Developing an Automated Bangla Parts Of Speech Tagged Dictionary

Sabir Ismail, M. Shahidur Rahman, Md. Abdullah Al Mumin
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
sabir-cse@sust.edu, rahmanms@sust.edu, mumin-cse@sust.edu

Abstract— **This paper develops an algorithm for making an automated Bangla Parts Of Speech (POS) tagged dictionary. Natural Language Processing is one of the most vigorous research areas of computer science. It enables to communicate and retrieve information form computer based system more effectively and efficiently. Researches on Bangla language processing have started long back. However, this research area still suffers from resource scarcity. A POS tagged corpus is a cardinal element for language processing. POS tagging is the process of categorizing a particular word to a particular part of speech or syntactic category. In Bangla, we do not have any large POS tagged dictionary. In this paper we develop an automated way to make a POS tagged dictionary of Noun, Verb and Adjective. Initially, a suffix (or postfix) list is created manually for Bangla language. Based on this suffix list the POS tagged dictionary is developed. The proposed algorithm is evaluated using a paragraph consisting of manually tagged 10,000 words with 11 tags. We found that POS tagging is obtained more accurately for Verb than Noun and Adjective.**

Keywords— Bangla Language Processing, Parts Of Speech Tagging, Machine Learning, Bangla Corpus.

I. INTRODUCTION

Natural Language Processing [1-2] comprises large area of researches like information retrieval, document categorization, machine translation, question answering, human-computer interaction, automatic summarization, coherence resolution, discourse analysis, morphological segmentation, naming entity recognition and so on. Natural language processing starts with separating each word from a sentence. These individual words are further processed for more advance analysis. The mere individual words do not contain enough information for further processing. POS tag of word helps profound parsing of text, semantic processing and morphological analysis. There are various ways for POS tagging such as supervised, semi supervised or unsupervised methods. In supervised methods there are fixed rules for tagging. Unsupervised methods used machine learning techniques like Hidden Markov Model (HMM), Maximum Likelihood Estimation and Maximum Entropy (ME) based tagging. Semi-supervised methods use both supervised and unsupervised methods. In this paper, we propose a semi-supervised method to develop POS tag dictionary for noun, verb and adjectives. There is no large size POS tag dictionary currently available for Bangla. This POS tag dictionary can be employed for word categorization, word sense disambiguation, morphological analysis of verb and noun, analyse suitable POS tag set for Bangla. It can be used as training corpus for unsupervised models as well.

The rest of the paper is organized as follows. We start with an overview of previous work on Bangla corpus and Bangla POS tagging in Section II. Then we discuss Bangla language and use of parts-of-speech in Bangla grammar in Section III. Section IV describes the proposed algorithm and Section V analyses the results. Finally we conclude the paper in Section VI.

II. RELATED WORKS

Bangla language is the 7th most spoken language [3] in the world. Compared to other languages like English, Chinese, German, and French, Bangla has scarcity of large corpus. Different types of corpuses are available for those languages. Unfortunately, effort for building a rich POS tagged dictionary of Bangla has just started. The Centre for Research on Bangla Language Processing of BRAC University has conducted some works. They proposed a way to implement a corpus by collecting data form online resources [4]. They implemented POS tagger based on HMM, n-gram and Brill's tagger [5]. The result is analized with a small corpus of 5000 words giving accuracy of only 55%. Since Bangla language is a head-final specifier-initial language, another study shows that backward n-gram gives better result than forward n-gram [6]. Ansari, Selim and Iqbal [7] also make some statistical analysis on Bangla Academy Dictionaries corpus. Their corpus also contains only 150 tagged words. Hammad Ali [8] developed Baum-Welch trained HMM tagger for the Bangla POS tagging. His aim was to check whether rule based taggers was working better than stochastic taggers. The author used 41 tag sets and 50,000 tokens and outcome shows that rule based tagger is more effective than tagger. Sandipan et al [9] build an automatic POS tagger using HMM, ME based stochastic taggers. They also used morphological and contextual information of the words to improve performance. They summarized that morphological features are helpful to develop Bangla POS tagger when tagged resources are limited. Asif et al [10] proposed a Bangla POS tagger using the statistical Conditional Random Fields and they also conclude that different contextual information of the words along with the variety of features are helpful in predicting POS tag. They also developed a ME based tagger [11] with a training corpus of 72,341 words and 26 tag sets. They compare result with HMM and ME outperforms the HMM based tagger. They also added that lexicon, naming entity recognizer and different word suffixes are effective in handling unknown word tagging problems. Another attempt by Asif and Sivaji [12] is lexicon development and POS tagging by HMM with approximately 31,190 word forms and 5967 manually tagged corpuses. They achieved 83.04% accuracy and accuracy increased up to 91.6% with inclusion of different techniques adopted for handling unknown words. Sandipan [13] used ME based static parser for Bangla, Hindi and Telugu, and also used morphological analyzer for Bangla that led better result for Bangla tagger than Hindi and Telugu. Another layer based POS tagging was attempted by Chakrabarti [14], he used a rule based tagger and 4 layers of chunking to handle multi-verb exception and reduced ambiguous tagging. Here contextual and morphological cue plays vital role in different layers. All of them used a manually created tagged corpus for training semi-supervised and unsupervised model or used rule based tagger, but no attempt is made to develop a large tagged dictionary.

III. BANGLA LANGUAGE

Bangla Language arose from eastern Middle Indo-Arayn dialects of the Indian subcontinent, which is now the main language of Bangladesh and some western parts of India. Bangla language has rich grammatical rules including Parts-of-speech (পদ), Case (কারক), Suffix (বিভক্তি), Prefix (উপসর্গ), Number (বচন), Gender (লিঙ্গ), Person (পুরুষ), Tense (কাল), Primary Suffix (কৃৎ প্রত্যয়) and Secondary Suffix (তদ্ধিত প্রত্যয়) [17-22]. The structure of Bangla language is quite different form English language. Bangla words are highly inflectional particularly for verb and noun. A Bangla verb has more than 200 inflectional forms based on tense, aspect, mood and person. Mostly those words are formed by adding suffix with some exceptions. Same is true for noun. Based on case, gender and number, noun also has several inflectional forms. This makes vast variation of Bangla words. However, this highly inflection of word has also some benefits. We use this inflection of words to build POS tag dictionary.

POS tagging is the process of tagging every word in a text to a particular parts of speech based on relation with neighboring words and phrase. English language has mainly 8 parts of speech: Noun, Pronoun, Verb, Adverb, Adjective, Preposition, Conjunction and Interjection. However, number of tags varies from corpus to corpus when tagging POS. English language, for example, has several corpuses where every corpus has its own tag set. The Brown Corpus has 87 tag set, the Penn Treebank has 45 and the C7 has 146 tag set [15]. Bangla language has only five parts of speech: Noun, Pronoun, Adjective, Verb and Conjunction. Due to absence of large corpus, the number of tag set for Bangla is not well defined. In this paper, we work only with Noun, Verb and Adjective. The *root word* is a primary lexical unit of a word, which contains most significant information about semantic meaning and cannot be reduced to smaller part. Sometimes, the term *root word* is also used to describe the word with its inflectional endings discarded, but with its lexical endings in place. This *root word* is also called lemma or stems and inflectional endings are called suffix. In Bangla language a word can have more than one suffix or inflectional endings added to it. Our proposed method uses this suffix or inflectional ending to determine the POS tag.

IV. DEVELOPING BANGLA POS TAG DICTIONARY

As discussed earlier, Bangla words are highly inflectional. Almost every word has more than one inflectional form. We use this inflectional form to determine the POS tag. First, a list of Bangla suffix [18-22] is created from various resources on Bangla grammars. This list contains both the Primary Suffix (কৃৎ প্রত্যয়) which are added with verb and Secondary Suffix (তদ্ধিত প্রত্যয়) which are added with noun. This suffix list also contains some suffixes which are used to change number (বচন), gender (লিঙ্গ) and some suffixes are used to change POS tag of a word. We listed about 500 suffixes. Then we collect large number of words from various sources, like online newspapers, blogs and other web sites [23-27]. Primarily, total number of words collected are about 1000000. After eliminating the duplicate words the number is reduced to 320443 unique words. Then the words are sorted according to the Unicode values of Bangla characters [16]. As data structure, a hash table (also called hash map) is used that map the keys to values for storing data.

A. The Proposed Algorithm:

This algorithm is described here for a number of words and suffixes as given in Table 1 and Table 2.

1. Initially we introduced three hash tables: Hash map-1, Hash map-2 and Hash map-3. Hash map-1 (Key: root word, Value: suffix) contains the root word as *Key* and suffixes as the *Values.* Hash map-2 (Key: root word, Value: inflectional form) stores the root word as *Key* and the inflectional form of that root word as *Values*. Hash map-3 (Key: suffix, Value: root word) contains suffix as the *Key* and the root words as *Values*. Initially all the hash maps are set empty.

TABLE 1. INITIAL WORD LIST

Word List
অংশ,অংশই,অংশও,অংশকে,অংশগুলির,অংশগুলো,অংশগুলোও,অংশ অংশগ্রহণ,অংশগ্রহণও,অংশগ্রহণকারী,অংশগ্রহণকারীদের,অংশগ্রহণব অংশগ্রহণকারীরা,অংশগ্রহণকারীরাই,অংশগ্রহণকে, অংশগ্রহণরত,অংশগ্রহণে,অংশগ্রহণের

TABLE 2. SUFFIX LIST

Suffix list:
◌া, ◌ি, ই, ও, কে, গুলি, গুলির, গুলো, গুলোও, গুলোতে, দের, র, রা, রাই, কে, ত, ◌ে, ◌ের

2. Then for every word in the word list we check if the starting of a word matches with any root word stored in Hash map-1 (Key: root word, Value: suffix) then we chop the root part from that word and retrieve rest of the characters.

3. If rest of the characters are found in the suffix list that indicates the word is generated from that root word, then this word is added in Hash map-1, Hash map-2 and Hash map-3. The example 1 as given below explains the case.

4. If a word matches with the root word more than once then the longest match is considered. Again, if a word matches with the root word but there are some other words that starts with this word, then this word itself is a root word. Explanation is given in example 2.

Example 1: অংশ is the first word in our word list, so this is a root word which is stored in hash map. Table 3 shows the content of hash map.

TABLE 3. CONTENT OF THE HASH MAPS AFTER PROCESSING THE FIRST WORD অংশ

Hash map-1 (Key: root word, Value: suffix)	Hash map-2 (Key: root word, Value: inflectional form)	Hash map-3 (Key: suffix, Value: root word)
Key: অংশ, Value: root	Key: অংশ, Value: root	Key: root, Value: অংশ

The next word is অংশই, now first we check in the Hash map-1 (Key: root word, Value: suffix) if there is any word which is a substring and also a prefix of অংশই. In this case, অংশ matches both the conditions. Then we chop rest of the characters from অংশই which is ই and check whether it is in suffix list. In this case it is found and অংশই is asserted as an inflectional form of অংশ and the hash maps are updated as shown in Table 4.

TABLE 4. CONTENT OF THE HASH MAPS AFTER PROCESSING FIRST TWO WORDS অংশ, অংশই

Hash map-1 (Key: root word, Value: suffix)	Hash map-2 (Key: root word, Value: inflectional form)	hash map3 (Key: suffix, Value: root word)
Key: অংশ, Value: root, ই	Key: অংশ, Value: root, অংশই	Key: root, Value: অংশ Key: ই, Value: অংশ

Example 2: Let us now assume the word অংশগুলো in the word list. Before we start processing the word অংশগুলো, the words অংশ, অংশই, অংশও, অংশকে, অংশগুলির are already processed and Table 5 shows the content of the hash maps.

TABLE 5. CONTENT OF HASH MAPS BEFORE PROCESSING THE WORD অংশগুলো

Hash map-1 (Key: root word, Value: suffix)	Hash map-2 (Key: root word, Value: inflectional form)	Hash map-3 (Key: suffix, Value: root word)
Key: অংশ, Value: root, ই, ও, কে, গুলির	Key: অংশ, Value: root, অংশই, অংশও, অংশকে, অংশগুলির	Key: root, Value: অংশ Key: ই, Value: অংশ Key: ও, Value: অংশ Key: কে, Value: অংশ Key: গুলির, Value: অংশ

5. Now অংশগুলো matches with root word অংশ and গুলো is a valid suffix, but there are some more words (অংশগুলোও, অংশগুলোতে) also start with অংশগুলো. In this case অংশগুলো is considered as a root word. Table 6 shows content of the hash map after processing the word অংশগুলো

TABLE 6. CONTENT OF THE HASH MAPS AFTER PROCESSING THE FIRST WORD অংশগুলো

Hash map-1 (Key: root word, Value: suffix)	Hash map-2 (Key: root word, Value: inflectional form)	Hash map-3 (Key: suffix, Value: root word)
Key: অংশ, Value: root, ই, ও, কে, গুলির Key: অংশগুলো, Value: root	Key: অংশ, Value: root, অংশই, অংশও, অংশকে, অংশগুলির Key: অংশগুলো, Value: root	Key: root, Value: অংশ, অংশগুলো Key: ই, Value: অংশ Key: ও, Value: অংশ Key: কে, Value: অংশ Key: গুলির, Value: অংশ

Table 7 shows the content of the hash map after processing all the words in the wordlist.

TABLE 7. CONTENT OF THE HASH MAPS AFTER PROCESSING ALL WORDS IN THE WORDLIST

Hash map-1 (Key: root word, Value: suffix)	Hash map-2(Key: root word, Value: inflectional form)	Hash map-3 (Key: suffix, Value: root word)
Key: অংশ, Value: root, ই, ও, কে, গুলির Key: অংশগুলো, Value: root, ও, তে Key: অংশগ্রহণ, Value: root, ও, কে, ◌ে, ◌ের Key: অংশগ্রহণকারী, Value: root, দের Key: অংশগ্রহণকারীর, Value: root, ◌া, ◌াই Key: অংশগ্রহণরত, Value: root	Key: অংশ, Value: root, অংশই, অংশও, অংশকে, অংশগুলির Key: অংশগুলো, Value: root, অংশগুলোও, অংশগুলোতে Key: অংশগ্রহণ, Value: root, অংশগ্রহণও,অংশগ্রহণকে, অংশগ্রহণে, অংশগ্রহণের Key: অংশগ্রহণকারী, Value: root, অংশগ্রহণকারীদের Key: অংশগ্রহণকারীর, Value: root,অংশগ্রহণকারীরা, অংশগ্রহণকারীরাই Key: অংশগ্রহণরত, Value: root	Key: ◌ের, Value: অংশগ্রহণ Key: ◌ে, Value: অংশগ্রহণ Key: ই, Value: অংশ Key: ও, Value: অংশ, অংশগুলো, অংশগ্রহণ Key: কে, Value: অংশ, অংশগ্রহণ Key: গুলির, Value: অংশ Key: তে, Value: অংশগুলো Key: দের , Value: অংশগ্রহণকারী Key: ◌া, Value: অংশগ্রহণকারীর Key: ◌াই, Value: অংশগ্রহণকারীর Key: root, Value:

		অংশ, অংশগুলো, অংশগ্রহণ, অংশগ্রহণকারী, অংশগ্রহণকারীর, অংশগ্রহণরত

6. We analyse data stored in the Hash map-3 (Key: suffix, Value: root word) and it is revealed that some suffixes are very reliable candidate for POS tagging a word. For example, গুলি, টি, তে suffixes are secondary suffix and are added with only noun. Similarly ছিল, িব, িতেছিল suffix are primary suffix and are added with only verb. We, therefore, categorize the suffixes as verb, noun and adjective. We tagged nouns as 'N', verb as 'V', adjective as 'A' and some suffixes are not suitable for detecting POS tag which are tagged as 'U' as shown in Table 8.

TABLE 8: POS TAGGED SUFFIX LIST

Suffixes with POS tag					
িয়াছ V িয়াছি V িয়াছিলাম V কারী N তর A	কি U কী U কুল U তা A তাছি V	গুলিতে N গুলিন N গুলির N গুলো N তম A	তাম V তার V তি U তিও U তাও U	তিছি V তিছেন V তিস V কুলের N কে N	তুত V তুম V তাছেন V গুলি N গুলিকে N

Then we create two more hash maps.

- Hash map-4 (Key: root word, Value: POS tag) from Hash map-1 (Key: root word, Value: suffix) , by replacing suffixes with POS tag of Table 8 as shown in Table 9
- Hash map-5 (Key: root word, Value: inflectional form + POS tag) from Hash map-2 (Key: root word, Value: inflectional form), by replacing suffixes with POS tag of Table 8 as shown in Table 10.

TABLE 9. HASH MAP WITH POS TAG

Hash map-1 (Key: root word, Value: suffix)	**Hash map-4 (Key: root word, Value: POS tag)**
Key: অংশ, Value: root, ই, ও, কে, গুলির Key: অংশগুলো, Value: root, ও, তে Key: অংশগ্রহণ, Value: root, ও, কে, ে, ের Key: অংশগ্রহণকারী, Value: root, দের Key: অংশগ্রহণকারীর, Value: root, া, াই Key: অংশগ্রহণরত, Value: root	Key: অংশ, Value: root, U, U, N, N Key: অংশগুলো, Value: root, U, N Key: অংশগ্রহণ, Value: root, U, N, U, N Key: অংশগ্রহণকারী, Value: root, N Key: অংশগ্রহণকারীর, Value: root, U, N Key: অংশগ্রহণরত, Value: root

TABLE 10. HASH MAP WITH POS TAG AND INFLECTIONAL FORM

Hash map-2 (Key: root word, Value: inflectional form)	**Hash map-5 (Key: root word, Value: inflectional word and POS tag)**
Key: অংশ, Value: root, অংশই, অংশও, অংশকে, অংশগুলির Key: অংশগুলো, Value: root, অংশগুলোও, অংশগুলোতে Key: অংশগ্রহণ, Value: root, অংশগ্রহণও,অংশগ্রহণকে, অংশগ্রহণে, অংশগ্রহণের Key: অংশগ্রহণকারী, Value: root, অংশগ্রহণকারীদের Key: অংশগ্রহণকারীর, Value: root, অংশগ্রহণকারীরা, অংশগ্রহণকারীরাই Key: অংশগ্রহণরত, Value: root	Key: অংশ, Value: root, অংশই U, অংশও U, অংশকে N, অংশগুলির N Key: অংশগুলো, Value: root, অংশগুলোও U, অংশগুলোতে N Key: অংশগ্রহণ, Value: root, অংশগ্রহণও U, অংশগ্রহণকে N, অংশগ্রহণে U, অংশগ্রহণের N Key: অংশগ্রহণকারী, Value: root, অংশগ্রহণকারীদের N Key: অংশগ্রহণকারীর, Value: root, অংশগ্রহণকারীরা U, অংশগ্রহণকারীরাই N Key: অংশগ্রহণরত, Value: root

7. We eliminate multiple occurrences of same tag (e.g. multiple 'N' or 'U') form the Hash map-4 (Key: root word, Value: POS tag) as shown in Table 11. We remove 'U' tag, because our analysis on suffix list reveals that these suffixes cannot determine POS tag and they do not also change POS tag of a root word. Thus these inflectional words have same POS tag as root word.

TABLE: 11 HASH MAP WITH POS TAG FOR ROOT WORD

Hash map-6 (Key: root word, Value: tag)	
অংশ [root, N]	অংশগ্রহণকারী [root, N]
অংশগুলো [root, N]	অংশগ্রহণকারীর [root, N]
অংশগ্রহণ [root, N]	অংশগ্রহণরত [root]

Finally we regenerate all the words from Hash map-2 (Key: root word, Value: inflectional form) and Hash map-6 (Key: root word, Value: tag). Table 12 shows the POS tag dictionary.

TABLE 12. FINAL POS TAGGED DICTIONARY

POS tagged Dictionary		
অংশ [N] অংশই [N] অংশও [N] অংশকে [N] অংশগুলির [N] অংশগুলো [N] অংশগুলোও [N]	অংশগুলোতে [N] অংশগ্রহণ [N] অংশগ্রহণও [N] অংশগ্রহণকে [N] অংশগ্রহণে [N] অংশগ্রহণের [N]	অংশগ্রহণকারী [N] অংশগ্রহণকারীদের [N] অংশগ্রহণকারীর [N] অংশগ্রহণকারীরা [N] অংশগ্রহণকারীরাই [N] অংশগ্রহণরত [U]

V. EXPERIMENT AND RESULT ANALYSIS

We have collected more than 10,00,000 words from online Bangla newspapers, blogs and other Bangla web sites and 3,20,443 words are extracted as unique. The proposed technique successfully tagged 1,34,749 nouns, 11,067 verbs, and 8,435 adjectives. Some POS has very small amount of words, compare with noun or verb. Example: pronoun, conjunction, interjection, *Wh* question, negative, preposition has small amount of words. So our algorithm do not try to predict those tag set, rather we make a list of words having those tag set. The number of words are increasing every day that will further improve the accuracy of POS tag. We evaluate the proposed algorithm using a paragraph consisting of manually tagged 10,000 words with 11 tags as shown in Table 13.

TABLE 13. POS TAG LIST FOR TAGGED CORPUS

POS	Tag	POS	Tag
Noun	NN	Adverb	RB
Pronoun	PRP	Adjective	JJ
Verb	VB	Postposition	PSP
Auxiliary Verb	VA	Conjunction	coNN
Wh question	WH	Interjection	INT
Negative	NEG	Unknown	U

Table 14 shows the Confusion Matrix resulted from the experiment where the horizontal tags represent the actual and the vertical tags represent the obtained tags. We work for Noun (NN), Verb (VB), Adjective (JJ), unknown tag (U) and others are tagged as (O). A number in a particular cell represents the number of tags obtained using the proposed method with respect to the actual tag. For example 3201 NN tags are determined successfully 2589 times.

TABLE 14. CONFUSION MATRIX OBTAINED USING A MANUALLY TAGGED CORPUS OF 10000 WORDS

		Actual POS tag											
POS tag obtained using proposed algorithm		VB (2296)	WH (51)	coNN (125)	NEG (135)	NN (3201)	U (0)	VA (763)	INT (86)	RB (312)	PRP (1298)	JJ (966)	PSP (0)
	VB	2036	0	0	0	224	0	711	0	6	0	5	0
	NN	219	0	0	0	2589	0	14	3	5	12	513	0
	JJ	9	0	1	0	39	0	8	0	0	0	311	0
	U	3	2	2	2	26	767	4	77	45	29	81	0
	O	29	49	122	133	123	0	26	6	256	1257	56	0

Table 15 shows the accuracy of the proposed algorithm.

TABLE 15. OVERALL PERFORMANCE

Tag	True Positive	False Positive	False Negative
Verb (VB)	2747	235	260
Noun (NN)	2589	766	412
Adjective (JJ)	311	57	655

Table 15 shows that POS tagging is obtained more accurately for both Verb (VB) and Auxiliary Verb (VA) (2036+711= 2747). But for Noun (NN), False Positive and for Adjective (JJ), False Negative is high. This is because Noun and Adjective tag are interchangeable for a word based on its contextual meaning and surrounding words. Here we do not consider surrounding words which are usually used in machine learning based tagging methods.

VI. CONCLUSION

In this paper, we have developed method for tagging noun, verb and adjective. The proposed algorithm is evaluated using a paragraph consisting of manually tagged 10,000 words with 11 different tags. We found that POS tagging is obtained more accurately for Verb than Noun and Adjective.

Resource is prerequisite to any research. Unfortunately, Bangla language still suffers from scarcity of resource. The resulted POS tag dictionary can play a vital role for future research on Bangla language processing specially on POS tagging and making a canonical tag set. Unsupervised model like HMM needs large tagged corpus for training data. The POS tag dictionary can serve this purpose.

REFERENCES

[1] A free encyclopedia built collaboratively using wiki software: https://en.wikipedia.org/wiki/Natural_language_processing :Last accessed 14th November, 2013

[2] Natural Language Processing: http://research.google.com/pubs/NaturalLanguageProcessing.html : Last accessed 14th November, 2013

[3] "Statistical Summaries". Ethnologue. Retrieved 2013-07-31. : Last accessed 14th November, 2013

[4] Sarkar, Asif Iqbal, Dewan Shahriar Hossain Pavel, and Mumit Khan."Automatic Bangla corpus creation". Center for research on Bangla language processing (CRBLP), BRAC University, 2007.

[5] Hasan, Fahim Muhammad, Naushad UzZaman, and Mumit Khan. "Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla." Advances and Innovations in Systems, Computing Sciences and Software Engineering. 2007.

[6] Khan, Naira, et al. "History (Forward N-Gram) or future (Backward N-Gram)? Which model to consider for N-Gram analysis in Bangla?" 2006.

[7] MD. Abdul Awal Ansary,Mohammad Reza Selim, Muhammad Zafar Iqbal. "Bangla Academic Dictionaries (BAD) Corpus with some Applications and Statistical Analysis". Journal of Emerging Trends in Computing and Information Sciences. VOL. 3, NO.11 Nov, 2012.

[8] Ali, Hammad. "An unsupervised parts-of-speech tagger for the bangla language." Department of Computer Science, University of British Columbia, 2010.

[9] Dandapat, Sandipan, Sudeshna Sarkar, and Anupam Basu. "Automatic part-of-speech tagging for Bengali: An approach for

morphologically rich languages in a poor resource scenario." Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, 2007.

[10] Chakrabarti, Ekbal, Asif, Rejwanul Haque, and Sivaji Bandyopadhyay. "Bengali part of speech tagging using conditional random field.” Proceedings of Seventh International Symposium on Natural Language Processing, 2007.

[11] Ekbal, Asif, Rejwanul Haque, and Sivaji Bandyopadhyay. "Maximum Entropy Based Bengali Part of Speech Tagging." RCS Journal 33, pp.67-78, 2008.

[12] Ekbal, Asif, and Sivaji Bandyopadhyay. "Lexicon Development and POS Tagging Using a Tagged Bengali News Corpus." FLAIRS Conference, 2007.

[13] Dandapat, Sandipan. "Part of specch tagging and chunking with maximum entropy model." Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages, Hyderabad, India, 2007.

[14] Chakrabarti, Debasri, and Pune CDAC. "Layered Parts of Speech Tagging for Bangla." Language in India, www. languageinindia. com, Special Volume: Problems of Parsing in Indian Languages, 2011.

[15] Daniel Jurafsky, James H. Martin – “Speech and Language Processing”(2nd Edition), Page: 8; Chapter: 5, 2008.

[16] Amin, Md Ruhul, Asif Mohammed Samir, Madhusodan Chakraborty, and Mahfuzur Rahaman. "An Efficient Unicode based Sorting Algorithm for Bengali Words." *International Journal of Computer Applications* 24, no. 7, 2011.

[17] প্রফেসর ড. আলাউদ্দিন আল আজাদ, ড. মনন অধিকারী, রুহুল আমীন বাবুল - “উচ্চতর বাংলা ভাষারীতি” লেকচার পাবলিকেশন, ২০০৮ ইং .

[18] নবম-দশম শ্রেনী ২০০০ - জাতীয় শিক্ষাক্রম ও পাঠ্যপুস্তক বোর্ড - “বাংলা ব্যাকরন”, ২০১১ ইং .

[19] ড. মুহম্মদ শহীদুল্লাহ - “বাংলা ব্যাকরন”, মাওলা বার্দ্রাস, ১৪১০ বাং .

[20] ড. হুমায়ুন আজাদ - “বাক্য তত্ত্ব”, আগামী প্রকাশনী, ২০১০ ইং .

[21] মহাম্মদ দানীউল হক - “ভাষা বিজ্ঞানের কথা” , মাওলা বার্দ্রাস, ২০১১ ইং .

[22] উদয়কুমার চক্রবর্তী - “ বাংলা পদগুচ্ছের সংগঠন” , দে'জ পাবলিশিং কলকাতা .

[23] মুহম্মদ জাফর ইকবাল - “আমার বন্ধু রাশেদ”, কাকলী প্রকাশনী, ১৯৯৮ ইং .

[24] বাংলা ব্লগ: http://www.somewhereinblog.net/ : Last accessed 14th November, 2013.

[25] Bangla Newspaper, প্রথম আলো: http://www.prothom-alo.com/ : Last accessed 14th November, 2013.

[26] Bangla Newspaper, কালের কণ্ঠ: http://www.kalerkantho.com/ : Last accessed 14th November, 2013.

[27] অনলাইন বাংলা নিউজ : http://www.rtnn.net/ : Last accessed 14th November, 2013.