Hindi Morphological Analyzer and Generator

Vishal Goyal[1], Gurpreet Singh Lehal[2]
[1] Lecturer, Department of Computer Science, Punjabi University, Patiala
[2]Professor and Head, Department of Computer Science, Punjabi University, Patiala
[1]vishal.pup@gmail.com, [2]gslehal@gmail.com

Abstract

Morphology is the field of the linguistics that studies the internal structure of the words. Morphological Analysis and generation are essential steps in any NLP Application. Morphological analysis means taking a word as input and identifying their stems and affixes. Morphological Analysis provides information about a word's semantics and the syntactic role it plays in a sentence. Morphological Analysis is essential for Hindi it has a rich system of inflectional morphology as like other Indo-Aryan family languages. Morphological Analyzer and generator is a tool for analyzing the given word and generator for generating word given the stem and its features (like affixes). This paper presents the morphological analysis and generator tool for Hindi Language using paradigm approach for Windows platform having GUI. This project has been developed as part of the Development of a machine translation System from Hindi to Punjabi Language.

1. Introduction

A word can be defined as a sequence of characters delimited by spaces, punctuation marks, etc. in case of the written text. A word can be of two types: simple and compound. A simple word or word consists of a root or stem together with suffixes and prefixes. A compound word (also called conjoined word) can be broken into two or more independent words. Each of the constituent words in a compound word is either a compound word or a simple word and may be used independently as a word. On the other hand, the root and the affixes, which are constituents of a simple word, are not independent words and cannot occur as separate words in the text. Constituents of a simple word are called morphemes or meaning units. The overall meaning of a simple word comes from the morphemes and their relationships [1]. Morphological Analysis is the process of finding the constituent morphemes in a word like cat +N +PL for word cats [2]. Morphological generator is the process of generating the word form taking stem word and its features (affixes) as input. Morphological Analysis is essential for Hindi it has a rich system of inflectional morphology as like other Indo-Aryan family languages. Main concern here is on the grammatical information of words and this grammatical information like gender, number, person etc. is marked through the inflectional suffixes. The rest of the paper is organized as follows. The next section discusses the motivation behind the development of the tool. Then brief about the Hindi morphology and Punjabi word classes along with the grammatical information will be explained. Next Section will cover the major drawbacks of existing Hindi Morphlogical Analyzer will be explained. Then a very brief overview of the approaches commonly followed for Morphological analysis is provided and the approach that we have followed is discussed. Next section provides the database design for this morphological analyzer and generator. After that implementation of this tool, and some of its features and its working scheme is provided. Providing a summary of the results concludes this paper.

2. Hindi Morphology

Morphology involves the study of inner structure of words and their forms in different uses and constructions. It can be mainly divided into two branches – derivational morphology and inflectional morphology. Derivational morphology involves the processes by which new lexemes are built from existing ones mainly through the addition of affixes. As an example in Hindi म + मेरा = ममेरा (Pronoun to Adjective) , like in English – go + at = goat (verb to noun) etc. Inflectional morphology involves the processes by which various inflectional forms are formed from a lexical stem. As an example in Hindi – inflectional forms of noun अतिथि (guest) are अतिथि (masculine-singular-direct), अतिथि (masculine-

978-0-7695-3267-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICETET.2008.11

oblique-singular), अतिथि(masculine-direct-plural), अतिथियों (masculine-oblique-plural). Hindi is very rich in inflectional morphology can be witnessed from the fact that in English usually there are maximum of 7-8 inflected word forms of noun but in Hindi it can be up to 40 and even more than that.

3. Hindi Word Classes

The first step while developing a morphological analyzer is to define the word classes and the grammatical information that will be required for words of these word classes natural language processing application for that language. After defining the word classes for hindi and the grammatical information that is required from the words of these word classes, various paradigms for these word classes were developed. Paradigm for a root word gives information about its possible word forms, in a particular word class, and their respective grammatical information. All the words of a word class may not follow the same paradigm. Like, it is not that all nouns will follow the same inflectional pattern. So, the first task was to find out the various paradigms for a word class and then group the words of that word class according to those paradigms. Proceeding this way paradigms were developed for the word classes which show inflection. For developing the paradigms the inflectional patterns of the root words of a word class were studied. And, then on their basis, the root words which inflect in the similar way were grouped. The inflection patterns for those groups constitute the set of paradigms for that word class. Following is the list of word classes along with their grammatical information that are being used for Hindi :

Noun: Grammatical information required for Hindi nouns is - gender, number and case. Gender can be masculine, feminine or both (as some nouns can be used both as masculine and as feminine). Number can be singular or plural. Case can be two types (in present work) – direct and oblique.

Pronoun: Grammatical information required is – number, case, person, and gender. Gender can be masculine, feminine or both. Number can be singular or plural. Case can be two types (in present work) – direct and oblique. Person can take first, second and third person.

Adjective: Grammatical information required for Hindi nouns is - gender, number and case. Gender can be masculine, feminine. Number can be singular or plural. Case can be two types – direct and oblique.

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

Adverb: There are two classes of adverbs, inflected and uninflected. Inflected adverbs behave like nouns so no separate paradigms were required for these. Grammatical information required for inflected adverbs will be same as required for nouns and for uninflected adverbs no grammatical information is to be stored.

Sharisthi Pronoun: Grammatical information required is – number, case, person, and gender, parsarg. Gender can be masculine, feminine or both. Number can be singular or plural. Case can be two types (in present work) – direct and oblique. Person can take first, second and third person. Parsarg will be shashthi.

4. Approaches Followed

In this paper a morphological analyzer and generator for Hindi is discussed, which is developed as part of the project on Development of Hindi to Punjabi Machine Translation System. The morphological analyzer that is discussed here gives preference to the second parameter i.e. time taken to search for a word in the database to know its grammatical information, and also accuracy of returned results. In the database used by this tool all the possible word forms of all root words are stored. Though it takes a bit more disk space but the search time is very less. It was seen that inflections patterns followed by Hindi words are not infinite or very large. A typical Hindi noun can have at the most 8 inflectional forms though on an average Hindi nouns take 3-4 forms. And Hindi verbs have on an average 48 forms. So it is very much possible to store all the word forms. Arguably it is not possible to store all the proper names i.e. person, place names etc. That is not taken into account here. Main focus here is on words of other word classes and also nouns, but excluding the proper nouns to some extent.

5. Motivation

First, The Morphological Analyzer is an integral part of any Natural Language Processing system, especially in the context of Indian. Indian Languages like Hindi, Punjabi etc are morphologically and inflectionally very rich languages. For developing the machine translation between two Indian languages, we need to have lexicon as the back bone of the system. If we had an exhaustive lexicon which listed all the word forms of all the roots, and along with each word form it listed its feature values then clearly we do not need a

morphological analyzer. Then we need only to look a given word in the lexicon and retrieve its feature values. But this method has several problems. First, it is extremely wasteful of memory space. Every form of the word is listed which contribute to the large number of enteries 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 relationship among different roots that have similar word forms. Thus, it fails to represent a linguistic generalization. Third, some languages have a rich and productive morphology like Hindi. The number of word forms might well be infinite in such a case. Clearly, this will not work with such category of languages.

The linguist or the language expert is asked to provide different tables of word forms covering the words in a language. Each word forms table covers a set of roots which means that the root follow the pattern (or paradigm) implicit in the table for generating their word forms. For example in Hindi the paradigm for laDkaa and other roots in its class can be specified by giving its word forms. Other word forms like kapaDaa (cloth) behave like laDkaa and belong to the same paradigm. Thus, paradigm can be extracted from the word forms of laDkaa by identifying the number of characters to be deleted from the root and the characters to be added to obtain the word forms. This leads to efficient storage because there is only one paradigm table for class of roots rather than a separate word forms table for each root.

Second, The morphological analyzer has also been developed by IIIT Hyderabad and it is available free to download from their LTRC website for Linux and Windows platform both. But it has lot of drawbacks (explained in next section) that motivates us for developing it. But, it is worth mentioning that the data has also been used from this morphological analyzer as well. Most of the NLP Community I interacted and got the feedback about their tool is that they faced a lot of problem in using installing it and using it.

6. Drawbacks of Existing System

The existing morphological analyzer has been developed by IIIT Hyderabad. It is freely downloadable for use. This is available for Windows and Linux platforms. The Linux version has been developed in CGI-Perl and Windows platform has been developed in C Programming language. During installation, the user has to face number of problems like sometimes, there are hard coded directory paths and face problems, it requires number of modules to be present for installation that a layman cannot understand how to install. Then this morphological analyzer has been developed for WX Encoding. The WX Encoding is defined by IIIT, Hyderabad and is not commonly used. It is difficult to first find the WX Encoding specification to use it. Like for example w to त , W to थ etc. So, a layman will not be using this morph until and unless one does not have the complete knowledge for computers. Then if we talk from the technical point of view, the software is following data dependence approach. Dependence approach means that the coding is dependent on the data file formats. If data file format is changed, the programming coded needs to be changed. This can also lead to inconsistent results. There are number of like Ca (For storing hindi grammatical categories), Ce (For storing the various features related to each grammatical category stored in Ca File), .p files (for storing the paradigms for all roots of the grammatical categories in the Ca file). Each Category listed in Ca file has corresponding .p file for storing the roots along with its word forms. Root file is used for storing the roots.

7. Features of Developed Morphological Analyzer and Generator

The above drawbacks has been tried to eliminate in the developed Morphological Analyzer and generator for Hindi language for Windows Platform. It has easy to use GUI for the users to operate and need not to have much knowledge about computers, platform and any programming language. Users just need some basic computer operation knowledge for software installation and operate. It has on the screen keyboard for typing in the word for analyzing and generation. It has been developed for Unicode format. If we talk from the technical point of view, it has been developed using Visual Basic and Front End and MS-Access as back End. The database follows database independence approach. It has been normalized up to third normal form of normalization.

8. Morphological Analyzer and Generator Database Schema

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. Details of six tables in database schema is described below(<Table Name>{<Comma Seperated list of columns>}):

1. Categories {catId (Primary Key), categoryName, mappedName} : All the possible categories in Hindi is stored.

2. CatFeatures{catFeatureID (Primary Key), catId (Foreign Key), gender, number, case, TAM, person, vibhakti, parsarg) : All the possible features for the categories are stored. If some features are not applicable for certain categories, they will be marked '—'
3. ParadigmRootDetails{paradigmRootID (Primary Key), paradigmRootName, catId (Foreign Key), paradigmRoot}: As we are following paradigm approach, all the possible paradigms for stem words will stored in this table.
4. SuffixInfo{paradigmId(Primary Key), paradigmName, paradigmRootId (Foreign Key), catFeatureId (Foreign Key) , suffDel, charCountDelete, suff_Add,}: All the possible paradigms are stored.
5. root{rootId (Primary Key), catId (Foreign Key), paradigmRootId (Foreign Key), rootWord, paradigmId (Foreign Key)}: All the possible root words are stored along with the inform that which paradigm it follows.

9. Conclusions

The Hindi morphological analyzer and generator discussed in this paper stores all the commonly used word forms for all Hindi root words in its database. This approach prefers time and accuracy to memory space. With the memory space not being a problem these days, neither in terms of cost nor in terms of storage requirements, this approach will perform better than the other approaches in which only root words and paradigms are stored in database. In those approaches search time is quite high due to the obvious reasons though they take less memory space. But the approach discussed in this paper prefers time to space. The search time based on this approach is very less. Another advantage of this approach is that the user will always get the accurate results. As sometimes suffix trimming approach to get possible root can result in some extra and indifferent results also. Therefore, this approach is recommended at least for the languages in which the number
of possible inflections for a word is not infinite or very high.

Acknowledgements. We thank Dr. Amba Kulkari, Reader and Head of the Department, Department of Sanskrit, University of Hyderabad, Hyderabad for solving my problems during the development of this morph. We thank Prof. Rajeev Sangal, Director, IIIT , Hyderabad and his team for making available morph developed at IIIT freely downloadable to NLP researchers.

10. References

[1] Bharati, Akshar, Vineet Chaitanya and Rajeev Sangal. (1995). Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi.
[2] Bharati, Akshar, Amba P. Kulkarni, Vineet Chaitanya. (1998a). Challenges in Developing Word Analyzers for Indian Languages, Presented at Workshop on Morphology, CIEFL, Hyderabad, July 1998.
[3] Bharati, Akshar, Rajeev Sangal and S.M. Bendre (1998b). Some Observations on Corpora of Some Indian Languages. Knowledge Based Computing Systems, Tata McGraw-Hill.
[4] Goldsmith, John. (2001). Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics, Vol 27, No. 2, pp 153-198.
[5] Daniel Jurafsky, James H. Martin.Speech and Language Processing: An introduction to speech recognition, natural language processing, and computational linguistics.
[6] LTRC, IIIT Hyderabad http://ltrc.iiit.ac.in
[7] Gill Mandeep Singh, Lehal Gurpreet Singh, Joshi S.S., A full form lexicon based Morphological Analysis and generation tool for Punjabi,International Journal of Cybernatics and Informatics, Hyderabad,India,October 2007, pp 38-47