

# Importance of linguistic constraints in statistical dependency parsing

**Bharat Ram Ambati**

Language Technologies Research Centre, IIT-Hyderabad,  
Gachibowli, Hyderabad, India – 500032.

ambati@research.iit.ac.in

## 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 and Czech.

## 1 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 (Marcus et al., 1993) 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 (Hajicova, 1998). 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 (Hudson, 1984; Shieber, 1985; Mel'čuk, 1988, Bharati et al., 1995). The basic

difference between a constituent based representation and a dependency representation is the lack of nonterminal nodes in the latter. 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 (Buchholz and Marsi, 2006; Nivre et al., 2007a) were held aiming at building state-of-the-art dependency parsers for different languages. Recently in NLP Tools Contest in ICON-2009 (Husain, 2009 and references therein), 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 two data-driven parsers, namely, Malt (Nivre et al., 2007b) and MST (McDonald et al., 2006). The major limitation of both these parsers is that they 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 these parsers handle linguistic constraints also, then they become 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 parsers for Hindi and Czech 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.

## 2 Motivation

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 Translation System<sup>1</sup> 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 post-positions 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".

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. 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 object of the verb 'khaatha'. This information helps in getting the correct translation. If the parser for the source sentence assigns the label 'subject' to both 'raama' and 'seba', the MT system can not give the correct output.

There were some attempts at handling these kind of linguistic constraints using integer programming approaches (Riedel et al., 2006; Bharati et al., 2008). In these approaches dependency parsing is formulated as solving an integer program as McDonald et al. (2006) has formulated dependency parsing as MST 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/objects for a verb.

## 3 Approaches

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

### 3.1 Naive Approach (NA)

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

<sup>1</sup> <http://sampark.iiit.ac.in/>

avoided. We repeat the same procedure for ‘object’.

Main criterion to avoid multiple subjects/objects in this approach is position of the node in the sentence. Consider the following example,

Eg. 3: *raama seba khaatha hai*  
‘Ram’ ‘apple’ ‘eats’ ‘is’  
‘Ram eats an apple’

Suppose the parser assigns the label ‘subject’ to both the nouns, ‘*raama*’ and ‘*seba*’. Then naive approach assigns the label subject 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/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 object of the verb ‘*khaatha*’. The only difference in these two sentences is the order of the word. In (2a), subject precedes object. Whereas in (2b), object precedes subject. Suppose the parser identifies both ‘*raama*’ and ‘*seba*’ as subjects. NA 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 use a probabilistic approach.

### 3.2 Probabilistic Approach (PA)

The probabilistic approach is similar to naive approach except that the main criterion to avoid multiple subjects/objects in this approach is probability of the node having a particular label. Whereas in naive approach, position of the node is the main criterion to avoid multiple subjects/objects. In this approach, for each node in the sentence, we extract the k-best labels along with their probabilities. Similar to NA, 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 probability value. 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 ‘object’.

Consider (2a) and (2b). Suppose the parser identifies both ‘*raama*’ and ‘*seba*’ as subjects. Probability 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, NA couldn’t identify ‘*raama*’ as subject in (2b).

## 4 Experiments

We evaluate our approaches on the state-of-the-art parsers for two languages namely, Hindi and Czech. First we calculate the instances of multiple subjects/objects in the output of the state-of-the-art parsers for these two languages. Then we apply our approaches and analyze the results.

### 4.1 Hindi

Recently in NLP Tools Contest in ICON-2009 (Husain, 2009 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 Ambati et al. (2009) for Hindi. They used two well-known data-driven parsers, Malt<sup>2</sup> (Nivre et al., 2007b), and MST<sup>3</sup> (McDonald et al., 2006) for their experiments. As the accuracy of the labeler of MST parser is very low, they used maximum entropy classification algorithm, MAXENT<sup>4</sup> for labeling.

For Hindi, dependency annotation is done using paninian framework (Begum et al., 2008; Bharati et al., 1995). So, in Hindi, the equivalent labels for subject and object are ‘*karta (k1)*’ and ‘*karma (k2)*’. ‘*karta*’ and ‘*karma*’ are syntactico-semantic labels which have some properties of both grammatical roles and thematic roles. *k1* behaves similar to subject and agent. *k2* behaves similar to object and patient (Bharati et al., 1995; Bharati et al., 2009). Here, by object we mean

<sup>2</sup> Malt Version 1.3.1

<sup>3</sup> MST Version 0.4b

<sup>4</sup> [http://homepages.inf.ed.ac.uk/lzhang10/maxent\\_toolkit.htm](http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.htm)

only direct object. Thus we consider only *k1* and *k2* labels which are equivalent of subject and direct object. Annotation scheme is such that there wouldn't be multiple subjects/objects for a verb in any case (Bharati et al., 2009). For example, even in case of coordination, coordinating conjunction is the head and conjuncts are children of the coordinating conjunction. The coordinating conjunction is attached to the verb with *k1/k2* label and the conjuncts get attached to the coordinating conjunction with a dependency label '*ccof*'.

We replicated the experiments of Ambati et al. (2009) on test set (150 sentences) of Hindi and analyzed the outputs of Malt and MST+MaxEnt. We consider this as the baseline. In the output of Malt, there are 39 instances of multiple subjects/objects. There are 51 such instances in the output of MST+MAXENT.

Malt is good at short distance labeling and MST is good at long distance labeling (McDonald and Nivre, 2007). As '*k1*' and '*k2*' are short distance labels, Malt could able predict these labels more accurately than MST. Because of this output of MST has higher number of instances of multiple subjects/objects than Malt.

	<i>Total Instances</i>
<i>Malt</i>	<b>39</b>
<i>MST + MAXENT</i>	<b>51</b>

Table 1: Number of instances of multiple subjects or objects in the output of the state-of-the-art parsers for Hindi

Both the parsers output first best label for each node in the sentence. 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. In case of MST+MAXENT, labeling is performed by MAXENT. We used a java version of MAXENT<sup>5</sup> to extract all possible tags with their scores. We applied both the naive and probabilistic approaches to avoid multiple subjects/objects. We evaluated our experiments based on unlabeled attachment score (UAS), labeled attachment score (LAS) and labeled score

(LS) (Nivre et al., 2007a). Results are presented in Table 2.

As expected, PA performs better than NA. With PA we got an improvement of 0.26% in LAS over the previous best results for Malt. In case of MST+MAXENT we got an improvement of 0.61% in LAS over the previous best results. Note that in case of MST+MAXENT, the slight difference between state-of-the-art results of Ambati et al. (2009) and our baseline accuracy is due different MAXENT package used.

	<i>Malt</i>			<i>MST+MAXENT</i>		
	<i>UAS</i>	<i>LAS</i>	<i>LS</i>	<i>UAS</i>	<i>LAS</i>	<i>LS</i>
<i>Baseline</i>	90.14	74.48	76.38	91.26	72.75	75.26
<i>NA</i>	90.14	74.57	76.38	91.26	72.84	75.26
<i>PA</i>	<b>90.14</b>	<b>74.74</b>	<b>76.56</b>	<b>91.26</b>	<b>73.36</b>	<b>75.87</b>

Table 2: Comparison of NA and PA with previous best results for Hindi

Improvement in case of MST+MAXENT is greater than that of Malt. One reason is because of more number of instances of multiple subjects/objects in case of MST+MAXENT. Other reason is use of probabilities in case MST+MAXENT. Whereas in case of Malt, we interpreted the scores as probabilities which is not a good way to do. But, in case of Malt, that is the only option available.

## 4.2 Czech

In case of Czech, we replicated the experiments of Hall et al. (2007) using latest version of Malt (version 1.3.1) and analyzed the output. We consider this as the baseline. The minor variation of the baseline results from the results of CoNLL-2007 shared task is due to different version Malt parser being used. Due to practical reasons we couldn't use the older version. In the output of Malt, there are 39 instances of multiple subjects/objects out of 286 sentences in the testing data. In case of Czech, the equivalent labels for subject and object are 'agent' and 'theme'.

Czech is a free-word-order language similar to Hindi. So as expected, PA performed better than NA. Interestingly, accuracy of PA is lower than the baseline. Main reason for this is scores of libsvm of Malt. We explain the reason for this using the following example, consider a verb 'V' has two children 'C1' and 'C2' with dependency label subject. Assume that the label for 'C1' is subject and the label of 'C2' is object in the gold-data. As the parser marked 'C1' with subject, this

<sup>5</sup> <http://maxent.sourceforge.net/>

adds to the accuracy of the parser. While avoiding multiple subjects, if ‘C1’ is marked as subject, then the accuracy doesn’t drop. If ‘C2’ is marked as object then the accuracy increases. But, if ‘C2’ is marked as subject and ‘C1’ is marked as object then the accuracy drops. This could happen if probability of ‘C1’ having subject as label is lower than ‘C1’ having subject as the label. This is because of two reasons, (a) parser itself wrongly predicted the probabilities, and (b) parser predicted correctly, but due to the limitation of libsvm, we couldn’t get the scores correctly.

	<i>UAS</i>	<i>LAS</i>	<i>LS</i>
<i>Baseline</i>	<b>82.92</b>	<b>76.32</b>	<b>83.69</b>
<i>NA</i>	82.92	75.92	83.35
<i>PA</i>	82.92	75.97	83.40

Table 3: Comparison of NA and PA with previous best results for Czech

## 5 Discussion and Future Work

Results show that the probabilistic approach performs consistently better than the naive approach. For Hindi, we could able to achieve an improvement 0.26% and 0.61% in *LAS* over the previous best results using Malt and MST respectively. We couldn’t able to achieve any improvement in case of Czech due to the limitation of libsvm learner used in Malt.

We plan to evaluate our approaches on all the data-sets of CoNLL-X and CoNLL-2007 shared tasks using Malt. Settings of MST parser are available only for CoNLL-X shared task data sets. So, we plan to evaluate our approaches on CoNLL-X shared task data using MST also. Malt has the limitation for extracting probabilities due to libsvm learner. Latest version of Malt (version 1.3.1) provides option for liblinear learner also. Liblinear provides option for extracting probabilities. So we can also use liblinear learning algorithm for Malt and explore the usefulness of our approaches. Currently, we are handling only two labels, subject and object. Apart from subject and 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.

## 6 Conclusion

Statistical systems with high accuracy are very useful in practical applications. If these systems can capture basic linguistic information, then the usefulness of the statistical system improves a lot. In this 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/objects as its children. We proposed two approaches, one based on position and the other based on probabilities to handle this. We evaluated our approaches on state-of-the-art dependency parsers for Hindi and Czech.

## Acknowledgments

I would like to express my gratitude to Prof. Joakim Nivre and Prof. Rajeev Sangal for their guidance and support. I would also like to thank Mr. Samar Husain for his valuable suggestions.

## References

- 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.
- R. Begum, S. Husain, A. Dhvaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In *Proceedings of IJCNLP-2008*.
- A. Bharati, V. Chaitanya and R. Sangal. 1995. *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, New Delhi, pp. 65-106.
- 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.
- 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)*.
- E. Hajicova. 1998. Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In *Proc. TSD ’98*.
- 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*.
- R. Hudson. 1984. *Word Grammar*, Basil Blackwell, 108 Cowley Rd, Oxford, OX4 1JF, England.

- S. Husain. 2009. Dependency Parsers for Indian Languages. In *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*. Hyderabad, India.
- M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank, *Computational Linguistics* 1993.
- I. A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*, State University, Press of New York.
- 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.
- R. McDonald and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proc. of EMNLP-CoNLL*.
- 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*.
- 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.
- S. Riedel, Ruket Çakıcı and Ivan 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)*.
- S. M. Shieber. 1985. Evidence against the context-freeness of natural language. In *Linguistics and Philosophy*, p. 8, 334–343.