A Supervised Framework for Classifying Dependency Relations from Bengali Shallow Parsed Sentences

Anupam Mondal and Dipankar Das(✉)

Computer Science and Engineering, Jadavpur University, Kolkata, India
link.anupam@gmail.com, ddas@cse.jdvu.ac.in

Abstract. Natural Language Processing, one of the contemporary research area has adopted parsing technologies for various languages across the world for different objectives. In the present task, a new approach has been introduced for classifying the dependency parsed relations for a morphologically rich and free-phrase-ordered Indian language like Bengali. The pair of dependency parsed relations (also referred as *kaarakas* 'cases') are classified based on different features like *vibhaktis* (inflections), Part-of-Speech (POS), punctuation, gender, number and post-position. It is observed that the consecutive and non-consecutive occurrences of such relations play a vital role in the classification. We employed three different machine-learning classifiers, namely NaiveBayes, Sequential Minimal Optimization (SMO) and Conditional Random Field (CRF) which obtained the average F-Scores of 0.895, 0.869 and 0.697, respectively for classifying relation pairs of three primary *kaarakas* and one primary *vibhakti* relation. We have also conducted the error analysis for such primary relations using confusion matrices.

Keywords: Dependency relations · *Kaaraka* · *Vibhakti* · Machine-learning classifiers

1 Introduction

Dependency Parsing, a challenging task for processing any natural language seems an obvious milestone while dealing with morphologically rich and free-phrase ordered languages, especially Indian Languages. Bengali, the seventh popular language[1] in the World, second in India and the national language of Bangladesh is morphologically rich and resource constrained. Thus, to the best of our knowledge, at present, there is no such full-fledged parser available in Indian languages and especially for Bengali. However, Bengali is one of the important Indo-Iranian languages spoken by a population that now exceeds 211 million or 3.11 % of the world population. Geographically, Bengali-speaking population percentages[2] are as follows: Bangladesh (over 95 %), Indian States of Andaman and Nicobar Islands (26 %), Assam (28 %), Tripura (67 %), and West Bengal (85 %). The development of parsers for Indian languages in general

[1] http://listverse.com/2008/06/26/top-10-most-spoken-languages-in-the-world/.
[2] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers.

R. Prasath et al. (Eds.): MIKE 2015, LNAI 9468, pp. 597–606, 2015.
DOI: 10.1007/978-3-319-26832-3_56

and Bengali in particular is difficult and challenging as the language is (1) inflectional language providing the richest and most challenging sets of linguistic and statistical features resulting in long and complex word forms, and (2) relatively free phrase order and less computerized compared to English [2].

Till date, due to the scarcity of reliable annotated data, it is observed that several attempts that were used to develop the parsers for Indian languages mainly depend on linguistic rules [2, 13, 14, 17, 18]. A hybrid dependency parser that proposed two stage parsing system [1] and a data driven parser that identifies the dependency relations between chunks in a sentence using Treebank are found in the literature. The respective researchers conducted their experiments to improve the mistakes of the data driven parser based on the effects of case frames. However, none of the approaches has considered the classification of the relation pairs using machine learning approaches. Therefore, the present task aims to identify the chunks and phrases and their intra relationships from sentences using a data-driven approach. In addition, we have also classified the dependency relations that are considered as the prerequisites towards developing a full-fledged parser. It is observed that different relations like *kaaraka* and *vibhakti* use to play an important role for constructing the sentences. In Bengali grammar, *kaaraka* is the relationship between verb and noun or verb and pronoun in a sentence. There are seven different *kaaraka* relations such as *kartaa*, *karma*, *karana, sampradana, apadana, nimito, adhikarana* that are represented in the paper as K1, K2, K3, K4, K5, K6 and K7, respectively. Here, we have dealt with only three *kaaraka* relations e.g., *kartaa* (K1), *karma* (K2) and *adhikarana* (K7) as per the frequency. The examples of the *kaaraka* have been illustrated in Fig. 1.

Kaaraka:

***Kartaa Kaaraka*:-**

Chatok kator swarey daake. চাতক কাতর স্বরে ডাকে।

(*The bird is singing very sadly.*)

Illustration: The bird "chatok" (চাতক) is active.

***Karma Kaaraka*:-**

Ami tomakei khunjchilam. আমি তোমাকেই খুঁজেছিলাম।

(*I was searching you only.*)

Illustration: It emphasizes "you" (তোমাকেই) because it makes a sense that, the person I was searching for is 'you'.

Adhikarana Kaaraka:-

Takata ghore pore thakte dekhechi. টাকাটা ঘরে পড়ে থাকতে দেখেছি।

(*I have seen the money in the room.*)

Illustration: Where I have seen the money? The answer is room (ঘরে).

Fig. 1. *Kaaraka* examples with Illustration

In case of making the Bengali sentences, *vibhakti* plays an important role. There are presently ten symbols that are considered for indicating the *vibhaktis* in Bengali viz. *ay, ke, re, te, sunyo* etc. and are represented as R1, R2, R3, …, R10, respectively in the

Vibhakti:

য় (ay) Vibhakti:-

Takai ki na hoy! টাকায় কি না হয়! (*What can't be bought by money?*)

Illustration: টাকায় " টাকা (money) + য় (ay) = টাকায় (by money) ", য় (ay) *Vibhakti* is used.

কে (ke) *Vibhakti:-*

Maake aar desher matike valobaste sekho. মাকে আর দেশের মাটিকে ভালবাসতে শেখ। (*Learn how to love mother and motherland.*)

Illustration: মাকে " মা (mother) + কে (ke) = মাকে (to mother) ", (কে) ke *Vibhakti* is used and in মাটিকে " মাটি (motherland) + কে (ke) = মাটিকে (to motherland) ", (কে) ke *Vibhakti* is used.

তে (te) *Vibhakti:-*

Somitite kichu dite hbe. সমিতিতে কিছু দিতে হবে। (*Some money has to be given to the Committee.*)

Illustration: সমিতিতে " সমিতি (Committee) + তে (te) = সমিতিতে (Somitite) (to the Committee) ", তে (te) *Vibhakti* is used.

Fig. 2. *Vibhakti* examples with Illustration

annotated corpus [12]. It also provides the information related to the respective *kaarakas* as shown in Fig. 2. Therefore, in order to deal with such relations using a machine learning framework, we always need to extract the linguistic features at different levels of granularities (word, chunk and or sentence).

The first obvious question was how to select the important *kaarakas* in order to identify the dependency parsed relations. To start with the top four frequent dependency relations (K1, K2, K7 and R6), we have initiated the inclusion of associated features viz. Part-of-Speech (POS), punctuations, number, gender and post- position for implementing the machine learning framework. An exhaustive error analysis with respect to different classifiers and mode of operations were performed to achieve the maximum F-Scores of 52 %, 42 %, 45 % and 69 % for K1, K2, K7 and R6 dependency relations, respectively.

The rest of the Sections are as follows. In Sect. 2, we have discussed the related attempts made in developing the parsing technologies for Bengali and other Indian languages. Preprocessing of the corpus and selection of top-frequent relations are discussed under Sect. 3. In the next Section, we have mentioned the methodologies to extract the related features which were supplied with the relations. The system framework for classifying the dependency relations is discussed in Sect. 5 while in Sect. 6 we have analyzed the errors in terms of the confusion matrices. Finally, Sect. 7 concludes the task and mentioned the possibilities of future scopes.

2 Related Work

In literature survey, we have found the development of a predictive parser in an efficient way for morphological rich and free word order languages viz. Bengali [3, 15]. The identification of structured Bengali sentences purposes the symbols (constituents) based on Context Free Grammar (CFG) rules. The recognition of Bengali grammar from the sentences was a contributory attempt of this task for the reason of availability of different grammars. In contrast, the grammar driven parser was developed for Bengali language which achieved a score near to 90 % [4] in a Shared Task[3]. A group of researchers were trying to generate a new dataset for reducing the gap between structured and unstructured form of data with the help of Treebank's which contains approximately 1500 sentences. They had not used the developed dataset for extracting linguistics rules for the task. A comparative analysis had done between grammar driven and data driven approaches of Bengali language for developing a dependency parser of Bengali language [10, 11].

Lexical Functional Grammar (LFG) [6] based linguistic phenomena has been applied in a wide range for Bengali language. The Constituent phrase Structure (C-structure) and Functional structure (F-structure) is considered as primary features for LFG technique.

In concern, the Paninian model and dependency based framework were introduced as an effective technique for parsing the Bengali sentences [5]. The researchers were taking help of demand-source concept under Paninian grammar with six different types of *kaaraka* and verb. The *kaaraka* and verb group of words are treated as source and demand groups, respectively for the task where the dependency tree root indicates as verb along with appropriate *kaaraka* labels. Several researchers had analyzed the dependency parsers [7, 18] for Indian languages and remarked that the development of dependency parsers can be carried out either using grammar driven approaches or data driven approaches [16]. In case of morphologically rich and free word order languages, the grammar driven approach is difficult than the data driven approach. In several cases, Malt parser[4] has been used as transition based approach for dependency parsing and it mainly consists of the transition and classifier based on prediction approaches.

In this report, we have introduced the dependency relations (*kaaraka, Vibhakti*) based classification approaches with several features viz. POS, punctuation, number etc. for developing a Bengali dependency parser.

3 Resource Preparation

3.1 Corpus

In order to develop a dependency parser for any language, we need to identify the linguistic rules that guide us how to relate different chunks of a sentence using the grammar of that language. The effect of morphological richness and free phrase ordered

---

[3] http://ltrc.iiit.ac.in/mtpil2012/.

[4] www.maltparser.org.

<af=e, drel=nmod: NP2/name=NP>
<af=keu, drel=k1: VGF/name=NP2>
<af=biRayZa, drel=k7: VGF/name=NP3>
<af=AgrahI, drel=k1s: VGF/name=JJP>

Fig. 3. A sample of dependency relations of words based on Shakti Standard Format (SSF)

structure make the rule identification difficult. Therefore, in the present task, we have mainly tried to design the language dependent rules without considering the structure of the sentences in the input corpus. We have observed that in Bengali language, the words of a sentence appear in the form of any of the seven *kaarakas* and ten *vibhaktis* as per the guidelines [8]. The dependency relation (*drel*) tag of a sentence has shown in Fig. 3. We have adopted two different techniques based on consecutive and non-consecutive occurrences of such primary dependency relations, as described in the next section.

3.2 Selection of Consecutive and Non-consecutive Occurrences

In order to derive the classification features, we have evaluated the occurrence probabilities of the dependency relations from the corpus in terms of consecutive and non-consecutive appearances. In case of consecutive dependency relation, we have considered the dependency relations of neighboring words where as two or three word gap was considered for identifying dependency relations in case of non-consecutive occurrences. We have observed that, in case of morphologically rich and free phrase ordered language, the occurrence probability of the dependency relation (*kaaraka*) is high in case of consecutive words. Similarly, the non-consecutive presence of the dependency relations also plays a crucial role. The non-consecutive appearances help to identify the implicit co-reference exists among the long-distanced words in order to develop a full-fledged dependency parser. Table 1 has illustrated the dependency relations of consecutive and non-consecutive appearances of the words whereas Fig. 4 shows the steps to identify consecutive dependency relations and similar steps we have considered for identifying the non-consecutive relations.

In order to implement any data-driven model, we need to analyze the data based on different statistics prior to start applying the supervised algorithms. In the present report, the whole corpus was collected from the articles published in newspapers, text books by a group of members of IIIT-H and annotated with different relations based on *kaarakas* (e.g. K1, K2, K7) and *vibhaktis* (e.g. R1, R2, R6) [9]. The corpus was provided by IIIT-H in a shared task challenge[5] in order to build the shallow parser for Bengali. We split the corpus in three different sets, namely training, development and test randomly with a distribution of 50 %, 20 % and 30 %, respectively. The important distributions of the sentential relations, POS tags and their combinations in these three sets are mentioned in Table 2 with their corresponding distributions. Therefore, we attempted to identify other features that are available from the annotated corpus.

[5] http://shiva.iiit.ac.in/SPSAL2007/.

Table 1. Important dependency relation combinations for consecutive (non-consecutive) words

Important relation combinations	Training	Development	Test
K1-K2	139 (376)	20 (53)	44 (106)
K2-k2	126 (274)	7 (22)	83 (165)
R6-K1	121 (234)	11 (14)	77 (182)

Step1: *Take a list of dependency relations and termed it as a DRL.*
DRL= { K1, K2, K3, K4, K5, K6, K7, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, vmod and ccof }
Step2: *Extract the dependency relation from each of the words of a sentence and store in a list called L.*
Step3: *Pick up two consecutive relations,* R_i *and* R_{i+1} *from the list L.*
Step3.1: *If both* R_i *and* R_{i+1} *belong to DRL, take the relation-pair* R_i *and* R_{i+1} *as our candidate pair.*
Step3.2: *Else move to next relation pair,* R_{i+1} *and* R_{i+2}*.*
Step4: *Repeat until the list L is exhausted.*

Fig. 4. Steps for identifying the consecutive dependency relations

Table 2. Important primary relations, their POS tags and combinations in Training, Development and Test Data Sets

	Training	Development	Test
Words	2329	660	1854
Sentences	700	150	280
Top 4 relations			
K1	734	174	252
K2	756	101	386
K7	395	91	273
R6	446	58	297
POS tags			
Noun	795	165	569
Pronoun	384	61	77
Unk	709	76	332
POS tag combinations			
Adverb-Noun	869	175	623
Noun-Pronoun	1179	226	673

4 Feature Extraction

While analyzing the training data with 700 sentences, we have found that a total 2329 instances are present in the top-4 relations (K1, K2, K7 and R6). These relations are appeared with an average of 3.3 relational instances per sentence and therefore considered as our key instances. The distributions of such top-4 relations are mentioned in

Table 3. Important Feature analysis for (Training/Development/Test) Datasets

Relations on resource ($T/$D/$Te)	POS (f1)			Punc (f2)	Gender (f3)		Number (f4)			Post position (f5)	
	Adj	Adv	Noun	unk	Sg	Pl	4	5	a	D	O
K1 (734/174/252)	47/44/7	22/1/5	211/61/128	**265/34/79**	323/77/140	**37/10/15**	6/0/3	0/0/0	0/0/0	359/87/154	4/0/1
K2 (756/101/386)	**80/5/93**	15/4/13	**295/41/179**	206/18/73	**346/44/177**	14/4/10	7/1/3	21/6/6	3/2/0	348/46/183	12/2/4
K7 (395/91/273)	36/6/20	**25/5/3**	157/39/153	81/8/58	148/39/152	0/0/3	0/1/0	1/2/1	4/0/3	195/50/164	0/1/2
R6 (443/58/297)	9/2/2	12/0/6	141/24/136	**157/16/121**	221/31/141	32/6/14	0/0/0	0/0/0	6/0/11	3/0/1	250/37/153
Total (2329/660/1854)	172/57/122	74/10/27	795/165/596	709/76/331	1038/191/610	83/20/42	13/2/6	22/8/7	13/4/14	905/183/502	266/40/160

$T → Training $D → Development $Te → Test Punc (f2) → Punctuation
Adj → Adjective Sg → Singular Adv → Adverb Pl → Plural a → any number

Table 3. After an initial investigation on the training, test and development data sets, we extracted five features (POS, punctuation, Gender, Number and post-position) that play important roles in distinguishably identifying the top-4 primary relations. The POS tag feature produces remarkable output for identifying K1, K2 and K7 relations. Mainly, the adjective, adverb, noun, verb and WQ tags are notable for identifying the relation pairs of K1-K2 and K2-K2 pairs. In case of gender feature, K2 mainly appears as singular whereas K1 represents the plural. The above derived observations played the vital roles for designing the dependency parser for Bengali language.

5 System Framework

We have used the Weka[6] tool and employed two different classifiers viz. NaiveBayes and SMO for classifying the relations. Along with the extracted features described in the previous section, we also included the consecutive and non-consecutive occurrences of the relation pairs and their POS tag combinations as features for developing the classification framework. It is observed that the inclusion of the features related to the consecutiveness improves the accuracy of the system as illustrated in Table 4. We have adopted four different modes of operations namely Use training set, Supplied test set, Cross validation Folds-10 and Percentage split 66 % on each of the classifiers in Weka toolkit. NaiveBayes classifier produced the remarkable accuracy (70 %), average precision, recall and F-measure with top 6 features and all features with respect to all types of operations. Similarly, in case of SMO classifier, the accuracy (75 %), average precision, recall and F-measure are notable with top 5, top 6 and all features set with respect to all types of operations.

In addition to different classifiers in Weka, we also used Conditional Random Field (CRF[7]) for classifying the primary dependency relations. The precision and F-measure with respect to top 4 features are high for K1 whereas recall is low with top 6 features for relation K2. In case of identifying the R6 relation using CRF, the precision, recall and F-score are notable with top 5, top 6 and all features. The detail observations of precision, recall and F-Score for all primary relations (K1, K2, K7 and R6) along with secondary relations (SR) is mentioned in Table 5.

6 www.cs.waikato.ac.nz/ml/weka.

7 nlp.stanford.edu/software/CRF-NER.shtml.

Table 4. System generated results with important mode of operation for different classifiers

A	B	C	D	E	F
NaiveBayes classifier					
Cross-validation Folds-10 [661]	9#	498	75.34	0.80	0.753
	8$	496	75.04	0.79	0.750
	7@	439	66.41	0.64	0.664
	6**	412	62.33	0.60	0.623
	5***	399	60.36	0.57	0.604
SMO classifier					
Use training set [661]	9#	577	87.29	0.89	0.873
	8$	575	86.99	0.88	0.870
	7@	558	84.42	0.86	0.844
	6**	524	79.27	0.80	0.793
	5***	524	79.27	0.80	0.793

all features $ top 6-features @ top 5-features
** top 4-features *** top 3-features
A → Important Mode of Operation [No. of Instances]
B → No. of Attributes (No. of features)
C → No. of Correctly Classified Instances
D → Avg. Precision E → Avg. Recall F → Avg. F-Measure

Table 5. System generated important results based on CRF tool

Dependency relation with no. of occurrence		Precision	Recall	F-Score
K1 (734)	All 7	0.431	0.6	0.5
	Top 5	0.434	0.7	0.5
	Top 6	0.478	0.7	0.6
K2 (756)	All 7	0.518	0.3	0.4
	Top 5	0.522	0.3	0.4
	Top 6	0.444	0.4	0.4
K7 (396)	All 7	0.570	0.3	0.4
	Top 5	0.588	0.4	0.4
	Top 6	0.696	0.3	0.4
R6 (443)	All 7	0.713	0.7	0.7
	Top 5	0.714	0.7	0.7
	Top 6	0.673	0.6	0.6
SR (1986)	All 7	0.976	1	0.9
	Top 5	0.978	1	0.9
	Top 6	0.975	1	0.9

6 Error Analysis

We have also conducted an error analysis based on the confusion matrices for the classified dependency relations in the form of graphical representation. Figure 5 shows the occurrences of the relations (K1, K2, K7, R6 and SR) in the confusion matrices for different classifiers, NaiveBayes, SMO and CRF respectively with respect to all features. However, we have observed that the occurrences are high when K1 relation appears as K2, K2 appears as K1 or K7, K7 appears as K1 and R6 as SR (Secondary relations) relation for their important modes of operation for all classifiers.

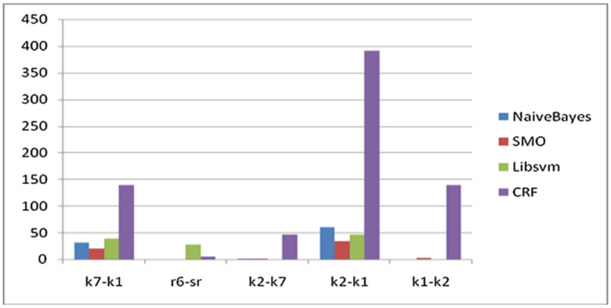

Fig. 5. Confusion matrix for different classifiers w.r.t all features of important modes

7 Conclusion and Future Work

In this paper, we have introduced the approaches for classifying the dependency parsed relations based on *kaarakas* and *vibhaktis* for the morphologically rich and free-word order language, Bengali. The consecutive and non-consecutive techniques have been used for identifying the important dependency relations from the sentences. The dependency relations based on chunks or phrases also gives satisfactory output. Finally, we prepared a machine-learning framework for classifying the dependency relations followed by an exhaustive error analysis that shows crucial insights towards developing a full-fledged parser.

In future, we will include the semantic relationships for extracting the suitable chunks from the sentences which can guide to develop the full-fledged dependency parser in efficient manner.

References

1. Dhar, A., Chatterji, S., Sarkar, S., Basu, S.: A hybrid dependency parser for Bangla. In: Proceedings of the 10th Workshop on Asian Language Resources, COLING Mumbai, pp. 55–64, India (2012)

2. Ghosh, A., Bhaskar, P., Das, A., Bandyopadhyay, S.: Dependency parser for Bengali. In: JU System at ICON (2009)
3. Chatterji, S., Sonare, P., Sarkar, S., Roy, D.: Grammar driven rules for hybrid Bengali dependency parsing. In: Proceedings of ICON 2009 NLP Tools Contest: Indian Language Dependency Parsing, Hyderabad, India (2009)
4. Das, A., Shee, A., Garain, U.: Evaluation of two Bengali dependency parsers. In: Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL), COLING, pp. 133–142 (2012)
5. Garain, U., De. S.: Dependency Parsing in Bangla. IGI Global (2013)
6. Haque, M.N., Khan, M.: Parsing Bangla using LFG. In: Proceedings of Association for Computational Linguistic (1997)
7. Kosaraju, P., Kesidi, S.R., Ainavolu, V.B.R., Kukkadapu, P.: Experiments on Indian language dependency parsing. In: Proceedings of ICON (2010)
8. Bharati, A., Sangal, R., Sharma, D.M.: SSF: Shakti Standard Format Guide (2007)
9. Das, D., Choudhury, M.: Chunker and shallow parser for free word order languages: an approach based on valency theory and feature structures. In: Proceedings of ICON (2004)
10. Begum, R., Husain, S., Sharma, D.M., Bai, L.: Developing verb frames in Hindi. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco (2008)
11. Chatterji, S., Sarkar, T.M., Sarkar, S., Chakrabory, J.: Kaaraka relations in Bengali. In: Proceedings of 31st All-India Conference of Linguists (AICL), Hyderabad, pp. 33–36, India (2009)
12. Bharati, R., Sangal, D.M., Bai, L.: AnnCorra: annotating corpora guidelines for POS and chunk annotation for Indian languages. Technical report (TR-LTRC-31), LTRC, IIIT Hyderabad, India (2006)
13. Ghosh, A., Das, A., Bhaskar, P., Bandyopadhyay, S.: Bengali parsing system. In: ICON NLP Tool Contest (2010)
14. Rao, P.R.K., Vijay, S.R.R., Vijaykrishna, R., Sobha, L.: A text chunker and hybrid POS tagger for Indian languages. In: Proceedings of IJCAI Workshop on Shallow Parsing for South Asian Languages (2007)
15. De, S., Dhar, A., Garain, U.: Structure simplification and demand satisfaction approach to dependency parsing in Bangla. In: Proceedings of ICON 2009 NLP Tools Contest: Indian Language Dependency Parsing, Hyderabad, India (2009)
16. Bandyopadhyay, S., Ekbal, A., Halder, D.: HMM based POS tagger and rule-based chunker for Bengali. In: Proceedings of NLPAI Machine Learning Workshop on Part of Speech and Chunking for Indian Languages (2006)
17. Das, D., Ekbal, A., Bandyopadhyay, S.: Acquiring verb subcategorization frames in Bengali from corpora. In: Li, W., Mollá-Aliod, D. (eds.) ICCPOL 2009. LNCS, vol. 5459, pp. 386–393. Springer, Heidelberg (2009)
18. Begum, R., Husain, S., Dhwaj, A., Sharma, D.M., Bai, L., Sangal, R.: Dependency annotation scheme for Indian Languages. In: Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP), Hyderabad, India (2008)