BI-DIRECTIONAL HEAD-DRIVEN PARSING FOR ENGLISH TO INDIAN LANGUAGES MACHINE TRANSLATION

Pavan Kurariya, Prashant Chaudhary, Jahnavi Bodhankar, Lenali Singh[1] and Ajai Kumar[1]

Centre for Development of Advanced Computing, Pune, India

ABSTRACT

In the age of Artificial Intelligence (AI), a significant breakthrough occurred as machines demonstrated their ability to communicate in human languages. This marked the beginning of a ground-breaking era in Natural Language Processing. defined by unparalleled computational capabilities. Amidst this evolution, parsers stand as an indispensable component, facilitating syntactic comprehension and empowering various NLP applications, from Machine Translation to sentiment analysis. Parser plays a crucial role in deciphering the complex syntactic structures inherent in human languages. With the use of a parser, machines can comprehend human language, extract meaning, and facilitate a variety of natural language processing (NLP) applications, such as information retrieval, sentiment analysis, and machine translation. This research paper presents the implementation of Bi-Directional Head-Driven Parser, aiming to expand the horizons of NLP beyond the constraints of traditional early-type L-TAG (Lexicalized Tree Adjoining Grammar) Parsing. While effective, conventional Parsers encounter inherent limitations in grappling with the intricacies and subtleties of natural language. Through the utilization of Bi-Directional principles, Head-Driven techniques offer a revolutionary breakthrough in computational frameworks for large-scale grammar parsing, enabling complex NLP tasks such as discourse analysis and semantic parsing, and guaranteeing reliable linguistic analysis for practical applications. The performance of the Bi-Directional Parser has been examined on the data set of 15000 sentences and observed a reduction in the variation of derivations for sentences of the same length compared to the conventional TAG Parser, this research showcases how Head-Driven Parser facilitates breakthrough in language processing, syntactic analysis, semantic comprehension, and beyond. Moreover, it underscores the structural implications of integrating Head-Driven Parsing. Traditional approaches, such as Tree Adjoining Grammar (TAG), while valuable, often encounter limitations in capturing the full spectrum of linguistic phenomena, particularly in the context of cross-linguistic transfer between English and Indian languages. In light of the significance of natural language processing (NLP) in addressing these issues, this research introduces a Bi-Directional Head-Driven Parser implementation. Drawing upon the rich foundation of TAG and acknowledging its constraints, our approach transcends these limitations by harnessing advanced parsing traversal techniques and linguistic theories. By bridging the gap between theory and application, our approach not only enhances our understanding of syntactic parsing across language families but also surpasses the performance of an 'Early-type Parser' in terms of time and memory. Through rigorous experimentation and evaluation, this research contributes to the ongoing discourse on expanding the frontiers of Tree Adjoining Grammar-based research and shaping the trajectory of Machine Translation

David C. Wyld et al. (Eds): AISO, SIP, SOFTFM, NLDM, CRBL, BIOM, COMIT– 2024
pp. 71-81, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141506

KEYWORDS

Artificial intelligence (AI), Natural Language Processing (NLP), Tree Adjoining Grammar (TAG), L-TAG (Lexicalized Tree Adjoining Grammar)

1. INTRODUCTION

The rapid progress of machine translation (MT) technology has transformed human communication by permitting a seamless flow of information across linguistic boundaries. Classical machine translation (MT) systems generate translations using rule-based methods, Statistical Models, or neural networks, The intricacies of human languages are still difficult for these methods to fully capture nuances of Indian languages, especially for Low Resource Languages. Head-Driven parsing can be emerged as a significant Parsing Technique that can transform traditional Parser by utilizing the Bi-Directional method to perform computations at levels that were previously unattainable. This research introduces a Head-Driven Bi-Directional parsing for language translation to explore the potential advantages of bottom-up traversal. A traditional parser works from the left and typically requires three inputs: an unknown end position, a given start position, and a Part-of-Speech that has to be parsed. Two pairs of positions are provided by the algorithm in a bidirectional parser: one pair of indices shows the extreme positions between which the category must be identified, and the other pair of indices provides the precise position of the category once it has been identified. One of the extreme positions corresponds to the actual situation, depending on whether we are parsing to the left or the right.
Parsing is initiated by making top-down predictions on certain nodes and proceeds by moving bottom-up from the head-corner associated with the goal node (root node). In parsing right siblings are parsed from left to right and left siblings are parsed from right to left. Our objective is capturing the nuances of syntactic structure by integrating Bi-Directional Tree traversal into the existing machine translation architecture. The purpose of this study is to investigate the potential benefits of Head-Driven- approaches in improving translation accuracy, fluency, and efficiency.

2. LITERATURE SURVEY

In the early years of Machine Translation, parsing large-scale grammars posed a significant challenge to researchers in the field of Natural Language Processing (NLP). Joshi's imperial work on Tree Adjoining Grammar (TAG) [1] emerged as a pioneering solution, offering a framework that facilitated the parsing of complex linguistic structures. Building upon Joshi's foundation, early endeavours in NLP research also saw the implementation of the Early type Parser, originally proposed by Vijay-Shanker [2], which further enriched the TAG Parser available to computational linguists. Furthermore, in pursuit of language-agnostic solutions, we were inspired to develop a Language-Independent Generator [3] for Natural Languages, aiming to transcend linguistic boundaries and enhance the versatility of computational models. This endeavour broadened the applicability of NLP techniques and contributed to the optimization of Tree Adjoining Grammar-based Machine Translation systems [4], fostering advancements in cross-linguistic communication. Continuing the trajectory of innovation within the TAG framework, the research community delved into exploring the full potential of TAG structures. This quest led us to conceptualize vTAG [5], an initiative focused on discovering fresh insights and capabilities inherent in TAG formalisms. Additionally, we introduced sTAG [6] enriching the discourse on TAG-based parsing and generation techniques. A substantial amount of work has been done with a variety of parsing approaches, laying the foundation for real-world applications. Early rule-based approaches, most notably Chomsky's transformational grammar, provided foundational principles for syntactic analysis [7]. Beyond rule-based approaches, Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs), revolutionized parsing by enabling parsers to learn from given corpora [8]. Dependency parsing also emerged as an effective alternative to traditional parsing, offering simpler yet effective representations of syntactic structures [9]. In recent years, Head-Driven parsing has gained attention for its emphasis on hierarchical structures and the identification of key syntactic heads [10]. The integration of linguistic principles, such as Tree Adjoining Grammar (TAG), with machine learning techniques has shown promise in addressing the limitations of traditional parsers, particularly in cross-linguistic parsing scenarios [11]. Bi-Directional parsing methods, as proposed in [12][13], represent a paradigm shift in NLP, offering enhanced capabilities to capture a broader range of syntactic phenomena through both left-to-right and right-to-left parsing strategies. These advancements in parsing techniques have profound implications for various NLP applications, including machine translation, corpus analysis and classification, and information retrieval [14]. Through various experimentation and evaluations, researchers continue to push the boundaries of computational linguistics, shaping the future of NLP and advancing our understanding of human language.

3. IMPLEMENTATION OF BI-DIRECTIONAL HEAD-DRIVEN PARSING FOR TRANSLATING ENGLISH TO INDIAN LANGUAGES

Tree Adjoining Grammar (TAG) is a highly expressive formalism used in computational linguistics for syntactic analysis of Natural Languages. Combining TAG with Bi-Directional Head-Driven parsing creates a powerful method for translating English to Indian languages. Figure 1 depicts the comprehensive pipeline of English to Indian Languages Machine Translation, accompanied by concise descriptions outlining the fundamental NLP components of the translation system.

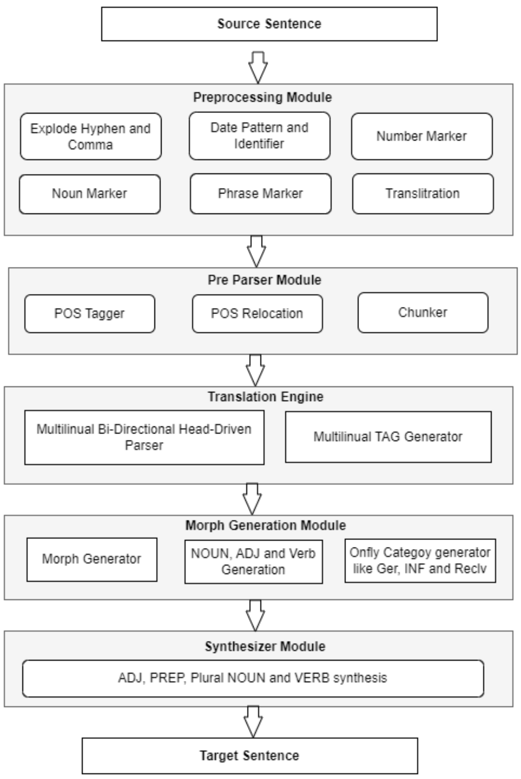

Figure 1: Bidirectional Head-Driven Parsing-based Machine Translation System

3.1. Pre-processing

Pre-processing of source sentences in machine translation involves several critical steps to ensure accurate and contextually appropriate translations. The process begins with exploding hyphens and commas, which splits compound words connected by hyphens into individual words and separates items in lists connected by commas. Next, the Date Patterns Identifier detects and normalizes date formats into a consistent structure, facilitating the correct translation of date-related information. The Number Marker then identifies and tags numerical values, ensuring they are preserved accurately in the translation. Noun Marker follows by tagging nouns to help maintain their meaning and context. Phrase Marker is used to identify and mark idiomatic expressions or multi-word phrases that need to be treated as single units to retain their specific meanings. Finally, Transliteration converts words from the source script to the target script, preserving phonetic properties for proper nouns, brand names, or words without direct translations. Together, these pre-processing steps enhance the machine translation system’s ability to handle complex linguistic elements, ensuring a more precise and coherent translation.

3.2. Pre-Parser Module

The pre-parser module in natural language processing plays a pivotal role in preparing text for further linguistic analysis and understanding. It includes three essential components: the Part-of-Speech (POS) Tagger, POS Relocation, and the Chunker. Every word in a phrase, including nouns, verbs, adjectives, and so on, must have their parts of speech assigned by the POS Tagger in order to provide an understanding of the grammatical structure. Following this, the POS Relocation step adjusts the positioning of these tags to resolve ambiguities and correct inaccuracies, ensuring that the grammatical roles assigned during POS tagging align correctly with the context. Last but not least, the Chunker converts word sequences into meaningful units that correspond to the grammatical structure of the sentence, such as noun or verb phrases. This chunking process is crucial for understanding the hierarchical relationships within the text and facilitating more advanced parsing tasks. Together, these components of the pre-parser module enhance the system's ability to interpret and process natural language accurately, laying a strong foundation for effective linguistic analysis and subsequent natural language processing tasks.

3.3. Translation Engine

3.3.1. Bidirectional Head-Driven Parser

In Bidirectional Head-Driven Parsing, tree vector serves as a crucial structure for both parsing and generating outputs. Think of it as a reservoir of trees specifically designed for Tree Adjoining Grammars (TAG), where lexicalized trees are drawn for parsing and generation processes. This structure, known as the tree vector, is implicitly defined and essential for the parser's operations. It manages mappings between trees, their names, and lexicons and incorporates a string array that stores the segmented sentence, with each word acting as a key in the mapping.

The Multithreaded Bidirectional Head-Driven Parser is a Bi-Directional, Head-Driven parser designed for constraint-based Lexicalized Tree-Adjoining Grammars (L-TAG) with multithreading capabilities. Parser selects a node in an elementary tree—using a lexical node for an initial tree and a foot node for an auxiliary tree—and treats it as the <Head>. Parsing begins with top-down predictions on specific nodes and proceeds bottom-up from the Head Node associated with the goal node (root node). During parsing, right siblings are parsed from left to right, while left siblings are parsed from right to left. The use of multithreading enhances the parser's efficiency and speed by allowing multiple parsing operations to be conducted simultaneously. Figure 2 demonstrates the Bi-Directional Head-Driven Parsing process incorporating substitution, adjunction operations, and the generation of a state chart.

These functions operate on instances of the State class, taking an object (s) as input and generating a new object (s1) of the same class as output. What makes these functions unique is that they don't rely on parameters passed directly to them. Instead, they access and modify global variables to perform their tasks. This design choice allows them to manipulate and update data within the program's broader scope, potentially affecting multiple parts of the system simultaneously.

a. **Reoperation:** This function is designed to evaluate if a node possesses a substitution or null adjunction constraint. Subsequently, it selects the suitable trees corresponding to that specific node and executes either an adjunction, substitution, or no adjunction operation accordingly. Following the performed operation, it modifies the state to reflect the action taken.
b. **CheckSiblings:** This function serves as a higher-level function that internally calls the following two subordinate functions.
c. **MoveDotRight:** This function verifies whether a node has a right sibling. If a right sibling exists, the function updates the state, prompting the parser to begin processing with the right sibling.
d. **MoveDotLeft:** This function serves to determine if a node possesses a left sibling. If such a sibling exists, the function updates the state, directing the parser to commence processing with the left sibling.

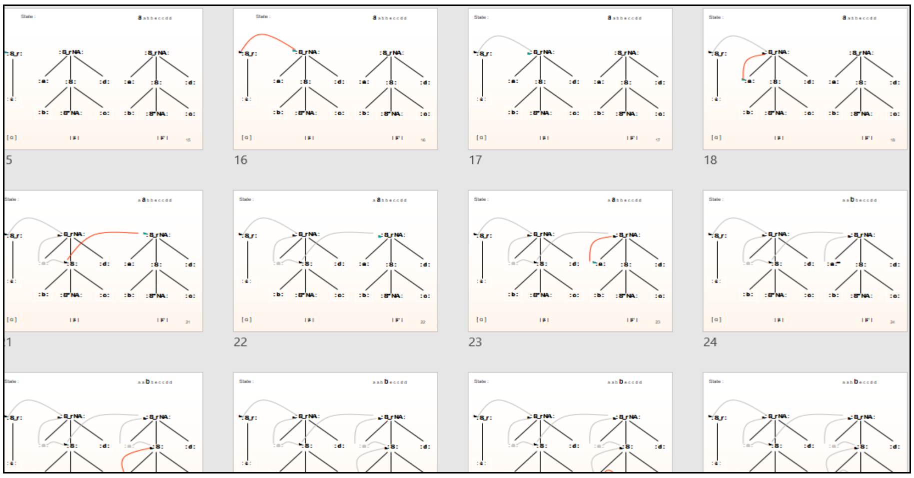

Figure 2: Snapshot of Bi-Direction Head-Driven Paring

e. **PRightCompletor:** This function plays a crucial role during the parser's processing of the root node within an elementary tree. It assesses whether any adjunction or substitution has taken place within the current tree. If such operations have occurred, the function retrieves the state from before the adjunction or substitution occurred. Conversely, if there have been no adjunctions or substitutions, it proceeds with the completion process as usual.

f. **PLeftCompletor:** This function is invoked when the parser processes the root node of an elementary tree. It verifies if any adjunctions or substitutions have taken place within the current tree. If such operations have occurred, it retrieves the state prior to the adjunction or substitution. In cases where no adjunctions or substitutions have taken place, it proceeds with the completion process**.**

g. **MoveDotUp:** When the parser is processing an elementary tree's lexical node, this function is invoked. This function marks the position associated with the elementary tree and modifies the left and right boundaries.

The Parser stores all parsing information within states organized in a state chart, but the Generator lacks knowledge of the parsing algorithm, states, and state chart. To process this information effectively, a structure is needed that summarizes the state chart coherently for the Generator, enabling interpretation in the target language. In order to accomplish this, we used the derivation structure, in which the addresses within the parent tree where child trees are either replaced or adjoined are labelled on edges, and nodes are labelled with the names of the elementary trees. This format allows the 'Generator' to understand and process parsed data, facilitating accurate target language generation. The derived and derivation Trees, depicted in Figures 3 and 4 respectively, are generated by the Bi-Directional Head-Driven Parser while parsing the sentence described in Figure 4.

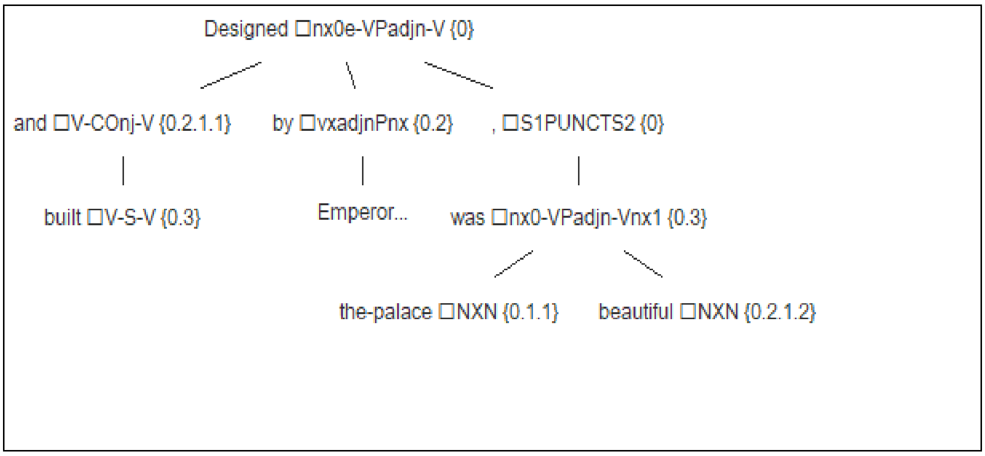

Figure 3: Source deviation tree generated by Bi-Direction Head-driven Parser

3.3.2. Multilingual Tag Generator

The derivation tree encapsulates all translation information stored in the state chart, compressing it to a minimal size, and making it easier to manipulate.

To understand the detailed derivation of how a sentence emerges from the grammar trees, we need the translation tree, or simply the translation. This involves stitching together the entire set of trees used in deriving the particular sentence. The Tree-Adjoining Grammar (TAG) derives the sentence in lexical order, and no other order should be assumed. To make it manipulatable in post-order and pre-order spaces, it provides information on the subject, object, and main verb; possessive case markers in Indian languages; multiple synonym generation; and multiple output generation based on context. The figure below illustrates the generated derivation of the source sentence into the target sentence using the multilingual TAG generator.

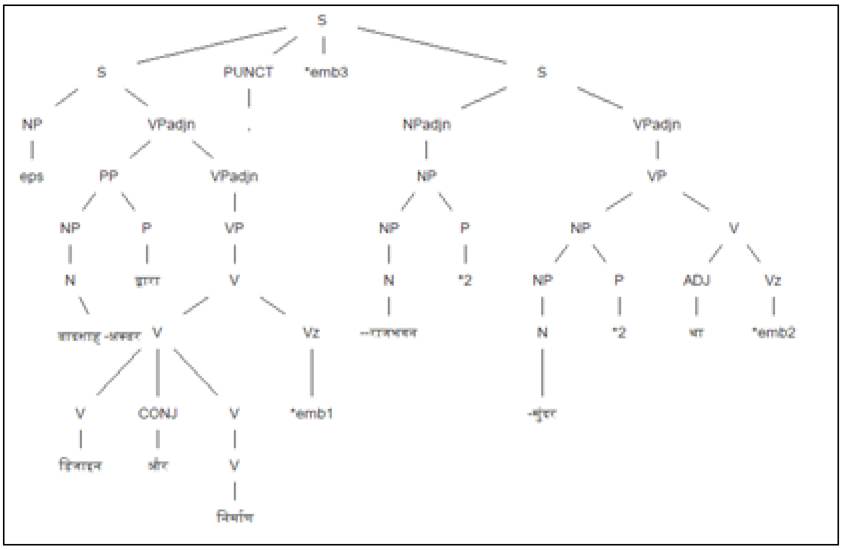

Figure 4: Derived tree in Target Language

4. Experiment with a Bi-Directional Head-Driven Parser with Tree Bank

In this section, illustrates the experimentation of the Bi-Directional head-driven parser-based translation system with the source sentence (English) which passes through pre-processing, POS tagging, Parsing, and translating into a target sentence (Hindi). In order to analyse the effectiveness of the Bi-Directional Head-Driven Parser, which makes use of a Multilingual Grammar developed by language specialists, a specialized experimental setup has been established, as illustrated in Fig. 6. Throughout these experiments, we closely monitored CPU usage and memory utilization. A dataset consisting of 15,000 sentences was employed for this purpose. Notably, we compared the performance of the Bi-Directional Head-Driven Parser with that of our previously implemented 'Early TAG Parser', particularly focusing on longer sentences, as illustrated in Fig. 7. The following are the outcomes of these experiments.

Sentence:
Designed and built by Emperor Akbar, the palace was beautiful.
Pre-processing:
[Designed and built by Emperor-Akbar, the palace was beautiful]
Part-of-speech:
[0^Designed-and-built@@PtPART 1^by@@PREP 2^Emperor-Akbar@@NOUN 3^,@@PUNCT 4^the-palace@@NOUN 5^was@@VERB 6^beautiful@@ADJ]
Bidirectional tree vector:
0^PtPART--> [PtPart-V-PtPart(Initial), nx0e-VPadjn-PtPart-nx1-s(Auxiliary), nx0e-VPadjn-PtPart-s(Auxiliary), nx0e-VPadjn-PtPart-px1-s(Auxiliary)]
1^PREP--> [Pnx(Initial), vxadjnPnx(Auxiliary)]
2^NOUN--> [NXN(Initial), NPnx(Auxiliary)]
3^PUNCT--> [PPUNCT(Initial), nx-Punct-S(Auxiliary), nx-Punct-nx1-Punct(Auxiliary), nxPUNCTpx1(Auxiliary), Px-Punct-S(Auxiliary), np1PUNCTnp2(Auxiliary), S1PUNCTS2(Auxiliary)]
4^NOUN--> [NXN(Initial), NPnx(Auxiliary)]
5^TYPE--> [nx0-VPadjn-VP-AdjP(Initial), nx0e-VPadjn-VP-AdjP(Initial), nx0e-VPadjn-VP-AdjP-S(Initial), nx0-VPadjn-VP-AdjP-S(Initial)]
6^ADJ--> [AdjP(Initial)]
Bi-directional head corner parser output:
(((("S"."r")))(((("NP"."0")):constraints"NA")(((("eps"."")):lexipT)))(((("VPadjn".""))))(((("VP".""))))(((("V"."PtPART")):headpT)(((("0^PtPART".""))))))(((("S"."f")):constraints"NA")(((("PP"."r")))(((("NP"."r")))(((("N"."")):headpT)(((("2^NOUN"."")))))))(((("P"."")):headpT)(((("1^PREP"."")))))))(((("PUNCT"."")):headp(((("3^PUNCT"."")))(((("S"."f")))(((("NPadjn".""))))(((("NP"."r")))(((("N"."")):headp)(((("4^NOUN".""))))))))(((("VPadjn".""))))(((("VP".""))))(((("AdjP"."r")))(((("Adj"."")):headp T)(((("6^ADJ"."")))))))(((("V"."")):headp T)(((("5^TYPE".""))))))))))
Bi-directional head corner generator output:
(((("S"."r")))(((("NP"."0")):constraints"NA")(((("eps"."")):lexip)))(((("VPadjn".""))))(((("VP".""))))(((("V"."PtPART")):headpT)(((("0^PtPART".""))))))(((("S"."f")):constraints"NA"(((("PP"."r")))(((("NP"."r")))(((("N"."")):headpT)(((("2^NOUN"."")))))))(((("P"."")):headpT)(((("1^PREP"."")))))))(((("PUNCT"."")):headp)(((("3^PUNCT"."")))))(((("S"."f")))(((("NPadjn".""))))(((("NP"."r")))(((("N"."")):headp)(((("4^NOUN"."")))))))(((("VPadjn".""))))(((("VP".""))))(((("AdjP"."r")))(((("Adj"."")):headp T)(((("6^ADJ"."")))))))(((("V"."")):headp T)(((("5^TYPE".""))))))))))
Generated output:
बादशाह अकबर द्वारा डिज़ाइन और निर्माण किया गया , महल सुंदर था

Figure 5: Machine Translation Process using Bidirectional Head-Driven Parser

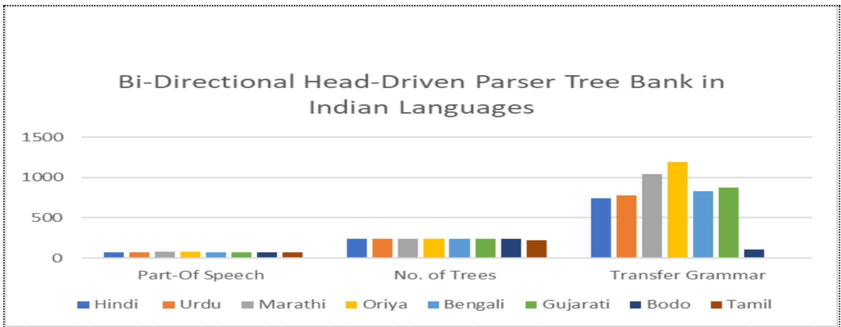

Figure 6: Multi-Lingual Tree Bank for Bi-Directional Head-Driven Parser

- 11500 out of 15000 sentences have been successfully Parse and generated on given grammar

- 4531 out of 11500 Sentences having multiple parse derivations.
- 3000 sentences having better output in comparison to the existing Multi-Threaded TAG Parser
- The performance of the Parser has been examined, and it was observed that it requires approximately 40 minutes to parse a total of 11,500 sentences. In comparison, the existing "Early Type Parser" takes around 120 minutes to parse an equivalent set of sentences

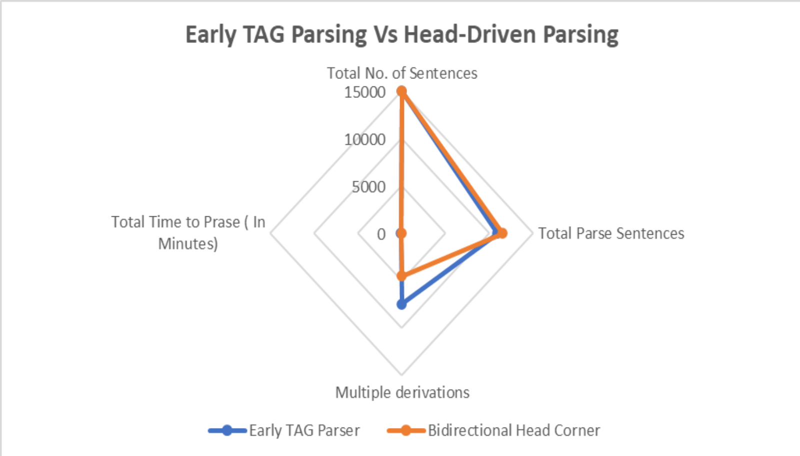

Figure 7: Comparison between 'Early TAG' Parser Vs l Head-Driven Parser

5. Conclusions

This research paper begins by discussing the limitations of the conventional TAG Parser. It also examines the advancements made by various researchers in parsing techniques. We also discuss how these limitations can be overcome. Furthermore, we describe an implementation of a Bi-Directional Head-Driven Parser in detail, along with its advantages over traditional TAG parsing. We have conducted a series of experiments with the Bi-Directional Head-Driven Parser, for which we used a multilingual Tree Grammar created by language experts. We also provide a detailed implementation of a Bi-Directional Head-Driven parser, along with its advantages over conventional TAG Parser. We ran empirical tests by parsing 15,000 English sentences from the General domain. Out of 15,000 sentences, 11,500 were successfully parsed and generated using the given grammar, while 4,531 out of the 11,500 sentences had multiple parse derivations. In comparison to the traditional TAG Parser, we observed a decrease in the variety of derivations for sentences of the same length throughout the Bi-Directional Head-Driven parser's investigations of parse derivations. Experimental results have shown that the performance of the Bi-Directional Head-Driven Parser is better on multi-clause structures compared to the conventional Early-Type TAG Parser. We have also discussed an end-to-end Machine Translation System using the Bi-Directional Head-Driven parsing algorithm. In summary, the evolution of parsers in NLP reflects a continuous effort to improve the computational understanding of human languages. Bi-Directional Head-Driven parsing techniques represent a significant step forward in this field, offering new insights into syntactic parsing and paving the way for enhanced language processing capabilities. Despite the growing popularity of Large Language Models (LLMs), the research described here underlines the importance of Bi-Directional Head-Driven parsing approaches, especially for low-resource languages. While Large Language Models (LLMs) excel in numerous natural language processing tasks, they often require extensive amounts of data and computational power, which can be challenging for low-resource languages. On the other hand, this research's exploration of the Bi-Directional Head-

Driven parsing algorithm is useful for specific applications with limited datasets. Furthermore, these methods open the door for specialized applications where exact syntactic comprehension is critical, such as machine translation systems.

REFERENCES

[1] Joshi, A. K. (1985). Tree Adjoining Grammars: How Much Context-Sensitivity Is Required To Provide Reasonable Structural Descriptions? In Proceedings Of The 21st Annual Meeting Of The Association For Computational Linguistics (Pp. 154-160).

[2] Vijay-Shanker, K., & Weir, D. J. (1994). The Equivalence Of Four Extensions Of Context-Free Grammars. Mathematical Systems Theory, 27(2), 101-120.

[3] Kurariya, P., Chaudhary, P., Jain, P., Lele, A., Kumar, A., & Darbari, H. (2015, September). File Model Approach To Optimize The Performance Of Tree Adjoining Grammar Based Machine Translation. In 2015 International Conference On Computer, Communication And Control (IC4) (Pp. 1-6). IEEE.

[4] Kurariya, P., Chaudhary, P., Bodhankar, J., Singh, L., Kumar, A., & Darbari, H. (2020, December). TREE ADJOINING GRAMMAR BASED “LANGUAGE INDEPENDENT GENERATOR”. In Proceedings Of The 17th International Conference On Natural Language Processing (ICON) (Pp. 138-143).

[5] Kurariya, P., Chaudhary, P., Bodhankar, J., Singh, L., Kumar, A., & Darbari, H. (2022, October). VTAG: Virtual Lab For Tree-Adjoining Grammar-Based Research. In International Conference On Information And Communication Technology For Competitive Strategies (Pp. 765-777). Singapore: Springer Nature Singapore.

[6] Kurariya, P., Chaudhary, P., Bodhankar, J., Singh, L., & Kumar, A. (2023, August). Unveiling The Power Of TAG Using Statistical Parsing For Natural Languages. In CS & IT Conference Proceedings (Vol. 13, No. 14). CS & IT Conference Proceedings.

[7] Chomsky, N. (1956). Three Models For The Description Of Language. IRE Transactions On Information Theory, 2(3), 113-124.

[8] Lafferty, J., Mccallum, A., & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models For Segmenting And Labeling Sequence Data. In Proceedings Of The Eighteenth International Conference On Machine Learning (Pp. 282-289).

[9] Eisner, J. (2012). Three New Probabilistic Models For Dependency Parsing: An Exploration. In Proceedings Of The 16th Conference On Computational Natural Language Learning (Pp. 25-36).

[10] Hale, J. (2001). A Probabilistic Earley Parser As A Psycholinguistic Model. In Proceedings Of The North American Chapter Of The Association For Computational Linguistics Conference (Pp. 159-166).

[11] Joshi, A., Levy, L. S., & Takahashi, M. (1992). Tree Adjoining Grammars. In A. Von Stechow & D. Wunderlich (Eds.), Handbook Of Contemporary Syntactic Theory (Pp. 65-130). Berlin, Germany: De Gruyter Mouton.

[12] Satta, G., & Stock, O. (1994). Bidirectional Context-Free Grammar Parsing For Natural Language Processing. Artificial Intelligence, 69(1-2), 123-164.

[13] Zhou, J., & Zhao, H. (2019). Head-Driven Phrase Structure Grammar Parsing On Penn Treebank. Arxiv Preprint Arxiv:1907.02684.

[14] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction To Information Retrieval. Cambridge University Press.

AUTHORS

Mr. Pavan Kurariya Is A Scientist 'E' Working In The HPC AI-IT Infra. & Operations Group At C-DAC Pune And Has More Than 15 Years Of Experience. He Is A Distinguished Researcher, And His Expertise Lies In Various Domains Such As Natural Language Processing, Cyber Security, Cryptography, And Quantum Computing. He Has Contributed Significantly To The Advancements Of Machine Translation, Cyber Security, And Quantum Computing. His Primary Area Of Interest Centers Around Machine Translation And Cryptography, Where He Investigates Novel Techniques And Cutting-Edge Methodologies To Enhance The Accuracy And Efficiency Of Various NLP Applications.

Mr. Prashant Chaudhary is a Scientist 'E' working in the AAI & GIST Group at C-DAC Pune and has more than 15 years of experience. He is a distinguished researcher, and his expertise lies in various domains such as Natural Language Processing, Machine Translation, and Cyber Security. Through his numerous research papers, he has made significant contributions to the field of Machine Translation by investigating both theoretical aspects and practical applications. His primary area of interest centers around Natural Language Processing, where he investigates cutting-edge techniques and methodologies to enhance the accuracy and efficiency of various NLP applications.

Ms. Jahnavi Bodhankar is a Scientist ‘F’ working in the HPC AI-IT Infra. & Operations Group at C-DAC Pune and has more than 18 years of experience. She is a distinguished researcher, and her expertise lies in various domains such as Natural Language Processing, Cyber Security, Machine Learning, and Blockchain Technology. She has contributed significantly to the advancements and understanding of NLP, E-Signature, and Blockchain through her numerous research papers and intricate work.

Ms. Lenali Singh is a Scientist ‘F’ working in the AAI & GIST Group at C-DAC Pune and has more than 20 years of experience. Her key role is in initiating and executing various projects in the areas of Natural Language Processing and Speech Technology. She is a distinguished researcher, and her expertise lies in various domains such as Natural Language Processing and Speech Technology. She has contributed significantly to the advancements and understanding of the NLP field through her numerous research papers and intricate work.

Dr. Ajai Kumar is a Scientist ‘G’ and Head of the AAI & GIST Group at C-DAC Pune, with more than 20 years of experience working in Natural Language Processing, including Machine Translation, Speech Technology, Information Extraction & Retrieval, and E-learning systems. His key role is in initiating mission mode consortium projects in the areas of Natural Language Processing, Speech Technology, Video Surveillance, etc. Through his meticulous research, he aims to bridge the gap between different languages and enable seamless communication across linguistic boundaries.

**** C-DAC**: Centre for Development of Advanced Computing is the premier R&D organization of the Ministry of Electronics and Information Technology (MeitY), Government of India*