Simple Parser for Indian Languages in a Dependency Framework

Akshar Bharati, Mridul Gupta, Vineet Yadav, Karthik Gali and Dipti Misra Sharma
Language Technologies Research Center,
IIIT-Hyderabad, India
{mridulgupta,vineetyadav}@students.iiit.ac.in,
karthikg@research.iiit.ac.in,dipti@iiit.ac.in

Abstract

This paper is an attempt to show that an intermediary level of analysis is an effective way for carrying out various NLP tasks for linguistically similar languages. We describe a process for developing a simple parser for doing such tasks. This parser uses a grammar driven approach to annotate dependency relations (both inter and intra chunk) at an intermediary level. Ease in identifying a particular dependency relation dictates the degree of analysis reached by the parser. To establish efficiency of the simple parser we show the improvement in its results over previous grammar driven dependency parsing approaches for Indian languages like Hindi. We also propose the possibility of usefulness of the simple parser for Indian languages that are similar in nature.

1 Introduction and Related Work

Broad coverage parsing is a challenging task. For languages such as the Indian languages, it becomes all the more difficult as these languages are morphologically richer and the word order for these languages is relatively variable and less bound. Although dependency grammar driven parsing is much better suited for such type of languages (Hudson, 1984; Mel'Cuk, 1988), robust broad coverage parsing (Bharati et. al, 2008) still involves extensive analysis. Achieving good results in parsing for these languages may require large amount of linguistic resources such as annotated corpora, verb frames, lexicon etc. On the other hand, pure shallow parsing techniques (PVS and Gali, 2007) are not enough for providing sufficient information for applications such as machine translation, query answering etc.

It is here that the notion of a simple parser is born where the idea is to parse a sentence at a coarser level. One could go to a finer level of parse depending on the ease with which such a parse can be generated. The simple parser that we describe here is a grammar oriented model that makes use of linguistic features to identify relations. We have modeled the simple parser on the Paninian grammatical model (Begum et al., 2008; Bharati et al., 1995) which provides a dependency grammar framework. Paninian dependency grammar works well for analyzing Indian languages (Bharati et al., 1993). We have followed *karaka*[1] based approach for parsing.

An effort has been previously made in grammar driven parsing for Hindi by us (Gupta et al., 2008) where the focus was not to mark relations in a broad coverage sense but to mark certain easily identifiable relations using a rule base. In this paper, we show improvements in results over our previous work by including some additional linguistic features which help in identifying relations better. Our previous work focused only on inter-chunk annotation. In this paper, however, we have worked on both inter as well as intra chunk annotation. We later show their effectiveness and results at different levels of dependency annotation. We also propose how useful the simple parser is for Indian languages which are similar in nature.

2 Paninian Dependency Annotation Scheme at Various Levels

Paninian dependency scheme is based on a modifier-modified relationship (Bharati et al., 1995). The modified chunk (or group) is classified on the basis of its part of speech category. A hierarchy of dependency relations is thus established on the basis of this category. For example, all those relations whose parent (modified group) is a verb are classified under the verb modifier (vmod) category. Subsequent levels further classify these relations (or labels). Depth of a level in the hierarchy reflects the fineness of the dependency relations/labels. There are five labels at the

[1] The elements modifying the verb participate in the action specified by the verb. These participant relations with the verb are called *karakas*.

Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP 2009, pages 162–165,
Suntec, Singapore, 6-7 August 2009.

coarsest level namely, vmod, nmod (noun modifier), jjmod (adjective modifier), advmod (adverbial modifier) and ccof (conjunct of). Athough, ccof is not strictly a dependency relation (Begum et al., 2008). Figure 1 shows the hierarchy of relations used in the scheme.

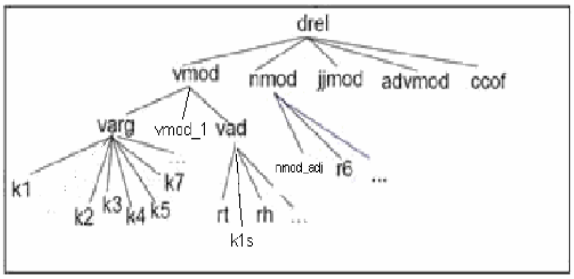

Figure 1: Hierarchy of Dependency Labels.

The next level comprises of varg (verb argument), vad (verb adjunct) and vmod_1[2] labels under vmod. Under the nmod label, nmod_adj (adjective), r6 (genitive) are classified. At the most fine grained level, varg and vad further branch out into labels like k1, k2, k3, k5, k7 and rh, rt, rd, k1s etc. The relations under varg are the six karakas that are the most essential participants in an action. All the other dependency labels[3] are non-karakas (for a more detailed explanation see Begum et al. (2008) and Bharati et al. (1995)).

Languages often have constructions that are ambiguous, owing to similar feature and context distribution. Thus, in such cases, it is appropriate to under-specify the relations (labels) or group some of them together. Also, some labels have very less frequency of occurrence in the corpus and it is thus appropriate to leave them out for marking by the simple parser. One can later, on the availability of more information, try to identify and mark such instances with appropriate labels.

The dependency tagset described in this section is used to mark inter-chunk relations. For marking relations between words within a chunk (intra-chunk), a similar tagset has been developed.

[2] vmod_1: A dependency relation in the vmod category, that exists between a non-finite verb and its parent verb. It has been under-specified for simplicity.

[3] A comprehensive list of the dependency tagset can be found at http://ltrc.iiit.ac.in/MachineTrans/research/tb/dep-tagset.pdf

3 Procedure

Our approach is corpus based where rules have been crafted after studying the corpus. We used the Hyderabad Dependency Treebank (HyDT) for development and testing our rules. The treebank consists of about 2100 sentences in Hindi, of which 1800 were part of the development set and 300 were used as test data. Each sentence is POS tagged and chunked (Bharati et al., 2006) in SSF format (Bharati et al., 2005).

3.1 Approach

The simple parser we propose here is a language independent engine that takes a rule file specific for a particular language (Gupta et. al, 2008). Indian languages are similar in various respects (Emeneau 1956; 1980). Hence, rules made for one language can be efficiently transferred for other similar languages. However, there can be cases where rules for one language may not work for another. These cases can be handled by adding some new rules for that particular language. The relative closeness among such languages, determines the efficiency of transference of rules from one language to another. We have taken Hindi and Punjabi, as example languages to support our proposal. 1(a) below is in Hindi,

1(a). *raama ko mithaaii acchii nahii*
„Ram - *dat*' „sweets' 'good' „not'
lagatii.
„appear'

"Ram does not like sweets."

Its corresponding Punjabi sentence,
1(b). *raama nuu mitthaai changii nii*
„Ram - *dat*' „sweets' „good' „not'
lagadii.
„appear'

"Ram does not like sweets."

Now, the rules for identifying k1[4] and k2 in Hindi are similar to that of Punjabi. For instance, in both the cases, the noun chunk possessing a nominative case marker (chunks take the properties of their heads) and the TAM (tense, aspect and modality of the main verb) should agree in

[4] 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. Likewise, k2 behaves like object/theme (Begum et al., 2008)

GNP for the noun to be a k2. It is easy to see how rules made for identifying certain relations in Hindi can be transferred to identify the same relations in Punjabi and similarly for other languages. However, not all rules can be transferred from one language to another.

3.2 Intra-chunk Relations

We also mark intra-chunk dependency relations. The procedure of marking intra-chunk labels is also rule based. Rules have been crafted using a common POS[5] tagset for Indian languages (Bharati et al., 2006). Rules can be applied to other languages. However, some rules may not work. In those cases we need to add some rules specific to the language. The rule format is a five-tuple containing the following fields,

1. Modified word
2. Modified constraints
3. Modifier word
4. Modifier constraints
5. Dependency relation

Rules for marking intra-chunk relations have been marked studying the POS tagged and chunked corpus. Commonly occurring linguistic patterns between two or more nodes are drawn out in the form of statistics and their figures are collected. Such patterns are then converted into robust rules.

4 Experiments and Results

We conducted experiments using the simple parser to establish its efficacy in identifying a particular set of relations explained in section 2. Experiments were conducted on gold standard test data derived from HyDT. The experiments were carried out on Hindi.

4.1 Marking Relations at Various Levels

We marked dependency labels at various levels described above using the proposed simple parser. The results are shown below We report two measures for evaluation, labeled (L) and labeled attachment (LA). Table 1 shows results for marking relations at the top most level (cf. Figure 1).

It should be noted that we have not marked relations like jjmod and advmod because the frequency of their occurrence in the treebank is quite low. The focus is only on those relations whose frequency of occurrence is above a bare minimum (>15). The frequency of labels like jjmod and advmod is not above that threshold value (Relations like k1 and k2 occur more than 1500 times in the treebank).

Relation	Precision		Recall	
	L	LA	L	LA
vmod	**93.7%**	**83.0%**	**76.1%**	67.4%
nmod	83.6%	79.1%	77.5%	**73.3%**
ccof	92.9%	82.9%	53.5%	50.4%
Total	91.8%	82.3%	72.9%	65.4%

Table 1. Figures for relations at the highest level.

Table 2 below depicts the figures obtained for the next level.

Relation	Precision		Recall	
	L	LA	L	LA
varg	77.7%	69.3%	77.9%	69.4%
vad	75.2%	66.6%	30.3%	26.9%
vmod_1	**89.6%**	75.8%	46.0%	38.9%
r6	83.2%	78.5%	**90.2%**	**85.2%**
nmod__adj	77.8%	**77.8%**	10.9%	10.9%
Total	79.1%	71.2%	64.6%	58.2%

Table 2. Figures for level 2.

In section 1, improvement in marking certain relations over our previous attempt (Gupta et. al, 2008) was mentioned. We provide a comparison of the results for the simple parser as opposed to the previous results. Figures shown in table 3 have been reproduced for comparing them against the results of the simple parser shown in this paper.

Relation	Precision		Recall	
	L	LA	L	LA
k1	66.0%	57.7%	65.1%	57.6%
k2	31.3%	28.3%	27.8%	25.1%
k7(p/t)	80.8%	77.2%	61.0%	58.4%
r6	82.1%	78.7%	89.6%	85.8%
nmod__adj	23.2%	21.9%	27.4%	25.8%

Table 3. Figures reproduced from our previous work.

Table 4 shows results of the simple parser. Note the improvement in precision values for all the relations.

Relation	Precision		Recall	
	L	LA	L	LA
k1	72.6%	68.0%	67.9%	63.5%
k2	61.6%	54.1%	29.9%	26.2%
k7(p/t)	84.6%	77.9%	73.5%	68.7%
r6	83.2%	78.6%	90.2%	85.5%
nmod__adj	77.8%	77.8%	10.9%	10.9%
pof	89.4%	87.7%	25.7%	25.2%

Table 4. Figures for simple parser.

[5] POS: Part of Speech

4.2 Intra-chunk Experiments

We also carried out some experiments to determine the efficiency of the simple parser with respect to annotating intra-chunk relations for Hindi. Results shown below were obtained after testing the simple parser using gold standard test data of about 200 sentences. Table 5 shows figures for labeled accuracy as well as labeled attachment accuracy.

Relation	Precision		Recall	
	L	LA	L	LA
nmod	100%	89.3%	70.0%	62.5%
nmod__adj	100%	92.7%	85.2%	79.0%
nmod__dem	100%	100%	100%	100%
nmod__qf	97.0%	92.4%	80.0%	76.2%
pof	84.5%	82.1%	94.5%	92.0%
ccof	91.8%	80.0%	70.9%	62.0%
jjmod__intf	100%	100%	100%	100%
Total	96.2%	90.4%	82.6%	77.7%

Table 5. Figures for intra-chunk annotation.

5 Conclusion

We introduced the notion of a simple parser for Indian languages which follows a grammar driven methodology. We compared its performance against previous similar attempts and reported its efficiency. We showed how by using simple yet robust rules one can achieve high performance in the identification of various levels of dependency relations.

The immediate tasks for the near future would be to identify relative clauses in order to reduce labeled attachment errors and hence to come up with rules for better identification of clauses. We also intend to thoroughly test our rules for Indian languages that are similar in nature and hence evaluate the efficiency of the simple parser.

Acknowledgements

We sincerely thank Samar Husain, for his important role in providing us with valuable linguistic inputs and ideas. The treebank (Hyderabad dependency treebank, version 0.05) used, was prepared at LTRC, IIIT-Hyderabad.

References

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for Indian languages. In *Proceedings of IJCNLP-2008*.

Akshar Bharati, Vineet Chaitanya and Rajeev Sangal. 1995. *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, New Delhi, pp. 65-106.

Akshar Bharati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal. 2008. A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In *Proc. of the COLIPS International Conference on Asian Language Processing 2008 (IALP)*. Chiang Mai, Thailand. 2008.

Akshar Bharati and Rajeev Sangal. 1993. Parsing Free Word Order Languages in the Paninian Framework, *ACL93: Proc. of Annual Meeting of Association for Computational Linguistics*.

Akshar Bharati, Rajeev Sangal and Dipti M. Sharma. 2005. ShaktiAnalyser: SSF Representation.

Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma and Lakshmi Bai. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. *Technical Report (TR-LTRC-31), Language Technologies Research Centre IIIT, Hyderabad http://ltrc.iiit.ac.in/MachineTrans/publications/technicalReports/tr031/posguidelines.pdf*

Murray B. Emeneau. 1956. India as a linguistic area. *Linguistics,* 32:3-16.

Murray B. Emeneau. 1980. *Language and linguistic area. Essays by Murray B. Emeneau. Selected and introduced by Anwar S. Dil.* Stanford University Press.

Mridul Gupta, Vineet Yadav, Samar Husain and Dipti M. Sharma. 2008. A Rule Based Approach for Automatic Annotation of a Hindi Treebank. In *Proc. Of the 6th International Conference on Natural Language Processing (ICON-08),* CDAC Pune, India.

R. Hudson. 1984. *Word Grammar*, Basil Blackwell, Oxford, OX4 1JF, England.

I. Mel'cuk . 1988. *Dependency Syntax: Theory and Practice*, State University, Press of New York.

Avinesh PVS and Karthik Gali. 2007. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. In *Proc. Of IJCAI-07 Workshop on "Shallow Parsing in South Asian Languages", 2007.*