A Survey on Various Methods to Develop Morph Analyzer

Pooja C. Patel  
Department of Computer Engineering  
IIET, Dharmaj, India

Jatayu H. Baxi  
Department of Computer Engineering  
DDU, Nadiad, India

Rohan C. Prajapati  
Department of Computer Engineering  
IIET, Dharmaj, India

Abstract

A Morph Analyzer is a software tool which is used to detect morphemes from given sentence and provides grammatical analysis of each word present in the sentence. Morph analyzer is one of the essential tools for natural language processing and it is used for applications like machine translation. Lots of work has been reported in this field of morph analyzer for the foreign languages as well as some Indian languages. Previously for developing morph analyzer, both supervised and unsupervised methods had been tried. In this paper we compare various existing methods like Rule based Method, Suffix Stripping method, Finite state transducer approach and machine learning method. Then we also compare existing methods with respect to Gujarati language based on various parameters and check feasibility of implementing morph analyzer for Gujarati with each method.

Keywords: Morph Analyzer, Gujarati Language.

I. INTRODUCTION

Morphological Analyzer is a software component capable of detecting morphemes in a piece of text and providing grammatical information of a word given. The Morph analyzer takes any valid word form as an input and returns root word and grammatical information about all the possible word forms for a given word.

A. Applications of Morph Analyzer
A highly inflectional language has capability of generating hundreds of words from a single root. Hence to identify structure of various Word forms, the Morph analyzer is essential tool.

The Morph Analyzer is used in the applications like POS tagging, Chinking and Machine Translation. While building machine translation systems involving highly inflectional languages it is important to pick the correct word form of given word.

B. Gujarati Language
Gujarati is an Indo-Aryan language, and part of the greater Indo-European language family. It is derived from a language called Old Western Rajasthani (1100 - 1500 AD). It is native to the Indian state of Gujarat, and is its chief language, as well as of the adjacent union territories of Daman and Diu and Dadra and Nagar Haveli.

C. Morph Analyzer Develop for Various Languages –

1) Finite State Non-Concatenative Morphotactics
Proposed concept of Finite state Morphology. And this approach was used to build Morphological analyzers for Many Languages. In this approach, XFST is used as an interface. It is an interface which gives access to finite state operations. The interface of XFST includes a lookup operation and generation operation, Hence this approach can be used for Generation and Analysis of Morphology. The Language grammar is modelled in form of Finite state automata. There are basically three components of this approach. The Lexicon, Regular Expression and the Interface. It should be noted that FSM transducers produce all possible analyses of the word.

2) Automated Paradigm Selection for FSA Based Konkani Verb Morphological Analyzer
Discusses work in which the morphological analyzer for Konkani has been developed using FSA based approach with Word paradigm model. In traditional finite state transducer approaches the characters are sequenced. In this approach instead of characters, the morphemes are sequenced. Word paradigm model is used which stores root words and corresponding paradigm information. Mapping of word to the paradigm is done automatically using corpus based approach. There is verb paradigm...
repository. If the word is not found in that repository then only the paradigm selector module is invoked which maps correct paradigm to the word automatically. If the word belongs to multiple paradigms then all the word forms are searched in the corpus and the highest matching paradigm is selected. The basic focus of this paper is to build Morphological analyzer for Verbs. Also it has been observed that Finite state automata are computationally efficient.

3) A Paradigm-Based Finite State Morphological Analyzer for Marathi
Developed paradigm based Finite state Morphological analyzer was developed for Marathi. In Hindi the inflected forms do not allow further suffixes to be added but in Marathi even the inflected forms can also take suffixes, hence the traditional paradigm based approach does not work for this language. The linguistic resources required by the morphological analyzer include a lexicon and inflection rules for all paradigms. Inflection rules specify the inflectional suffixes to be inserted (or deleted) to (or from) different positions in the root to get its inflected form.

Various features used to analyze a word are Case, Gender, Number, Person, Tense, Aspect, Mood etc. The FSM is used for morphological recognition of Marathi. SFST (Stuttgart Finite State Transducer) is used to model the Finite state transducer. This system exploits the regularity in the inflectional paradigms while employing the Finite State Systems for modelling the language in an elegant way.

4) Hybrid Stemmer for Gujarati
Have developed lightweight stemmer for Gujarati language using hand crafted suffix rules. Hybrid approach has been used in this system. Basic approaches to build stemmer are Rule based approach, unsupervised approach and hybrid approach. List of hand crafted Gujarati suffixes which contain the postpositions and the inflectional suffixes for nouns, adjectives and verbs are created for use in this approach. In this approach the optimal split point is determined using frequency based equation for given word. Once the optimal split point has been determined, the frequencies of stem and suffixes are updated. After the training phase is over, the list of stems and along with frequencies are obtained and they are used to create signatures. For stemming any unknown word, the value of function is obtained for each split point and the split point for which function value is maximum is selected as optimal split point.

5) Hybrid Inflectional Stemmer and Rule-Based Derivational Stemmer for Guajaratii
Developed Inflectional and derivational stemmer. The inflectional stemmer is built using hybrid approach and derivational stemmer is built using rule based approach. Four lists of suffixes which contain postpositions and inflectional suffixes respectively for Nouns, Verbs, Adjectives and adverbs are created. For inflectional stemming the Goldsmith approach is used as an inspiration and along with that two more modules have been incorporated ,One performing POS based stemming and other doing suffix stripping based on Linguistic rules. During the training phase some Gujarati words are used in order to learn probable suffixes and stems. This information is used in stemming an unknown word. The inflectional stemmer has an accuracy around 90. For derivational stemming, four step algorithms is used, the algorithm checks the input word with Orthographic rule match, Substitution Rule, Suffix stripping rule and finally the word is checked by FSA. This derivational stemmer has accuracy around 70.7 and can be further improved by rule enhancement

6) Developing Morphological Analysers for South Asian Languages: Experimenting With the Hindi and Gujarati Languages
Used rule based approach that takes both prefixes as well as suffixes into account. Given a corpus and a dictionary, this method can be used to obtain a set of suffix replacement rules for deriving an inflected word’s root form. This system is basically built for Hindi but some experiments are done on Gujarati language also. In this approach first the dictionary of root forms are obtained. Then the training corpus is taken and for each of the word in the training corpus, the suffixes are removed from the end and every time it is checked in the root dictionary. If the match is found then the suffix which was removed is noted and hence the suffix replacement rule is formed. For Gujarati also the similar approach is tried. The major limitation of this approach is that if the root form does not exist in the dictionary then the rule cannot be formed.

7) A Suffix Stripping Based Morph Analyser for Malayalam Language
Developed Morphological analyzer using suffix stripping approach for Malayalam language. The finite state transducer is used to sequence the morphemes and to validate the ordering. In this system the Suffix stripping method with sandhi rules are used that does not require any lookup tables. Also the rules are stored to find root/stem form. This approach is analyzed to be simpler than any other approach implementation wise. The Noun and verbs are analyzed with the grammatical features and sandhi rules are formed. The overall algorithm works as follows: The word is taken as an input First it is checked whether the word is in the root directory, If the word is found in the root directory then it is already in root form so the algorithm stops, otherwise the suffix is separated from the right hand size, If any suffix is present in the word then check the suffix in the dictionary and remove the suffix and reinitialize the word. This process is repeated until the word is found in the dictionary.
8) **Hindi Morphological Analyzer and Generator**

This project Hindi morphological analyzer and generator has been developed as a part of the development of a machine translation system from Hindi to Punjabi language. This Morphological analyzer and generator follow database driven approach. It does not use any files for storing the database rather all the required data is stored in the table in normalized form.

## II. EXISTING METHODS FOR MORPH ANALYZER

### A. Rule based Approach

In rule based approach the suffix rules for particular language are defined. When an input word is given to the system, the system will try to match rules which are already defined. If the rule is matched, then the feature defined along with that rule is given as an output.

The advantage of using this approach is that it is less complex and easy to implement but the major concern is rule building for particular language. If the nature is highly inflectional in nature then the number of rules is very large. Also the rules may be ambiguous. One rule may correspond to more than one feature of the word. So resolving this ambiguity becomes important while using rule based approaches. Rule based approach is also known as paradigm based approach. In this approach various paradigms are defined. Paradigms are built based on similarities in word form generation. If all the word forms of different roots have same characteristics then all these roots are grouped into single paradigm. For particular paradigm one representing word is listed along with all suffix rules which are applicable for that particular paradigm.

Also in this approach one dictionary is created which contains root words along with the paradigm to which that root word belongs. When an input word is given to the system, based on the suffixes the system will try to determine the paradigm to which the word belongs. Once the paradigm is decided then the root form is obtained and checked against the dictionary of roots. If the match is found then the paradigm is correct and the grammatical features are returned. The major concern of this approach is mapping word to the correct paradigm. If the manual dictionary is prepared and each root has to be manually mapped to paradigm then it is very time consuming process. The accuracy of this approach for high inflectional languages is very good.

### B. Suffix Stripping Approach

In suffix stripping approach the suffixes from the end of word is removed recursively until the word is found in some dictionary. This approach requires dictionary of root forms in back end. If the word is not present in the dictionary then the matching fails and hence analysis cannot be done. The advantage of using this approach is that it is simple to implement but its complexity is very high as every time the word has to be searched in the dictionary. So dictionary has to be built. If the language is less inflectional and has less number of suffixes then this approach is suitable. But for high inflectional languages where more than one suffixes are attached to the word this approach might not be suitable.

### C. Finite state Transducer Approach

In FST approach, first step is to create FSA for all possible word forms in particular language. The basic purpose of FST is to check whether the word form which is given as an input is in proper order or not. The input word will be treated like string which is to be matched against regular expression which defines grammar of particular language. In this approach also once all the suffixes are determined, then the root form must be validated against dictionary of root forms for the language. this approach is analyzed to be computationally efficient. The only difficulty in this approach can be modelling of language as FSA in which all the word forms accepted by the language should be covered.

### D. Machine Learning Approach

Machine learning based Approach is basically further divided into two sub categories: Supervised Approach and Unsupervised approach. Supervised approach requires training data and rules are extracted from the training data. Whereas unsupervised approach does not require any sort of training data.

Now we look at one sample implementation of both approach. A semi-automatic approach which makes use of supervised learning has been tried for Hindi. In this work, the SVM classifier is used to build derivational Morphological analyzer. Training data was prepared having tuple (A, B) meaning Word B derived from A. This training list was prepared manually. Features like Document Co-occurrence frequency, Length of the noun, Difference between lengths of the noun and suffix, Occurrence in Hindi WordNet definition, Occurrence in Hindi Wiktionary definition, Occurrence in a Wikipedia page, Occurrence in anchor and body. Inflection Morph analyzer is also built using paradigm based approach as machine learning techniques performs poorly in classifying huge amount of suffixes.

Goldsmith built Morph analyzer for English using unsupervised learning approach. This work reports on a new algorithm for automatic and unsupervised morphological analysis of a corpus it determines the prefixes, suffixes, and stems. Work is divided into 3 problems. The determination of the correct morphological split for individual words; the establishment of accurate categories of stems based on the range of suffixes that they accept and the identification of different forms of the same stem. The limitation of Present unsupervised methods are that they are typically developed for large corpora and are not suitable for small corpora of a few hundred thousand or even just thousands of words.
III. COMPARISON OF VARIOUS APPROACHES WITH RESPECT TO GUJARATI LANGUAGE

In this section we discuss an example of an inflected Gujarati word and see how that word is processed using various approaches discussed above:

The word in this example is છોકરાઓ (Chhokrao, Boys)

Now we see processing of this word using different approaches:

A. Rule based approach:

In this approach, various suffix rules are defined. On the basis of these rules grammatical features can be extracted. Here in this case one rule is defined that if the ending suffix is ઓ (o), then the Noun is in plural form. So the system will remove suffix and then the word which is left is છોકર (Chhokra). Now we have one more rule that if the word ends with ાા (aa), then remove that suffix and append ા (u) suffix to form the root word. So the root word is છોકર ા (Chhokru).

In paradigm based approach various paradigms are defined. Paradigms are built based on similarities in word form generation. If all the word forms of different roots have same characteristics then all these roots are grouped into single paradigm. The suffix rules are also defined. So first a suffix ઓ (o) is removed and then the root word is obtained in the same way as obtained in rule based system. Here the root word is છોકર (chhokru, Child), so it is searched in the dictionary of root words. In the dictionary of root words, the word છોકર (chhokra, Boys) is mapped with paradigm ID to which it belongs. So from that paradigm ID we can get information about root word, its grammatical features and other possible forms of that root word. If more than one paradigm matches for given word then we need to select correct paradigm based on some techniques like Dictionary based technique, Probability based technique etc.

B. Suffix Stripping Approach:

In this approach the suffixes from the end of word is removed recursively until

The word is found in some dictionary. So in this example it will first scan the character from the end. The character is ઓ (o), first the system will check whether ઓ (o) is valid suffix or not. If it is not valid suffix then the program will terminate. Here ઓ (o) is proper suffix so it will be removed from the end. Now the word left is છોકર (chhokra, Boys). Now the word છોકર (chhokra, Boys) is searched in the dictionary. If the word is found in the dictionary then the system will return the matched word otherwise system will keep on removing suffixes until match is found in the dictionary.

C. Finite State Transducer Approach:

In this approach Finite state automaton is prepared for particular language. This FSA will check correct sequencing of Morphemes in the word. Here the word is છોકરાઓ (chhokrao, Boys), so the FSA will recognize it as proper word and sequence the morphemes as છોકર (chhokra, Boys) and ઓ (o). The sample FSA for this is shown in 1.1. So from the transition from છોકર (chhokra) to છોકરાઓ (chhokrao) we can extract the plural feature of the word.

![Fig. 1.1: Sample FSA](image)

D. Machine Learning Approach:

In unsupervised approach the lexicons given to the system first and from this lexicon the system understands relation between morphemes using some machine learning algorithm. A popular algorithm for that is based on differences between two consecutive words in the lexicon. After this model is prepared, unseen word is given to the system and system gives output with help of some parameters, thresholds, human intervention, and model selection during development. In our example also if the system has some learning regarding structure of word છોકરાઓ (chhokrao, Boys), it gives output accordingly.
Title | Approach | Algorithm | Result
--- | --- | --- | ---
Hybrid Inflectional Stemmer and Rule-Based Derivational Stemmer for Gujarati | Rule based | Goldsmith | Derivational-70.7%  
Inflection – 90%
Developing Morphological Analysers for South Asian Languages: Experimenting With the Hindi and Gujarati Languages | Rule based | Suffix Stripping | -
Hybrid Stemmer for Gujarati | Rule based + Machine Learning(Unsupervised) | Split Point | -

IV. CONCLUSION

As per our knowledge and understanding, No work has been reported in the direction of building Morphological analyzer for Gujarati Language. As Gujarati has very rich Morphology, various word forms are possible and Gujarati is my mother tongue so we try to develop morph analyzer for Guajarati language using rule based approach and this is suitable for highly inflection language.

BIBLIOGRAPHY

[3] A Paradigm-Based Finite State Morphological Analyzer for Marathi, Mugdha Bapat, Harshada gune, Pushpak Bhattacharya, Proceeding of the 1st workshop on South and Southeast Asian Natural Language Processing(WSSANLP), Pages 26-34, the 23rd International Conference on Computational Linguistics(COLING), Beijing, August 2010.
[4] Hybrid Stemmer for Guajarati, Pratik Kumar Patel, Kashyap Popat, Pushpak Bhattacharya, Proceeding of the 1st workshop on South and Southeast Asian Natural Language Processing(WSSANLP), Pages 26-34, the 23rd International Conference on Computational Linguistics(COLING), Beijing, August 2010.