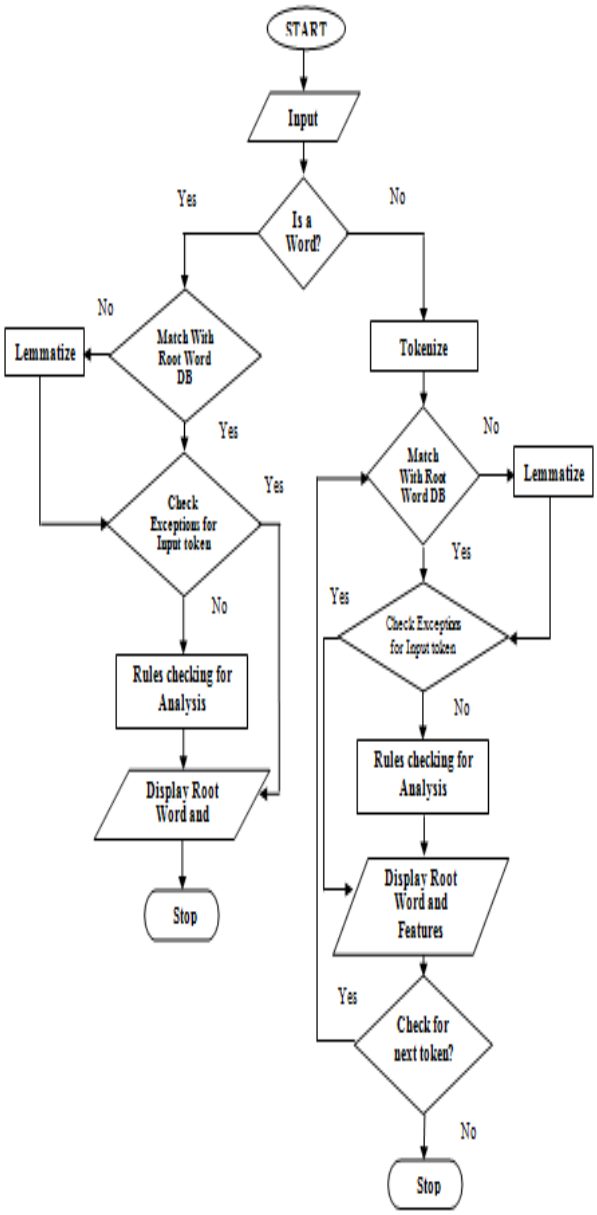

Figure 1: Flowchart of our morphological analyser

5. Illustrations

The above algorithm is illustrated with the help of following examples:

CASE I: If the input given isनिपुणता then in step 2 it is checked whether it is a word or sentence. As a word is recognized, so next step matches it with the database. Match is found and then from the further step morphemes of निपुणता are produced which is the final result obtained as:

निपुणता = निपुण(root word) + Noun(Category) + Feminine(Gender) + Any(Number)

CASE II: If the input given is "अंकुरतेज़दौड़ताहै", Input is first checked in step2 and it recognizes the input as a sentence. So next step tokenizes the sentence into tokens and each token is matched with database in step by step iteratively. अंकुर is matched and its features are generated and displayed after further steps being executed. After that next token तेज़ is checked and so on. All tokens were matched and hence all features are displayed as follows:

अंकुर = अंकुर(root word) + Noun(Category) + Mascuilne(Gender) + Singular(Number)

तेज़=तेज़(root word) + Adj(Category) + Any(Gender) + Singular(Number)

दौड़ता=दौड़(root word)+ Verb(Category) + Mascuilne(Gender) + Singular(Number)

है= indeclinable

CASE III: If input given is"ऑपरेटिग".It is recognized as a word by step2 but no match is found by further steps.So a"invalid word" message is displayed as a result.

6. Conclusions

We have discussed in this paper a Hindi Morphological Analyzer which is basically based on rule based approach but also utilizes the corpus when exception occurs. We have incorporated almost all the possible rules for the different word formations of Hindi as described in the Section 3 be it inflectional or derivational. As there is no problem of memory space these days, this analyzer is performing better than others. The accuracy of the system is very high as all the possible exceptions are also covered. As a future work, we can integrate the word sense disambiguation with this analyzer so that the words having multiple senses could be analyzed accurately too.

References

[1] Nikhil Kanuparthi, AbhilashInumella, DiptiMisra Sharma, "Hindi Derivational Morphological Analyzer", Proceedings of the twelfth meeting of the Special Interest Group on Computational Morphology & Phonology, Canada, pp.10-16, June, 2012.

[2] Vishal Goyal, Gurpreet Singh Lehal, "Hindi Morphological Analyzer and Generator", First International Conference on Emerging Trends in Engineering and Technology, USA, pp.1156–1159, 2008.

[3] Deepak Kumar, Manjeet Singh, SeemaShukla, "FST Based Morphological Analyzer for Hindi Language", International Journal of Computer Science Issues(IJCSI), Vol. 9, pp.349-353, July, 2012.

[4] Antony P.J, Dr. Soman KP, "Computational Morphology and Natural Language Parsing for Indian Languages: A Literature Survey", International Journal of Computer Science and Engineering Technology(IJCSET), Vol. 3, pp.136-146, April, 2012.

[5] Teena Bajaj, Parteek Bhatia, "Semisupervised Learning Approach of Hindi Morphology", All India Conference on Advances in Communication Computers, Control & Knowledge Management(AICACCC-KM), Bahadurgarh, Feb, 2008.

[6] NirajAswani, Robert Gaizauskas, "Developing Morphological Analyzers for South Asian Languages: Experimenting with the Hindi and Gujrati Languages", Proceedings of the Seventh International Conference on Language Resources and Evaluation, Valleta, Malta pp.811-815, May, 2010.

[7] Omkar N. Koul, "Modern Hindi Grammar", Dunwoody Press, USA, 2008.

[8] AnkitaAgarwal, "Morphological Analyser", Online at http://www.studymode.com/essays/Morphological-Analyser-39264244.html (as of October 2013).

Ms. Ankita Agarwal is a Research Scholar and doing her internship from C-DAC Pune under M.Tech, Computer Science IInd year Curriculum which is being pursued from Banasthali University, Rajasthan India. She has completed her B.Tech degree in Computer Science from Integral University, Lucknow. She has also been an ex-Lecturer of Sherwood College of Engineering, Lucknow for 2 years.

Ms. Pramila Yadav is a Research Scholar and doing her internship from IIIT Hyderabad under M.Tech Computer Science IInd year Curriculum which is being pursued from Banasthali University, Rajasthan, India. She has completed her B.Tech degree in Computer Science from Rajasthan Technical University, Kota.

Mr. Shashi Pal Singh is working as STO, AAI Group, C-DAC, Pune. He has completed his B.Tech and M.Tech in Computer Science & Engg. and has published various national & international papers. He is specialised in Natural Language Processing (NLP), Machine assisted Translation (MT), Cloud Computing and Mobile Computing.

Mr. Ajai Kumar is working as Associate Director and Head, AAI Group, C-DAC, Pune. He is handling various projects in the area of Natural Language Processing, Information Extraction and Retrieval, Intelligent Language Teaching/Tutoring, Speech Technology [Synthesis & Recognition ASR], Mobile Computing, Decision Support Systems & Simulations and has published various national & international papers

Dr. Hemant Darbari is working as Executive Director in C-DAC, Pune. He is one of the founding members of C-DAC, an R&D institute set up by the Department of Electronics and Information Technology; Govt. of India for carrying out advanced research in new and emerging technological domains. He has to his credit, 85 Technical Papers that have been published in national & international Journals & Conference Proceedings.