Parsing Subordinate Clauses in Telugu using Rule-based Dependency Parser

[1]Sangeetha Perugu, [2]Parameswari Krishnamurthy, [3]Amba Kulkarni
[1,2]Centre for Applied Linguistics and Translation Studies, [3]Centre for Sanskrit Studies
University of Hyderabad
[1]geethanjali.sheldon@gmail.com, {[2]pksh, [3]ambakulkarni}@uohyd.ac.in

Abstract

Parsing has been gaining popularity in recent years and attracted the interest of NLP researchers around the world. It is challenging when language under study is a free-word order language which allows ellipsis like Telugu. In this paper, an attempt is made to parse subordinate clauses especially, non-finite verb clauses and relative clauses in Telugu which are highly productive and constitute a large chunk in parsing task. This study adopts a knowledge-driven approach to parse subordinate structures using linguistic cues as rules. Challenges faced in parsing ambiguous structures are elaborated alongside providing enhanced tags to handle them. Results are encouraging and this parser proves to be efficient for Telugu.

1 Introduction

Parsing, the word derived from Latin (*pars orationis*), was originally used in elementary schools for grammatical explication of sentences (Nivre, 2006). Currently, parsing is a well-known and well-researched area in natural language processing (NLP) which involves analyzing sentences syntactically or syntactico-semantically. Building parsers and treebanks have attracted several researchers for its utility in various larger NLP applications. An efficient and ready-to-use parser for languages like Telugu, one of the most widely spoken Dravidian languages is still under development, though a handful of resources are traced.

Telugu is a south-central Dravidian language with free-word order and well-known for its agglutinating morphology. Agglutination allows carrying multiple grammatical information on words in Telugu. This grammatical information is quite helpful in parsing and stands as a rationale behind building the rule-based parser, despite multiple challenges. Parsing free-word order and agglutinating languages like Telugu is particularly challenging as they allow pro-drops, ellipsis and complex constructions. Earlier attempts in developing Telugu dependency parsers include mostly data-driven approaches (Ambati et al., 2009; Husain, 2009; Bharati et al., 2009; Kesidi et al., 2013; Kanneganti et al., 2016; Gatla, 2019; Nallani et al., 2020; Rama and Vajjala, 2018). Among the attempts made, UDPipe for Telugu[1] which is trained using Telugu-MTG UD treebank (Rama and Vajjala, 2018) is the only publicly accessible parser. There is an attempt in developing a rule-based parser with linguistic knowledge-driven approach (Sangeetha et al., 2021) for simple sentences. In this paper, we present our experiment in parsing subordinate clauses, particularly, non-finite verb clauses and relative participle clauses in Telugu using rule-based dependency parser.

2 A Rule-Based Dependency Parser

This study uses a rule-based parser (RBP) which takes input from sentences that are morphologically analysed. Telugu POS tagger, pruning and pick-one-morph modules are used to select one analysis per token (Rao, 1999). The RBP follows dependency approach based on the Indian theories of verbal cognition where three factors viz. *ākānksā* (expectancy), *yōgyata* (meaning compatibility), and *sannidhi* (proximity) are used and implemented initially for Sanskrit (Kulkarni, 2019). Telugu RBP is adopted from Sanskrit RBP and modified for Telugu parsing (Sangeetha et al., 2021). We model the parser as a tree where the nodes of a tree correspond to a word and the edges between nodes correspond to a relation between the corresponding words. Parser is implemented using the functional programming language Ocaml [2] to write rules and

[1] http://lindat.mff.cuni.cz/services/udpipe/

[2] https://ocaml.org/

Proceedings of the ICON 2021 Workshop on Parsing and its Applications for Indian Languages (PAIL) , pages 12–19
December 16, 2021. Silchar, India. ©2021 NLP Association of India (NLPAI)

Perl to generate dependency trees as graphs. The figure 1 explains the architecture of the RBP.

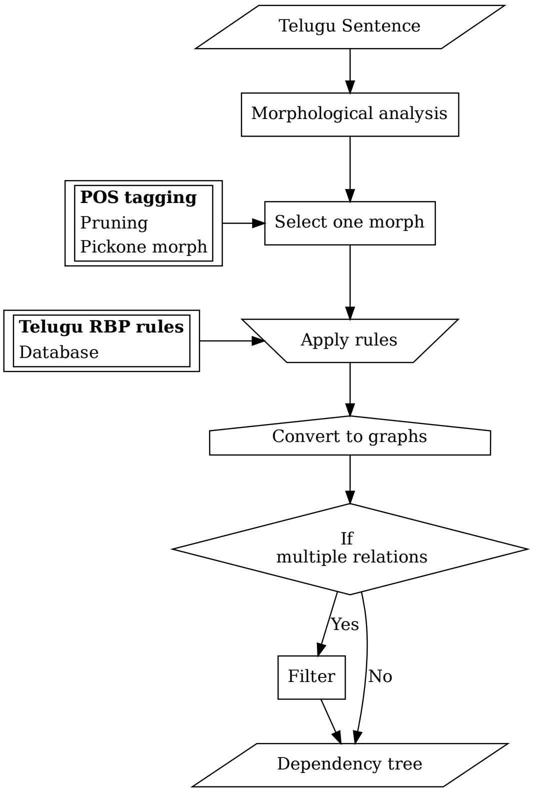

Figure 1: Architecture

In parsing simple sentences, 29 dependency labels are used and they are divided into kāraka(K) relations (for example, kartā (roughly equivalant to subject) (k1), karmā (object) (k2) etc.) and non-kāraka (for example, genitive (r6), associative(ras) etc.) labels. The dependency tree for the sentence (1) is seen in figure 2.

(1) mā nānna rēpu ūri nuMci
our father tomorrow village from
vas-tā-ru
come-FUT-3.SG.HON.
'My father will come from village tomorrow'

3 Subordinate Clauses in Telugu

Subordinate clauses in Telugu include non-finite verb clauses, relative participle clauses and complementizer clauses. Subordinate clauses in Telugu do express ambiguity with different syntactico-semantic relations.

Non-finite verb clauses are highly productive in the formation of sentences in Telugu and they constitute a large chunk in parsing task. They are dependent clauses which cannot stand alone in a sentence. They are realised as subordinate clauses which are derived from simple sentences with certain structural changes and precede the matrix clause by occurring to their left side. The verb of subordinate clause is syntactically the head of the clause but does not exhibit person-number-gender agreement with respective subjects, however it is marked for appropriate tense, aspect and mood. They are classified into conjunctive participles, conditionals, concessives and infinitives in Telugu (Krishnamurti and Gwynn, 1985). Conjunctive participles are divided into past, durative and negative. Conditionals and concessives clauses can have both affirmative and negative forms whereas infinitives can have only affirmative form.

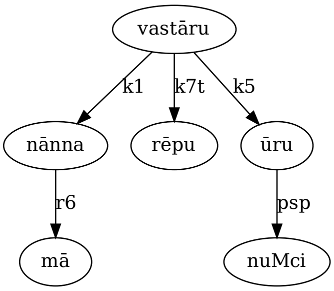

Figure 2: Dependency tree for sentence(1)

Relative participle clauses are primarily noun phrases which are further divided into past, durative, future/habitual and negative participles. Negative participles do not differentiate for tense. Complementizer clauses are formed by the quotative form i.e. *ani* 'that' which links both finite clauses. Figure 3 provides the classification of subordinate clauses in Telugu. Examples of various types of subordinate clauses are provided in the table 1.

In this paper, we present challenges in parsing non-finite verb clauses and relative participle clauses using rule-based parsing. We use the anncora tagset for tagging the dependency relations (Version 2.5) (Bharati et al., 2009). There is a great requirement for the enhancement of tags for Telugu to disambiguate various functions of subordinate clauses. An attempt is made to build enhanced tags and implemented using linguistic cues as rules in RBP.

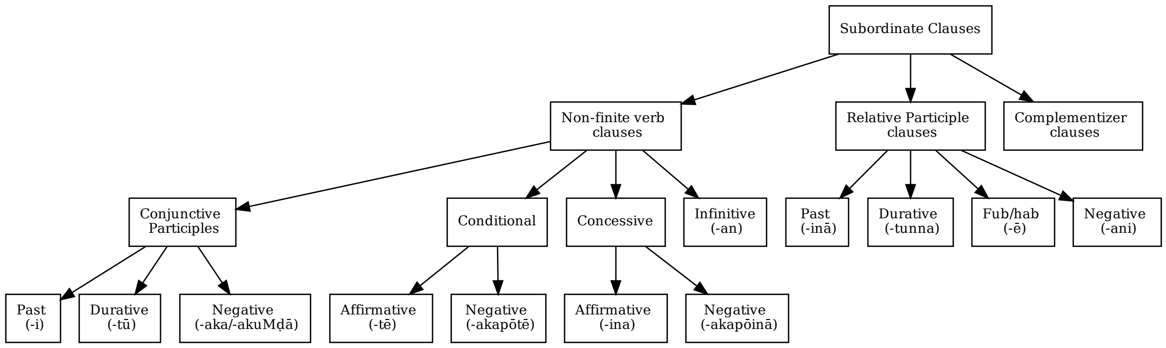

Figure 3: Types of non-finite clauses in Telugu

Type of subordinate clause	**Example**
I. Non-finite verb clauses	
Conjunctive Participle	
Past	*tin-**i*** ‘having eaten‘
Durative	*tin-**ṭū*** ‘along with eating‘
Negative 1 (*-akuMḍa*)	*tina-**kuMḍā*** ‘not having eaten‘
Negative 2 (*-aka*)	*tin-**aka*** ‘due to not having eaten‘
Conditional	
Affirmative	*tin-**tē*** ‘if one eats‘
Negative	*tin-**akapōtē*** ‘if one does not eat‘
Concessive	
Affirmative	*tin-**inā*** ‘inspite of having eaten‘
Negative	*tin-**akapōinā*** ‘inspite of not having eaten‘
Infinitive	*tin-**(an)*** ‘to eat‘
II. Relative Participle	
Past	*tin-**ina** abbāyi* ‘the boy who ate‘
Durative	*tin-**tunna** abbāyi* ‘the boy who is eating‘
Future-habitual	*tin-**ē** abbāyi*‘the boy who will eat‘
Negative	*tin-**ani** abbāyi* ‘the boy who did not eat‘

Table 1: Examples of subordinate clauses

.

4 Challenges in Parsing Subordinate Clauses

Subordinate clauses in Telugu are ambiguous across certain sub-types. These ambiguous constructions pose various parsing challenges mainly due to multiple functions or interpretations of a non-finite marker which causes ambiguity. Certain ambiguous constructions with non-finite verb clauses and relative participle clauses in Telugu are discussed in this section.

4.1 Conjunctive participle clause

The conjunctive participle clause occurs as a subordinate clause and modifies the matrix clause. This conjunctive participle clause can be used to express verbal modifier (`vmod`) functions such as serial action, manner and simultaneous action in Telugu. Example (2) explicates conjunctive participle as a serial verb. The figure 4 is shown with the tag `vmod:cp_serial` for the sentence (2) with conjunctive participle expressing serial action.

(2) rāmuḍu.∅ annaM.∅ **tin-i**
Ram.NOM food.ACC eat-CP.PST
paḍukunn-ā-ḍu
sleep-PST-3.SG.M
‘Ram ate food and slept’

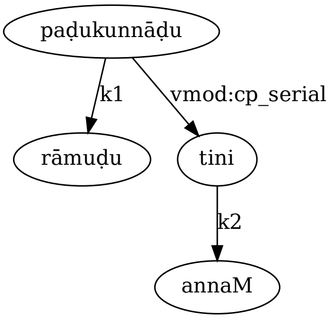

Figure 4: Dependency tree for (2)

The conjunctive participle can express manner as explicated in the sentence (3) with the Figure 5. Here, the verb class i.e. *motion verbs* is used as a cue to identify the manner in the verb modification with the tag `vmod:cp_manner`.

(3) vimala.∅ āphīsu-ku nadic-i
vimala.NOM office-DAT walk-CP.PST
veḷt-uM-di
go-HAB-3.SG.F
‘Vimala goes to office by walk’

The conjunctive participles express simultaneous action when the participle is durative as in the sentence (4).

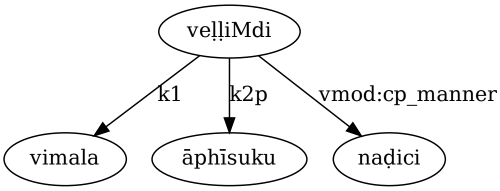

Figure 5: Dependency tree for (3)

(4) prakāsh.∅ sinimā **cūs-tū**
prakash.NOM cinema watch-CP.DUR
cūldriMk tāg-ā-ḍu
cool-drink drink-PST-3.SG.M
'Prakash drank cool drink while watching a cinema'

Figure 6 shows a dependency tree of the sentence (4) adding a new tag `vmod:cp_simul`.

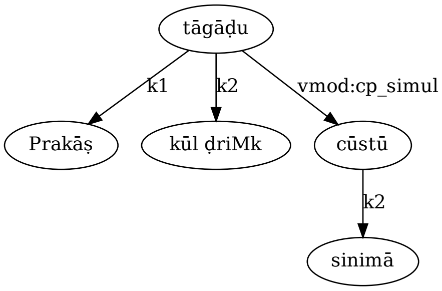

Figure 6: Dependency tree for (4)

However, when the active form of conjunctive participle verb is followed by the passive matrix verb, it renders an ambiguous interpretation. Consider example (5) from (Ramarao, 2017, pg. 116) and its dependency tree in the Figure 7.

(5) sujāta **tiraskariMc-i**
sujata.NOM reject-CP.PST
avamāniMc-a-baḍ-iM-di
insult-PASS-PST-3.SG.F
'Sujata rejected (someone) and was insulted' or 'Sujata got rejected and was insulted'.

Example (5) is ambiguous due to argument ellipsis. This can be interpreted in two different ways by supplying either a passive subject (as in (6)) or the object (as in (7)) in the non-finite clause. This ambiguity is represented in Figure 7.

(6) sujāta vāḍi cēta **tiraskariMc-(abaḏ)i**
sujata.NOM he by reject-(PASS).CP.PST
avamāniMc-abaḍ-iM-di
insult-PASS-PST-3.SG.F
'Sujata got rejected by him and was insulted'

(7) sujāta vāḍi-ni **tiraskariMc-i**
sujata.NOM he-ACC reject-CP.PST
avamāniMc-abaḍ-iM-di
insult-PASS-PST-3.SG.F
'Sujata rejected him and was insulted'

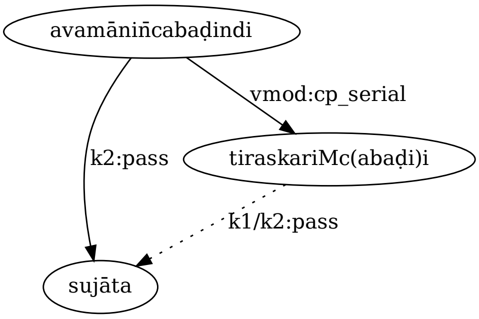

Figure 7: Dependency tree for (5)

Other cases include constructions with negative matrix verb percolating its features to the conjunctive participle resulting in ambiguity as in the sentence (8).

(8) ravi.∅ kāphī.∅ **tāgi**
Ravi.NOM coffee.ACC drink-CP.PST
skūl-ki veḷḷ-a-lēdu
school-DAT go-PST-NEG
'Ravi drank coffee but he did not go to school/ It is not coffee that Ravi drank (but something else) and went to school'

Since disambiguating senses in (8) is not in the scope of parsing and it requires deep semantic analysis, the dependency tree does not show the difference in meaning as in the figure 8.

However, the occurrence of the particle *kūḍa* 'also' after the participle form helps in disambiguating and the negative percolation from the matrix to subordinate clause is prevented.

(9) ravi.∅ kāphī.∅ **tāg-i** **kūḏa**
Ravi.NOM coffee.ACC drink-CP.PST also
skūl-ki veḷḷ-a-lēdu
school-DAT go-PST-NEG
'Ravi drank coffee but he did not go to school'

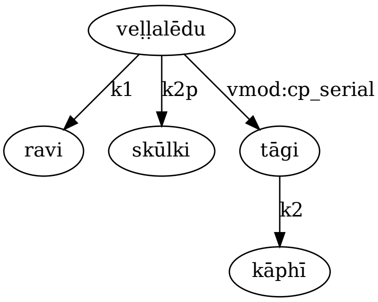

Figure 8: Dependency tree for (8)

4.2 Conditional clauses

Conditional clauses in Telugu not only express conditional sense but also show other interpretations leading to several parsing analyses. Such constructions are identified and tagged differently in the RBP.

Sentences (10) and (11) differ with the use of tense in finite verb and render different senses. If the finite verb of a complex sentence is in non-past tense, it is considered as a conditional clause and will be tagged with `vmod:cond`. Whereas, if the matrix verb is in the past tense, the conditional verb expresses the serial action and is given the tag `vmod:cond_serial` as the sentence (11).

(10) rāyi-tō **koḍi-tē** kāya kiMda
stone-INST hit-COND fruit-NOM down
padu-tuM-di
fall-NON.PST-3.N.SG
'If you hit with a stone, the fruit falls'

(11) rāyi-tō **koḍi-tē** kāya kiMda
stone-INST hit-COND fruit-NOM down
pad-iM-di
fall-PST-3.N.SG
'The fruit fell when hit with a stone'

Other exceptional case of conditional suffix rendering non-conditional sense include the causal meaning. In the sentence (12) (Ramarao, 2017, pg. 129), the verb of non-finite clause *tiM-tē* expresses the cause for the main action and can be alternated with conjuctive participle form *tini* 'having eaten'. The subject *subbārāvu* 'Subbarao' is shared with both non-finite and matrix clauses. Shared subject constraint is used as a syntactic cue in order to parse these constructions and tag `vmod:cond_cause` is attached in the dependency tree as in 9.

(12) subbārāvu guḍlu **tiMṭe**
Subbarao-NOM eggs eat-NF-COND
baliṣ-ā-ḍu
fat-become-PST-3.SG.M
'Subbarao became strong by eating eggs'

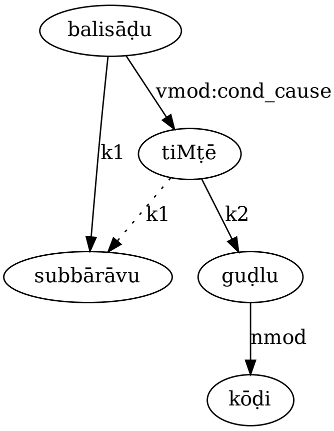

Figure 9: Dependency tree for (12)

4.3 Concessive clauses

Concessive clauses in Telugu are formed by adding the suffix *-inā* to the verb stem and express the meaning 'even if/even though'. It functions as adverbial modifiers to the matrix verb. The negative concessive form is formed by the suffix 'akapoyinā'. This clause is tagged as `vmod:conc` in the rule-based parser.

(13) nēnu cadiv-inā pāsu
I-NOM study-NF-CONC
avva-lēdu
become-NEG
'Even after studying, I did not pass (the examination)'

4.4 Infinitive clauses

Infinitive clauses are not very common in Telugu. The infinitive suffix in Telugu is *-an* and the tag `vinf:k1` is used in tagging infinite clauses when they occur in the subject position as in the sentence (14) and the respective dependency tree in Figure 11.

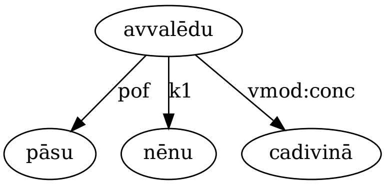

Figure 10: Dependency tree for (13)

(14) mīru nā-tō ā viṣayaM cepp-an
I-HON I-INST that matter tell-INF
akkar-lēdu
need-NEG
'You need not tell me that matter'

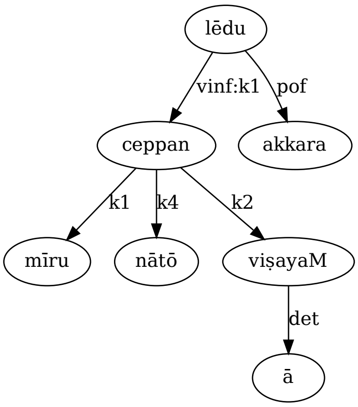

Figure 11: Dependency tree for (14)

4.5 Relative Participle Clauses

A simple sentence can be changed into a relative clause by replacing its finite verb by a relative participle (or verbal adjective) in the corresponding tense-mode and shifting the noun that it qualifies as head of the construction (Krishnamurti and Gwynn, 1985). Relative participle clauses occur immediately before nouns which they qualify. In Telugu, they show the distinction in tense in affirmative construction whereas in negative they do not show the tense.Relative participles are tagged as `nmod:relcl` in RBP. `nmod:relcl` is added with the argument relation of the noun which is relativized. In the sentence (15), the relativized nouns holds the object (k2) relation with the relative participle whereas the sentence (16) with the subject (k1) relation. There are tagged as `nmod:relcl_k2` and `nmod:relcl_k1` respectively in Figures 12 and 13.

(15) nēnu cūs-ina maniṣi iMṭi-ki
I.NOM see-RP.PST man home-DAT
vacc-ā-ḍu
came-PST-3.SG.M
'The man whom I saw came home'

(16) nan-nu cūsina maniṣi iMṭi-ki
I-ACC see-RP.PST man home-DAT
vacc-āḍu
come-PST-3.SG.M
'The man who saw me came home'

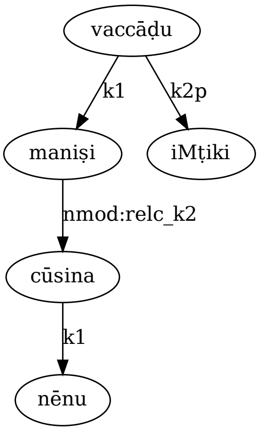

Figure 12: Dependency tree for (15)

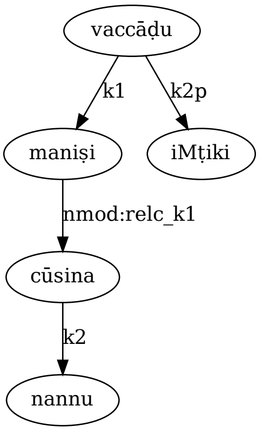

Figure 13: Dependency tree for (16)

Relative participle clause constructions are ambiguous when the noun in the relative clause has the potential to be an agent followed by the relative participle form of the verb which is transitive.

(17) nēnu tin-ina kaMcaM pāta-di
I.NOM eat-RP.PST plate old-3.SG.N
'The plate in which I ate is old'/'The plate which I ate is old'

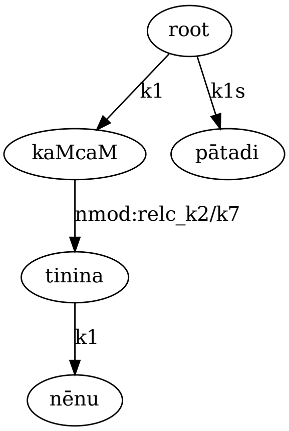

Figure 14: Dependency tree for (17)

The token *kaMcaM* 'plate' can be interpreted with the tag `k7` (location) as well as `nmod:relc_k2` as in figure 14. However, we use selectional restriction rules to rule out one of the analysis as eating *kaMcaM* 'plate' with the tag `nmod:relc_k2` is semantically not possible.

5 Enhanced Anncora Tagset

Anncora guidelines (Bharati et al., 2009) suggest the tag `vmod` for conjunctive participles, concessives, conditionals and *nmod* for relative participles. In this study, we have used multiple linguistic cues and enhanced subordinate clause tags as shown in the table 2. Around 41 rules with linguistic cues have been used to parse both simple and subordinate clauses in Telugu.

6 Evaluation

Rules of RBP are framed based on the model sentences collected from various Telugu grammar books Krishnamurti and Gwynn (1985), Ramarao (1975), Krishnamurti (2003) & (Ramarao, 2017). The purpose of choosing grammar texts for building rules is due to the wide-range of exceptions that are covered. These exceptions enabled us to segregate several cases of subordinate clause occurrences and providing fine-grain tags. Around 250 sentences were collected from news paper data for testing subordinate clauses. The labelled attachment score (LAS) is **72%** and unlabelled attachment score is **81%**. The Table 3 shows the LAS and UAS various sub-type of subordinate clauses.

Subordinate clause	**Enhanced Tag for Telugu**
conjunctive participle	vmod
serial action	vmod:cp_serial
simultaneous action	vmod:cp_simul
Manner	vmod:cp_manner
conditional clauses	
condition	vmod:cond
serial action	vmod:cond_serial
cause	vmod:cond_cause
concessive clause	vmod:conc
infinitive clause	vinf:k1
Relative participle clause	
relativization of subject	nmod:relcl_k1
relativization of object	nmod:relcl_k2
relativization of location	nmod:relcl_k7

Table 2: Dependency Tags for Subordinate Clauses in Telugu

Type of clauses	**LAS**	**UAS**
Conjunctive participle clauses	77.7%	86.2%
Conditional clauses	70.5%	82%
Concessive clauses	69.6%	80%
Infinitive clauses	64%	64%
Relative participle clauses	66.7%	73.2%

Table 3: Results of various subordinate clauses

RBP works on the linguistic cues (verbal/nominal databases, grammatical information) provided to it. RBP fails when these linguistic cues are not included as part of database or when it encounters an exception. But these cues can be updated as and when RBP encounters a new corpus. Another case in which RBP fails to deliver a correct parse is when pre-processing tools like morphological analyser, POS, pruning, pick-one morph provide an erroneous output.

7 Conclusion

Parsing of non-finite verb clauses and relative participle constructions in Telugu is attempted in this paper using a rule-based parser. It is observed that knowledge-driven parser works better for agglutinating languages like Telugu as many linguistic cues can be seen in the structure. Parsing of subordinate clauses is challenging due to its diverse interpretations and usage. Various ambiguous constructions are considered in this paper alongside adding enhanced/fine-grain tags to the existing Ancora tagset. These tags are beneficial as the tag vmod is quite under-specified. Results prove that RBP serves as an efficient parser for Telugu and addition of linguistic cues can improve the performance further. Parsing of other complex structures will be carried out in the future work.

Acknowledgments

We thank the reviewers for their critical comments which immensely helped us in improving this paper.

References

Bharat Ram Ambati, Phani Gadde, and Karan Jindal. 2009. Experiments in indian language dependency parsing. *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pages 32–37.

Akshar Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begum, and Rajeev Sangal. 2009. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank. *LTRC, IIIT Hyderabad, India*. Version 2.

Praveen Gatla. 2019. Dependency parsing for telugu using data-driven parsers. *Language in India*, 19(1).

Samar Husain. 2009. Dependency parsers for indian languages. In *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*.

Silpa Kanneganti, Himani Chaudhry, and Dipti Misra Sharma. 2016. Comparative error analysis of parser outputs on telugu dependency treebank. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 397–408. Springer.

Sruthilaya Reddy Kesidi, Prudhvi Kosaraju, Meher Vijay, and Samar Husain. 2013. *Constraint-based hybrid dependency Parser for Telugu*. Ph.D. thesis, Ph. D. thesis, International Institute of Information Technology Hyderabad.

Bhadriraju Krishnamurti. 2003. *The dravidian languages*. Cambridge University Press.

Bhadriraju Krishnamurti and John Peter Lucius Gwynn. 1985. *A grammar of modern Telugu*. Oxford University Press, USA.

Amba Kulkarni. 2019. *Sanskrit Parsing: Based on the Theories of Śābdabodha*. DK Printworld (P) Ltd.

Sneha Nallani, Manish Shrivastava, and Dipti Misra Sharma. 2020. A simple and effective dependency parser for telugu. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149.

Joakim Nivre. 2006. *Inductive dependency parsing*. Springer.

Taraka Rama and Sowmya Vajjala. 2018. A dependency treebank for telugu. In *English Conference Papers, Posters and Proceedings*, 8, pages 119–128.

C Ramarao. 1975. *Telugu vākhyam*. A.P. Sahitya Academy.

Chekuri Ramarao. 2017. *A reference grammar of modern Telugu*. EMESCO books Pvt. ltd., Hyderabad.

G. Uma Maheshwar Rao. 1999. *A Morphological Analyzer for Telugu*. (electronic form). Hyderabad: University of Hyderabad. Accessible at.

P Sangeetha, Parameswari K., and Amba Kulkarni. 2021. A rule-based dependency parser for telugu: An experiment with simple sentences. *Translation Today*, 15(1).