Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.3

www.iiste.org

Parts Of Speech Tagger and Chunker for Malayalam – Statistical Approach

Jisha P Jayan
Department of Tamil University
Tamil University, Thanjavur
E-mail: jishapjayan@gmail.com

Rajeev R R
Department of Tamil University
Tamil University, Thanjavur
E-mail: rajeevrrraj@gmail.com

Abstract

Parts of Speech Tagger (POS) is the task of assigning to each word of a text the proper POS tag in its context of appearance in sentences. The Chunking is the process of identifying and assigning different types of phrases in sentences. In this paper, a statistical approach with the Hidden Markov Model following the Viterbi algorithm is described. The corpus both tagged and untagged used for training and testing the system is in the Unicode UTF-8 format.

Keywords: Chunker, Malayalam, Statistical Approach, TnT Tagger, Unicode

1. Introduction

Part of Speech Tagging and Chunking are two well-known problems in Natural Language Processing. A Tagger can be considered as a translator that reads sentences from certain language and outputs the corresponding sequences of part of speech (POS) tags, taking into account the context in which each word of the sentence appears. A Chunker involves dividing sentences into non-overlapping segments on the basis of very superficial analysis. It includes discovering the main constituents of the sentences and their heads. It can include determining syntactical relationships such as subject-verb, verb-object, etc., Chunking which always follows the tagging process, is used as a fast and reliable processing phase for full or partial parsing. It can be used for information Retrieval Systems, Information Extraction, Text Summarization and Bilingual Alignment. In addition, it is also used to solve computational linguistics tasks such as disambiguation problems.

Parts of Speech Tagging, a grammatical tagging, is a process of marking the words in a text as corresponding to a particular part of speech, based on its definition and context. This is the first step towards understanding any languages. It finds its major application in the speech and NLP like Speech Recognition, Speech Synthesis, Information retrieval etc. A lot of work has been done relating to this in NLP field.

Chunking is the task of identifying and then segmenting the text into a syntactically correlated word groups. Chunking can be viewed as shallow parsing. This text chunking can be considered as the first step towards full parsing. Mostly Chunking occur after POS tagging. This is very important for activities relating to Language processing.

A lot of work has been done in part of speech tagging of western languages. These taggers vary in accuracy and also in their implementation. A lot of techniques have also been explored to make tagging more and more accurate. These techniques vary from being purely rule based in their approach to being completely stochastic. Some of these taggers achieve good accuracy for certain languages. But unfortunately, not much work has been done with regard to Indian languages especially Malayalam. The existing taggers cannot be used for Indian languages. The reasons for this are: 1) The rule-based taggers would not work because the structure of Indian languages differs vastly from the Western languages and 2) The stochastic taggers can be used in a very crude form. But it has been observed that the taggers give best results when there is some knowledge about the structure of the language.

The paper presented here is as follows. The second section deals with the statistical approach towards POS tagging and Chunking. The third section explains the tagset for POS tagging and chunking for Malayalam. Fourth section deals with the result. The fifth section concludes the paper.

2. Statistical Approaches towards Tagging and Chunking

The statistical methods are mainly based on the probability measures including the unigram, bigram, trigram and n-grams.

A Hidden Markov Model (HMM) is a statistical model in which the system modeled is thought to be a Markov process with the unknown parameters. In this model, the assumptions on which it works are the probability of the word in a sequence may depend on its immediate word presiding it and both the observed and hidden words must be in a sequence. This model can represent the observable situations and in POS tagging and Chunking, the words can be seen themselves, but the tags cannot. So HMM are used as it allows observed words in input sentence and hidden tags to be build into a model, each of the hidden tag state produces a word in a sentence.

With HMM, Viterbi algorithm, a search algorithm is used for various lexical calculations. It is a dynamic programming algorithm that is mainly used to find the most likely of the hidden states, results in a sequence of the observed words. This is one of the most common algorithms that implement the n-grams approach. This algorithm mainly work based on the number of assumptions it makes.

The algorithm assumes that both the observed and the hidden word must be in a sequence, which corresponds to the time. It also assumes that the tag sequence must be aligned. One of the basic views behind this algorithm is to compute most likely tag sequence occurring with the unambiguous tag until the correct tag is obtained. At each level most appropriate sequence and the probability including these are calculated.

In Malayalam, there are many chances where each word may come up with different tags. This is since Malayalam is morphologically rich and agglutinative language. According to the context also there are chances where the tags may be given differently.

3. Tagset for POS Tagging and Chunking

Malayalam belongs to the Dravidian family of languages, inflectionally mainly adding of suffixes with the root or the stem word forms rich in the morphology.

Since words are formed by the suffix addition with root, the word can take the POS tag based on the root / stem. Hence it can be stated that the suffixes play major role in deciding the POS of the word.

Table 1. Tagset for Parts of Speech Tagging

Sl.No	Main Tags	Representation
1	Noun	NN
2	Noun Location	NST
3	Proper Noun	NNP
4	Pronoun	PRP
5	Compound Words	XC
6	Demonstration	DEM
7	Post Position	PSP
8	Conjuncts	CC
9	Verb	VM
10	Adverb	RB
11	Particles	RP
12	Adjectives	JJ
13	Auxiliary Verb	VAUX
14	Negation	NEG
15	Quantifiers	QF
16	Cardinal	QC
17	Ordinal	QO
18	Question Words	WQ
19	Intensifiers	INTF
20	Interjection	INJ
21	Reduplication	RDP
22	Unknown Words	UNK
23	Symbol	SYM

For Chunking, mainly six tags are used. This is based on the grammatical or the syntactical category.

Table 2. Tagset for Chunking

Sl.No	Main Tags	Representation
1	Noun Phrase Chunk	NNP
2	Verb Finite Chunk	VGF
3	Non-Finite Verb Chunk	VGNF
4	Conjunction Chunk	CCP
5	Verb Chunk Gerund	VGNN
6	Negation Chunk	NEGP

4. Trigrams N Tag (Tnt)

TnT tagger is proposed by Thorsten Brants and in literature its efficiency is reported as one of the best and fastest on different languages such as German, English, Slovene and Spanish. TnT is a statistical approach, based on a Hidden Markov Model that uses the Viterbi algorithm with beam search for fast processing. TnT is trained with different smoothing methods and suffix analysis. The parameter generation component trains on tagged corpora. The system uses several techniques for smoothing and handling of unknown words.

TnT can be used for any language, adapting the tagger to a new language; new domain or new tagset is very easy. The tagger is implemented using Viterbi algorithm for second order Markov models. Linear interpolation is the main paradigm used for smoothing and the weights are determined by deleted interpolation. To handle the unknown words, suffix trie and successive abstraction are used. There are two types of file formats used in TnT, untagged input for tagger and the tagged input for tagger.

Trigrams N Tags (TNT) is a stochastic HMM tagger based on trigram analysis, which uses a suffix analysis technique based on properties of words like, suffices in the training corpora, to estimate lexical probabilities for unknown words that have the same suffices. Its greatest advantage is its speed, important both for fast tuning cycle and when dealing with large corpora. The strong side of TnT is its suffix guessing algorithm that is triggered by unseen words. From the training set TnT builds a trie from the endings of words appearing less than n times in the corpus, memorizes the tag distribution for each matrix. A clear advantage of this approach is the probabilistic weighting of each label, however, under default settings the algorithm proposes a lot more possible tags than a morphological analyzer would.

5. System Testing and Result

The application of TnT has two steps. In step 1, the model parameters are created from a tagged training corpus. In step 2 the model parameters are applied to the new text and actual tagging is performed. The parameter generation requires a tagged training corpus in the prescribed format.. The training corpus should be large and the accuracy of assigned tags should be as high as possible.

The system is trained using the manually tagged corpus. The words and tags are taken from the training file to build a suffix tree data structure. In this tree structure the word and tag frequency are stored and the letter tree is build taking the word and its frequency as the argument. While training, the transition and emission property matrix are calculated and the models of the language are building. The lexicon file created during the generation of the parameter contains the frequencies of the words and its tags, which occurred in the training corpus. A hash of the tag sequence and its frequency is build. This is used for determining the lexical probability. The n-gram file that is also generated during the parameter generation contains the contextual frequencies for the unigrams, bigrams, and trigrams. While testing Viterbi algorithm is applied to find best tag sequence for a sentence. If tag sequence is not present smoothing techniques are applied according to runtime arguments of the postagger.

5.1 System Testing

After training the system using the manually tagged corpus, the system can be now tested with the raw or untagged corpus. For the tagging of the raw corpus, both the files, which contain the modal parameter for the lexical and the contextual frequencies, are required.

5.2Result

Following results were obtained while testing the raw corpus with in the system. The raw corpus used for testing was in Unicode.

For training the system, ie for in the training phase, the tagger and chunker were trained with using about 15,245 tokens. Increasing the accuracy of the system can increase this further to any extent there.

In case of Parts of Speech Tagging, Comparing 200 tokens,

Overall result,

Equal : 181/200 (90.5%)
Different : 19/200 (9.5%)

In case of Chunking, Comparing 200 tokens,

Overall result,

Equal : 184/200 (92.00%)
Different : 16/200 (8.00%)

For chunking, the system gives about 92% accuracy while for POS tagging it gave about 90.5% accuracy.

6. Conclusion

Part-of-speech tagging now is a relatively mature field. Many techniques have been explored. Taggers originally intended as a pre-processing step for chunking and parsing but today it issued for named entity recognition, recognition in message extraction systems, for text-based information retrieval, for speech recognition, for generating intonation in speech production systems and as a component in many other applications. It has aided in many linguistic research on language usage. The Parts of Speech Tagging and Chunking for Malayalam using the statistical approach has been discussed. The system works fine with the Unicode data. The POS and Chunker were able tot assign tags to all the words in the test case. These also focus on the point that a statistical approach ca also work well with highly morphologically and inflectionally rich languages like Malayalam.

References

T. Brants. TnT — A Statistical Part of-Speech Tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP-2000)*, pages 224–231, 2000.

Eric Brill, A simple rule-based part of speech tagger. In *Third Conference on Applied Natural Language Processing*. 1992.

S. Abney. Tagging and Partial Parsing. In K. Church, S. Young, and G. Bloothooft (eds.), Corpus-Based Methods in Language and Speech. 1996.

Abney S. P., The English Noun Phrase in its Sentential Aspect, Ph.D. Thesis, MIT, 1987.

Collins, M., "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms", Proceedings of EMNLP, 2002.

B. Merialdo. Tagging English Text with a Probablistic Model, Computaional Linguistics.vol 20.