Accurate Identification of the Karta (Subject) Relation in Bangla

Arnab Dhar
Computer Science and Engineering
Indian Institute of Technology
Kharagpur, WB, India
arnab832007@gmail.com

Sudeshna Sarkar
Computer Science and Engineering
Indian Institute of Technology
Kharagpur, WB, India
shudeshna@gmail.com

Abstract

This paper presents an accurate identification of different types of *karta (subject)* in Bangla. Due to the limited amount of annotated data of dependency relations, we have built a baseline parser for Bangla using data driven method. Then a rule based post processor is applied on the output of baseline parser. As a result, average labeled attachment score improvement of *karta (subject)* based on F-measure on KGPBenTreeBank and ICON 2010 Treebank are 25.35% and 9.53%, respectively.

1 Introduction

Machine translation, anaphora resolution, question answering, etc., are the major application areas under natural language processing. While translating a source language to target language, the dependency structure of a sentence of source language plays a key role. Dependency grammar is a form of syntactic representation, where the syntactic structure consists of lexical elements linked by binary dependency relations. Dependency parsing involves syntactic analysis based on dependency representation (Nivre, 2005). The dependency structure is more suitable for handling highly inflected languages and the languages where word order is not very rigid.

The objective of our work is to build a high accuracy dependency parser for Bangla to facilitate Bangla to Hindi Machine Translation (BHMT) system. We have built a baseline dependency parser for Bangla using data driven method which implements inductive dependency parsing using the framework of MaltParser (Nivre et al., 2006; Nivre et al., 2007), in which we adapted the parameters and features for Bangla sentence parsing. We have analyzed different types of errors in the output of this baseline parser. We note that the correct identification of *karta (subject)* is a very important task for good quality BHMT system. We have analyzed different types of errors of *karta (subject)* and proposed some methods to rectify those errors by post processing the output of the baseline parser.

The rest of the paper is organized as follows. Section 2 describes the previous work related to dependency parsing. Section 3 describes the motivation and objective of our work. Section 4 describes the development of dependency parser for Bangla using data driven method. Section 5 presents rule based post processing. Section 6 presents the conclusion and the future directions of this research.

2 Literature Survey

Dependency parsing approaches can be broadly classified into three categories, namely, grammar driven, data driven and hybrid approaches. Grammar driven parsers have been developed based on context free dependency grammar (Hays, 1964) and constraint dependency grammar (Maruyama, 1990). Graph-based (McDonald et al., 2005) and transition-based parsing (Nivre et al., 2007) are some methods of data driven parsing. Marneffe et al. (2006) has proposed a system[1] which extracts dependency parses

[1] http://nlp.stanford.edu:8080/parser/

from phrase structure parses of English sentences.

We now discuss work on parsing of Indian languages. Bharati et al. (1993) has described a constraint based Hindi parser by applying the Paninian framework (Bharati et al., 1995). Bharati et al. (2002) have also used the computational Paninian framework for parsing Hindi sentences (Bharati and Sangal, 1993) without using lakshan charts (discrimination nets) for nouns and verbs. Bharati et al. (2009) has described a two stage constraint based approach for parsing Hindi sentences. Dhar et al. (2012) has described a two-stage approach for parsing Bangla sentences.

The Tool Contests of ICON 2009 (Husain, 2009) and ICON 2010 (Husain et al., 2010) released three Indian language Treebanks for Hindi, Bengali and Telugu. The system of De et al. (2009b) had the best performance for Bangla. They used a grammar driven approach for parsing. They have used 500 demand frames in Bangla (De et al., 2009a) for parsing. A hybrid approach has been suggested by Chatterji et al. (2009) and Ghosh et al. (2010) where data driven parser used as a baseline system and followed by a rule based post processor. We have also followed a hybrid approach (Chatterji et al., 2009; Ghosh et al., 2010; Dhar et al., 2012), but in rule based post processing we mainly focus on correct identification of different types of *karta (subject)* in a sentence. Kolachina et al. (2010) and Kosaraju et al. (2010) have built a dependency parser for Indian languages using data driven method. For this, they have used the framework of MaltParser.

3 Motivation and Objective

The objective of our work is to build a high accuracy dependency parser for Bangla to facilitate BHMT system. The Bangla verb form does not depend on gender and number of *karta (subject)* of a sentence, but sometimes in Hindi the verb form depends on gender, number and person of the *karta (subject)* of a sentence. Bangla *karta (subject)* takes different types of vibhaktis (suffixes) such as কে (ke), র (ra), 0 (shUNya) [zero]. Identifying *karta (subject)* is a non-trivial task.

There is no one to one correspondence between Bangla sentence and its corresponding Hindi translation. Sometimes in Hindi, the *karta (subject)* is followed by post position markers but it is absent in corresponding Bangla sentence. For example, in Hindi, when the transitive verb is in the past tense a post position marker ने (ne) is added to the *karta (subject)*. So, correct identification of *karta (subject)* is very useful for good quality BHMT system.

4 Our Approach

In this section, we describe the development of our basic data driven parser for Bangla.

4.1 Development of Dependency Parser

In our work, we have developed a Bangla dependency parser using the framework of MaltParser. We follow the MaltParser settings for Bangla used in Kosaraju et al. (2010). They used Covington's algorithm and run this algorithm in a non-projective mode which allows crossing edges in dependency structure.

4.2 Feature Description for Data Driven Parser

Features are very important element of statistical modeling of data. We follow the basic features used in Kolachina et al. (2010), and we added two additional features, namely, named entity (NE) tag and semantic class (SC). Named entity tag indicates the class in which a proper noun belongs. Class here refers to person name, locations, organizations, times etc. Named entity tag also helps to identify that a proper noun is animate if that proper noun belongs to person name class. Semantic class contains semantic property of each word (mainly noun) i.e. it is either animate or inanimate. Sometimes, vibhakti (suffix) information fails to resolve the ambiguity in identifying *karta (subject)* and *karma (object)* in a sentence when both have same vibhakti (suffix). In this case, semantic property of the words help to resolve the ambiguity. Basic features refer to root (LEMMA), word (FORM), part-of-speech (POSTAG), chunk (CPOSTAG) and morphological (MORPH) features. Morphological features include lexical category, number, person, case and vibhakti (suffix). We have

used some tools and resources, namely, Tokenizer, Morph Analyzer, Chunker, Head Computation, Named Entity Recognizer, Clause Boundary Identifier and Dictionary resource, for extracting the above features of word token in training and test data. We have used varying window length of basic features on LEFT, RIGHT, LEFTCONTEXT and RIGHTCONTEXT data structures which are used in MaltParser. The feature template for Bangla which is used in our experiments, is shown below.

1. A set of FORM features over LEFT and RIGHT of length 2.
2. A set of FORM features of dependent word and head word over LEFT and RIGHT of length 1.
3. A set of LEMMA features over LEFT and RIGHT of length 2.
4. A set of POSTAG features over LEFT and RIGHT of length 4.
5. A set of POSTAG features of dependent word and head word over LEFT and RIGHT of length 1.
6. A set of POSTAG features over LEFTCONTEXT and RIGHTCONTEXT of length 1.
7. A set of CPOSTAG features over LEFT and RIGHT of length 1.
8. A set of CPOSTAG features of dependent word and head word over LEFT and RIGHT of length 1.
9. A set of combinations of the POSTAG and FORM features over LEFT and RIGHT of length 2.
10. A set of DEPREL (dependency relation) features over LEFT and RIGHT of length 1.
11. A set of DEPREL features of dependent word over LEFT and RIGHT of length 1.
12. A set of MORPH features over LEFT and RIGHT of length 3.
13. A set of NE features over LEFT and RIGHT of length 3.
14. A set of SC features over LEFT and RIGHT of length 3.

4.3 Data Set

We discuss the description of the data sets which are used in our experiments. The Treebanks used in our experiment are KGPBenTreeBank (Chatterji et al., 2013) and ICON 2010 Treebank (Husain et al., 2010). We follow the dependency relations used in Chatterji et al. (2013) and Husain et al. (2010). Dependency relations in KGPBenTreeBank are assigned between words in a sentence. Dependency relations in ICON 2010 Treebank are assigned between chunks in a sentence.

Chatterji et al. (2013) categorize these dependency relations in Bangla into three main types, namely, intrachunk relations, interchunk relations and interclause relations. Interchunk relations include *karta (subject)*, *karma (object)*, *karan (instrumental)*, *adhikaran (locative)* etc. *Karta (Subject)* is further subdivided into six categories, namely, *sadharan karta (general subject)*, *kriya sampadak karta (doer subject)*, *anubhav karta (experiencer subject)*, *paroksha karta (passive subject)*, *samanadhikaran (noun of proposition)* and *sahakari karta (associate subject)*. Data sets are shown in table 1.

	TB1	TB2
No of tags	55	37
No of training sentences	2905	1130
No of test sentences	322	150
Average sentence length	13.78	10.03

Table 1: Data sets
TB1: KGPBenTreeBank, TB2: ICON 2010 Treebank

4.4 Experimental Results

The metrics used to evaluate parser are labeled attachment score (LAS), unlabeled attachment score (UAS) and label accuracy (LA).

We have done experiments on KGPBenTreeBank using MaltParser settings and the features of Kolachina et al. (2010) and Kosaraju et al. (2010). The evaluation results are shown in the table 2. It is observed from the experiments that MaltParser settings and the features of Kosaraju et al. (2010) give the better result.

Experimental result of our data driven parser: We have done a set of experiments on KGPBenTreeBank and ICON 2010 Treebank using different combinations of basic features. The best experimental results on KGPBenTreeBank and ICON 2010 Treebank are shown in table 3. It is observed from

Systems	LAS	UAS	LA
S1	59.34	72.30	67.28
S2	61.19	72.97	69.36

Table 2: Evaluation results on KGPBenTreeBank using settings of different systems
S1: Kolachina et al. (2010), S2: Kosaraju et al. (2010)

the experiments that the following feature combinations FORM, LEMMA, POSTAG, CPOSTAG, number, person, vibhakti, NE and SC give the best result. It is observed in table 3 that the above experiments give better result on ICON 2010 Treebank than KGPBenTreeBank because sentences in KGPBenTreeBank are very complex.

Corpus	LAS	UAS	LA
TB1	61.51	73.20	69.52
TB2	75.75	88.97	79.08

Table 3: Parser evaluation results for KGPBenTreeBank and ICON 2010 Treebank

4.5 Analyzing the Mistakes of Data Driven Parser

We have analyzed the major errors that occur in the output of the baseline parser, and some of them are described below. The *sadharan karta (general subject)* is sometimes wrongly identified as *karma (object)*, the *vidheya karta (noun of proposition)* is wrongly identified as *sadharan karta (general subject)*, the *karma (object)* is wrongly identified as *sadharan karta (general subject)* or *kriya antargata bisheshya (part of relation)* and some *kriya antargata bisheshya (part of relation)* are wrongly identified as *sadharan karta (general subject)*.

Example sentence with mistake is shown below. In the following example, নগেন (nagena) [Nagen] is the *sadharan karta (general subject)* and নৃপতি (nRRipati) [king] is the *vidheya karta (noun of proposition)* of the verb ছিলেন (chhilena) [was]. But the data driven parser identifies both নগেন (nagena) [Nagen] and নৃপতি (nRRipati) [king] as *sadharan karta (general subject)*.
নগেন নামে এক নৃপতি ছিলেন. (nagena nAme eka nRRipati chhilena) [There was a king named Nagen.]

5 Rule Based Improvement

As discussed in previous section, the data driven approach has limited data. For this reason, the data driven approach fails to produce a good quality parser. Since it is time consuming to get a large annotated Treebank, can be improved the quality of the baseline parser in other way.

We used the parser with BHMT system and observed many errors related to adding incorrect vibhakti (suffix) to the noun phrases in the translated Hindi sentences. After analysis it is found that such errors occur because of incorrect identification of *karta (subject)* in the source Bangla sentences.

So, correct identification of *karta (subject)* is very useful for BHMT system. As an initial problem, we decided to work on accurate identification of *karta (subject)*. We have classified the major types of errors of *karta (subject)*. Some of them are discussed below. The relation between the noun phrase and an intransitive verb is often wrongly labeled as *karma (object)* instead of *karta (subject)*. We also observed several cases where two noun phrases related to the same verb, one of which is *sadharan karta (general subject)*, and the other is *vidheya karta (noun of proposition)*, are both wrongly identified as *karta (subject)*.

Some of these errors can be fixed if we have the argument structure and constraints associated with the different values. For these reasons, we have classified Bangla verbs based on valency, which is the number of arguments taken by a verb. The arguments of a verb include subject and all the objects of that verb. There are three basic classification of verbs based on valency, namely, intransitive, transitive and ditransitive. We have also classified the verbs based on action. The action of verbs indicates either physical action or mental action. We have created a list of mental verbs and a list of linking verbs, which are also known as copula verb, join the subject of a sentence with its complement.

We also created the karaka (case) frames of 29 common verbs with the help of Bangla corpus IL-POST (Baskaran et al., 2008). We study this corpus to know which verbs take which dependency relations and its relation with vibhakti (suffix), lexical type, named en-

tity and semantic class. For Bangla, we follow the argument structure of karaka (case) frame for each verb entry used in Begum et al. (2008) and De et al.(2009a). We kept the following information in the karaka (case) frame for each verb entry, name of the verb, type of the verb i.e. transitive, intransitive, ditransitive, mental verb or linking verb, karaka (case) relations, necessity of the arguments which can be either mandatory (M) or desirable (D), vibhakti (suffix) information, lexical category, named entity tag and semantic class of each arguments. Table 4 shows the karaka (case) frame of the verb যা (yA) [go].

Dep rel	Necessity	Vibhakti	Lexical type	NET	Semantic class
k1d	M	0	NN |NNP |PRP	0 |PERSON	Animate |Inanimate
k7p	D	0|এ (e) |য় (Ya)	NN |NNP |PRP	0 |LOCATION	0
k7t	D	0|এ (e) |য় (Ya) |পর (para)	PRP |NN	0 |TIMEX	0

Table 4: Karaka frame of the verb যা (yA) [go]
NN: Noun, NNP: Proper Noun, PRP: Pronoun, Dep rel: Dependency Relation, NET: Named Entity Tag, k1d: *kriya sampadak karta (doer subject)*, k7p: *sthanadhikaran (spatial locative)*, k7t: *kaladhikaran (temporal locative)*

In table 4, the features of three dependents, namely, *kriya sampadak karta (doer subject)*, *sthanadhikaran (spatial locative)*, and *kaladhikaran (Temporal Locative)* of the verb যা (yA) [go] are shown. The *karta (subject)* is mandatory (M) and the other two dependents are desirable (D) for this verb. The possible values of the features are separated by |(pipe) symbol. Zero (0) indicates that the corresponding value of the feature is either null or unknown.

We have proposed some methods to improve the accuracy of *karta (subject)* using karaka (case) frames and Bangla specific rules, which are discussed in the next sections.

5.1 Correction of Improper Relations using Karaka Frames

In this section, we discuss the methods for detection and correction of improper dependency relations in the output of the data driven parser using karaka (case) frame of the verbs.

Karaka (case) frame of a verb consists of mandatory karaka (case) relations and desirable karaka (case) relations. We first assign every mandatory karaka (case) relations in the karaka (case) frame of a verb to the noun phrases in a sentence. After assigning the mandatory karaka (case) relations to the noun phrases in a sentence, if there exists any noun phrases in a sentence that are not assigned by the mandatory karaka (case) relations then from these noun phrases in a sentence, some or all are assigned by the desirable karaka (case) relations in the karaka (case) frame. The detail study is discussed below.

Preprocessing steps of the Algorithm: A sentence in the output of the data driven parser is taken. We split up the sentence into n clauses using clause boundary identifier. We consider the karaka (case) frame of the verbs in each sentence.

Description of feature structure: Consider a noun phrase np with head $h(np)$ in the output of data driven parser is related to a verb vg with dependency relation dr. The relevant features of $h(np)$ refer to the root, person, number, vibhakti (suffix), lexical type, named entity tag and semantic class. The relevant features of vg refer to the root, person and vibhakti (suffix). The relevant features of dr in the karaka (case) frame refer to set of vibhakti (suffix), set of lexical type, set of named entity tag and set of semantic class.

Definition of Match: If *vibhakti*, *lexical_type*, NET and *semantic_class* of $h(np)$ belong to *vibhakti* list, *lexical_type* list, NET list and *semantic_class* list of dr in the karaka (case) frame of vg, respectively, then we say that this instance of the relation dr between $h(np)$ and vg are matched, else we call them unmatched. This procedure is outlined in Procedure Match.

This is explained in more detail below: Initially we mark each relation type ka_r_m in the karaka (case) frame $k\_f(vg_i)$ of each verb vg_i and each $h(np_j)$ in a sentence as unmatched. For each clause cl_i in a sentence s, we consider each $h(np_j)$ with dr_j and check whether the features of $h(np_j)$ and features of dr_j in $k\_f(vg_i)$ are matched. If it is matched then we mark both the $h(np_j)$ and dr_j in

Input: Features of $h(np)$ and features of dr in karaka (case) frame of vg.

let $fs(h(np))$ be the features of $h(np)$.
let $k\_f(vg)$ be the karaka (case) frame of verb vg.
let $fs(dr, k\_f(vg))$ be the features of dr in karaka (case) frame of vg.

if $fs(h(np)).vibhakti \in fs(dr, k\_f(vg)).vibhakti\ list$ ***and*** $fs(h(np)).lexical\_type \in fs(dr, k\_f(vg)).lexical\_type\ list$ ***and*** $fs(h(np)).NET \in fs(dr, k\_f(vg)).NET\ list$ ***and*** $fs(h(np)).semantic\_class \in fs(dr, k\_f(vg)).semantic\_class\ list$ **then**
  return *Matched*
else
  return *Unmatched*

Procedure Match

$k\_f(vg_i)$ as matched. This method is shown in Algorithm 1.

Input: A sentence.
Resources used: Karaka (case) frame of verbs.
Step 1: Run data driven parser on the input sentence.
Step 2: Run clause boundary identifier on the input sentence.
Initialize: Mark each ka_r_m in the karaka frame and each $h(np_j)$ in a sentence as unmatched.

begin
  for *each* cl_i *in* s **do**
    for *each* np_j *in* cl_i **do**
      if dr_j *is in* $k\_f(vg_i)$ **then**
        if $fs(h(np_j))$ *matched with* $fs(dr_j, k\_f(vg_i))$ **then**
          mark both $h(np_j)$ and dr_j in $k\_f(vg_i)$ as matched .
    end
  end
end

Algorithm 1: Correction of Improper Relations using Karaka Frames- Part 1

If any unmatched ka_r_m exists in $k\_f(vg_i)$ then for each of unmatched ka_r_m in $k\_f(vg_i)$, first we consider mandatory karaka (case) relation Mr_j. Then we search for the np in a clause whose features of $h(np)$ are matched with the features of the unmatched Mr_j in $k\_f(vg_i)$. If multiple records are found, we pick up the first one f_np and assign that unmatched Mr_j to the dependency relation of $h(f\_np)$. We also assign vg_i to the parent of $h(f\_np)$. We mark both the $h(f\_np)$ and that unmatched Mr_j in $k\_f(vg_i)$ as matched.

Now we consider the desirable karaka (case) relation Dr_j. If Dr_j is found unmatched then, we search for the unmatched np in a clause whose features of $h(np)$ are matched with the features of the unmatched Dr_j in $k\_f(vg_i)$. If multiple records are found, we pick up the first one f_np and assign that unmatched Dr_j to the dependency relation of $h(f\_np)$. We also assign vg_i to the parent of $h(f\_np)$. We mark both the $h(f\_np)$ and that unmatched Dr_j in $k\_f(vg_i)$ as matched. This method is shown in Algorithm 2.

5.2 Correction of Improper Relations using Rules

In this section, we discuss the correction of improper *karta (subject)* relation in the output of baseline parser using Bangla specific rules. We observed and classified major types of errors and formulated 45 rules. Some of them are discussed below.

Observation 1: We observed that in several cases *anubhav karta (experiencer subject)* is incorrectly identified. We observed in the output of data driven parser is that some head of noun phrases with genitive marker (র) (ra), which is related to the noun of mental verbs with the relation *sambandha (genitive relation).*

The *anubhav karta (experiencer subject)* takes genitive marker র (ra) and nominative marker. The *anubhav karta (experiencer subject)* is always animate entity. A noun phrase with genitive marker র (ra), it's semantic class is animate and it is followed by a mental verb, we say that this noun phrase is *anubhav karta (experiencer subject).*

In the following example, আমার (AmAra) [my] with র (ra) vibhakti (suffix) is related to the mental verb শীত করছে (shIta karachhe) [getting cold] with the relation *anubhav karta (experiencer subject).* Rule 1 takes care of this observation.

Input: Each token in the parsed sentence and each dependency types in the karaka (case) frame of verbs mark with matched or unmatched.

begin
  for *each* cl_i *in* s **do**
    `/* Mandatory karakas */`
    for *each* Mr_j *in* $k\_f(vg_i)$ **do**
      if Mr_j *is unmatched* **then**
        Search for np in cl_i whose $fs(h(np))$ matched with $fs(Mr_j, k\_f(vg_i))$.
        if *one or more records are found* **then**
          pick up f_np.
          assign Mr_j to dependency relation of $h(f\_np)$ and vg_i to parent of $h(f\_np)$.
          mark $h(f\_np)$ and Mr_j as matched.
    end
    `/* Desirable karakas */`
    for *each* Dr_j *in* $k\_f(vg_i)$ **do**
      if Dr_j *is unmatched* **then**
        Search for unmatched np in cl_i whose $fs(h(np))$ matched with $fs(Dr_j, k\_f(vg_i))$.
        if *one or more records are found* **then**
          pick up f_np.
          assign Dr_j to dependency relation of $h(f\_np)$ and vg_i to parent of $h(f\_np)$.
          mark $h(f\_np)$ and Dr_j as matched.
    end
  end
end
Output: Corrected output of the output of data driven parser.

Algorithm 2: Correction of Improper Relations using Karaka Frames- Part 2

আমার শীত করছে. (AmAra shIta karachhe) [I am getting cold.]

Observation 2: We observed several cases where two noun phrases related to the same linking verb, one of which is *sadharan karta (general subject)*, and the other is *vidheya karta (noun of proposition)*, are both wrongly identified as *sadharan karta (general subject).*

There are many sentences which have linking verbs. These sentences have different structures. We discuss one of them. A noun phrase with null marker is followed by a noun phrase with genitive marker র (ra) is followed by another noun phrase with null marker which is followed by a linking verb, we say that the first noun phrase is *sadharan karta (general subject)* and the third noun phrase is *vidheya karta (noun of proposition).*

In the following example, এটাই (eTAi) [this] is *sadharan karta (general subject)* and বই (ba_i) [book] is *vidheya karta (noun of proposition).* Both are related to the linking verb ছিল (chhila) [was]. Rule 2 takes care of this observation.

এটাই আমার বই ছিল. (eTAi AmAra ba_i chhila) [This was my book.]

Observation 3: We observed that a noun phrase is immediately followed by a verbal noun is identified as *karta (subject)* instead of *karma (object)* or *sthanadhikaran (place related locative).*

Vibhakti (suffix) of the noun phrase is এ (e) or য় (Ya), type of the noun phrase is location and it is immediately followed by a verbal noun, we say that this noun phrase is *sthanadhikaran (place related locative).*

In the following example, পাহাড়ে (pAhA.De) [hill] takes এ (e) vibhakti (suffix) and it is related to the verbal noun ওঠার (oThAra) [climbing] with relation *sthanadhikaran (place related locative).* Rule 3 takes care of this observation.

পাহাড়ে ওঠার পর সে হাঁপিয়ে গেল. (pAhA.De oThAra para se hA.NpiYe gela) [He became tired after climbing the hill.]

Format of the rules: The rules have two parts, namely, LHS (left hand side) and RHS (right hand side), which are separated by ⇒ symbol. The format of the rule is shown below.

CNA1 < feature1: value1 /value2, feature2: value3, ... > CNA2 < feature1: value4, ... > CN CNB1 < feature1: value5, ... > ⇒ CNA1 < feature5: value6, ... > CNA2 < > CN* CNB1 < >*

LHS consists of chunks ids with features of the head of the chunk which are enclosed

within < >. Features are separated by ',' (comma). Multiple values of the features are separated by '|'. Chunk id consists of name of the chunk followed by number. Same chunk names are distinguished by numbers i.e. CNA1, CNA2. When new chunk name comes it's number starts from 1 i.e. CNB1. '...' inside the < > indicates multiple combinations of different features with values can be included in the rule. CN* indicates that there exists none or more number of chunks in between CNA2 and CNB1. Those chunks have no significance in the rule.

RHS consists of same number of chunk ids as in LHS. If the features in LHS of the rule are satisfied then the rule is fired and the required modification of the value of the features are done in RHS of the rule. Empty < > and features with values inside the < > after chunk id in the RHS indicate value of the features remain same as the value of the features of the corresponding chunk in LHS and only value of those features of the corresponding chunk in LHS are modified, respectively.

Rule 1: *NP1 < pos: PRN |NN |NNP, vibhakti: র (ra), ner: 0 |PERSON, animacy: animate > VGF1 < class: mental verb > ⇒ NP1 < dep_rel: k1e, parent: VGF1 > VGF1 < >*
Rule 2: *NP1 < pos: PRN |NNP, vibhakti: 0 > NP2 < pos: PRN |NN |NNP, vibhakti: র (ra) > NP3 < pos: NN |NNP, vibhakti: 0 > VGF1 < class: linking verb > ⇒ NP1 < dep_rel: k1, parent: VGF1 > NP2 < > NP3 < dep_rel: k1s, parent: VGF1 > VGF1 < >*
Rule 3: *NP1 < pos: NN |NNP |PRP, vibhakti: এ (e) |য় (Ya), ner: 0 |LOCATION > VGNN1 < pos: NN > ⇒ NP1 < dep_rel: k7p, parent: VGNN1 > VGNN1 < >*

5.3 Experimental Results after Post Processing

We improved our results by post processing the output of the data driven parser using karaka (case) frames and Bangla specific rules. Two different stages (baseline parser and after rule based post processing) of the overall evaluation results are shown in table 5. LAS and LA of different types of *karta (subject)* are shown in table 6 and table 7, respectively. Average LAS improvement of *karta (subject)* based on F-measure on KGPBenTreeBank and ICON 2010 Treebank are 25.35% and 9.53%, respectively. Average LA improvement of *karta (subject)* based on F-measure on KGPBenTreeBank and ICON 2010 Treebank are 25.5% and 9.79%, respectively.

6 Conclusion and Future Work

A hybrid approach of dependency parsing for Bangla is presented in this paper. We have combined two methods i.e. data driven method and rule based post processing for development of dependency parser for Bangla.

In future, we may extend this work to other dependency relations. We may analyze in depth the errors of other dependency relations in order to get more effective features for the development of more karaka (case) frames and develop more Bangla specific rules. We may improve dependency parser for Bangla through unsupervised learning as manually annotated data of dependency relations is very limited.

Acknowledgment

This work is partially supported by the ILMT project sponsored by TDIL program of MCIT, Govt. of India. We would like to thank all the members in Communication Empowerment Lab, IIT Kharagpur.

References

Sankaran Baskaran, Kalika Bali, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, et al. 2008. A common parts-of-speech tagset framework for indian languages. In *In Proc. of LREC 2008*. Citeseer.

Rafiya Begum, Samar Husain, Lakshmi Bai, and Dipti Misra Sharma. 2008. Developing verb frames for hindi. In *LREC*.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 105–111. Association for Computational Linguistics.

Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: a Paninian perspective*. Prentice-Hall of India New Delhi.

Akshar Bharati, Rajeev Sangal, and T Papi Reddy. 2002. A constraint based parser using integer programming. *Proc. of ICON*.

	Baseline parser			After rule based post processing		
Corpus	LAS	UAS	LA	LAS	UAS	LA
TB1	61.51	73.20	69.52	65.38	74.64	73.22
TB2	75.75	88.97	79.08	80.35	89.63	84.20

Table 5: Evaluation results on KGPBenTreeBank and ICON 2010 Treebank

			Baseline parser			After rule based post processing		
corpus	deprel	#occ	recall	precision	F-measure	recall	precision	F-measure
TB1	k1	278	60.79	57.68	59.19	76.98	79.55	78.24
	k1a	5	20.00	100.00	33.33	100.00	83.33	90.91
	k1d	90	46.67	41.58	43.97	82.22	77.08	79.56
	k1e	10	20.00	28.57	23.53	100.00	66.67	80.00
	k1s	40	25.00	50.00	33.33	60.00	77.42	67.61
TB2	k1	166	76.85	72.51	74.62	86.06	81.61	83.78
	k1s	18	66.67	80.00	72.73	83.33	88.24	85.71

Table 6: Labeled attachment score of different types of *karta (subject)*

deprel: Dependency Relation, #occ: no of occurrences in the text, k1: *sadharan karta (general subject)*, k1a: *sahakari karta (associate subject)*, k1d: *kriya sampadak karta (doer subject)*, k1e: *anubhav karta (experiencer subject)*, k1s: *samanadhikaran (noun of proposition)*

		Baseline parser			After rule based post processing		
corpus	deprel	recall	precision	F-measure	recall	precision	F-measure
TB1	k1	62.95	59.73	61.29	80.22	82.90	81.54
	k1a	20.00	100.00	33.33	100.00	83.33	90.91
	k1d	56.67	50.50	53.41	88.89	83.33	86.02
	k1e	20.00	28.57	23.53	100.00	66.67	80.00
	k1s	25.00	50.00	33.33	60.00	77.42	67.61
TB2	k1	81.03	74.02	77.37	87.35	86.30	86.82
	k1s	66.67	80.00	72.73	83.33	88.24	85.71

Table 7: Label accuracy score of different types of *karta (subject)*

Akshar Bharati, Samar Husain, Meher Vijay, Kalyan Deepak, Dipti Misra Sharma, and Rajeev Sangal. 2009. Constraint based hybrid approach to parsing indian languages. In *PACLIC*, pages 614–621.

Sanjay Chatterji, Praveen Sonare, Sudeshna Sarkar, and Devshri Roy. 2009. Grammar driven rules for hybrid bengali dependency parsing. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing, Hyderabad, India*.

Sanjay Chatterji, Tanaya Mukherjee Sarkar, Pragati Dhang, Samhita Deb, Sudeshna Sarkar, Jayashree Chakraborty, and Anupam Basu. 2013. A dependency annotation scheme for bangla treebank. In *Language Resources and Evaluation*.

Sankar De, Arnab Dhar, and Utpal Garain. 2009a. Karaka frames and their transformations for bangla verbs. In *31st All-India Conference of Linguists-2009 (to appear)*.

Sankar De, Arnab Dhar, and Utpal Garain. 2009b. Structure simplification and demand satisfaction approach to dependency parsing for bangla. In *Proc. of 6th Int. Conf. on Natural Language Processing (ICON) tool contest: Indian Language Dependency Parsing*, pages 25–31.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Arnab Dhar, Sanjay Chatterji, Sudeshna Sarkar, and Anupam Basu. 2012. A hybrid dependency parser for bangla. In *10th Workshop on Asian Language Resources, COLING 2012*, page 55.

Aniruddha Ghosh, Amitava Das, Pinaki Bhaskar,

and Sivaji Bandyopadhyay. 2010. Bengali parsing system at icon nlp tool contest 2010. *ICON10 NLP TOOLS CONTEST: INDIAN LANGUAGE DEPENDENCY PARSING*, 960(7269):20.

David G Hays. 1964. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525.

Samar Husain, Prashanth Mannem, Bharat Ram Ambati, and Phani Gadde. 2010. The icon-2010 tools contest on indian language dependency parsing. *Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing, ICON*, 10:1–8.

Samar Husain. 2009. Dependency parsers for indian languages. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*.

Sudheer Kolachina, Prasanth Kolachina, Manish Agarwal, and Samar Husain. 2010. Experiments with maltparser for parsing indian languages. *Proc of ICON-2010 tools contest on Indian language dependency parsing. Kharagpur, India*.

Prudhvi Kosaraju, Sruthilaya Reddy Kesidi, Vinay Bhargav Reddy Ainavolu, and Puneeth Kukkadapu. 2010. Experiments on indian language dependency parsing. *Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing*.

Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 31–38. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical report, Växjö University.