Comparative Error Analysis of Parser Outputs on Telugu Dependency Treebank

by

Silpa Kanneganti, Himani Chaudhry, Dipti Misra Sharma

in

17th International Conference on Intelligent Text Processing and Computational Linguistics

Konya, Turkey

Report No: IIIT/TR/2016/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2016

Comparative Error Analysis of Parser Outputs on Telugu Dependency Treebank

Silpa Kanneganti, Himani Chaudhry, Dipti M. Sharma

Kohli Center on Intelligent Systems (KCIS)
International Institute of Information Technology, Hyderabad (IIIT Hyderabad)
Gachibowli, Hyderabad, Telangana 500032
silpa.kanneganti@research.iiit.ac.in, himani@research.iiit.ac.in, dipti@iiit.ac.in

Abstract. We present a comparative error analysis of two parsers - MALT and MST on Telugu Dependency Treebank data. MALT and MST are currently two of the most dominant data-driven dependency parsers. We discuss the performances of both the parsers in relation to Telugu language. we also talk in detail about both the algorithmic issues of the parsers as well as the language specific constraints of Telugu.The purpose is, to better understand how to help the parsers deal with complex structures, make sense of implicit language specific cues and build a more informed Treebank.

1 Introduction

In this paper we present a comparative error analysis of two dependency parsers - MALT[1] and MST[2], on Telugu Treebank data. We discuss the performances of both the parsers in relation to Telugu language, to better understand language specific workings of each of the models as well as the nuances of the language much better. The purpose is, to figure out better ways to help the parsers deal with complex structures, make sense of implicit language specific cues and build a more informed Treebank. McDonald and Nivre, (2007) [3] talk in detail about the errors of data driven parsing models, (Husain, 2012)[4] reports error analysis of MALT on Hindi language data. MALT and MST are currently two of the most dominant data-driven dependency parsers. MALT adopts a local, greedy, transition-based approach, and MST takes to global, exhaustive, graph based modeling. While they are neck and neck when it comes to parsing accuracy, their approaches could not be more different from each other. Further, (Vempaty et al., 2010) [5] discuss their observations with regard to Telugu language, in their attempt to build a Telugu treebank.

2 The Parsers

MALT [1] is a classifier based Shift/Reduce parser, that employs transition-based inference algorithms [6] for parsing. History based feature models are used for predicting the next parser action [7] and Support vector machines to map histories to parser actions [8]. MST uses Maximum Spanning Tree algorithm [9],[10] for non-projective parsing and Eisner's algorithm for projective parsing [11]. It uses large-margin structured learning [12] as the learning algorithm.

3 The Data

3.1 Telugu Language

Telugu, a morphologically,syntactically complex language, is highly inflectional and agglutinative. It is a nominative-accusative language, with SOV as its default word order. The verbs exhibit a rich inflectional morphology. Hence it encodes various grammatical categories like tense, case, gender, number, person, negatives, imperatives etc. morphologically. [13]

3.2 The Treebank Data

The data set we use is from the ICON10 parsing contest (Husain et al.,2010)[14]. The data, comprising of 1451 sentences was annotated using the Computational Pāṇinian Grammar (CPG) dependency annotation scheme [15,**?**]. We use inter-chunk dependency trees. A description of the dependency relations used in the CPG framework has been provided by Chaudhry et al., (2013). [16]. Some of the issues that arise while trying to parse a free word order language like Telugu are [4]

- difficulty in extracting relevant linguistic features
- lack of explicit cues
- long distance dependencies
- complex linguistic phenomena

3.3 Error Classification

The error analysis is carried out on MALT and MST parsers trained on Telugu treebank based on 4 properties: Sentence length, Arc length, Arc type, Arc depth[3][4]. While analyzing errors we tried to neglect the ones caused by algorithmic limitations, learning and data sparsity issues. Our focus is on the errors that could be occurring due to lack of robust features and complex structures too difficult to learn.

3.4 Experiment Setup

For experiment purposes, we are using MALT [1] [1] with parser settings from Ambati et al. (2010b)[17] and best settings from Ambati et al.(2009)[18] for MST[2]. We are using maxent tool [3] for labeling MST parsed trees. The train and the test set contain 1161, 290 sentences respectively. Error classification is based on the test data. Table 1 depicts the accuracies of 5fold cross validation and the

[1] MALT version 1.8.1
[2] MST version 0.5.0
[3] http://homepages.inf.ed.ac.uk/lzhang10/maxent.html

train and test set being used for the analysis. The evaluation is based on LAS[4], although LAS, UAS[5] and LS[6] are being reported.

	LAS		UAS		LS	
	MALT	MST	MALT	MST	MALT	MST
5fold	67.36%	67.24%	90.5%	89.1%	69.3%	68.6%
Best	76.14%	75.84%	95.63%	92.18%	77.36%	78.62%

Table 1. Accuracies

4 Error Analysis

Error Analysis in this paper is discussed in two parts. In Section 5.1 we talk about linguistic errors that are common and different between both the parsers as well as Telugu specific parsing issues, while in section 5.2 we talk about algorithm related parser statistics and errors of both the parsers.

4.1 Linguistic Analysis of Errors

In this section we discuss the linguistic aspect of the errors of both the parsers. Below, we discuss the types and causes of these errors.

4.1.1 Errors Common to Both the Parsers

Complements and Adjuncts:
Ambiguity between a subject and its complement due to lack of identification of post position.

(1) neVlalaku piMdam velaMwa uMtuMdi
In-months embryo size-of-a-finger is
'In months the embryo is the size of a finger.'

The post-position 'aMwa' could be a cue to identify 'velaMwa' as the subject compliment 'k1s' here. Instead, it is being marked as another subject 'k1' along with 'piMdam', which is the actual 'k1' in the sentence, as seen in example 1. The post-position, in turn is perhaps not being identified due to the language's tendency to manifest post-positions agglutinated with the word.

(2) neVlalaku piMdam vel**aMwa** uMtuMdi
In-months embryo size-of-finger is
'In months the embryo is the size of a finger.'

[4] LAS – Labeled Attachment Score
[5] UAS – Unlabeled Attachment Score
[6] LS - Labeled Score

Being a null subject language, Telugu tends to drop the pronominal subject of a finite clause. Also, agglutination of the subject marker with the verb further adds to the complexity of the sentence structures. We observe that in cases where the inherent subject is absent, the object is being misannotated as the subject.

(3) oVkati wulasiki iccAdu
one(thing) to-Tulasi he-gave
'He gave one to Tulasi.'

Due to the absence of inherent subject and agglutination of the subject indicator 'du' (he) with the verb, in example 3 the NP 'oVkati' is being misidentified as subject.

Further, not just Telugu verbs agglutinate with the subject markers such as 'ru' (they), 'du' (he), 'di' (she/it), but both, Telugu nouns and verbs can also agglutinate conditionals such as 'we' (if - example 6) and emphatic markers such as 'lone' (itself - example 5)

(4) illu KAlYI ceSA**du** ani ceVppA**ru**
house vacate he-did that they-said
'They said that he vacated the house.'

(5) yavvanaM**lone** kAlYlu cewulu AdawAyi
In-youth-itself legs hands move
'youth is when legs and hands move.'

In example 5, 'yavvanaMlone' that is,'yavvanaM' (youth) + 'lone' (in-itself) is a case of agglutination of the emphatic marker 'lone' with the noun 'yavvanaM'.

In example 6, instead of identifying the relations of 'OVwwidi' (stress) as subject (k1) and of 'padi' (put/falls) as the adjunct reason (rh), it is mislabeling them as the 'noun of a complex predicate' (pof), and 'verb modifier' (vmod), respectively

(6) manasumIxa oVwwidi padiwe vaswuMxi
on-the-heart stress if-put/falls comes
'It comes when the heart is stressed out.'

Adjuncts:

Quantifiers (QF) and Quotatives (UT) are being mislabeled as 'karma' instead of verb modifiers.

(7) waggipowuMxi ani veVlYlipoyAdu
will-go-away saying he-left
'He left saying it will go away.' In example 7 'ani' (quotative that) is mislabeled with the object label (k2), instead of being identified as a verb modifier and being labeled 'vmod'.

Noun Modifiers:

(8) maMxula RApuku poyAdu
medical to-shop he-went
'he went to the medical shop.'

In example 8 'maMxula' (Medical) is a modifier of the noun 'RApu-ku' (to-shop) and should be annotated as the noun-modier (nmod). Instead, it is being marked object (k2) of 'poyAdu' (he went).

Genitives:
In Telugu, the genitive marker is often covert (dropped), and the relation must be inferred. In such cases, error in identifying a genitive relation between a subject (noun) and its complement (noun) is observed.

(9) bidda walli kadupulo peVruguwuMxi
baby mother stomach grows
'Baby grows in its mom's stomach.'

In example 9, the noun 'walli' does not have an overt genitive marker. Thus the parsers are unable to infer the relation between 'walli' (mother) and 'kadupulo' (stomach) as a genitive relation, and marks it as a child of the verb 'peVruguwuMxi' (grows).

(10) rAmulu kUwuru ramA caxuvukuMxi
Ramulu daughter Rama studied
'Ramulu's daughter Rama studied.'

In the phrase 'rAmulu kUwuru' in 10, 'rAmulu' (Ram) has a genitive relation with 'kUwuru' (daughter) but the parsers are unable to infer it in the absence of an explicit case marker.

However, if the sentence had an overt genitive marker, say 'yoVkka' ('s), as in example 11 it would perhaps be easier for the parsers to identify the relation correctly.

(11) rAmulu yoVkka kUwuru ramA caxuvukuMxi
rAmulu 's daughter ramA studied
'Ramulu's daughter Rama studied.'

Complex Predicates:
We note that the parsers are erroneously annotating noun components of complex predicates with other relations, such as location, subject, object, etc.

(12) maMxulu vAdiwe rakwapotu axupulo uMtuMxi ani weVlusukunnAru
medicines if-used blood-pressure in-control will-be that-they have-learnt
'They have learnt that if medicines are used blood pressure can be kept in control.'

In example 12, 'axupulo' is annotated as a location, with the label (k7p), instead of being marked as a part of (pof) the complex predicate. Thus, more context and semantic information is needed to handle Complex predicates.

Conjunctions:
Broadly, the word ‘ani’ in Telugu, either occurs as a subordinating conjunct or as a complementizer (that). It is a Quotative (UT) quite loaded semantically. Instances of ‘ani’ are being misidentified as the head of the sentences, as seen in example 13

(13) nIku nagalu ceyiswAnu ani javAbu uMxi
for-you jewellery i-will-get-made ani answer is-there
‘“I will get jewellery made for you” the answer is there.’

Erroneous errors:
Parser tree error are seen to be occurring in the output broadly as occurrence of multiple children with same dependency label.

Error in differentiating between a subject and its complement, due to lack of identification of the post-position, as seen (and discussed) in example 2 also falls under the purview of the parse tree error type.

4.1.2 Errors Different to Both the Parsers

As discussed elsewhere in the paper, each of the parsers performs better over the other, for some categories. Below, we report some such errors:

Complements:

(14) nIvu beVMga peVttukovaxxu
you worry don’t-keep
glt ‘you don’t worry.’

In example 14, the NP beVMga ‘worry’, instead of being annotated, the object of the verb is being marked with the relation ‘part-of’ (pof) to indicate that it is the Nominal component of a Complex predicate by MST. Whereas MALT is able to annotate such cases correctly, by and large.

Noun modifiers:

(15) sAkRi saMwakaM uMxi
witness signature is-there
‘The signature of the witness is there.’

In example 15, ‘sAkRi’ (witness), a modifier of the nominal node ‘saMwakaM’ (signature), is being annotated as the subject of the verb ‘uMxi’ (is there) by MALT. MST is annotating such cases correctly. Thus, in such cases, not only does the verb have two children annotated with the relation ‘kartā’ (k1), (making it a case for ’Erroneous errors’), noun modifiers are also being incorrectly annotated. (MALT is seen to prefer longer arcs over more convenient shorter arcs in case of nmods and genitives)

Genitives:

(16) gexe pAlu wIsi pattukupoyevAdu
buffalo milk after-taking he-used-to-take-with-him
'After taking the buffalo milk, he used to carry it with him.'

We see in example 16, that though 'gexe' (buffalo) has a genitive relation with the Noun 'pAlu' (milk), it is being incorrectly annotated as a place/location (k7p) by MALT. MST, on the other hand, is seen to annotate such cases correctly.

In section 4.1 we discussed some examples of the parser outputs comparing the performance of the two parsers. Though there are other categories in which one of the parsers performs better than the other, it is beyond the scope of the paper to discuss all of the examples.

4.2 Parser Specific Errors

In this section we discuss the algorithm specific statistics and errors of both the parsers. The purpose of this is to analyze the type of sentences each parser is comfortable with. The parameters for this are Sentence Length, Arc Length, Arc Depth and Arc Type, four major factors known to contribute to errors in data driven dependency parsing[3]. Due to lack of Non-projective sentences in Telugu treebank data[19], we are not discussing Non-projectivity in this paper.

4.2.1 Sentence Length

Table 2 shows the percentage of errors for both the parsers relative to sentence length.

S.L	MALT	MST
1	0	0
2	38	34
3	56	58
4	68	70
5	65	75
6	100	100
7	100	100
8	0	100

Table 2. Parser errors for different Sentence Lengths

Observations:
As can be noted from the above table 2, MALT performs better than MST on

shorter sentences. The reason for this could be that the greedy inference algorithm that Malt parser employs has to make less parsing decisions with shorter sentences as compared to longer sentences. For the longer sentences, the near exhaustive search algorithm that MST uses is more effective. Also, the rich feature representation of MALT helps with it having less error propagation issues than MST parser[3].

4.2.2 Arc Length

Arc length corresponds to the linear distance between a head and its child. Table 3 shows the number of labeled arcs each parser is getting wrong over all arc lengths against the type of the arcs.

	MALT					MST				
Arc.length	1	2	3	4	5-8	1	2	3	4	5-8
main(290)	1	0	0	2	0	1	0	0	1	0
Complements(379)	68	27	5	0	0	62	34	8	1	0
Adjuncts(183)	22	23	1	1	0	34	25	4	1	0
Noun Mods(10)	10	0	0	0	0	4	0	0	0	0
Adj Mods(0)	0	0	0	0	0	0	0	0	0	0
Apposns(0)	0	0	0	0	0	0	0	0	0	0
Genitive(16)	5	0	0	0	0	2	0	0	0	0
Conj(25)	0	0	0	0	0	4	0	0	0	0
Comp Preds(32)	27	0	0	0	0	25	0	0	0	0
Rel Clause(4)	3	0	0	0	0	4	0	0	0	0
Coord(8)	0	2	0	0	0	0	0	0	0	0
subord(0)	0	0	0	0	0	0	0	0	0	0

Table 3. Parser errors for different Arc Lengths

- Most of the errors for both MALT and MST are at Arc lengths 1 and 2.
- For MALT, 4% of arcs with length 3 and 4, 34% with length 1 and 23% with length 2 are incorrect. For MST, 7% arcs with length 3, 5 % with length 4, 35% with length 1 and 25% with length 2 are incorrect.
- More than 95% of verbal Complement and Adjunct errors are being shared by lengths 1 and 2 for both the parsers
- Both report Noun modification, Genitive, Complex Predicate and Relative clausal errors at length 1.
- Intra-clausal coordination errors do not show in MALT, but MST is getting 16% of them wrong at length 1.
- While MALT reports Inter-clausal coordination errors at length 2 MST does not have any

Observations:
From table 3, MALT performs better than MST for short arcs. This can be explained by the fact that shorter arcs are created before the longer arcs in MALT's greedy parsing algorithm and hence are less prone to error propagation.[3]. Given that the average sentence length in the treebank is relatively small , nothing conclusive can be said about longer arcs.

4.2.3 Arc Depth

The depth of an edge is the level at which it is situated in the tree. Table 4 shows the number of labeled arcs each parser is getting wrong over all arc depths against the type of the arcs.

	MALT					MST				
Arc.Depth	1	2	3	4	5-8	1	2	3	4	5-8
Complements	79	15	6	0	0	85	15	4	1	0
Adjuncts	40	5	0	2	0	52	8	1	3	0
Noun Mods	0	10	0	0	0	0	4	0	0	0
Adj Mods	0	0	0	0	0	0	0	0	0	0
Apposns	0	0	0	0	0	0	0	0	0	0
Genitive	0	4	1	0	0	0	1	1	0	0
Conj	0	0	0	0	0	0	4	0	0	0
Comp Preds	24	2	1	0	0	22	2	1	0	0
Rel Clause	0	2	0	1	0	3	0	1	0	0
Coord	2	0	0	0	0	0	0	0	0	0
subord	0	0	0	0	0	0	0	0	0	0

Table 4. Parser errors for different Arc Depths

- Most of the errors occur at Arc depth 1 and 2 for both the parsers.
- For MALT, 25% and 33 % of arcs with depth 3 and 4 respectively, 21% with depth 1 and 29% with depth 2 are incorrect. For MST 22% arcs with depth 3, 55% with depth 4, 23% with depth 1 and 27% with depth 2 are incorrect.
- For both the parsers around 95% of verbal complement errors are shared between Arc Depths 1 and 2.
- 85% of Adjunct errors are at depth 1, for MALT, MST stands at 81%.
- Both parsers report all the nmod errors at depth 2.
- All the Genitive errors for both occur at arc depths 2 and 3 with MALT showing more errors at depth 2 than MST.
- Intra-clausal coordination errors do not crop up in MALT parser, MST is reporting all of its errors at depth 2.
- Both the parsers are reporting most of their Complex Predicate errors at depths 1.
- All Relative clausal errors at for MALT are at depths 2 and 4, while MST is showing them at 1 and 3.

– MALT reports Inter-clausal coordination errors, at depth 1 while MST does not have any.

Observations:
Table 4 shows that for arcs close to the root, MST is more precise than MALT and vice-versa for arcs further away from the root. This could be because the dependency arcs further away from the root are being constructed early in the parsing algorithm of MALT[3]. MALT's reduced likelihood of error propagation and rich feature representation help the parser in this case.

4.2.4 Arc Type

Arc type is defined by its dependency label. Appendix 1[4] lists the different types of labels that are used to annotate the treebank. Given their number, they are classified into coarser classes (Husain, 2012)[4].(Table 5).

Table 5. Percentages of Arc Type Errors

	Telugu	
Arc.Type	MALT	MST
Main	1	.6
Complements	25	27
Adjuncts	25	35
Noun Modifiers	100	40
Adj Modifiers	0	0
Apposition	0	0
Genitives	31	12
Conjunctions	0	16
Complex Predicates	84	78
Relative Clauses	75	100
Coordination	25	0
Subordination	0	0

4.2.5 Observations

It can be inferred from table 5 that MALT performs better on Main, complements, Adjuncts, Intra-clausal coordination and Rel clause type of arcs while MST is better on Noun Modifiers, Genitives, Complex Predicates, Inter-clausal coordination arcs. It is surprising that MST is doing better with classes that require shorter arcs like Noun modifiers and Genitives given that MALT is meant to perform better for shorter arcs. Due to lack of implicit cues in Telugu, the words that are failing to attach as modifiers are being connected to roots. MALT over-predicting root modifiers could be blamed for this. Also MST doing better with Inter-clausal errors, could be due to the their arc lengths being usually big and close proximity to the root. [3]

5 Conclusion and Future Work

We presented the differences in errors caused by both MALT and MST parsers for Telugu Language. Through the insights we gained from these experiments, we hope to build a better parsing system as well as a more informed treebank for Telugu. In the future we plan to extend these experiments to other Indian languages - Hindi, Bengali and Urdu etc.

6 Acknowledgments

We thank Riyaz Ahmad Bhat, Vigneshwaran Muralidharan and Irshad Ahmad Bhat for their assistance and comments that greatly improved the manuscript.

References

1. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering **13** (2007) 95–135
2. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 523–530
3. McDonald, R.T., Nivre, J.: Characterizing the errors of data-driven dependency parsing models. In: EMNLP-CoNLL. (2007) 122–131
4. Husain, S., Agrawal, B.: Analyzing parser errors to improve parsing accuracy and to inform tree banking decisions. Linguistic Issues in Language Technology **7** (2012)
5. Vempaty, C., Naidu, V., Husain, S., Kiran, R., Bai, L., Sharma, D.M., Sangal, R.: Issues in analyzing telugu sentences towards building a telugu treebank. In: Computational Linguistics and Intelligent Text Processing. Springer (2010) 50–59
6. Nivre, J.: Inductive dependency parsing, volume 34 of text, speech and language technology (2006)
7. Black, E., Jelinek, F., Lafferty, J., Magerman, D.M., Mercer, R., Roukos, S.: Towards history-based grammars: Using richer models for probabilistic parsing. In: Proceedings of the Workshop on Speech and Natural Language. HLT '91, Stroudsburg, PA, USA, Association for Computational Linguistics (1992) 134–139
8. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: Proceedings of the 6th Conference on Natural Language Learning - Volume 20. COLING-02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 1–7
9. Chu, Y.J., Liu, T.H.: On shortest arborescence of a directed graph. Scientia Sinica **14** (1965) 1396
10. Edmonds, J.: Optimum branchings. Journal of Research of the National Bureau of Standards B **71** (1967) 233–240

11. Eisner, J.M.: Three new probabilistic models for dependency parsing: An exploration. In: Proceedings of the 16th conference on Computational linguistics-Volume 1, Association for Computational Linguistics (1996) 340–345
12. McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: Proceedings of the 43rd annual meeting on association for computational linguistics, Association for Computational Linguistics (2005) 91–98
13. Garapati, U.R., Koppaka, R., Addanki, S.: (Dative case in telugu: A parsing perspective)
14. Husain, S., Mannem, P., Ambati, B.R., Gadde, P.: The ICON-2010 tools contest on Indian language dependency parsing. Volume 10. Citeseer (2010)
15. Bharati, A., Chaitanya, V., Sangal, R., Ramakrishnamacharyulu, K.: Natural Language Processing: A Paninian Perspective. Prentice-Hall of India (1995)
16. Chaudhry, H., Sharma, H., Sharma, D.M.: Divergences in english-hindi parallel dependency treebanks. DepLing 2013 (2013) 33
17. Ambati, B.R., Husain, S., Nivre, J., Sangal, R.: On the role of morphosyntactic features in hindi dependency parsing. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, Association for Computational Linguistics (2010) 94–102
18. Ambati, B.R., Gadde, P., Jindal, K.: Experiments in indian language dependency parsing. Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing (2009) 32–37
19. Bhat, R.A., Sharma, D.M.: Non-projective structures in indian language treebanks. In: The 11th International Workshop on Treebanks and Linguistic Theories, Edições Colibri (2012) 25–30