A Rule-based Dependency Parser for Telugu: An Experiment with Simple Sentences

SANGEETHA P., PARAMESWARI K.
& AMBA KULKARNI

Abstract

This paper is an attempt in building a rule-based dependency parser for Telugu which can parse simple sentences. This study adopts Pāṇini's Grammatical (PG) tradition i.e., the dependency model to parse sentences. A detailed description of mapping semantic relations to vibhaktis (case suffixes and postpositions) in Telugu using PG is presented. The paper describes the algorithm and the linguistic knowledge employed while developing the parser. The research further provides results, which suggest that enriching the current parser with linguistic inputs can increase the accuracy and tackle ambiguity better than existing data-driven methods.

1. Introduction

Parsing is a challenging task especially when languages under investigation are morphologically rich and have relatively free-word order. A parser is an automated Natural Language Processing (NLP) tool that analyses the input sentences based on the grammar formalism adopted in implementation and provides the output in constructed parse trees. The most frequently adopted grammar formalisms include constituency and dependency models. This study adopts the dependency model that has proved to be an efficient model for Indian languages that are morphologically rich with free-word order (Bharati & Sangal 1993; Kulkarni 2013; Kulkarni & Ramakrishnamacharyulu 2013; Kulkarni 2019).

Telugu is a South-central Dravidian language with agglutinating morphology and with relatively free word order. Hence, dependency grammar formalism was adopted for this study which proved to be useful for other free-word order languages. Apart from grammar formalism, the technique used for the implementation of a parser also stands as equally important. The implementation techniques majorly include grammar-driven or data-driven. The present study uses a grammar-driven technique that handles a wide range of language ambiguities.

This paper discusses various problematic cases in parsing Telugu simple sentence structures which consist of a clause that includes covering constructions such as copula, imperative, passive, dubitative, interrogative, non-nominative subjects, reflexive, and coordinating noun phrases. This paper is the first attempt (to the authors' best knowledge) in building a rule-based parser for Telugu using a dependency framework.

This paper is organized as follows: Section-2 provide the literature survey of parsing in Telugu; section-3 describes the theoretical background for the study involving a discussion on the mapping from *kāraka* to *vibhakti* in Telugu, taking insights from PG; Section-4 provides a detailed description on building the current parser, algorithm, and constraints (both local and global); Section-5 provides the evaluation of the rule-based parser and Knowledge-based parser, further discussing the error analysis and some observations; finally, Section-6 concludes and explores the future scope of the study.

2. Brief Survey

A few attempts were made in developing a Telugu dependency parser based on data-driven approaches. Some of them include Vempaty Chaitanya, Viswanatha Naidu, Samar Husain, Ravi Kiran, Lakshmi Bai, Dipti Mishra Sharma & Rajeev Sangal (2010) who discussed issues in parsing various linguistic constructions like copula, genitive, implicit and explicit conjunct, and complementizer constructions. Garapati, Uma Maheshwar Rao, Rajyarama Koppaka & Srinivas Addanki

(2012) analysed dative case marker (*-ki)* with various functions in Telugu in parsing perspective. Kesidi, Sruthilaya Reddy, Prudhvi Kosaraju, Meher Vijay & Samar Husain (2013) implemented a constraint-based dependency parser for Telugu which was earlier used for languages like Hindi. This parser deals with relations in two different stages wherein stage-1 handles intra-clausal relations and stage-2 handles inter-clausal relations. Kumari, B. V. S., & Ramisetty Rajeshwara Rao (2015) had developed combinatory categorial grammar supertags using which they claim the enhancement of identification of verbal arguments. Nagaraju, B, N. Mangathayaru & B. Padmaja Rani 2016), Kumari B. V. S. & Ramisetty Rajeshwara Rao 2017, Kanneganti S., Himani Chaudhry & Dipti Misra Sharma (2018) worked on various statistical approaches of parsers. Rama, Taraka & Sowmya, Vajjala (2018) developed a Telugu treebank using Universal Dependency (UD) tagset with an addition of language-specific tags to handle compound and conjunct verb phrases for Telugu. Gatla (2019) developed a treebank for Telugu which was trained using data-driven parsers, namely, Minimum-Spanning Tree (MST) parser and Models and Algorithms for Language Technology (MALT) parser. Nallani, Sneha, Manish Shrivastava & Dipti Mishra Sharma (2020) expanded treebank by adding language-specific intra-chunk tags to the existing annotation guidelines based on the Pāṇinian framework. In addition to improving the existing tagset, Nallani, Sneha, Manish Shrivastava & Dipti Mishra Sharma (2020b), also developed a Telugu parser using a minimal feature Bidirectional Encoder Representations from Transformers (BERT) model providing considerable results. The highest Label Attachment Score (LAS) reported so far has been 93.7% (Nallani, Sneha, Manish Shrivastava & Dipti Mishra Sharma 2020) and the approaches have been data-driven. However, the results of the above-mentioned systems prove that there should be continuous improvement in the annotated corpus size to improve the results further in data-driven approaches. Hence, the effort in building the parser for Telugu using grammar-driven approaches is attempted in this paper to study its feasibility and advantages.

3. Theoretical Background

The dependency model follows the grammatical tradition of dependency, tracing back to Pāṇini`s grammar. The dependency grammatical model represents the relation between the head and its dependents through directed arcs and arc labels. The relation between content words is marked by dependency relations; functional words are attached to the content words they modify. The parse thus generated is a tree, where the nodes of the parse tree stand for words in an utterance and the link between words represents the relation between pairs of words. All such dependencies in a sentence can either be argument dependencies (subject, object, indirect object, etc.) or modifier dependencies (determiner, noun modifier, verb modifier, etc.). The peculiar feature of the dependency model is to provide syntactico-semantic relations, unlike the other grammar formalisms, which are purely syntactic (Bresnan 1982; Gazdar Gerald, Ewan Klein, Geoffrey k. Pullum, & Ivan A. Sag, 1985). Based on these syntactico-semantic relations, Bharati Akshar, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begum & Rajeev Sangal (2009) have developed a dependency tagset known as Anncora tagset which can be used for almost all major Indian languages. This tagset consists of around 19 fine-grained tags for *karaka* (K) relations and 25 fine-grained tags for non-*kāraka* (r) relations. This study adopts the Anncora tagset in order to label dependency relations.

The most common dependency relation in a simple sentence structure includes the dependency between a noun and a verb or a noun and a noun. PG uses syntactico-semantic relations called *kāraka* relations expressed through *vibhakti*s to capture dependencies between noun-verb and non-*kāraka* relations to capture noun-noun dependencies. The pāṇinian treatment of *kāraka* relations considers a system of default *vibhakti* for each relation. This *vibhakti* assignment is independent of verb semantics. Table-1 provides the default *vibhakti* for *kāraka* relations in Telugu. In addition to this, the other tags used for the current parser are listed as part of the Appendix.

Sl.No	***kāraka* Relation**	***Vibhakti***
1	*kartā* (k1)	-0
2	*karma* (k2)	*-ni/-nu*
3	*karaṇa* (k3)	*-tō*
4	*sampradāna* (k4)	*-ki/ku*
5	*apādāna* (k5)	*nuMḍi/nuMci/niMci*
6	*ṣaṣṭhī* (r6)	*Yokka*
7	*viṣaya-adhikaraṇam (k7)*	*-lō*

Table-1: *kāraka* relations and default *vibhakti*s in Telugu

Apart from these default *vibhakti*s, there exist cases of deviation in Telugu in which there is no one-one mapping between the *vibhakti* and *kāraka* relation. These deviations arise when the verbs do not follow linguistic generalizations or when a structure is out of the scope of linguistic generalisation. In order to handle these deviations, Panini employs a model wherein he proposes two methods (Preeti 2010) viz.

1. Assigning a different *vibhakti*
2. Imposing a new *kāraka* relation

Preeti (2010) summarizes the ways of mapping semantic relations to *vibhakti*s through *kāraka*s in PG. Consider the following figure:

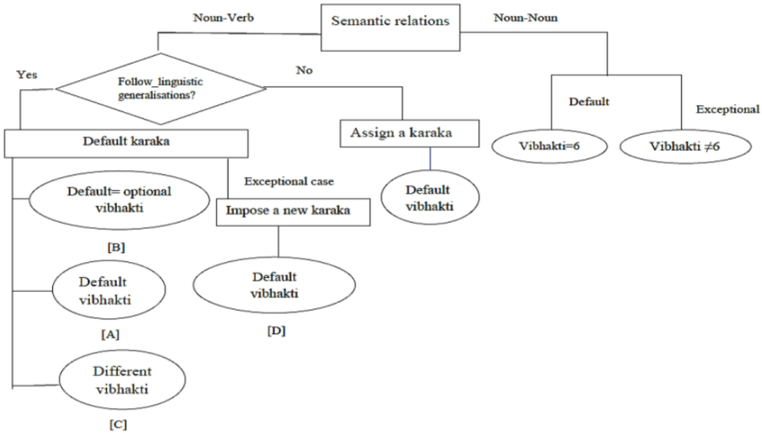

Figure-1 Semantic Relations

Based on fig-1, the semantic relations between noun-verb are divided into the following types:

3.1. Type-A

The first type of semantic relation is when the language follows the linguistic generalisation and takes a default *kāraka* as listed in Table-1. In example (1) as explicated, *kartā* (k1) and *karma* (k2) are marked with the default *vibhakti* i.e ∅ and -*ni* respectively.

1.	*nēnu.∅*	*ravi-ni*	*cūs-ā-nu.*
	I.NOM	Ravi-ACC	do-PST-1.SG.
	‘I saw Ravi’		

3.2. Type-B

In certain relations, there exist instances of verbs in addition to the default case marking which deviate from the default case marking and assign optionally other case-suffixes as in (2) and (3). The verb *ceppu* ‘to tell’ assigns either *vibhakti -ki* or *–tō* to express the relation *sampradāna* (k4) i.e the recipient of an action as in (2).

2.	*nēnu.∅*	*prakās-**ki / -tō***	*ā viSayaM*	*cepp-ā-nu*
	I.NOM	Prakash-DAT/ASS	that matter	tell-PST-1.SG
	‘I told that matter to Prakash’			

Similarly, the verb *ekku* ‘to climb’ in Telugu, has an expectancy of a noun expressing the location ‘to climb‘. In this case, the noun is marked either with the *vibhakti -nu* or *mīda* as in (3).

3.	*nēnu*	*ēnugu-*	***nu / mīda***	*ekk-ā-nu.*
	I.NOM	elephant-	ACC/on	climb up-PST-1.SG.
	‘I climbed an elephant’			

3.3. Type-C

In certain cases, it is found that a different *vibhakti* is assigned instead of the default one to indicate a particular semantic relation. For instance, the default *vibhakti* indicates the source of separation, *apādānā* i.e. the ablative case as in example (4). However, in the case of mental separation as in (5) where the *kartā,vāḍu* ‘he’ separates himself mentally due to the fear of *siMhaM* ‘lion’ which is considered as *apadānā* in PG but it is realized by the different *vibhakti* i.e. *-ki*, not by *–nuMḍi*

4.	*Cet̲t̲u*	*nuMḍi*	*ākulu*	*rālā-yi*
	Tree	From	Leaves	fall-3.PL
	‘Leaves fell from the tree’			

5.	*vāḍu*	*siMhāni-ki*	*bhayapaḍatā-ḍu*
	He	lion-	scare-3.SG.M
	‘He is scared of a lion’		

3.4. Type-D

In certain exceptional cases, it is found that a new *kāraka* is imposed using a default *vibhakti.* This can be due to the extension of the case relation as explicated in (6) where *iḷḷu* 'home' is the *karma* to the verb *veḷḷu* as per PG, however it is marked with the *vibhakti -ki.*

6.	*nēnu*	*iMti-ki*	*veḷḷ-ā-nu*
	I.NOM	house-DAT	go-PST-1.SG.
	'I went home'		

The other case as shown in Figure-1 is when the sentence does not follow linguistic generalizations and a new *kāraka* is assigned. We have not come across such cases so far in Telugu; hence no explanation is provided in this paper.

When the semantic relationship is found between noun-noun, non-*kāraka* relation i.e. *ṣaṣṭhī* (the tag 'r6') is expressed by *yokka* or the default oblique marker or by the *vibhakti-ki* in certain cases in Telugu as in (7) i.e *vādi-**ki** 'his'*.

7.	*vādi-**ki***	*kāli-ki*	*debba*	*tagil-iM-di*
	He-**DAT**	Leg-DAT	Wound-NOM	Hit-PST-3.SG.N
	'He got a wound on his leg"			

4. Parser and Algorithm

The parser takes input from sentences that are morphologically analysed and Parts of Speech (POS) tagged. Telugu morphological analyzer and POS tagger (Garapati 1999) are used as pre-processing tools. POS tagger helps in selecting the best possible morphological analysis of each word. The parser is built following the Indian theories of verbal cognition where three factors viz. *ākānksā* (expectancy), *yōgyatā* (meaning compatibility), and *sannidhi* (proximity) are used. 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. For instance, the parsed tree of the example (1) is provided as below:

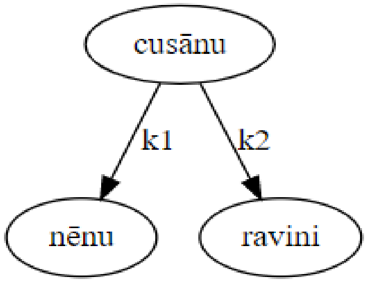

Fig-2 Parsed tree for example (1)

The basic algorithm for parsing which is followed is given below (Kulkarni 2019)

1. Define one node each corresponding to every word in a sentence
2. Establish directed edges between the nodes, if there is either a mutual or unilateral expectancy (*ākānksā*) between the corresponding words. In order to hypothesize a possible edge between two words, we refer to the expectancies of the verbs and the corresponding *vibhakti*s and then postulate a possible relation
3. Define constraints, both local on each node as well as global on the graph as a whole. One of these constraints corresponds to *sannidhi* (Proximity)
4. Use semantic constraints to filter out the meaning-wise non-congruent solutions
5. Extract all possible trees from this graph that satisfy both local and global constraints

6. Produce the most probable solution as the first solution by defining an appropriate cost function. The cost C associated with a solution tree is defined as $\boldsymbol{C = \sum_e d_e \times r_k}$ an edge from a word w_j to a word w_i with label k, $\boldsymbol{d_e = |j\text{-}i|}$, r_k rank of the role with label k. Then the problem of parsing a sentence may be modelled as the task of finding a sub-graph T of G such that T is a Directed Tree (or a Directed Acyclic Graph).

4.1 Algorithm: An Elaboration

In this section, we explain steps 2, 3, and 4 of the algorithms in detail. The step-2 corresponds to the use of lexical semantics of nouns and verbs, step-3 is the use of constraints, and step-4 is the use of selectional restriction or mutual congruity.

The **step-2** of the algorithm deals with the expectancies of verbs and the corresponding *vibhakti*s which enable the parser to postulate a possible relation. We notice that the mapping of semantic relations to *vibhakti*s is one-one except for the optional case marking (see Section 2.2), however the reverse mapping viz. *vibhakti* to semantic relation is not one-one. Case-suffixes as small as 7 (see table-1 and *ṣaṣṭhī)* in number are used to express around 40 case relations which lead to ambiguity. Ambiguities hence occurred are resolved by augmenting linguistic information such as the lexical semantics of verbs and nouns. (i) Lexical semantics of verbs. The lexical semantics of verbs provides cues in certain cases to disambiguate *vibhakti*s with their corresponding semantic relation. Consider the examples (8) & (9)

8.	*nēnu-∅*	*vāḍi-ki*	*pustakaṁ*	*icc-ā-nu*
	I.NOM	He-DAT	Book-ACC	Give-PST-1.SG
	"I gave a book to him".			

9.	*nēnu-Ø*	*baḍi-ki*	*veḷl-ā-nu*
	I-NOM	School-DAT	Go-PST-1.SG.
	"I went to the school"		

The *vibhakti -ki* is used to express two different relations viz. *sampradānā* (k4) as in (8) and goal/destination (k2p) as in (9). In such cases, the semantics of the verb is considered to disambiguate the *vibhakti*. In example (9), the verb belongs to the class of [+motion] hence it has a requirement of k2p unlike the example (8). This semantic information is augmented with syntactic rules in order to mark the appropriate relation.

(ii) Lexical Semantics of Nouns

In some cases, it is the lexical choice of nouns that helps in resolving the ambiguity. For instance, when the *vibhakti-ki/-ku* is marked with *kāla-adhikaraṇam* (k7t) or *deśa-adhikaraṇam* (k7p) relation, corresponding nouns should be either place or time denoting terms as in example (10).

10.	*ravi-Ø*	*padi-Ø*	*gaMṭalaku*	*haidarabādu-ku*	*cērukuṇ-ṭā-ḍu*
	Ravi-NOM	10	Hour-DAT	Hyderabad-DAT	Reach-FUT-3.SG.M
	"Ravi will reach Hyderabad at 10'o'clock"				

Here, the noun expressing time i.e. *padi gaMtalu* '10 'o clock', and the place i.e. *haidarabādu* 'Hyderabad' are marked with -*ku*, however, they are marked as k7p and k7t respectively based on their semantics. In such cases, a list of these terms is maintained as linguistic cues to access the information.

The step-3 of the algorithm is to define local and global constraints. The local constraints used in the parser to postulate the best possible result are given below (Kulkarni 2019):

1. A node can have one and only one incoming edge.

2. There cannot be more than one outgoing edge with the same label from the same node if the relation corresponds to a *kāraka* relation.

3. There cannot be self-loops in a graph. In addition to the local constraints, we also use global constraints like *sannidhi* 'proximity' which is a constraint that restricts crossing of edges. The sample graph satisfying all the above local and global constraints is provided below:

11.	*nēnu-Ø*	*prasādu-tō*	*rēpu*	*madrāsu-lō*	*telugu*	*sinimā-ki*	*veḷ-tā-nu*
	I-NOM	Prasad-ASS	tomorrow	Madras-LOC	Telugu	Movie-DAT	Go-FUT-1.SG
	'I will go to a Telugu movie with Prasad in Madras tomorrow.'						

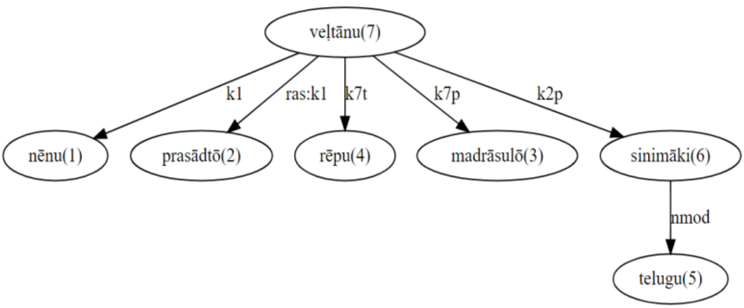

Figure-2 Sample graph for the example (11)

The use of semantic constraints is dealt with in step4 of the algorithm. It is quite important to include semantic constraints in a parser to arrive at the correct solution. For instance, the sentence *colourless green ideas sleep furiously* (Chomsky 1957) is a syntactically well-formed sentence but semantically ill-formed. The natural language feature which enables the use of semantically well-formed constructions is termed as *yōgyatā* in PG or the selectional restriction in western terminology. The selectional restriction is defined as the semantic constraint imposed on the arguments of verbs. We use selectional restriction of arguments of the verb to prune out the non-congruent solutions and arrive at a single parse. Let us consider the following examples:

12.	*tūphānu-Ø*	*illu-Ø*	*kūlc-iM-di*
	Storm-NOM	House-ACC	destroy-PST-3.SG.N
	"The storm destroyed the houses"		

***13.**	*illu-Ø*	*tūphānu-Ø*	*kūlc-iM-di*
	house-NOM	storm-ACC	destroy-PST-3.SG.N
	"Houses destroyed the storm"		

Both examples (12) and (13) are syntactically well-formed sentences, when *yōgyatā* is applied, the example (13) stands semantically ill-formed because 'Houses destroying the storm' is a semantically unacceptable sentence. In order to solve such issues, the canonical word order of a language is used as a cue.

The other instance in which we use selectional restriction is to disambiguate *kartā* and karma in Telugu. When karma is [-animate], the *vibhakti* Ø is used which is synonymous with the marker for *kartā*. In such cases, two ontological features [+/-animate] and [+/- human] could resolve the ambiguity in Telugu as well as in other Indian languages as examined by (Bharati, Akshar; Samar, Husain; Bharat, Ambati; Sambhav, Jain; Dipti, Sharma; & Rajeev, Sangal 2008). *kartā* is considered to be higher in its animacy hierarchical order in comparison with *karma*. Consider the following example:

14.	*nēnu*-∅	*pāta*-∅	*pāḍ-ā-nu*
	I.NOM	Song. ACC	sing-PST-3.SG.N
	“I sang a song”		

Here, the verb *pāḍu* ‘sing’ expects *kartā* with a semantic feature of [+human] thus, (n*ēnu*) ‘I’ is prioritized over a [-animate] entity (i.e. *patā*) ‘song’. These two semantic features proved to be quite helpful in resolving the most ambiguous relation of *kartā* and *karma*. As seen earlier, this parser exploits various linguistic information which stands crucial in disambiguating certain cases. In the next section, we present the results, which show the impact of linguistic information used in the parser.

5. Evaluation of the System

The parser is evaluated for its Labelled Attachment Score (LAS) and Unlabelled Attachment Score (UAS). In this section, the data used for evaluating parsers is presented followed by the results. Finally, we also present the error analysis and some observations.

5.1. Data

The present study selects 453 sentences to test parsers which are extracted from various sources such as (i) Telugu Grammar books viz. *telugu vākhyam* (Ramarao 1885) and A grammar of modern Telugu (Krishnamurti & Gwynn 1985) (ii) Random sentences from Telugu corpus (3 million words (CALTS[1]) corpus). The corpus contains sentences with intransitive verbs (223 sentences), transitive verbs (197 sentences), and ditransitive verbs (33 sentences). The sentences covering constructions such as copula, imperative, passive, dubitative,

1 Centre for Applied Linguistics and Translation Studies

interrogative, non-nominative subjects, reflexive and coordinating noun phrases are noticed.

5.2. Results

The results consist of the Unlabelled Attachment Score (UAS) where the dependency tree produced by the parser matches exactly with the tree from the gold data without considering the labels and the Labelled Attachment Score (LAS) which checks if the two relations and labels are correctly matched. Out of 453 sentences, 1043 relations are manually identified and annotated for the evaluation. MALT parser is developed with the data annotated. The rule-based parser produces correct dependency trees for 1001 relations and 969 correct labelled trees. Whereas MALT parser produces 928 relations, out of which 739 relations are correctly labelled. The results are provided in the table-2.

Parser type	**UAS**	**LAS**
Rule-Based Parser	96.5%	92.9%
MALT parser	89%	70.85%

Table 2: Results

Further, the rule-based parser output is analysed with different sentence structures as given in Table-3. The exact match and partial match of sentences are also identified.

Sentence Type	**No. of sentences**	**exact match**	**partial match**	**UAS**	**LAS**
Intransitive	223	208	18	97.6%	95.5%
Transitive	197	152	40	97%	92.4%
Ditransitive	33	20	11	86.6%	80%
Copula constructions	87	68	16	92.5%	80%

Imperative constructions	25	15	8	68%	52%
Dubitative constructions	56	36	18	64%	56%
Passive constructions	33	28	3q	90%	81%
Non-nominative subject constructions	66	38	25	48%	41%
Reflexive constructions	17	8	7	46%	33%
Interrogative constructions	62	48	10	85%	77%

Table-3 Simple sentence structures and results

The parsing errors in these simple sentence structures are studied which help in improving further the rules in the rule-based parser for Telugu.

5.2. Error Analysis and Observations

In this section, we discuss certain cases where the rule-based parser fails to provide the appropriate results. The current rule-based parser has a difficulty in dealing with the coordinating noun phrases and with certain pro-drop constructions. As seen in the example (15), the noun phrases *gāli nīru* 'air and water' are co-ordinating noun phrases, but the linguistic cue to express them as coordination such as either comma (,) (i.e., *gāli nīru*) or the vowel-length in the end (*gālī nīrū*) are not present. This makes the system identify them wrongly as separate relations.

15.	*ā*	*prāMtaM*-lō	*gāli nīru*	*lēvaṭa*
	that	place-LOC	water air	be-NEG-QUO
	"There is no water or air in that place"			

Certain verbs in Telugu do not show agreement with the *kartā*. In example (16), when the verb expresses the mood of possibility with the auxiliary verb *vaccu,* it does not show agreement with the verb. When the *kartā* is pro-dropped, the system identifies the *karma* (i.e. *cēpa* 'fish'), the zero-marked as *kartā*. Consider the example below:

16.	*cēpa.Ø*	*tin-a-vaccu*
	Fish.ACC	eat-INF-POSS
	"(subject) can eat fish"	

The other two reasons for the failure of the parser in certain cases are due to the wrong output from the pre-processing tools and the lack of a database for the parser. These are handled by correcting the pre-processing output and improving the database (vocabulary). Whereas, in data-driven parsers like MALT, it is difficult to improve the accuracy unless a huge annotated corpus is trained again.

6. Conclusion

This paper deals with building a rule-based parser for Telugu experimenting with simple sentences. A discussion on the application of the Pāṇinian grammatical model to Telugu and the algorithm is provided. This paper explains how the use of two semantic features viz. animacy and humanity enables the unambiguous marking of *kartā* and *karma* relations. The results show that the rule-based parser proves to be better than the data-driven parser due to the inclusion of linguistic information. Further, the study aims to improve the accuracy of the pre-processing tools and also build the required database for Telugu parsing. The next phase of the study will focus on implementing the rule-based parser for all the sentence structures in Telugu and extending this algorithm to other Indian languages.

Appendix - List of tags used in the Telugu Parser

k1 (*kartā* 'Agent') **k2** (*karma* 'patient/goal') **k3** (*karaṇa* 'instrument') **k4**(*sampradāna*'beneficiary') **k4a** (*anubhavāı kartā* 'Experiencer') **k5** (*apādāna* 'Source') **k7** (*viṣaya-adhikaraṇam* 'location elsewhere') **k7t** (*kāla-adhikaraṇam* location in time) **k7p** (*deśa-adhikaraṇam* 'location in space') **k2g**(*gounakarma*'secondary karma')	**r6**(*ṣaṣṭhī karma* 'genitive') **rh** (*hetuḥ* 'reason') **rt** (*tātparya* 'purpose') **k1s**(*kartṛsamānādhikaraṇam*'complement of a *kartā*') **k2s** (*karmasamānādhikaraṇam* 'complement of a *karma*') **adv** (*kriyāviśeṣaṇnam* adverbs) **k*u**(*sādrishya* 'similarity') **rd** ('direction')	**ras-k*** (*upapada sahakārakatwa* 'associative') **case** ('for postpositions') **det** ('determiner') **enm**(enumerator (number words)) **jjmod**('adjective modifier' **lwg**('local word grouping) **nmod** ('noun modifier') **r6v** ('verb and noun relation') **rsym** ('symbols) **title** ('titles of names') **vmod** ('verb modifier')

References

AMBATI, BHARAT RAM; PHANI GADDE & KARAN JINDAL. 2009. Experiments in Indian Language Dependency Parsing. *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*. 32-37.

BHARATI, AKSHAR & RAJEEV SANGAL. 1993. Parsing Free Word Order Languages in the Paninian Framework. *31st Annual Meeting of the Association for Computational Linguistics.* 105-111.

BHARATI, AKSHAR, VINEET CHAITANYA, RAJEEV SANGAL & K. V. RAMAKRISHNAMACHARYULU. 1995. *Natural Language Processing: A Paninian Perspective*. New Delhi: Prentice Hall of India.

BHARATI, AKSHAR, SAMAR HUSAIN, BHARAT AMBATI, SAMBHAV JAIN, DIPTI SHARMA & RAJEEV SANGAL. 2008. Two Semantic Features Make All the Difference in Parsing Accuracy. *Proceedings of ICON* 8.

BHARATI, AMBATI, DIPTI MISRA SHARMA, SAMAR HUSAIN, LAKSHMI BAI, RAFIYA BEGUM & RAJEEV SANGAL. 2009. AnnCorra: TreeBanks for Indian Languages, Guidelines for Annotating Hindi TreeBank (version–2.0). *LTRC. Hyderabad:* IIIT Hyderabad.

BRESNAN, JOAN. 1982. Control and Complementation. *Linguistic Inquiry* 13(3). 343-434. http://www.jstor.org/stable/4178286 (Accessed 2nd June 2021).

CHOMSKY, NOAM. 1957. *Syntactic Structures* (2nd edition). Mouton de Gruyter: The Hague.

GARAPATI, UMA MAHESHWAR RAO. 1999. *Morphological Analyzer for Telugu*. (Electronic form). Hyderabad: University of Hyderabad.

GARAPATI, UMA MAHESHWAR RAO., RAJYARAMA KOPPAKA & SRINIVAS ADDANKI. 2012. Dative case in Telugu: A Parsing

Perspective. *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages.* 123-132.

GATLA, PRAVEEN. 2019. Dependency Parsing for Telugu Using Data-driven Parsers. *Language in India* 19(1). 185-197.

GAZDAR, GERALD, EWAN KLEIN, GEOFFREY K. PULLUM, & IVAN A. SAG. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

HUSAIN, SAMAR. 2009. Dependency Parsers for Indian Languages. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*.

KRISHNAMURTI, BH. & J. P. L. GWYNN. 1985. A Grammar of Modern Telugu. New Delhi: Oxford University Press.

KANNEGANTI, SILPA, HIMANI CHAUDHRY & DIPTI MISRA SHARMA. 2018. Comparative Error Analysis of Parser Outputs on Telugu Dependency Treebank. In Gelbukh A. (ed.), *Computational Linguistics and Intelligent Text Processing. CICLing 2016*. Lecture Notes in Computer Science 9623. Springer Cham. https://doi.org/10.1007/978-3-319-75477-2_28.

KESIDI, SRUTHILAYA REDDY, PRUDHVI KOSARAJU, MEHER VIJAY & SAMAR HUSAIN. 2013. Constraint based Hybrid Dependency Parser for Telugu. Hyderabad: International Institute of Information Technology Hyderabad doctoral dissertation.

KUMARI, B. VENKATA SESHU & RAMISETTY RAJESHWARA RAO. 2015. Improving Telugu Dependency Parsing Using Combinatory Categorial Grammar Supertags. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 14(1), 1-10.

KUMARI, B. VENKATA SESHU & RAMISETTY RAJESHWARA RAO. 2017. Telugu Dependency Parsing Using Different Statistical Parsers. *Journal of King Saud University-Computer and Information Sciences* 29(1). 134-140.

KULKARNI, AMBA SHEETAL POKAR & DEVANAND SHUKL. 2010. Designing a Constraint-based Parser for Sanskrit. *International Sanskrit Computational Linguistics Symposium*. 70-90.

KULKARNI, AMBA., & K. V. RAMAKRISHNAMACHARYULU. 2013. Parsing Sanskrit Texts: Some Relation Specific Issues. *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium.*

KULKARNI, AMBA. 2013b. A Deterministic Dependency Parser with Dynamic Programming for Sanskrit. *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*. 157-166.

KULKARNI, AMBA. 2019. *Sanskrit Parsing Based on the Theories of Sabdabodha.* Indian Institute of Advanced Study, Shimla and DK Publishers (P) Ltd.

NAGARAJU, B., N. MANGATHAYARU & B PADMAJA RANI. 2016. Dependency Parser for Telugu Language. *ICTCS '16: Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. 1-5.

NALLANI, SNEHA, MANISH SHRIVASTAVA, M., & DIPTI MISHRA SHARMA. 2020. A Fully Expanded Dependency Treebank for Telugu. *Proceedings of the WILDRE5–5th Workshop on Indian Language Data: Resources and Evaluation*. 39-44.

NALLANI, SNEHA, MANISH SHRIVASTAVA, M., & DIPTI MISHRA SHARMA. 2020b. A Simple and Effective Dependency Parser for Telugu. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. 143-149.

NIVRE, JOAKIM .2000. Statistical Parsing. *Handbook of Natural Language Processing* 2. 525–543.

PATEL PREETI K. 2010. *Vibhakti Divergence Between Sanskrit and Hindi*. Hyderabad: University of Hyderabad dissertation.

POLLARD, CARL JESSE & IVAN A. SAG. 1994. *Head-driven Phrase Structure Grammar*. Chicago: University of Chicago Press.

RAMA, TARAKA & SOWMYA, VAJJALA. 2018. A Dependency Treebank for Telugu. *English Conference Papers, Posters and Proceedings* 8. https://lib.dr.iastate.edu/engl_conf/8.

RAMARAO, CHEKURI. 1885. *Telugu Vākyaṁ*. Hyderabad: AP Sahitya Academy.

VEMPATY CHAITANYA, VISWANATHA NAIDU, SAMAR HUSAIN, RAVI KIRAN, LAKSHMI BAI, DIPTI MISHRA SHARMA & RAJEEV SANGAL. 2010. Issues in Analyzing Telugu Sentences towards Building a Telugu Treebank. In Gelbukh A. (ed.), Computational Linguistics and Intelligent Text Processing. *CICLing* 2010. Lecture Notes in Computer Science 6008. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-12116-6_5.

***

Cite This Work:

P., SANGEETHA, PARAMESWARI K. & AMBA KULKARNI. A Rule-based Dependency Parser for Telugu: An Experiment with Simple Sentences. *Translation Today*, Vol. 15(1). 123-144. DOI:10.46623/tt/2021.15.1.ar5