2009 International Conference on Advances in Recent Technologies in Communication and Computing

DEVELOPMENT OF A POS TAGGER FOR MALAYALAM-An Experience

Manju K, Soumya S, Sumam Mary Idicula
Department of Computer Science
Cochin University of Science and Technology
Kochi-22, Kerala, India
Email: manju2007mec@gmail.com, ps.soumya02@gmail.com, sumam@cusat.ac.in

***Abstract*—A Parts of Speech tagger for Malayalam which uses a stochastic approach has been proposed. The tagger makes use of word frequencies and bigram statistics from a corpus. The morphological analyzer is used to generate a tagged corpus due to the unavailability of an annotated corpus in Malayalam. Although the experiments have been performed on a very small corpus, the results have shown that the statistical approach works well with a highly agglutinative language like Malayalam**

Keywords-Dravidian Language; Morphemes; HMM; Viterbi; Tagset.

I. INTRODUCTION

Parts of Speech Tagging (grammatical tagging), is a process of marking the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its context[1]. This is the most common step for creating an annotated corpus. Annotated corpora find its major application in various NLP related applications like Speech Recognition, Text to Speech Conversion, Information retrieval, Word sense disambiguation etc. This proves to be a basic building block for constructing statistical models for automatic processing of natural languages. Many such corpora are available for languages across the world and have proved to be a useful step towards natural language processing. Many works related to POS tagging are being carried out in the NLP field.

Parts of speech are defined based on the morphological and syntactic behavior of the words. Assigning a POS tag to each word of an un-annotated text manually is a tedious task. And that is why POS Tagging has become one of the well-studied problems in the field of NLP.

There are two distinct approaches for POS Tagging-Rule based and Stochastic approaches [1]. Rule based approach uses a large database of hand-written disambiguation rules considering the morpheme ordering and contextual information. The Stochastic approach uses an unambiguously tagged text to estimate the probabilities to select the most likely sequence. For selecting the maximum likelihood probability the lexical generation probability and the n-gram probability are considered. The most common algorithm for implementing an n-gram approach is the Viterbi Algorithm which follows a Hidden Markov Model [1][3]. A lot of work has been done in part of speech tagging of western languages. These taggers vary in accuracy and also in implementation.

In this paper, we propose a part of speech tagger for Malayalam which uses a stochastic approach. The word frequencies and bigram probabilities are calculated from the training corpus. Since Malayalam has no explicit annotated corpus available, we developed a morphological analyzer in our system to generate a tagged corpus. We can compare the output of the statistical method with the morph analyzer to verify the accuracy of the system. Morphological analyzer gives the probable tags for the words. The rule based tagger within the system makes the tags un-ambiguous.

The paper is organized as follows. Section 2 gives a brief description on Malayalam language. Section 3 describes the architecture of the proposed system. Section 4 discuss about the related works done in this area. Section 5 describes the generation of the tagged corpus using the Morph analyzer, while Section 6 describes on obtaining the statistical data from the corpus. In Section 7 we discuss the results we have obtained for a small number of experiments. Finally the paper ends with some concluding remarks.

II. MALAYALAM

Malayalam is spoken primarily in Southern Coastal India by over 35 million speakers. Malayalam has its own distinct script, a syllabic alphabet consisting of independent consonant and vowel graphemes plus diacritics. Malayalam belongs to the Dravidian family of languages and is one of the four major languages of this family with a rich literary tradition. Morphologically Malayalam is richly inflected by the addition of suffixes with the root/stem word. Malayalam is a language registering a heavy amount of agglutination. The origin of Malayalam as a distinct language may be traced to the last quarter of 9th Century A.D. Malayalam has a special place in the classification of world languages. It is from Tamil that Malayalam was born. However, it is from the traditions of Sanskrit, the Indo-Aryan language, that Malayalam draws its rich diversity of words and compound alphabets (conjuncts). This dynamic synthesis of diversities has been achieved by no other Indian languages [2].

There are at least five main regional dialects of Malayalam and a number of communal dialects. Many words have been borrowed from Sanskrit. There are 37 consonants and 16 vowels in the script [2][5]. Malayalam

978-0-7695-3845-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ARTCom.2009.98

has a written traditional dating back from the late 9th century and the earliest work dates from 13th century. The script used is called Kolezhethu (Rod-script) which is derived from ancient Grandha Script. Malayalam differs from other Dravidian language as the absence of personal endings on verbs. It has a one to one correspondence with the Indo Aryan Devanagari syllabary.

III. THE PROPOSED METHOD

The overall architecture of the system including the connections between the modules is shown in Fig: 1.

The process follows mainly three steps. If the training corpus is not available, as a first step, it uses the morphological analyzer to generate the tagged morphemes. Now on, this is the training corpus. In the second step the statistical analyzer module compiles the statistical data of the training corpus using the unigram and bigram probability. Following this, the main module of the system, the tagger module, determines the parts of speech of the morphemes of the Test set.

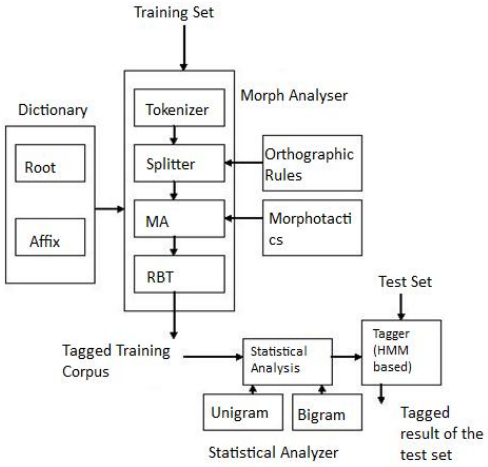

Figure 1. Architecture of the System

IV. RELATED WORKS

There are a number of attempts made for generating POS Taggers nationally and internationally. Most of the POS Taggers available are in English. POS taggers for English, such as Bril tagger, Tree tagger, CLAWS tagger, online tagger ENGTWOL are some classic examples [8]. These methods mostly used rule based, stochastic or morphological inputs. However, the analysis of Indian languages is a complex procedure. Many attempts have been made for different Indian languages. Hindi Morphological tagger, Marathi POS tagger, Tamil spell checker, Morphology driven Manipuri POS Tagger are some of the examples[6][7]. Rule based techniques, Finite state machine, morphological dictionary are some of the techniques used in works like ANUBHARATI, developed by IIT Kanpur, ANUVADHINI MT system for Bengali, and Tamil spell checker developed at AU, Chennai[9][10]. In Tamil spell checker, morphological dictionary is internally represented by a set of FSTs that are automatically generated from a more general dictionary containing morphological syntactic information. A Punjabi spell checker has been developed using Rule cum Dictionary based method. It is a dictionary with search algorithms, which searches string matching, fuzzy search and suffix stripping etc. Morphological Stripping Method, Paradigm based FSTs are some other techniques used in Text Analyzers.

In Indian Languages natural language processing tools are very less as compared to English and other European languages. In Hindi a rule based parts of speech tagger developed by IIT, Bombay and that has been used in stemmer and morphological analyzer for Word Net project. In Dravidian Languages, when compared to Tamil, only a very less work in Malayalam has done. In Tamil AU-KBC developed their own morph analyzer and parts of speech tagger. But it is a rule-based system and its accuracy is less, and it also showed word sense disambiguation problem for machine translation system. Through this work reported in this paper, it can be shown that these problems can be alleviated for Malayalam language using stochastic approach.

V. GENERATION OF THE TAGGED CORPUS

To perform parts of Speech tagging using stochastic technique, an annotated corpus is needed. Since the language Malayalam, has no annotated corpus and no explicit morphological analyzer to perform morphological analysis, a Morph Analyzer is developed in the system to generate a tagged corpus from the training set.

As the language Malayalam has a rich structure in the morphological sequences, these sequences are modeled using deterministic finite automata. The deterministic FSA is used to solve the problem of Morphological Recognition.

The Morph Analyzer accepts the input text through the soft keyboard. This text can have more than one sentence. On submitting the text, the text is transliterated to an intermediate representation and is stored as a file. This representation is used while traversing the FSA. Now each sentence is given to the Tokenizer. The token is checked with the dictionary to check if it is a valid word. If not, then the word (token) is given to the Splitter where the word is separated into root and affix based on the orthographic rules. After Identifying the Root, the analyzer searches the affix based on the morphotactics of the category of the root word. This is the morphologically Tagged result.

Malayalam is a language which is inflectionally rich; there is a very small possibility of ambiguity in the morphologically analyzed result. If any ambiguity exists that can be removed by giving the result to the Rule based Tagger. This is done by writing rules for those specific cases. By using the Morph Analyzer the tagged corpus is generated.

A. Dictionary

The different dictionaries are namely the affix which contains prefix or suffix information and root containing nearly 1500 entries are used by the system. The format of root is <root><Category>.

B. Tokenizer

Given a character sequence and defined documentation unit, tokenization is the job of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. A token is an instance of character sequence in some particular document. Tokenizer converts the input text into tokens.

C. Splitter

Splitter splits compounding words into pre-existing morphemes to form a new word. If the input word is not in the lexicon or it is a compound word then the splitter will handle such cases. The compound splitter works by recursively analyzing each word to see whether it can be split into a sequence of concatenated words.

E.g. ഞൊണ്ടിമലയാളിയാണ് is changed to ഞൊണ്ടിമലയാളിആണ്

The function split takes a string, i.e., a potentially complex word, as argument. Splitting is done by applying sandhi rules.

D. Rule based Tagger

Rule based tagging is used to resolve ambiguity, if any in the morphologically generated result. This is done by giving hand written rules for those specific cases. As far as the language Malayalam is concerned there is a very little probability of ambiguity after the morphologically analyzed result. Here those ambiguities are eliminated based on the affix attached to the root word. By a look ahead on the affix, the appropriate tag for the root morpheme is assigned.

VI. FSA FOR MALAYALAM

This section discusses how the FSA can be conceived by applying to Malayalam words [4]. The automaton is represented as a directed graph: a finite set of vertices (nodes), together with a set of directed links between pairs of vertices called arcs. Each node corresponds to a state. States are represented as circles with name tags in them. Arcs are represented by arrows going from one state to another state. The final states are represented by two concentric circles. The machine starts at the initial state, runs through a sequence of states by computing a morpheme in each transition. If it matches the symbol on an arc leaving the current state, then it crosses that arc, and moves to the next state, and thus, advances one symbol in the input.

Each state through which the speaker passes represents the grammatical restrictions that limit the choice of the next morpheme. Such a process gets iterated until the machine reaches the final state, successfully recognizing all the morphemes in the input string. But if the machine gets some input that does not match an arc, then it gets stuck there and never gets to the final state. This is considered as the FSA machine rejecting or failing to accept an input. The path moves from the initial point on the left to the final point on the right, proceeding in the direction of arrows. Once the arrow moves one step, there is no backward movement (Of course, recursion of an item can be shown by using closed loops). The resulting FSA is deterministic in the sense that given an input symbol and a current state, a unique next state is determined.

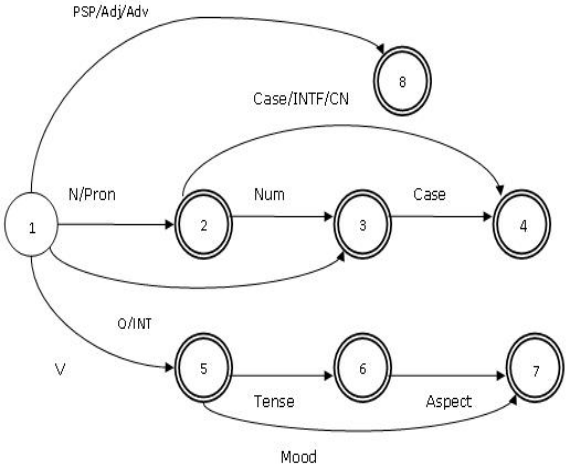

Figure 2. FSA for Malayalam

VII. OBTAINING STATISTICAL DATA FROM CORPUS

The statistical analyzer extracts unigram, bigram probabilities from the training corpus [3]. In calculating the n-gram probabilities, the number of times each word (unigram), two words sequence (bigram) occur in the corpus is determined, for each possible sequences of parts of speech of these words. As a result, given a word (unigram) or a word sequence (bigram), the probability that it occurs with a particular tag or tag sequence among all possible tags or tag sequences is determined

A. Unigram tagger

The unigram (n-gram, n = 1) tagger is a simple statistical tagging algorithm. For each token, it assigns the tag that is most likely for that token's text. Before a unigram tagger can be used to tag data, it must be trained on a training corpus. It uses the corpus to determine which tags are most common for each word. The unigram tagger will assign the default tag none to any token that was not encountered in the training data.

B. Bigram tagger

Bigram assumption: the probability of a tag appearing depends only on the previous tag.

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

Bigrams are groups of two written letters, two syllables, or two words; they are a special case of N-gram. Bigrams are used as the basis for simple statistical analysis of text. The bigram assumption is related to the first-order Markov assumption.

C. Tagging Using Statistical Data

After the training phase where relevant statistical data have been collected from the training corpus, the tagger is activated on the test corpus. The tagger employs a sentence based approach rather than a word based approach. That is, first all the possible tags for the words and the word sequences in the sentence are determined, and then the combination of the tags with the highest probability for the whole sentence is selected.

The intuition behind HMM (Hidden Markov Model) and all stochastic taggers is a simple generalization of the "pick the most likely tag for this word" approach. The unigram tagger only considers the probability of a word for a given tag t; the surrounding context of that word is not considered. On the other hand, for a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes the following formula:

P (word | tag) * P (tag | previous n tags).

For finding the maximum probability HMM uses the Viterbi Algorithm.

D. Viterbi for POS tagging

- Initialization step

For i=1 to N do
Seqscore(i,1)=Prob(w1|Li)*Prob (Li |Ø)
Backptr(i,1) =0;

- Iteration step

For t=2 to T
For i=1 to N
Seqscore(i,t)=$MAX_{j=1,N}$(Seqscore(j,t-1)*
Prob(L_i |L_j))*Prob(w_t| L_i)
Backptr(i,t) = index of j that gave the max above
Sequence

- Identification step

C (T) = i that maximizes Seqscore (i,T)
For i= T-1 to 1 do
C (i) = Backptr(C(i+1),i+1)

w_1, ….., w_T : Word sequence
L_i, ……., L_N : Lexical categories
Prob (w_t| L_i) : Lexical probability
Prob (L_i | L_j) : Bigram probability

E. POS tag set

Malayalam language is made inflectionally rich in morphology [5], by adding suffixes with the root / stem word. Since words are formed by the suffix addition with root, most of the words can take the POS tag based on the root or stem. Hence in Malayalam the suffixes play major role in deciding the POS of the word. The tag set based on Pen Treebank developed for this work is given in Table1 and the tag sequence considered for the tagger reported in this work is given in Table2.

TABLE I. TAGSET FOR TAGGING

TAG	DESCRIPTION
N	Noun
Pron	Pronoun
Acc	Accusative
Soc	Sociative
Dat	Dative
Gen	Gentive
Loc	Locative
Adj	Adjective
V	Verb
VN	Verbal Noun
Adv	Adverb
Dem	Demonstrative
Q	Quantifier
PSP	Postposition
RP	Particles
INTF	Intensifier
CN	Conjunction
INT	Wh Words

TABLE II. TAGSEQUENCE FOR TAGGING

Adj N PSP N N V VN PSP
Adv DEM N Q N Adj N PSP V
N Adj N N Adj N V
N VN PSP
PRON Adv N PSP N PSP V
Adv N PRON PSP V
Adv N N N PSP V
N Adj N PSP V N V
N V

VIII. RESULT

After training the system using the tagged corpus, the system was tested with the test case. For tagging the test case, both the lexical generation probability and the emission probability were used. Following results were obtained while testing the test data with the system.

The tagger was trained with using about 1,400 tokens. By increasing the tokens the accuracy of the system can be increased. The POS Tagger developed gave an accuracy of about 90%. For performing statistical tagging, we have considered only 10 tag sequences, and the result obtained from the Statistical Analyzer was very satisfactory. Almost

80% of the sequences generated automatically for the test case were found correct, when compared with the manually tagged result for those sentences.

IX. Conclusion

This Parts of Speech Tagger developed for Malayalam used the statistical approach, HMM. POS Tagging have good applications in processes like parsing, text-based information retrieval, speech recognition, etc. The POS Tagger developed in this work was able to assign tags to almost all the words in the test case. Due to the unavailability of an annotated corpus in Malayalam Language, in this work, we developed a morphological analyzer, which gave the parts of speech of words, independent of the corpus. This paper highlights that a statistical approach is very much suitable for highly agglutinative languages like Malayalam. The proposed system can be made more efficient by extending bigram probability to trigram or n-gram probability.

References

[1] Jurafsky D and Martin J H , Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition , Pearson Education Series 2002

[2] Dr. C J Roy, Kerala Paaniniiyam(Translation)

[3] JamesAllen, Natural Language Understanding, Benjamin/Cummings Publishing Company,1995

[4] Kalyanamalini Sahoo, Oriya Nominal Forms: a Finite State Processing, IEEE2003 (TENCON2003: Conference on Convergent Technologies For Asia-Pacific Region).

[5] Sumam Mary Idicula and Peter S David, A Morphological processor for Malayalam Language, South Asia Research, SAGE Publications, 2007

[6] Thoudem Doren Singh and shivaji Bandyopadhyay, Morphology Driven Manipuri POS tagger, In: Proceedings of the IJCNLP-08 Workshop on NLP for less privileged languages,pp 91-98, Hyderabad, India(2008).

[7] Manish Shrivastava and Pushpak Bhattacharya, Hindi POSTagger Using Naive Stemming : Harnessing Morphological Information Without Extensive Linguistic Knowledge, In: proceedings of ICON-2008

[8] Eric Brill, A simple rule-based part of speech tagger. In: Proceedings of Third conference of applied natural language processing, pp. 112 -- 116. Trento, Italy (1988).

[9] AU-KBC Research Centre WebPage http://www.au-kbc.org/research_areas/nlp/projects/postagger.html

[10] S. Lakshmana Pandian and T. V. Geetha, Morpheme based Language Model for Tamil Part-of-Speech Tagging, Webpage http://www.gelbukh.com/polibits/38_02.pdf