International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 04, April-2015

A Comprehensive Survey of Noun Phrase Chunking in Natural Languages

Biplav Sarma
Department of Information Technology,
Gauhati University, Assam, India.

Anup Kumar Barman
Department of Information Technology,
Gauhati University, Assam, India.

***Abstract*— Assamese being an official language in the Indian State Assam, is a computationally less aware language. Almost 13 Million peoples speak this language around the world. In spite of being a morphologically rich and agglutinative language, very limited research work has been done in Chunking of Assamese Sentences in the domain of Natural Language Processing. NP Chunker deals with extracting the Noun Phrases from a sentence. Though NP Chunker is much simpler than parsing, building an accurate and fast NP Chunker is difficult as well as a challenging task. The main objective of this paper is to highlight the works done in Chunking of Indian Languages and other than Foreign Languages.**

***Keywords*— NP, VP, Chunker, FSA, HMM, TBML, AIS.**

I. INTRODUCTION

In Natural Language Processing, a Chunking is the process of detecting syntactic constituents (such as Noun Phrase and Verb Phrase [2]. Chunking can be considered as a preprocessing step for complete parsing. Due to high ambiguity of natural language, exact parsing may become very complex. To resolve this ambiguity Chunking can be used as an intermediate step. Chunking can resolve this ambiguity partially if not completely. Chunking involves discovering the main constituents of the sentences and their heads. Chunking which always follows the tagging process, can be used as a fast and reliable processing phase for full or partial parsing. A chunker always follows tagging process to find the Parts of Speech tags of each token in a given input sentence. A Noun Phrases is a sequence of words that acts as a Noun in the given sentence. Noun Phrase can be used to refer to objects, places, concepts, events, qualities and so on [2]. The simplest Noun Phrase consists of Pronouns like মই, তুমি, তেওঁ, আমি, এখেত and so on. A Sentence in Assamese language is divided into subject (known as উদ্দেশ্য – pronounced Uddeshya) and the Predicate (known as বিধেয়- pronounced Viddeiya). Every sentence in Assamese language consists of Noun Phrase followed by a Verb Phrase and vice-versa [3]. A Noun Phrase denotes the Subject about which something. It is also represented by a constellation of words that acts as a noun in the sentence. A Verb Phrase on the other hand is used to say something about the subject /or describe some kind of action. Eg. In the assamese sentences given below–

1. যদুৱে বৰশী বায় । [3] is a simple sentence, which can be read as Yaduwe baraxi bai. Here যদুৱে denotes Noun phrase and বৰশী বায় denotes the Verb Phrase.
2. সি ব্যাকৰন পঢ়ে । [4] is a simple sentence, which can be read as Xi vyakaran parhe. Here সি denotes Noun phrase and ব্যাকৰন পঢ়ে denotes the Verb Phrase.
3. পঢ়োঁ । [3], read as parhu, is a sentence though it appears to be a single word. Here the Noun Phrase is hidden, which can be either মই or আমি.
4. কাজিৰঙা এখন অভয়াৰণ্য ।, read as Kajiranga ekon avayaranya. Here the Noun Phrase is কাজিৰঙা (Noun) এখন (Article) followed by Verb Phrase অভয়াৰণ্য (Noun) and hidden verb (হয়).

Chunking 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. Since very limited work has been done for Assamese Language, all existing taggers cannot be used for this language.

The remainder of this paper is organized as follows: Section II gives a brief description about Assamese language. Section III describes the various approaches which are used to perform Noun Phrase chunking in various languages.

II. ABOUT ASSAMESE LANGUAGE

A. Assamese Language

Assamese is the easternmost Indo-Aryan language used mainly in the state Assam. It is spoken by over 13 million native speakers. Apart from Assam, it is also spoken in other parts of North-East. Assamese is a free word order language, which is morpholgically rich as well as agglutinative. Assamese uses the Assamese script, a variant of the Eastern Nagari Script, which traces its descent from the Gupta Script. Developed from Brahmi through Devanagiri, Assamese script is similar to that of Bengali except the symbols for /ra (ৰ)/ and /wabba (ৱ)/ and highly resembles the Devanagiri script of Hindi, Sanskrit and other related Indic languages. As such it is a syllabary script and is written from left to right. The alphabet consists of 12 vowel graphemes and 52 consonant graphemes. Assamese spelling is not always phonetically based. Current Assamese spelling practices are based on Sanskrit spelling, as introduced in the second Assamese dictionary Hemkosh which was written in the middle of the

19th century. The Assamese script presently has a total of 11 vowels, 37 consonants. [4]

B. Features in Assamese Language

Assamese language involves some features that distinguishes itself from other Indo-Aryan languages and Dravidian Language. Unlike other Indo-Aryan language, Noun inflextion is observed. Eg ৰামে (Rame) and ৰামক (Ramak) are inflected from ৰাম (Ram). Inflexion are also observed in case of adjectives, verb and pronouns. The negation of verbs is another distinguishable property. Eg নায়াও (Najau) obtained by adding prefix (না-) to verb য়াও (jau). The addition of suffixes to obtain plural form of noun from singular ones is the third distinguishable property. Eg. মানুহবোৰ (Manuhbur) from মানুহ (Manuh). The extensive use of classifiers is also another feature. For almost everything or every shape the languages uses a different classifier. Eg. মানুহজন (Manuhjon)-Male, মানুহজনী/মানুহগৰাকী (Manuhjoni / manuhgoraki)-Female. The classifiers are also combined with all types of nouns and numerals occurring in the language resulting in the combination of following grammatical constructions- E+zon+manuh (numeral+classifier+noun) and manuh + e+ zon (noun+numeral + classifier). [4]

III. METHODOLOGIES USED IN CHUNKING

In performing NP chunking, in the domain of Natural Language Processing, several approaches are being used from being purely Rule-Based Approach to completely Stochastic ones. Abney did partial parsing using finite state cascades, the finite state cascades has sequence of levels. The basic idea of Abney was based on the fact that during reading a English sentence, the reader reads a sentence chunk by chunk. Ramshaw and Marcus used Eric Brill's transformation based learning for recognizing the noun chunks and other text chunking (Ramshaw, 1995) [7]. During survey of NP Chunker in various languages (Including Indian languages and Foreign languages) various approaches are being observed. The approaches varies from Rule Based one to Hybrid Approaches and are categorized as below-

A. Rule Based Chunking.
B. Statistical Based Chunking.
C. Hybrid Approach for Chunking.

A. Rule Based Chunking

A Rule Based approach is generally used when adequate data on particular language is not available. This approach needs linguistic knowledge that involves both syntactic and semantic elements. The rules used in this approach are defined by human or extracted from linguistic resources. The percentage of accuracy in implementing this approach is less. The following are the list of works done in chunking using Rule Based Approach.

Ramshaw (1995) proposed chunking for English Language. They used IOB tags for this purpose where "B" means beginning of a chunk, "I" means the word token is inside the chunk and "O" means the word token is outside chunk. They used Brill's Transformation Based Learning Mechanism (TMBL) for text chunking. Their entire learning process was based on template rules. The first step is derivation of rules, next is scoring of rules, and last step is selection of one rule with maximal positive effect. This process was iterative. They checked the candidate rules using this process to select all the rules which have maximum positive effect. Overall this approach achieves Recall and Precision of about 92% for Base NPs for English Language. [7]

Grover (2007) proposed Rule Based chunking using XML(Extensive Mark-up Language). The concern of their work is to develop a chunker which is reusable and easily configurable for any tag set. They used CoNLL2000 data which was based on newspaper data and trained the system on this data. Results show that the machine learning systems out-perform such a rule based system but only when trained and tested on a domain specific data. This approach has an additional advantage that of allowing the tagger to be retrained on new data or allowing a choice of Taggers. Whenever the domain will be changed the machine learning systems may require retraining for the new domain. The XML based system outperforms when data from different sources is collected. He reported 89.1% Precision and 88.57% Recall for Noun Group for English. [8]

Vijay Sundar Ram R and Sobha Lalitha Devi (2006) introduced Rule Based chunking using Finite State Automata (FSA). They built the FSA using manually crafted linguistic rules. When a tagged sentence is given, the current word's tag is considered as the transition symbol to have transition to the next state, in the next state the next word's tags is considered. Similarly the traversing in FSA happens till it reaches the end state. If it successfully reaches the end state this part of the text is chunked as noun phrase. The nouns with all the case markers have the same noun phrase structure except the nouns with possessive case marker. The noun phrase chunked text obtained after traversing through the FSA, is again processed using the heuristic rules. The system is evaluated with the data taken from the CIIL corpus (Central Institute of Indian languages corpus). The numbers of sentences under consideration are 500, which contains 2180 noun phrases. The number of noun phrases recognized correctly by the system is 2043, which is at a precision of 94.9% and the recall of 93.7%. The system takes 2.5 sec to chunk 1000 sentences [9]

Sivaji, Asif and Debashish presented HMM based POS Tagger and Rule Based Chunker for Bengali Lanuage. They used two sets namely ANNOT-A and ANNOT-B consisting of 40956 tokens to train their POS Tagger. They used a set ANNOT-D containing 5967 tokens to test POS Tagger and obtained tagging accuracy of about 85.42% of accuracy. Finally the Tagger was tested on unannotated set containing only 5129 tokens. They applied rule-based approach to perform chunking due to non-availability of large chunked corpus. They developed a chunking algorithm in two phases. The first phase involves Chunk Boundary Identification. They applied hand-crafted rules, to check whether two neighbouring POS tags belong to same chunk, if not, a chunk boundary is assigned between the words. The second phase

involves Chunk Labelling. On evaluating chunker in their test set ANNOT-D and obtained an accuracy for 97.5% for Chunk Boundary Identification and 96.9% for Chunk Labelling. However, on testing with unannotated test set they obtained an accuracy of 81.61% accuracy in Chunk Boundary Identification and Chunk Labelling. [10]

B. Statistical Approach in Chunking

The statistical approaches do not need linguistic knowledge. The success of following this approach highly depends on the availability of resources. This approach being language independent has an added advantage that it can be applied on languages with common features. This method extracts statistical information from the processed corpus, web pages, search engine outputs etc. The extracted statistical information consists of occurring phrases, frequency of occurrence of the words etc. The statistical methods are mainly based on the probability measures including the unigram, bigram, trigram and n-grams. Following are a list of works done using Statistical approaches to perform chunking.
Akshay Singh (2001) presented HMM Based Chunker for Hindi. They divided chunking task into two tasks. The first task was identification of chunk boundaries and the other task was to label the chunks with their syntactic boundaries. Three different tag schemes were introduce 2-tag Scheme {STRT,CNT}, 3-tag Scheme {STRT, CNT, STP} and 4-tag Scheme{STRT, CNT, STP,STRT_STP} where STRT denotes start of the chunk, CNT means the token lies in the middle of a Chunk, STP means token lies at the end of a chunk, STRT_STP means the token lies in a chunk of its own. He added four different types of input tokens which were– words only, POS tags only, Word_POS tags, Word_POS tags (words followed by tags) and POS_Word tags (POS tags followed by words). The data set contains 200,000 words out of which only 20,000 words were used for testing 20,000 words were kept for parameter tuning and remaining 150,000 words were used to train different HMM representations. The chunker was tested on 20,000 words of testing data and 92% precision with 100% recall achieved for chunk boundaries. They concluded that the machine learning technique is more suitable because of robustness. [11]

Jisha P Jayan and Rajeev presented HMM based chunker for Malayalam. They used a Hidden Markov Model (HMM) to model a system with unknown parameters. They build a model based on the assumption that the probability of a word in a sequence may depend on the word presiding it. They used viterbi algorithm to search various lexical calculations. To perform chunking they used only six tag-sets. They used TnT(Trigram n-tags) to estimate lexical probabilities for unknown words that have same probabilities. They performed their application using TnT in two steps. In the first step they created model parameters from tagged training corpus. In the second step they applied the model parameters to the new texts and performed the tagging. They trained the system using manually tagged corpus. They build a suffix tree data structure to store the words and tagged frequencies taken from the training set. The letter tree is built taking the word and its frequency as the argument. During training, a lexical file is created that contains frequencies of words and its tags which occurred in the training corpus. A n-gram file is also generated that contains frequencies for the unigrams, bigrams and trigrams. Viterbi algorithm is applied to find the best tag sequence for a sentence, and, if tag sequence is not present some smoothing techniques are applied based on runtime arguments of the pos-tagger. To perform tagging of the raw corpus two files are required. They are –file containing the lexical frequencies and file containing contextual frequencies of the modal parameters. The raw corpus they used for testing was in Unicode. For training the system, the tagger and chunker were trained with using about 15,245 tokens. For chunking, the system gives about 92% accuracy while for POS tagging it gave about 90.5% accuracy. [12]

Dhanalakshmi, Anand and Rajendran presented paper on Parts of Speech Tagger and Chunker for Tamil Language using machine based method. They used Support Vector Machine (SVM) based method to perform POS Tagging. They developed their own Tag set for annotating the corpus, which was used for both training and testing the POS generator and the chunker. They used 32 tags for performing POS and 9 tags for performing chunking. They used a corpus of size 2,25,000 words. They divided their corpus into training set (1,65,000 words) and test set (60,000 words). They developed Amrita Chunking Tagset, in which Noun Phrases are tagged with tag NP. In their training set each token in a line is separated by columns. The first column being a ‘word’, the second column being the corresponding ‘POS tags’, the third column ‘Chunk tag’ and so on. The last column represents the ‘answer’ tag which was going to be trained by the SVMTool. An SVM is a machine learning algorithm. The SVMTool package consists of three component SVMTlearn (Learning Model), SVMTagger (Tagger) and SVMTeval (Evaluator). They used Yamcha, an open source text Chunker and so called SVM. SVM based machine learning tool afforded the most encouraging result for Tamil POS tagger (95.64%) and chunker (95.82%). [13]

C. Hybrid Approach

Fang Xu, Chengqing and Jun (2006) Introduced an Hybrid approach for Chinese that involved combination of SVM and CRF. They used Yamcha and CRF++ to treat the testing data. They compared the original results from the two Chunkers, which used exactly the same format. They used conditional probability to detect the wrong IOB tags obtained and choose the most suitable output. All the experiments were performed on a Linux system with 3.2 GHz Pentium 4 and 2G memory. The total size of the Penn Chinese Treebank words is 13 MB, including about 500,000 Chinese words. The quantity of training corpus amounts to 300,000 Chinese words. Each word contains two Chinese characters in average. We mainly use five kinds of corpus, whose sizes include 30000, 40000, 50000, 60000 and 70,000 words. The hybrid error-pruning method achieves an obvious improvement F-scores by combining the outcome from SVM and CRF classifiers. The test F-scores are decreasing when the sizes of corpus increase. The best performance with F-score of 89.27% is achieved by using a test corpus of 30k words. [14]

Park and Zhang (2003) introduced an approach that uses combination of Hand-crafted rules and Machine-Based learning for chunking Korean language. The machine learning method mentioned here uses k-Nearest Neighbours (k-NN) algorithm. K-NN is a type of instance-based learning. The proposed method works in two stages. The first stage involves application of a rule to determine the chunk type of the word in the input sentence, that is, prediction of chunk type of the word. In the second stage that word is referred to a memory based classifier to check whether it is an exceptional case of the rule. True chunks are stored in the memory-based classifier. During training phase, some rules are applied to the input sentence are to predict the chunk types of the words in the sentence. Then the predicted chunk types are compared with the actual chunk types. In case the predicted chunk mismatches with the actual chunk, it is treated as an error and stored in the error-case library accompanied by the actual chunk type. They evaluated the proposed method STEP2000 Korean chunking dataset which was derived from STEP2000 project supported by Korean government. The corpus used consists of 12,092 sentences with 111,658 phrases and 321,328 words, and the vocabulary size is 16,808. When only Rules were applied the Noun Phrase chunker gave an accuracy of 97.99% and 91.89% F-score. On applying the hybrid method the F-score obtained is 94.21%. They applied the proposed approach to determine four kinds of phrases namely- Noun Phrase (NP), Verb Phrase(VP), Adverb Phrase(ADVP) and Independent Phrase(IP). [15]

Chen applied a probabilistic chunker to decide the implicit boundaries of constituents and utilize the linguistic knowledge to extract the noun phrases by a finite state mechanism. The test texts are SUSANNE Corpus and the results are evaluated by comparing the parse field of SUSANNE Corpus automatically. They defined three kinds of Noun Phrases (NP) namely- Maximum Noun Phrases (MNP), Minimal Noun Phrases (mNP) and Ordinary Noun Phrases (NP). MNP are those Noun Phrases which are not contained in other Noun Phrases, mNP are those Noun Phrases that do not contain any other Noun Phrases and NP are Noun Phrases without any restriction. They used a volume of around 150,000 words including punctuation marks. They extracted the Noun Phrase applying four steps. In the first step, they tagged the input sentences. In second step, they used a probabilistic partial parser to partition the tagged sentences into chunks. In the third step they decided the syntactic and semantic heads of each chunk. A syntactic head is the head of a phrase based on grammatical relations while a syntactic head is a head of a phrase according to their semantic usage. In the final step, a Noun Phrase is extracted from the chunks according to the syntactic and semantic heads. Their system resulted in average precision of 95% and recall of 95% for extracting noun phrases that exist independently in SUSANNE Corpus. [16]

Bindu and Sumam (2011) proposed a Hybrid based Chunking model that employs a combination of Artificial Immunity System (AIS) and Rule Based Approach for Malayalam Language. Since the majority of sentences in Malayalam documents are complex and compound sentences, the clauses are first separated and chunks are identified and labeled from each clause. For each chunk there is a head which is most often the right most word in the chunk. They developed a Artificial Immunity System (AIS) based chunker. AIS involves three main functions. They are POS Tagger, Clause Identifier and chunker. At first, a tag-set of 52 tags is developed. The POS tagger was designed using probabilistic approach Extended Conditional Random Field. Clause identifier identifies and separates clauses from sentences using handcrafted linguistic rules and forwards the output to the chunker. A clause generally ends with a verb/ auxiliary-verb/an adjectival-participle/ an adverbial-participle. All the phrases corresponding to each clause is identified and separated and labeled with phrase tags. The Phrase chunker is designed and implemented using J2SDK1.4.2 and MySQL. Its performance is evaluated using standardized techniques precision, recall and F-score where Precision is defined as a ratio of number of correct chunks to the number of chunks in the output and recall is the Ratio of number of correct chunks to the number of chunks in the test data. For a NP Chunker the system resulted in precision of 93.5%, recall of 92.6% and F-score of 93%. [17]

IV. CONCLUSION

The goal of NP Chunker is to find out the Noun Phrases from the sentences. Depending upon the availability of resources and linguistic knowledge about a particular language, approach is made to perform the chunking task. In this paper, we have seen the development of Noun Phrase Chunker (NP Chunker) for various Indian and Foreign Languages using various ones from purely Rule Based one to Hybrid Approaches. Till date, very limited research work has been observed in the development of NP Chunker for Assamese Language. Noun Phrase chunking work is Assamese Language is not reported so far. This work can be a start of a new sub area under Natural Language Processing domain. Chunking which always follows the tagging process, can be used as a fast and reliable processing phase for full or partial parsing. It can also be used for information Retrieval Systems, Information Extraction, Text Summarization. In other Indian Languages (Bengali, Tamil, Malayalam, Hindi) Various approaches have been followed for performing the Chunking.

REFERENCES

[1] James Allen., *Natural Language Understanding*., 2nd Edition, 1995, Pearson Education.

[2] Steven Bird, Ewan Klien, and Edward Loper., *Natural Language Processing with Python*.

[3] Dr.Dipti Phukan Patgiri., *Aadhunik Asomiya Vyakaran*. ,1st edition, Published by Book House ,Guwahati, 1999.

[4] Sri Bhagaban Moral., *Asomiya Vyakaran Jyoti*., 12th edition, Assam State Textbook Production and Publication Corporation Limited, Guwahati, 2013.

[5] "en.m.wikipedia.org/wiki/Assamese_language"- Wikipedia

[6] Sri Harendra Nath Sarma, *Satyanath Bora Rachita Asomiya Bhasar Bahal Vyakaran*., 4th edition, 2007.

[7] Lance A. Ramshaw and Mitchell P. Marcus., *Text Chunking using Transformation-Based Learning*., In the Proceedings of the Third Workshop on Very Large Corpora ,1995.

[8] Claire Grover and Richard Tobin, *Rule-Based Chunking and Reusability*, In the Proceedings of the Conference on Language Resources and Evaluation, 2006 .

[9] Vijay Sundar Ram R and Sobha Lalitha Devi, *Noun Phrase Chunker using Finite State Automata for an Agglutinative Language* , In the proceedings of the Tamil Internet-2010 at Coimbatore, India, June 23-27.

[10] Sivaji Banyopadhyay, Asif Ekbal and Debasish Halder, *HMM Based POS Tagger and Rule-Based Chunker for Bengali*, In Proceeding of the NLPAI Machine Learning contest workshop, National Workshop on Artificial Intelligence, Pune, India, 2006.

[11] Akshay Singh Sushma Bendre and Rajeev Sangal, *HMM Based Chunker for Hindi*, In the proceedings of 2nd International Joint Conference on Natural Language Processing-2005(IJCNLP-2005), Korea.

[12] Jisha P Jayan and Rajeev R R, *Parts of Speech Tagger and Chunker for Malayalam-A Statistical Approach*, Computer Engineering and Intelligent Systems ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol2,No.3, 2011.

[13] Dhanalakshmi.V, Padmavathy P, Anand Kumar M, Soman K P and Rajendran S (2009), *Chunker for Tamil using Machine Learning*., 7th International Conference on Natural Language Processing 2009(ICON2009), IIIT Hyderabad.

[14] Fang Xu Chengqing Zong Jun Zhao. (2006), *A Hybrid Approach to Chinese Base Noun Phrase Chunking* .,National Laboratory of Pattern Recognition Institute of Automation Chinese Academy of Sciences, Beijing 100080,China

[15] Seong-Bae Park and Byoung-Tak Zhang, *Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning* by , In the proceedings of 41st Annual Meeting of the Association for Computational Linguistics-2003.

[16] Kuang-Hua Chen and Hsin-Hsi Chen, *Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic valuation*, In the proceedings of workshop on Cognitive Models of Language Acquisition-1994 (ACL1994), Netherlands.

[17] Bindu.M.S and Sumam Mary Idicula, *A Hybrid Model for Phrase Chunking employing Artificial Immunity System and Rule Based Methods*, In the proceedings of the International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.4, October 2011.