ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

Lexicon Parser for syntactic and semantic analysis of Devanagari sentence using Hindi wordnet

Swati Ramteke[1], Komal Ramteke[2], Rajesh Dongare[3]

Student, Department of Computer Engineering, S.K.N. Sinhgad Institute of Technolgy and Science, Pune, India[1]

Assistant Professor, Department of Information Technology, Rajiv Gandhi College of Engineering and Research, Nagpur, India[2]

Programmer Analyst, ASP-BPI, Cognizant Technology Solutions Pvt. Ltd., Bangalore, India[3]

Abstract: The rule based approach used to developing lexicon parser rule, this approach uses the rules and a lexicon to resolve the word ambiguity. It developed for English, Sindhi and Chinese languages. In this paper we developed, Lexicon parser for syntactic and semantics analysis of Devanagari script sentence by using Hindi wordnet. The rule based parser generates a parse tree that is generated by using semantic relationship. In the given Devanagari (Hindi) sentence, the parser identifies grammatical meaning of the words. Testing is performed with new 150 sentences and accuracy at different levels is calculated. The accuracy of 89.33% was achieved from Lexicon parser.

Keywords: Rule based approach, Lexicon Parser, Hindi Wordnet

I. INTRODUCTION

The lexicon parser is developed for English, Sindhi and Chinese languages. But it has found that in Devanagari script there no lexicon parser is developed. The sentences should be syntactically and semantically correct in Devanagari. The sentences and words in this language are ambiguous, to this ambiguity problems are resolve by the lexicon parser rules.

Devanagari is the main script used to write standard Hindi, Marathi and Nepali. It has been the most commonly used script for Sanskrit. This script, having the origin in ancient Brahmi script.

Lexicon is a container for words belonging to the same language, where lots of words are common and where the differences may be marked word by word. It contains mostly words roots instead of complete words. The root are at the same time exclusive with respect to other words, where a root maps to a unique word and inclusive with to the different spellings of the same word. It is a declarative data structure which contains word entries and language rules.

Parser is analyzing the text, made of a sequence of tokens, to determine its grammatical structure with respect to a given formal grammar. Lexicon parser works on a grammar organized around lexemes, to which are attached syntactic and semantic processing instructions. Every lexicon uses some language of lexical descriptors to specify lexical information about each word or word sense. Although a language processing system's knowledge about a particular word may be quite extensive, often the bulk of the knowledge is about the meaning rather than the lexical constraints on the word. Therefore, lexical descriptions tend to be relatively small, and many words, though they may differ greatly in meaning, will have the same lexical description. Central to our method of mapping lexical structure is the notion of indiscernibility. Two words are indiscernible in a specific lexicon just in case that lexicon assigns the same lexical descriptors to the words. Parsing is the de-linearization of linguistic input. That is the use of grammatical rules and other knowledge sources to determine the functions of the words in the sentence. The need for unambiguous representation has lead to a great effort in stochastic parsing. Devanagari has a rich system of inflectional ending.

The proposed methodology uses a rule based parser. It consist of, process of analyzing input sentence of language syntactically and semantically. The main objective is to Design Lexicon Parser for syntactic and semantics analysis of Devanagari script sentence. Lexicon parser rules will be designed with the help of rule based approach, this approach uses the rules and a lexicon to resolve the word ambiguity.

A. Overview of Devanagari Script

Devanagari has evolved into a highly cursive script. Many languages in India, such as Hindi and Sanskrit, use Devanagari and many more languages throughout India use local variants of this script.

Hindu scriptures are written in Devanagari, a fact illustrated by the etymology of the name. "Devanagari" is a compound word with two roots: deva means "deity", and nagari means "city". Together it implies a script that is religious as well as urbane or sophisticated.

Devanagari is used in many Indian languages like Hindi, Nepali, Marathi, Sindhi etc. More than 300 million people

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

around the world use Devanagari script. This script forms the foundation of Indian languages. So Devanagari script plays a very major role in the development of literature and manuscripts. Devanagari script has about 11 vowels and 33 consonants.

B. Rule-based approach

In the rule-based approach, two components can usually be distinguished in an analyzer/generator [1]: a declarative component corresponding to linguistic knowledge and a procedural component which represent the analysis/generation strategy. Linguistic knowledge includes the grammar and the lexicon of the language while analysis/generation strategy is an algorithm which specifies in detail each of the operations involved in the process of analysis/generation.

Rule based consists of:
1. Process of analyzing input sentence of language syntactically and or semantically.
2. Process is controlled by dictionary and the rules.

Rule-based parsing proceeds by searching through the set of rules in a grammar (and lexicon) to determine which rules may jointly be applied to produce a well formed syntactic structure for a sentence of the language described in the grammar. If no analysis for a sentence can be found using the rules in the grammar, then the sentence is ungrammatical. If more than one analysis is found, then the sentence is syntactically ambiguous.

II. LEXICON PARSER FOR DEVANAGARI SENTENCES

A lexicon is a declarative data structure which contains word entries and language rules. We studied linguistic facts carefully from grammar books and observed that a word may have more than one meaning. In the traditional language lexicon, meaning and form of a word is stored. The information of words and their corresponding grammatical meanings are required. Therefore, two columns are defined in the lexicon database, one for words entry and second one for possible meaning of that word. The parser describe here is rule-based, not principle based, and in any case does not use any of the contemporary models of grammar.

Parsing is a process by which an input string is analyzed and assigned a suitable structure. The computation of the syntactic structure of a sentence involves the grammar and the parsing technique. The grammar is a formal specification of the structures allowable in the language and the parsing technique is the method of analyzing an input sentence to determine its structure according to the grammar. The parser takes input as a Hindi sentence and using the rule base, lexical analyzer, analyzes each word for the sentence .And returns the base form of each word along with their attributes. The information is analyzes to get relations among the words in the sentences using if-then rules.

The Modules are as follows:
1. Design Lexical Parser
2. Syntactic and Semantic analysis

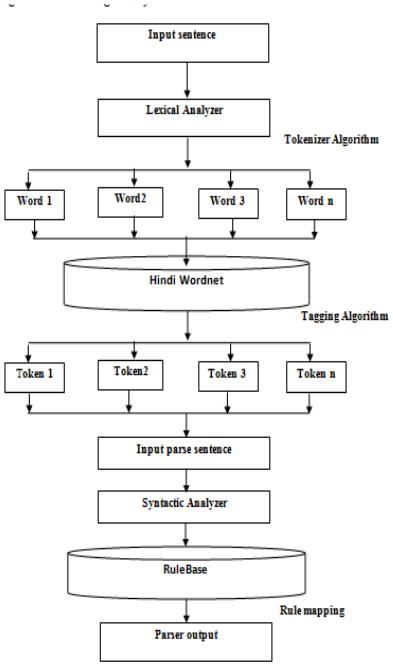

Fig.1 System architecture of Lexicon parser

A. Design Lexical Parser

The Devanagari (Hindi) sentence which is the input for our first module i.e. lexical parser it generates a parse tree that is generated by using semantic relationship.

A parser breaks data into smaller elements, according to a set of rules that describe its structure sequence of tokens (for example words), to determine its grammatical structure with respect to a given grammer. The semantic representation provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people without linguistic expertise who want to extract textual relations. The sentence relationships are represented uniformly as semantic relations between pairs of words. We have design our own lexical parser for getting the tag information and context-free structure grammar representation of source structure. The nouns, pronouns, verb, adverbs, adjectives etc are stored in a Data structure i.e. Collection (Array List).

1.1. Tokenization

The input Hindi Text is send to the tokenizer which divides the given phrase on the basis of spaces between them into constituents called tokens which are then passed to further phases.
For example, the sentence:
रवि बाहर जाता है is broken into tokens.

TABLE I
HOW THE SENTENCE BROKEN INTO TOKENS

रवि	Token1	संज्ञा (noun)
बाहर	Token2	क्रिया विशेषण (adverb)
जाता	Token3	क्रिया (verb)
है	Token4	क्रिया (verb)

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

1.2. Tagging Algorithm for Hindi text

1. Take input text
2. Tokenize input text.
3. Store all words into array WORD
4. Select each word one by one from array WORD.
5. Search and compare selected word from lexicon
6. If word is found one or more times, then store associated tag or tags of word into array TAGS
7. If one tag is stored in array TAGS, then display word with associated tag as an output.
8. Else select one or more linguistic rules and search most appropriate tag for a word by applying rules
9. Display word with associated tag as an output.

1.3. Ambiguity Resolution

The rules considering the tags for surrounding words are used for resolving ambiguities at different levels. Before the step of ambiguity resolution, each word is attached with number of tags. Since a particular word may have number of tags, there is need to check which tag is applicable to a particular word in a sentence, for example a word present in a nounlist in Hindi wordnet, can be tagged with a noun as well as an adjective tag. For this purpose, there is need to apply certain rules depending upon the grammatical category of preceding or succeeding words. These rules should be prioritized. First level of ambiguity exists when a particular word can have number of tags of different grammatical category. The rules should check the grammatical category for the surrounding words so that it can conclude the tag of that particular word. Eg. Consider the two sentences (Tum Khana khao) and (Tum Fhal khana). In the first sentence, 'khana' is a noun preceded by a verb. Whereas in the second sentence, verb followed by noun. Following figure shows the working of the first module i.e. lexical parser.

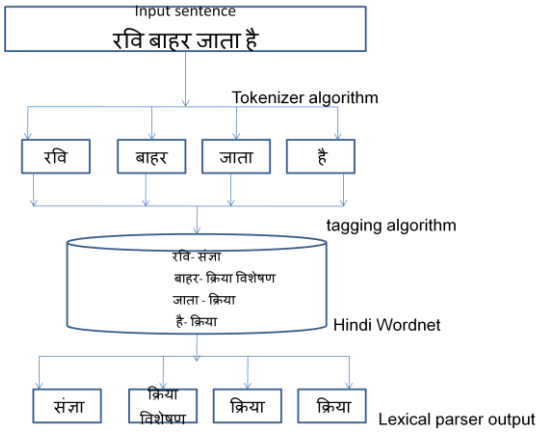

Fig.6 Illustration for Lexical Parser

B. Syntactic and semantic analysis

In the second module, Syntactic and semantic analysis. We have to Look up in Rule base we are mapping the tokens with Rules in the Rule base. This matching is semantic (meaningful) and check the structure of the sentence should be correct. The mapping is based on the relationship been established in the first module i.e. Lexical Parser. We have developed a ruled-based Hindi syntactic generator that determining the syntactic structure of the Hindi sentence. Structural mapping rules are used to determine the syntactic structure of the Hindi sentence by first recognizing a set of constituents from wordlist. Following figure illustrates a structural mapping rule that transforms the constituents into a feature structure that conforms to the syntactic rule that consists of a coordinator.

Rule 1:
Input: Hindi sentence
Output: syntactic structure of a Hindi Sentence
Structural mapping of SOV Rule:
If coordination from the input Words,
Subject (noun/pronoun+ ne) +
Object (noun/pronoun+ se/ko)+
verb
Then
Sentence= [coordination, subject, Object, verb]
End Rule

Fig.2 A structural mapping of SOV rule

Rule 2:
Input: Hindi sentence
Output: syntactic structure of a Hindi Sentence
Structural mapping of Subject+verb Rule:
If coordination from the input Words,
Subject (noun/pronoun+ ne)+
verb
Then Sentence= [coordination, subject, verb]
End Rule

Fig. 3 A structural mapping subject + verb rule

Rule 3:
Input: Hindi sentence
Output: syntactic structure of a Hindi Sentence
Structural mapping of noun / pronoun
+ adverb+verb Rule:
If coordination from the input Words,
noun/pronoun +
adverb+ verb
Then Sentence= [coordination, noun/pronoun, adverb, verb]
End Rule

Fig. 4 A structural mapping of noun/ pronoun +adverb +verb rule

Once the syntactic structure of a sentence is determined, another set of rules are used to ensure the agreement relations between various elements in the sentence. We have implemented rules for different types of agreement relationships, such as subject–verb, noun–adjective-verb, pronoun–noun-verb etc.

Algorithm for Semantic mapper

Step 1: The output from the first module i.e. Lexical parser acts as input to the Semantic/Syntactic Analyzer

Step 2: The tokens generated from the first module are stored in Data Structure i.e. Collection. These tokens have grammatical relations.

Step 3: Look up in Rule base we are mapping the tokens with Rules in the Rule base.

This matching is semantic (meaningful) and check the structure of the sentence should be correct.

Step 4: After matching the selected sentence kept as another data structure.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

Step 5: identify which rule satisfied the condition.
Step6: If satisfied with condition then display the message "The sentence is matched with defined rule".

Following figure shows the working of the second module i.e. Syntactic Analyzer

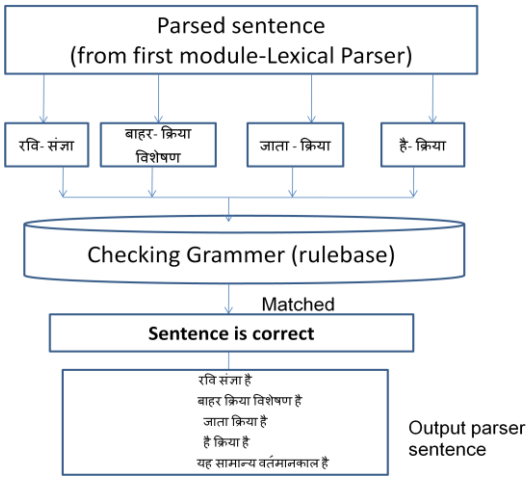

Fig. 7 Illustration for syntactic Analyzer

III. RESULT ANALYSIS

In this study, we measure the efficiency of Lexicon parser system for Hindi sentences. In this regard, Hindi sentences used for testing. The training corpus is gathered from "Hindi Wordnet" because all linguists of Hindi language are agreed that all Hindi words are found in this dictionary.

The corpora were manually tagged with our own tagset. 150 sentences were selected for testing. The lexicon contains approximate 36223 tagged words, among them the frequency of noun is 12890, pronoun is 137, verb is 11294, adjective is 6259, adverb is 5643.

During training and testing data, we have classified the words according to the part of speech and calculated the accuracy of each type of sentences.
The evaluated results are shown in Table:

TABLE 2
HOW THE SENTENCE BROKEN INTO TOTAL NUMBERS OF WORDS CONTAIN LEXICON LIMITED LEXICON AND RULES WERE USED IN SYSTEM.

Part of Speech	Total Words
Noun	12890
Pronoun	137
Verb	11294
Adverb	5643
Adjective	6259

When more sentences are tested and rules will be added then accuracy will be increased.
The GUI of Lexicon parser is classified into three windows:
(1) Browser Text box (2) Output window (3) List of words.

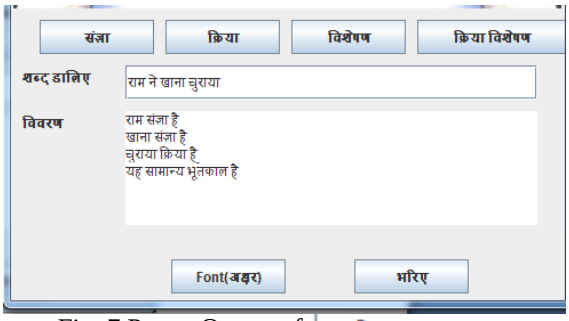

Fig. 7 Parser Output of राम ने खाना चुराया

The system read Hindi text; when terminator symbol is found then system compare each word of sentence with lexicon entries. If sentence is matched with the defined rule then information will display: "The sentence matched with defined rule" and output of the parser. If word in text found having multiple tags then system search most appropriate rule for the word. By applying rule, system selects most appropriate tagged word according to the context of sentence. If word is not found in the lexicon, then system will not display the information.

TABLE 3
CALCULATED ACCURACY OF SENTENCES

Tested sentences	Correct matched sentences	Accuracy (%)
150	134	89.33%

Testing is performed with new 150 sentences and accuracy at different levels is calculated. The first phase which resolves the ambiguity for different grammatical category and assigns tag to each word in a sentence was found to have approximately 89.33% accuracy.

IV. CONCLUSION

In this paper we developed lexicon Parser for Devanagari script (Hindi), it shows how a Hindi sentence is parsed into tokens and then find the relationship between tokens using grammar and by using semantic representation generate a parse tree. Rule based approach used to resolves the disambiguity of words. In this study, lexicon and word disambiguation rules were discussed and presented. Tagging and tokenization algorithms were developed and implemented for Devanagari text.
The accuracy of 89.33% was achieved from Lexicon parser. During the experiments, it has been observed that the accuracy was low when we tested more ambiguity sentences and sentences of future tense. Similarly, when we tested sentences of simple present and past tenses then the accuracy was very high.

REFERENCES

[1] Javed Ahmad MAHAR, Ghulam Qadir MEMON(2010), "Rule Based Part of Speech Tagging of Sindhi Language", International conference on Signal Acquisition and processing.,pp.101-106.

[2] Pawan Goyal, Vipul Arora, Laxmidhar Behera (2009), "ANALYSIS OF SANSKRIT TEXT:PARSING AND SEMANTIC RELATION" ,Springer-Verlag Berlin Heidaelberg., pp. 200-218.
[3] Ms Vaishali M. Barkade et. al. (2010) , "English to Sanskrit Machine Translation Semantic Mapper", International Journal of Engineering Science and Technologyvol.2(10).
[4] Ms Vaishali M. Barkade.Prof. Prakash R. Devale, Dr.Suhas H. Patil (2010) , "English to Sanskrit Machine Translator Lexical Parser and Semantic Mapper", National Conference On"Information and Communication Technology"(NCICT-10).
[5] Khaled Shaalan (2010), "Rule-based Approach in Arabic Natural Language Processing", International Journal on Information and Communication Technologies, Vol. 3, No. 3,.
[6] Ji, Luning; Lu, Qin; Li, Wenjie; Chen, YiRong (2007)," Automatic Construction of core Lexicon For Specific Domain", Sixth International Conference on Advanced Language Processing and Web Information Technology,pp.183-188.
[7] Drew McDermott(2005)," Lexiparse: A Lexicon-based parser for Lisp Application", Draft manual 0.93.
[8] De Lucia, A.; Di Penta, M.; Oliveto, R. (2011),"Improving Source code Lexicon via Traceability and Information Retrieval" IEEE Transaction on Software Engineering,pp.205-227.
[9] Akshar Bharati and Rajeev Sangal(1993), "Parsing Free Word Order Languages in the Paninian Framework", ACL93: Proc. of Annual Meeting of Association for Computational Linguistics,New Jersey, pp. 105-111.
[10] Chaudhuri, B.B.; Pal, U.(2002), An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi)" International conference on pattern recognition.
[11] Vijay Kumar, Pankaj K. Sengar(2010), "Segmentation of Printed Text in Devanagari Script and Gurmukhi Script", International Journal of Computer Applications Volume 3 – No.8, pp. 0975 – 8887.
[12] Shilpa Desai, Ramdas Karmali, Sushant Naik, Shantaram Walawalikar and Damodar Ghanekar, "Tools for IndoWordNet Development"