International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

National Conference on Recent Trends in Computer Science and Technology (NCRTCST)-2013

Improving the Usability of Statistical Parsers by Incorporating Linguistic Constraints

B.Venkata Seshu Kumari
Asst. Prof., Deptt. of CSE
St. Peter's Engineering College, Hyderabad, India
Email: hemakrishna.ambati@gmail.com

R. Rajeswara Rao
Assoc. Prof, Deptt. of CSE
JNTU, Vijayanagaram, India
Email: rajaraob4u@yahoo.com

***Abstract* – Statistical systems with high accuracy are very useful in real-world applications. If these systems can capture basic linguistic information, then the usefulness of these statistical systems improve a lot. This paper is an attempt at incorporating linguistic constraints in statistical dependency parsing. We consider a simple linguistic constraint that a verb should not have multiple subjects/objects as its children in the dependency tree. We first describe the importance of this constraint considering Machine Translation systems which use dependency parser output, as an example application. We then show how the current state-of-the-art dependency parsers violate this constraint. We present two new methods to handle this constraint. We evaluate our methods on the state-of-the-art dependency parsers for Hindi.**

***Keywords* – Dependency Parsing, Indian Languages, Statistical Parsing, Linguistic Constraints.**

I. Introduction

Parsing is one of the major tasks which helps in understanding the natural language. It is useful in several natural language applications. Machine translation, anaphora resolution, word sense disambiguation, question answering, summarization are few of them. This led to the development of grammar-driven, data-driven and hybrid parsers. Due to the availability of annotated corpora in recent years, data driven parsing has achieved considerable success. The availability of phrase structure treebank for English [12] has seen the development of many efficient parsers. Using the dependency analysis, a similar large scale annotation effort for Czech, has been the Prague Dependency Treebank. Unlike English, Czech is a free-word-order language and is also morphologically very rich. It has been suggested that free-word-order languages can be handled better using the dependency based framework than the constituency based one [3, 8, 21]. It has also been noted that use of appropriate edge labels gives a level of semantics. It is perhaps due to these reasons that the recent past has seen a surge in the development of dependency based treebanks.

Due to the availability of dependency treebanks, there are several recent attempts at building dependency parsers. Two CoNLL shared tasks [6, 18] were held aiming at building state-of-theart dependency parsers for different languages. Recently in NLP Tools Contest in ICON-2009 [9], rule-based, constraint based, statistical and hybrid approaches were explored towards building dependency parsers for three Indian languages namely, Telugu, Hindi and Bangla. In all these efforts, state-of-the-art accuracies are obtained by Malt [19]. The major limitation of Malt is that it won't take linguistic constraints into account explicitly. But, in real-world applications of the parsers, some basic linguistic constraints are very useful. If we can make this parser handle linguistic constraints also, then it becomes very useful in real-world applications.

This paper is an effort towards incorporating linguistic constraints in statistical dependency parser. We consider a simple constraint that a verb should not have multiple subjects/objects as its children. In section 2, we take machine translation using dependency parser as an example and explain the need of this linguistic constraint. In section 3, we propose two approaches to handle this case. We evaluate our approaches on the state-of-the-art dependency parser for Hindi and analyze the results in section 4. General discussion and future directions of the work are presented in section 5. We conclude our paper in section 6.

II. Motivation and Related Work

In this section we take Machine Translation (MT) systems that use dependency parser output as an example and explain the need of linguistic constraints. We take a simple constraint that a verb should not have multiple subjects/objects as its children in the dependency tree. Indian Language to Indian Language Machine Transtion System[1] is one such MT system which uses dependency parser output. In this system the general framework has three major components. a) dependency analysis of the source sentence. b) transfer from source dependency tree to target dependency tree, and c) sentence generation from the target dependency tree. In the transfer part several rules are framed based on the source language dependency tree. For instance, for Telugu to Hindi MT system, based on the dependency labels of the Telugu sentence postpositions markers that need to be added to the words are decided. Consider the following example,

(1) Telugu:	raamu	oka	pamdu	tinnaadu	
	'Ramu'	'one'	'fruit'	'ate'	
Hindi:	raamu	ne	eka	phala	khaayaa
	'Ramu'	'ERG'	'one'	'fruit'	'ate'

English: Ramu ate a fruit.

[1] http://sampark.iiit.ac.in/

In the above Telugu sentence, ‘raamu’ is the subject of the verb ‘tinnaadu’. While translating this sentence to Hindi, the post-position marker ‘ne’ is added to the subject node. If the dependency parser marks two subjects, both the words will have ‘ne’ marker. This affects the comprehensibility. If we can avoid such instances, then the output of the MT system will be improved.

This problem is not due to morphological richness or free-word-order nature of the target language. Consider an example of free-word-order language to fixed-word-order language MT system like Hindi to English MT system. The dependency labels help in identifying the position of the word in the target sentence. Consider the example sentences given below,

(2a) raama seba khaatha hai
‘Ram’ ‘apple’ ‘eats’ ‘is’
‘Ram eats an apple’
(2b) seba raama khaatha hai
‘apple’ ‘Ram’ ‘eats’ ‘is’
‘Ram eats an apple’

Though the source sentence is different, the target sentence is same. Even though the source sentences are different, the dependency tree is same for both the sentences. In both the cases, ‘raama’ is the subject and ‘seba’ is the direct object of the verb ‘khaatha’. This information helps in getting the correct translation. If the parser for the source sentence assigns subject label to both ‘raama’ and ‘seba’, the MT system cannot give the correct output.

There were some attempts at handling these kind of linguistic constraints using integer programming approaches [4, 20]. In these approaches dependency parsing is formulated as solving an integer program as [13] has formulated dependency parsing as Spanning Tree problem. All the linguistic constraints are encoded as constraints while solving the integer program. In other words, all the parses that violate these constraints are removed from the solution list. The parse which satisfies all the constraints is considered as the dependency tree for the sentence. In the following section, we describe two new approaches to avoid multiple subjects or direct objects for a verb.

III. Approaches

In this section, we describe the two different approaches for avoiding the cases of a verb having multiple subjects or direct objects as its children in the dependency tree.

A. *Position Based Approach (PBA)*

In this approach we first run a parser on the input sentence. Instead of first best dependency label, we extract the k-best labels for each token in the sentence. For each verb in the sentence, we check if there are multiple children with the dependency label ‘subject’. If there are any such cases, we extract the list of all the children with label ‘subject’. we find the node in this list which appears left most in the sentence with respect to other nodes. We assign ‘subject’ to this node. For the rest of the nodes in this list we assign the second best label and remove the first best label from their respective k-best list of labels. We check recursively, till all such instances are avoided. We repeat the same procedure for ‘direct object’.

Main criterion to avoid multiple subjects or direct objects in this approach is position of the node in the sentence. Consider the example sentence of (2a), Suppose the parser assigns the subject label to both the nouns, ‘raama’ and ‘seba’. Then position based approach assigns the subject label to ‘raama’ and second best label to ‘seba’ as ‘raama’ precedes ‘seba’. In this manner we can avoid a verb having multiple children with dependency labels subject or direct object. Limitation to this approach is word-order. The algorithm described here works well for fixed word order languages. For example, consider a language with fixed word order like English. English is a SVO (Subject, Verb, Object) language. Subject always occurs before the object. So, if a verb has multiple subjects, based on position we can say that the node that occurs first will be the subject. But if we consider a free-word order language like Hindi, this approach wouldn’t work always. Consider (2a) and (2b). In both these examples, ‘raama’ is the subject of the verb ‘khaatha’ and ‘seba’ is the direct object of the verb ‘khaatha’. The only difference in these two sentences is the word order. In (2a), subject precedes direct object. Whereas in (2b), direct object precedes subject. Suppose the parser identifies both ‘raama’ and ‘seba’ as subjects. Position based approach can correctly identify ‘raama’ as the subject in case of (2a). But in case of (2b), ‘seba’ is identified as the subject. To handle these kind of instances, we proposed a score based approach.

B. *Score Based Approach (SBA)*

The score based approach is similar to position based approach except that the main criterion to avoid multiple subjects or direct objects in this approach is score of the node having a particular label. Whereas in position based approach, position of the node is the main criterion to avoid multiple subjects or direct objects. In this approach, for each node in the sentence, we extract the k-best labels along with their scores. Similar to PBA, we first check for each verb if there are multiple children with the dependency label ‘subject’. If there are any such cases, we extract the list of all the children with label ‘subject’. We find the node in this list which has the highest score. We assign ‘subject’ to this node. For the rest of the nodes in this list we assign the second best label and remove the first best label from their respective k-best list of labels. We check recursively, till all such instances are avoided. We repeat the same procedure for ‘direct object’.

Consider (2a) and (2b). Suppose the parser identifies both ‘raama’ and ‘seba’ as subjects. Score of ‘raama’ being a subject will be more than ‘seba’ being a subject. So, the probabilistic approach correctly marks ‘raama’ as subject in both (2a) and (2b). But, PBA couldn’t identify ‘raama’ as subject in (2b). Figure 3, sketches the steps involved in both Position Based Approach and Score Based Approach.

IV. Experiments and Analysis

We evaluate our approaches on the state-of-the-art parser for Hindi. First we calculate the instances of multiple subjects/objects in the output of the state-of-the-art parser and then we apply our approaches and analyze the results.

Recently in NLP Tools Contest in ICON-2009 [9 and references herein], rule-based, constraint based, statistical and hybrid approaches were explored for parsing Hindi. All these attempts were at finding the inter-chunk dependency relations, given gold-standard POS and chunk tags. The state-of-the-art accuracy of 74.48% LAS (Labeled Attachment Score) is achieved by [1] for Hindi using Malt parser [19]. They used two well-known data-driven parsers, Malt [19] and MST [13] for their experiments.

For Hindi, data was annotated using the Computational Paninian Grammar [3]. The annotation scheme based on this grammar has been described in [2] and [5]. Subject and direct object equivalent dependencies in this framework are kartha karaka (k1) and karma karaka (k2). We replicated the experiments of [1] on test set (150 sentences) of Hindi and analyzed the output of Malt. We consider this as the baseline. In the output of Malt, there are 39 instances of multiple subjects/objects.

In case of Malt, we modified the implementation to extract all the possible dependency labels with their scores. As Malt uses libsvm for learning, we couldn't able to get the probabilities. Though interpreting the scores provided by libsvm as probabilities is not the correct way, that is the only option currently available with Malt. We applied both the PBA and SBA approaches to avoid multiple subjects/objects. We evaluated our experiments based on unlabeled attachment score (UAS), labeled attachment score (LAS) and labeled score (LS) [18]. Results are presented in Table 1. As Hindi is a free-word-order language, as expected, SBA performs better than PBA. With SBA we got an improvement of 0.26% in LAS over the previous best results for Malt.

Table 1: Comparison of Malt, PBA and SBA for Hindi

Approach	***Hindi***		
	UAS	***LAS***	***LS***
Malt	90.14	74.48	76.38
PBA	90.14	74.57	76.38
SBA	90.14	74.74	76.56

V. Discussion and Future Work

Our results show that the score based approach performs consistently better than the position based approach. Output after SBA is useful in applications like machine translation. Thus, using our approach, we can build a parser which is useful in real-world applications without compromising accuracy.

We plan to evaluate our approaches on all the data-sets of CoNLL-X and CoNLL-2007 shared tasks. Currently, we are handling only two labels, subject and direct object. Apart from subject and direct object there can be other labels for which multiple instances for a single verb is not valid. We can extend our approaches to handle such labels also. We tried to incorporate one simple linguistic constraint in the statistical dependency parsers. We can also explore the ways of incorporating other useful linguistic constraints.

VI. Conclusion

Statistical systems with high accuracy are very useful in practical real-world applications. If these systems can capture basic linguistic information, then the usefulness of the statistical systems improve a lot. In paper, we presented a new method of incorporating linguistic constraints into the statistical dependency parsers. We took a simple constraint that a verb should not have multiple subjects or direct objects as its children. We proposed two approaches, one based on position and the other based on scores to handle this. We evaluated our approaches on state-of-the-art dependency parser for Hindi. Our results show that using score based approach, we can build a statistical parser which handles linguistic constraints and thus useful in real-world applications without compromising accuracy.

Acknowledgment

We would like to thank Mr. Bharat and Language Technologies Research Institute (LTRC), International Institute of Information Technology, Hyderabad (IIIT-H) for providing the data and previous best settings for the parser.

References

[1] B. R. Ambati, P. Gadde and K. Jindal. 2009. Experiments in Indian Language Dependency Parsing. In Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing, pp 32-37.

[2] R. Begum, S. Husain, A. Dhwaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In Proceedings of IJCNLP-2008.

[3] A. Bharati, V. Chaitanya and R. Sangal. 1995. Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi, pp. 65-106.

[4] A. Bharati, S. Husain, D. M. Sharma, and R. Sangal. 2008. A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In Proceedings of the COLIPS International Conference on Asian Language Processing 2008 (IALP). Chiang Mai, Thailand.

[5] A. Bharati, D. M. Sharma, S. Husain, L. Bai, R. Begum and R. Sangal. 2009. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank (version 2.0).

http://ltrc.iiit.ac.in/MachineTrans/research/tb/DS-guidelines/DS-guidelinesver2-28-05-09.pdf.

[6] S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In Proc. of the Tenth Conf. on Computational Natural Language Learning (CoNLL).

[7] E. Hajicova. 1998. Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In Proc. TSD'98.

[8] R. Hudson. 1984. Word Grammar, Basil Blackwell, 108 Cowley Rd, Oxford, OX4 1JF, England.

[9] S. Husain. 2009. Dependency Parsers for Indian Languages. In ICON09 NLP Tools Contest: Indian Language Dependency Parsing. Hyderabad, India.

[10] S. Husain, P. Mannem, B. Ambati and P. Gadde. 2010. The ICON-2010 Tools Contest on Indian Language Dependency Parsing. In ICON-2010 Tools Contest on Indian Language Dependency Parsing. Kharagpur, India.

[11] P. Kosaraju, S. R. Kesidi, V. B. R. Ainavolu, and P. Kukkadapu. 2010. Experiments on Indian Language Dependency Parsing. In ICON-2010 tools contest on Indian language dependency parsing. Kharagpur, India.

[12] M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank, Computational Linguistics

[13] R. McDonald, K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), pp. 216–220.

[14] R. McDonald and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In Proc. of EMNLP-CoNLL.

[15] J. Nivre. 2003. An efficient algorithm for projective dependency parsing. In Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), pages 149–160.

[16] J. Nivre, and J. Nilsson. 2005. Pseudo-projective dependency parsing. In ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics,pages 99–106, Ann Arbor, Michigan.

[17] Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing. Computational Linguistics, 34(4):513–553.

[18] J. Nivre, J. Hall, S. Kubler, R. McDonald, J. Nilsson, S. Riedel and D. Yuret. 2007a. The CoNLL 2007 Shared Task on Dependency Parsing. In Proceedings of EMNLP/CoNLL-2007.

[19] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, 13(2), 95-135.

[20] S. Riedel, R. Çakıcı, and I. Meza-Ruiz. 2006. Multi-lingual Dependency Parsing with Incremental Integer Linear Programming. In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X).

[21] S. M. Shieber. 1985. Evidence against the contextfreeness of natural language. In Linguistics and Philosophy, p. 8, 334–343.