Various Approaches Used for Tagging and Chunking in Malayalam

Rinku T S, Merlin Rajan, Varunakshi Bhojane

Abstract: Parts of Speech (POS) tagging 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. This paper describes the analysis of different approaches that are used in tagging and chunking in Malayalam language.

Keywords: augmented transition network, chunking, Hidden Marcov Model, Malayalam, Supporting vector machines, Tagging, yamcha.

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.

Rinku T S
Information Technology Department, Mumbai University
Email: rinkusree@gmail.com
Merlin Rajan
Information Technology Department, Mumbai University
Email: merlintharakan@gmail.com
Varunakshi Bhojane
Computer Science Department, Mumbai University
Email: varunakshi_k@yahoo.com

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. 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 tagger give best results when

there is some knowledge about the structure of the language.

The paper described here is as follows. The second section deals with the statistical approach towards POS tagging and chunking. The third section describes rule based approach i.e., morpheme based augmented transition networks used for POS tagging and chunking.

2. STATISTICAL APPROACHES TOWARDS TAGGING AND CHUNKING

2.1 HMM

A Hidden Marcov Model (HMM) [3], [7] is a statistical model in which the system modeled is thought to be a Marcov 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 [3], 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.

2.2 SVM based Tools for Malayalam POS Tagger and chunker

The SVMTool [2] is a simple, flexible, and effective generator of sequential taggers based on Support Vector Machines and how it is being applied to the problem of part-of-speech tagging. This SVM-based tagger is robust and flexible for feature modeling (including lexicalization), trains efficiently, and is able to tag thousands of words per second. YamCha(Yet Another Multipurpose Chunk Annotator by Taku Kudo) is a generic, customizable, and open source text chunker. Yamcha is using a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), introduced by Vapnik.

2.2.1 Support Vector Machine

SVM is a machine learning algorithm for binary classification, which has been successfully applied to a number of practical problems, including NLP. Tagging a word in context is a multi-class classification problem. Since SVMs in general are binary classifiers, a binarization of the problem must be performed initially before applying them. Here a simple one-perclass binarization is applied, i.e., a SVM is trained for every POS tag in order to distinguish between examples of this class and all the rest. When tagging a word, the most possible tag according to the predictions of all binary SVMs is selected.

2.2.2 SVMTool for Malayalam POS Tagger

The SVMTool software package consists of three main components, namely the model learner (SVMTlearn), the tagger (SVMTagger) and the evaluator (SVMTeval). SVM model is learned from a training corpus using the SVMTlearn component. Different models are learned for the different tagging strategies. During tagging time, the SVMTagger component is used to choose the tagging strategy that is most suitable for the purpose of the tagging. Finally, when we give a correctly tagged corpus and the corresponding SVMTool predicted annotation, the SVMTeval component displays tagging results and reports. Tagged corpus is used for training a set of SVM classifiers. This is done using SVMlight, an implementation of Vapnik's SVMs in C, developed by Thorsten Joachims.

2.2.3 Yamcha for Malayalam Chunker

YamCha [3] is an open source text chunker and so called Support Vector machines (SVMs). SVMs are binary classifiers and thus must be extended to multiclass classifiers to classify three cases for NP chunking with (I, O, B). By mapping the n-dimensional input space into high dimensional feature space in which a linear classifier is then typically constructed. This approach is used for chunking, YamCha is used to perform the initial tagging, basic features in Yamcha are used, later all possible POS tag for the words in the corpus are added. This information is added to the training corpus and then it is trained using SVM therebypredicting the chunk boundary names using Yamcha, Finally the chunk labels and the chunk boundary names are merged to obtain the chunk tag.

3.RULE BASED APPROACH TOWARDS TAGGING AND CHUNKING

3.1Augmented Transition Networks

The chunking method discussed here works as a two stage process consisting of a compound word splitter and a chunker [6]. In the first stage the compound words in the input are split into its constituent morphemes. The sequence of morphemes is fed to the chunker for finding the chunks. The system uses a dictionary consisting of all the morphemes in the language along with their morpheme category.

3.1.1The splitting module

Malayalam is an agglutinative language where words of different syntactic categories are combined to form a single word. Formation of new words by combining a noun and a noun, noun and adjective, verb and noun, adverb and verb, adjective and noun and in some cases all the words of an entire sentence to reflect the semantics of the sentence are very common

3.1.2 The chunker

The chunker groups sequence of words or morphemes in the input sentence. The input to the chunker is the sentence which is split into sequence of morphemes with their lexical categories. The chunker uses sentence level grammatical rules for finding the morpheme groups. The rules are implemented as finite state cascades.

3.2 Augmented Transition Networks

A number of researchers have applied finite transducers to natural language parsing. An Augmented Transition Network (ATN) is a top down parsing technique that allows various kinds of knowledge to be incorporated into the parsing system and so it can operate efficiently [1]. It consists of a finite state cascade [4] consisting of a sequence of levels. Phrases at one level are built on phrases at the previous level. Abney has classified the rules that can be applied at each level so that only those set of rules are to be checked at a particular level [5]. Unlike Abney's rules the

phrases derived for our system recursive, i.e, a phrase at one level can contain the same phrase. The rules or the sequence of tags for each of these chunks are represented as finite automata.

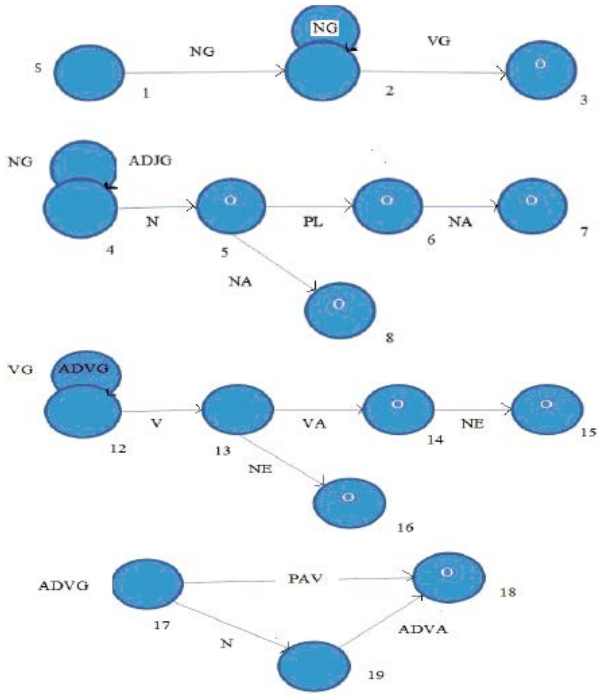

Fig.1 The Transitions Diagrams for the Chunks 1]

TABLE 1. Primitive Tags

No	Tag	Description
1	PL	Plural Suffix
2	ADJG	Adjective Group
3	NA	Post Position
4	PA	Pure Adjective
5	N	Noun
6	V	Verb
7	ADJA	Adjective Suffix
8	ADVA	Adverbial Suffix
9	PAV	Adjective
10	VN	Verbal Noun
11	VRP	Verbal Relative Participle
12	NCA	Noun Clause Suffix
13	ADVCA	Adverbal Clause Suffix
14	ADJCA	Adjectival Clause Suffix
15	INF	Infinitive
16	DJ	Disjunction
17	CNJ	Conjunction

TABLE 2. Chunk Tags

No	Tag	Description
1	NG	Noun Group
2	VG	Verb Group
3	NC	Noun Clause
4	ADVC	Adverbal Clause
5	ADJC	Adjectival Clause
6	NNC	Noun Conjunct

The chunk tags were chosen to reflect its function in the sentence. The chunk tag NG refers to the group of morphemes which function as a noun in the sentence. For example, noun group chunk(NG) consists of a sequence of adjectives followed by a noun, plural suffix and a case suffix or a sequence of adjectives followed by a noun and a case suffix. Similarly, the rules of other chunks are defined. Because of the feature of syntactic rules of the selected Malayalam language regrouping of tags needs to be tried only within a chunk and no inter chunk regrouping is needed. The algorithm proceeds in a depth first approach and when it cannot move forward since there is no transition on the input tag, it backtracks to the previous state and tries other options. The procedure is not a recursive descent one as it needs backtracking. Backtracking is needed because there can be multiple

transitions on the same input symbol from a state in the transition diagram.

3.3 Algorithm for detecting a chunk.

1. Begin from the starting state of the finite state hierarchy.
2. Intialise an array CL (chunk list) to store the sequence of chunks identified.
3. For each possible untried transition on the input symbol do steps 4, 5, 6 and 7. Longer matches are tried first before shorter ones.
4. If the transition is based on a lexical tag symbol then move to the next state if it matches with the morpheme tag in input and advance the input pointer goto step3.
5. Else if the transition is on a chunk tag then search for the sequence of tags corresponding to the chunk recursively. Add the chunk returned if it returns a list and go to next state.
Proceed to step 3.
6. Else if the currrent state is a final state then the Chunk name sequence of words for the chunk in CL and the current input pointer are returned.
7. Return failure.
8. Stop

4 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 to assign tags to all the words in the test case. These also focus on the point that a statistical approach can also work well with highly morphologically and inflectionally rich languages like Malayalam

REFERENCES

[1] Latha R Nair, David peter S, Shallow Parser for Malayalam Language using Finite State Cascades, 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)

[2]L.R Rabiner,B.H. Juang, An Introduction to Hidden Markov Models,0740-7467/86/0100 0004$01.00c.1986 IEEE

[3] Dhanalakshmi , Anand kumar , Rajendran , Soman K P, POS Tagger and Chunker for Tamil Language

[4]Vijay Sundar Ram R and Sobha Lalitha Devi, Noun Phrase Chunker using Finite State Automata for an Agglutinative Language

[5] Steven Abeny, Partial Parsing via Finite-State Cascades, Natural Language Engineering 1 (1): 000–000 °c 1995 Cambridge University Press

[6] Gilbert Krube – Two level representation for natural language, Research on Language and Social Interaction, volume 19, issue 2,1986, p.p.205- 286.

[7] Jisha P Jayan, Rajeev R R, Parts Of Speech Tagger and Chunker for Malayalam –Statistical Approach, Computer Engineering and Intelligent Systems ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol 2, No.3