

Effect of Minimal Semantics on Dependency Parsing

Bharat Ram Ambati
LTRC
IIIT-Hyderabad
India
ambati@students.iiit.ac.in

Pujitha Gade
LTRC
IIIT-Hyderabad
India
pujitha@students.iiit.ac.in

Chaitanya GSK
LTRC
IIIT-Hyderabad
India
chaitanya_gsk@students.iiit.ac.in

Samar Husain
LTRC
IIIT-Hyderabad
India
samar@research.iiit.ac.in

Abstract

In many languages general syntactic cues are insufficient to disambiguate crucial relations in the task of Parsing. In such cases semantics is necessary. In this paper we show the effect of minimal semantics on parsing. We did experiments on Hindi, a morphologically rich free word order language to show this effect. We conducted experiments with the two data-driven parsers MSTParser and MaltParser. We did all the experiments on a part of Hyderabad Dependency Treebank. With the introduction of minimal semantics we achieved an increase of 1.65% and 2.01% in labeled attachment score and labeled accuracy respectively over state-of-the-art data driven dependency parser.

Keywords

Minimal semantics, dependency parsing, free word order language, Malt parser, MST parser.

1. Introduction

Parsing morphologically rich free word order language like Hindi¹ is a challenging task. For such languages dependency based framework suits better than the constituency based one [10][19][15][2]. Data driven dependency parsers has achieved considerable success due to the availability of annotated corpora in recent years. In spite of availability of annotated treebanks, state-of-the-art parsers for these languages have not reached the performance obtained for English [16]. Small size of the treebanks, non-projectivity, complex linguistic phenomenon, long distance dependencies and lack of explicit cues are most frequently stated reasons for low performance [16][17][3].

Previously, [5] used semantic features and showed an improvement in error reduction in the LAS² up to 5.8%

¹ Hindi is a verb final language with free word order and a rich case marking system. It is one of the official languages of India, and is spoken by ~800 million people.

² Labeled attachment score

with a dependency parser trained on the WSJ Penn Treebank sections 2-21 [12]. For Hindi, [3] showed that two semantic features namely, animate and in-animate, can reduce the subject-object confusion to a large extent.

In many languages such as Hindi, general syntactic cues are sometimes insufficient to disambiguate crucial relations in the task of parsing. For such cases semantics is necessary. In this paper we try to investigate the role of minimal semantics in parsing and try to ascertain its contribution to parsing accuracy. All our experiments are on Hindi, a morphologically rich free word order language. We conducted experiments with the two data driven parsers MSTParser and MaltParser. Part of Hyderabad Dependency Treebank [1] has been used as the experimental data. With the introduction of minimal semantics there was an increase of 1.65% and 2.01% in labeled attachment score and labeled accuracy respectively over state-of-the-art data driven Hindi dependency parser [3].

The paper is arranged as follows, Section 2 gives a brief overview of the necessity of semantics. Section 3 briefly describes the semantic tagset. Section 4 presents the experiments. In Section 5 we discuss our observations. We conclude our paper with future work in Section 6.

2. Why Semantics?

To elegantly describe the varied phenomena of structure of Hindi, it is analyzed in Paninian framework [2][1]. This framework employs syntactico-semantic relations named *karakas*. *karakas* are syntactico-semantic in nature. Consequently, both Hindi dependency annotation [1] and dependency parsing follow this framework [3][4]. In this scheme, *vibhakti*³ and TAM⁴ are the crucial markers which help in identifying the correct dependency label [3]. Sometimes though, this information is unavailable to help

³ A generic term for prepositions, post-positions, suffixes.

⁴ Tense, aspect and modality.

disambiguate conflicting relations. In Hindi, this might happen, for example, when the lexical items have no postpositions. Take the following example:

- (1) *raama seba khaataa hai*
 ‘Ram’ ‘apple’ ‘eat’ ‘is’
 ‘Ram eats apple’

In (1), both ‘*raama*’ and ‘*seba*’ have \emptyset post-position and therefore there are no explicit syntactic cues that tell us that ‘*raama*’ is eating ‘*seba*’. Compare this with (2), where ‘*seba*’ is followed by a postposition that can help us identify the object/theme of the event ‘eat’,

- (2) *raama seba ko khaataa hai*
 ‘Ram’ ‘apple’ ‘ACC’ ‘eat’ ‘is’
 ‘Ram eats apple’

It should also be noted that in (1) the agreement information doesn’t help much as both the elements are masculine. Neither will word order help, as Hindi is a free word order language. In such cases where syntactic information fails, semantic features assist in disambiguating the labels, thus aiding parsing. In (1), for example, the information that ‘*raama*’ is a human or that ‘*seba*’ is an inanimate being will prove to be crucial. In fact, in (1), correct parsing is only possible if this semantic information is available. All the semantic features don’t contribute in identifying the dependency relations. Similarly, all the dependency relations are not benefited by semantic features. So an optimal set of semantic features should be used based on their positive contribution in dependency parsing. This paper is the first attempt in this direction.

3. Semantic Tags

The semantic tagset that we use in our experiments have been selected based on their closeness to the dependency labels that the parser is supposed to identify. In the Paninian framework all potential participants can participate in an action in six possible ways [2]. We have consequently kept this in mind while formulating the tagset. We call this set ‘minimal’ as the semantic labels broadly correspond to the semantic type of some of the core arguments. In this section we describe the semantic tags used for annotating the data. We also describe how these tags can help in disambiguating conflicts between some dependency labels.

3.1 Human

This tag is used to mark all the nouns which represent humans.

Eg: *mai* ‘I’, *lekhaka* ‘Author’

3.2 Non human

This tag is used to mark all the nouns which are animate but not human.

Eg: *kuttaa* ‘Dog’

3.3 Inanimate

This tag is used to mark all the inanimate nouns.

Eg: *kahaanii* ‘Story’, *seba* ‘apple’

The above three tags help in reducing k1-k2⁵ ambiguity. We saw the significance of such tags clearly in example (1) previously. Below we repeat (1) and show how k1-k2 disambiguation is done by semantic features.

- (3) *raama seba khaataa hai*
(sem-h) (sem-in)
 ‘Ram’ ‘apple’ ‘eat’ ‘is’
 ‘Ram eats apple’

In the above example both the nouns ‘*raama*’ and ‘*seba*’ have same vibhakti (\emptyset). Position cannot help in disambiguation due to free word order property of Hindi. Also, agreement info. doesn’t help either. Semantic tags of ‘*raama*’ and ‘*seba*’ are human and inanimate respectively. With the introduction of semantics, ‘*raama*’ that is marked as human can be identified as a k1 and ‘*seba*’ as k2.

These semantic tags also help in disambiguating pof (part-of relation) relation from k1 or k2. The relation between the nominal particle and its light verb is identified by a pof.

3.4 Time

All the nouns referring to time are marked with this tag. This helps in dependency labeling of the nouns with the label k7t. It indicates the time of the action.

Eg: *aaja* ‘Today’, *saala* ‘Year’

3.5 Place

All the nouns referring to place are marked with this tag. This helps in dependency labeling of the nouns with the label k7p. It is used to indicate the place where the action took place.

Eg: *skuula* ‘School’, *kheta* ‘Field’

⁵ k1 (*karta*) and k2 (*karma*) are syntactico-semantic labels which have some properties of both grammatical roles and thematic roles. k1 for example, behaves similar to subject and agent. k2 behaves similar to object and patient. For the complete tagset description, see: http://lrc.iiit.ac.in/MachineTrans/publications/technicalReports/t_r032/treebank.pdf

As nouns are explicitly marked with time and place semantic labels, identifying the dependency relations k7p and k7t becomes very easy. The features time and place not only reduce ambiguity between k7p and k7t but also reduce the ambiguity of these two labels with other labels like k1, k2 etc.

(4) *puraaNe samaya meM jMgala meM eka shera*
 ‘ancient’ ‘time’ ‘in’ ‘jungle’ ‘in’ ‘a’ ‘lion’
thaa
 ‘was’
 (sem-t) (sem-p)
 ‘In ancient times there was a lion in the jungle.’

In the above example *samaya* and *jMgala*, both the noun have ‘meM’ vibhakti. Semantic tags of ‘*samaya*’ and ‘*jMgala*’ are time and place respectively. Prediction of dependency labels k7p and k7t from these tags becomes straight forward.

3.6 Abstract

All the abstract nouns are annotated with the abstract tag. Eg: *kaama* ‘Work’, *racanaa* ‘Literary work’, *vicaara* ‘Thought’

3.7 Rest

The rest of the nouns which do not fall into any of the above categories are marked with the tag ‘rest’.

4. Experimental Setup

4.1 Parsers

We performed experiments with two data-driven parsers MaltParser⁶ [16] and MSTParser⁷ [14].

MaltParser is a transition based parser. It is a Shift/Reduce parser. It uses graph transformation to handle non-projective trees. MST has an implementation of Chu-Lui-Edmonds MST algorithm [6][8]. It uses online large margin learning as the learning algorithm [13]. Both the parsers provide an option for using different combinations of features.

4.2 Data

For our experiments we extracted 1221 sentences from HyDT [1], the dependency treebank for Hindi. Average length of these sentences is 17.83 words/sentence and 9.04 chunks/sentence. In HyDT, chunk heads appear as nodes. A chunk is a set of adjacent words which are in dependency relation with each other, and are connected to the rest of the words by a single incoming arc to the

⁶ Malt Version 1.2 (<http://maltparser.org/download.html>)

⁷ MST Version 0.4b (<http://sourceforge.net/projects/mstparser/>)

chunk. All noun chunks are manually annotated with one of the 7 semantic categories (see Section 3). We divided this data in to training, development and testing data each containing 850, 200 and 171 sentences respectively. We used experiment F3 for FEATS as this feature gives best results [3]. F3 uses TAM class for verb chunks, or vibhakti for noun chunks. We provide this feature in the FEATS column of the CoNLL format.

4.3 Experiments and Results

Both Malt and MST parsers were used during the experiments. We tried out different parser settings and applied the best one on test set. For MSTParser, non-projective algorithm, order=2 and training-k=5 gave best results. For feature model, we used conjoined feature set of [3]. For Malt Parser, arc-eager gave better performance over other algorithms. For feature model we tried out different combinations of best feature settings of the same parser on different languages in CoNLL-2007 shared task [9] and applied the best feature model on the test data. In FEATS column of CoNLL format in addition to F3, we appended manually annotated semantic categories. We name this feature as F5. Hence F5 is F3+semantic features. Results of both F3 and F5 can be seen in Table1. We evaluated our experiments based on unlabeled attachment score (UAS), labeled attachment score (LAS) and label accuracy (L).

Example 2:

puraaNe samaya meM jMgala meM eka shera
 ‘ancient’ ‘time’ ‘in’ ‘jungle’ ‘in’ ‘a’ ‘lion’
 (sem-t)(sem-p)
rehtha thaa
 ‘live’ ‘was’
 ‘In ancient times there lived a lion in the jungle.’

In the above example *samaya* and *jMgala*, both the noun chunks have ‘meM’ vibhakti. Semantic tags of ‘*samaya*’ and ‘*jMgala*’ are time and place respectively. Prediction of dependency labels k7p and k7t from place and time semantic labels is pretty easy.

Table 1. Results of MST and Malt Parsers

	MST			Malt		
	UAS	LAS	L	UAS	LAS	L

F3	86.77	65.28	69.16	88.57	69.81	72.68
F5	87.13	69.45	73.04	88.21	71.46	74.69

5. Discussion

With the introduction of semantic features there is a significant improvement in the performance of both the parsers. As expected, adding semantic features for nouns helps with label identification more than head identification. This is clearly shown by the improvements in LAS vs. UAS. For MST, there is an increase of 0.36% in UAS, 4.17% in LAS and 3.88% in L. Similarly, in case of Malt, there is an increase of 1.65% in LAS and 2.01% in L. These results clearly show that minimal semantics can help in improving the parsing accuracy. More importantly, this information in many instances cannot be done away with to get the correct parse.

Table 2 and Table 3 show the precision, recall and f-measures of the dependency labels k1, k2, pof, k7p and k7t for both the parsers. The results show that the seven semantic labels considered are indeed crucial to reduce ambiguities among these five labels.

Table 2. Tag-wise LAS for MSTParser

		MST Parser		
		Precision	Recall	F β =1
k1	F3	71.59	67.15	69.29
	F5	77.82	71.68	74.62
k2	F3	53.57	50.97	52.24
	F5	57.65	61.75	59.63
pof	F3	74.51	43.18	54.67
	F5	88.24	56.96	69.23
k7p	F3	50.00	64.18	56.21
	F5	67.44	68.24	67.84
k7t	F3	74.55	89.13	81.19
	F5	83.64	74.19	78.63

Table 3. Tag-wise LAS for Malt Parser

		Malt Parser		
		Precision	Recall	F β =1
k1	F3	77.22	77.82	77.52
	F5	77.86	82.10	79.92

k2	F3	59.91	66.33	62.96
	F5	63.05	65.31	64.16
pof	F3	44.71	74.51	55.89
	F5	51.32	76.47	61.42
k7p	F3	68.18	52.33	59.21
	F5	72.15	66.28	69.09
k7t	F3	80.85	69.09	74.51
	F5	73.21	74.55	73.87

6. Conclusion and Future Work

This paper clearly shows that minimal semantics helps in boosting the parsing accuracy. Instead of manually annotated semantic labels, we have to experiment with automatically extracted semantic labels. One way to get these labels is from the first sense of the words in Hindi WordNet [11]. Other method is to use an automatic semantic labeler [18]. We can experiment with iterative learning between Dependency Parsing and Semantic Labeling [7].

7. Acknowledgements

We would like to express our gratitude to Dr. Dipti Misra Sharma and Prof. Rajeev Sangal for their guidance and support. We would also like to thank Itisree Jena and Sushant Kishore for manually annotating the Treebank with semantic features.

References

- [1] R. Begum, S. Husain, A. Dhawaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In Proceedings of IJCNLP-2008.
- [2] A. Bharati, V. Chaitanya and R. Sangal. 1995. Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi.
- [3] A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal. 2008a. Two semantic features make all the difference in parsing accuracy. In Proceedings of ICON-08.
- [4] A. Bharati, S. Husain, D. M. Sharma, and R. Sangal. 2008b. A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In Proceedings of the COLIPS IALP, Chiang Mai, Thailand.
- [5] M. Ciaramita and G. Attardi. 2007. Dependency Parsing with Second-Order Feature Maps and Annotated Semantic Information. In *Proceedings of IWPT 2007*.

- [6] Y.J. Chu and T.H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.
- [7] Q. Dai, E. Chen, and L. Shi. 2009. An iterative approach for joint dependency parsing and semantic role labeling. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA. June 4-5.
- [8] J. Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.
- [9] J. Hall, J. Nilsson, J. Nivre, G. Eryigit, B. Megyesi, M. Nilsson and M. Saers. 2007. Single Malt or Blended? A Study in Multilingual Parser Optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*.
- [10] J. Hudson, 1984. 'Why English should be taught as a second language in Aboriginal schools in the Kimberleys', pp.99-106 in *Wikaru*, Vol.12.
- [11] S. Jha, D. Narayan, P. Pande and P. A. Bhattacharyya. 2001. WordNet for Hindi. *International Workshop on Lexical Resources in Natural Language Processing*, Hyderabad, India, January 2001.
- [12] M. Marcus, B. Santorini, and M. Marcinkiewicz. (1993). Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19:313-330.
- [13] R. McDonald, K. Crammer, and F. Pereira. 2005b. Online large-margin training of dependency parsers. In *Proceedings of ACL*. pp. 91–98.
- [14] R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005a. Non-projective dependency parsing using spanning tree algorithms. *Proceedings of HLT/EMNLP*, pp. 523–530.
- [15] I. A. Mel'cuk. 1988. *Dependency Syntax: Theory and Practice*, State University Press of New York.
- [16] J. Nivre, J. Hall, S. Kubler, R. McDonald, J. Nilsson, S. Riedel and D. Yuret. 2007b. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of EMNLP/CoNLL-2007*.
- [17] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E. Marsi. 2007a. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2), 95-135.
- [18] S. Reddy, A. Inumella, R. Sangal and S. Paul. 2009. All Words Unsupervised Semantic Category Labeling for Hindi. To be appeared in *Recent*
- [19] S. M. Shieber. 1985. Evidence against the context-freeness of natural language. In *Linguistics and Philosophy*, p. 8, 334–343.