Detection and Suggestion English Grammar Errors for Myanmar-English Machine Translation System

Nay Yee Lin, Khin Mar Soe, Ni Lar Thein
University of Computer Studies, Yangon
nayyeelynn@gmail.com

Abstract

This paper presents a system for detection and suggestion English grammar errors in Myanmar-English statistical machine translation system. There are very few spelling errors in the translation output, because all words come from Myanmar-English Bilingual corpus. However, the translated sentences might be incomplete in grammar because the syntactic structures of Myanmar and English language are totally different. Therefore, we propose a chunk-based grammar checker by using Naïve Bayesian theory and rule-based model. Context free grammar (CFG) is also applied to make chunk-based sentence pattern. The syntactic chunk structure of a sentence is used to recognize grammatical relations of chunks. This system evaluated the performance of detection and suggestion for simple, compound and complex sentence types. The effectiveness of the system can be confirmed through the experimental results.

1. Introduction

Grammar checkers check the grammatical structure of sentences based on morphological processing and syntactic processing. These two steps are part of natural language processing to understand the natural languages. Morphological processing is the step where individual words are analyzed into their components and non-word tokens such as punctuation. Syntactic processing is the analysis where linear sequences of words are transformed into structures that show grammatical relationships between the words in the sentence [8].

Grammar is the set of structural rules that govern the composition of clauses, phrases, chunks and words in any given natural language. Grammar checking is one of the most widely used tools within natural language processing (NLP) applications. Three methods are widely used for grammar checking in a language; syntax-based checking, statistics-based checking and rule-based checking. In *syntax based grammar checking*, each sentence is completely parsed to check the grammatical correctness of it. In *statistics-based approach*, POS tag sequences are built from an annotated corpus, and the frequency, and thus the probability, of these sequences are noted. In *rule-based approach*, it is very similar to the statistics based one, except that the rules must be handcrafted [6].

Syntax or the patterns of language defines the structure of sentence and recognizes the grammar rules. This system considers the syntactic structure of the sentence to check grammar and limits the detection of the semantic errors.

The rest of this paper is organized as follows. Section 2 summarizes the previous efforts related to grammar checking. Section 3 describes the proposed chunk-based grammar checker. Section 4 explains about English sentence structures. Section 5 discusses types of errors and the

experimental results are illustrated in section 6. Finally section 7 concludes the paper.

2. Related Work

This section discusses some related works of grammar checking for various languages.

Anuradha Sharma and Nishtha Jaiswal [1] proposed a system to reduce errors in Translation using Pre-editor for Indian English Sentences. A statistical grammar checker was developed by using n-gram based analysis of words and POS tags to decide whether the sentence is grammatically correct or not [10].

Authors [12] developed a grammar checking system for detecting various grammatical errors in Punjabi texts. Adriane [14] discussed an automatic diagnosis of written errors for beginning learners of German. Emi IZUMI [7] described a method of detecting grammatical and lexical errors made by Japanese learners of English that improve the accuracy of error detection with a limited amount of training data.

In [5], a new grammar checker was specifically developed for the needs of French speakers writing in English by using finite-state automata. A grammar and style checker is also developed for Spanish in [9]. They presented for detection and diagnosis besides a brief grammar error typology for Spanish based on the generalized use of PROLOG extensions to highly typed unification-based grammars. Berthold Crysmann [3] presented a hybrid approach for grammar and style checking, combining an industrial pattern based grammar and style checker with bidirectional, large-scale HPSG grammars for German and English. Bibekananda Kundu [4] considered syntactic and semantic analysis of Bangla language for developing grammar checker system by using rule-based and statistical approach.

3. Chunk-based Grammar Checker

In Myanmar-English statistical machine translation system, our proposed system is concerned with a part of the target language model to solve distortion, deficiency and make the translated English sentences smooth. Input Myanmar sentence has been processed in three models (source language model, alignment model and translation model), the translated English sentence is obtained in target language model. This translated sentence can often be ungrammatical. Therefore, the central goal of this system is to develop a target-dominant grammar checker for Myanmar-English machine translation system.

The proposed system consists of three main parts:

1. Part-of-speech (POS) tagging
2. Chunk Identification
3. Detection and Suggestion Grammar Errors

3.1. Part-of-Speech (POS) Tagging

POS tagging is needed for syntactic analysis because the system can be used for both as a complement of Myanmar-English machine translation system and as a standalone grammar checker. If the sentence has already tagged POS tags, these POS tags may not be enough for making chunk types. Therefore, we need to retag and segment again the translated sentences.

Translated English sentence is used as an input. Firstly, the input sentence is tokenized and set the corresponding Part of Speech (POS) tag such as noun, verb, pronoun, preposition, adverb, adjective or other tags to each word. There are many approaches to automated part of speech tagging. In this system, each word is tagged by using Tree Tagger. Sample output of TreeTagger is shown as follow:

Word	POS	Lemma
The	DT	the
TreeTagger	NP	TreeTagger
is	VBZ	be
easy	JJ	easy
to	TO	to
use	VB	use
.	SENT	.

Tree Tagger often fails to tag correctly some words when one word has more than one POS tag. For example, "*like*" can be a verb or a preposition such as "I *like* [VBP] candy." and "Time flies *like* [IN] an arrow.". POS tags are more vary depending on their location of a sentence. Therefore, the system also considers refining POS tags.

Refinement of POS tagging is based on the predefined rules for the positions of POS tags and root words of the consecutive neighbouring words. POS tag errors for some words are learnt at the training time. By comparing initial tagging output with manually annotated text, refinement rules of a certain pattern are learned to improve the quality (accuracy) of the POS tags.

Figure 1 illustrates how refinement of POS tagging learning works and some refinement rules of POS tagging are shown in Figure 2.

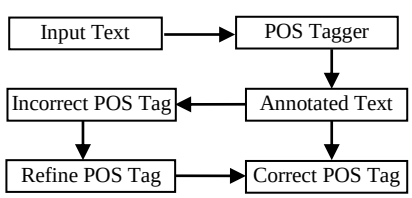

Figure 1. Refinement of POS Tagging

If previous tag is "**PP**" And current tag is "**NN**" And root word is "**bit**" Then change tag "**NN**" to "**VBD**"

If previous tag is "**DT**" And current tag is "**VB**" And root word is "**tailor**" Then change tag "**VB**" to "**NN**"

If previous tag is "**DT**" And current tag is "**JJ**" And root word is "**sweet**" Then change tag "**JJ**" to "**NN**"

If previous tag is "**PP**" And current tag is "**NN**" And root word is "**snow**" Then change tag "**NN**" to "**VBP**"

Figure 2. Sample Rules for Refinement of POS Tagging

3.2. Chunk Identification

Chunk identification is making groups of words that participate to form syntactic patterns. In this system, there are three levels of chunk identification: specific, common and general chunk types. *Specific chunk type* is the study of how POS tagged texts is combined to form meaningful chunks. It is marking up the chunks based on its definition and its context that is relationship with adjacent related words and POS tags. These specific chunk types are reconstructed to form *common chunk type* which is used for sentence patterns detection and analyzing chunk errors. These common chunk types are finally grouped into ten *general chunk types.* General chunk types are only used to know general sentence patterns. Descriptions and examples of these chunk types are illustrated in Appendix.

We identify the chunk types by using *Context-free grammar* (CFG). CFG's rules present a single symbol on the left-hand-side, are a sufficiently powerful formalism to describe most of the structure in natural language, while at the same time is sufficiently restricted as to allow efficient parsing. [15].

Chunking or shallow parsing segments a sentence into a sequence of syntactic constituents or chunks, i.e. sequences of adjacent words grouped on the basis of linguistic properties [16]. There are two methods for parsing such as Top-down parsing and Bottom-up parsing. In Top–down parsing, begin with the start symbol and attempt to derive the input sentence by substituting the right hand side of productions for non terminals. In Bottom–up (shift–reduce) parsing, begin with the input sentence and attempt to work back to the start symbol. Bottom-up parsers handle a large class of grammars [11].

The task of chunking is rule-based and relies on handcrafted rules written in formalism with a context-free backbone. This system can also know the long noun form or phrase by using CFG rules. There are about 600 rules for chunking. Chunks are identified using CFG based bottom-up parsing for assembling POS tags into higher level chunks, until a complete sentence has been found. Figure 3 describes the algorithm for chunk identification.

```
Let P={p1,p2,…,pm}be the set of POS tags in the input sentence
   S={ s1,s2, …,sn} be the set of specific chunk rules
   C={ c1,c2, …,ck} be the set of common chunk rules
        i =1                        x=1
        j =[i+1,i+2,…m]             y=[x+1,x+2,…,n]
Begin
     while ( pi is not equal to pm-1)
      do
       for each pj of POS tag P do
         combine pi and pj
         assign to pi
         if (pi exists in specific chunk rules) then Define pi as S
         end if
       end for
       i=j+1
     end while
     while ( sx is not equal to sn-1)
      do
       for each sy of specific chunk S do
         combine sx and sy
         assign to sx
         if (sx exists in common chunk rules) then Define sx as C
         end if
       end for
       x=y+1
     end while
End
```

Figure 3. Algorithm of Chunk Identification

Figure 4 depicts an example of how to recognize a noun chunk "a lovely girl" by using chunk rules.

[AA1N1]	→	[NCB1]	Common Chunk Type
[DT_JJ_NN]	→	[AA1N1]	Specific Chunk Type
a[DT] lovely[JJ] girl[NN]	→	[DT_JJ_NN]	POS Tagging

Figure 4. An example of a Noun Chunk

3.3. Detection and Suggestion Grammar Errors

This grammar checker detects the common chunk-based sentence patterns by comparing trained sentence rules. We have currently trained 500 general sentence patterns for declarative, interrogative and imperative types. Sample sentences are shown in Table 1. From these general sentence patterns, about 12000 common chunk-based sentence patterns are obtained for detection. Some common chunk-based sentence patterns for declarative and interrogative are described in Table 2.

Table 1. Some General Sentence Patterns

General Chunk-based Sentence Patterns	Examples
NC_VC_END=S	He plays.
NC_RC_VC_END=S	He first goes.
NC_VC_NC_END=S	He plays football.
NC_VC_AC_END=S	He is fat.
QC_VC_NC_VC_END=S	What is he doing?
QC_NC_VC_NC_END=S	Whose pen is this?
QC_VC_NC_VC_END=S	What is he doing?
VC_RC_END=S	Come here.
VC_NC_END=S	Close the window.
VC_AC_END=S	Don't be careless.

Table 2. Some Common Chunk-based Sentence Patterns

Declarative	Interrogative
NCB_PRV1_NCB_END=S	QC_PAV2_NCB2_VC_IEND=S
NCB_PRV1_NCB1_END=S	QC_PAV1_NCB1_VC_IEND=S
NCB_PRV1_NCB2_END=S	QC_PAV1_NCB_VC_IEND=S
NCB_PRV1_NCO_END=S	QC_PAV2_NCB_VC_IEND=S
NCB1_PRV1_NCB_END=S	QC_PAV2_NCS2_VC_IEND=S
NCB1_PRV1_NCB1_END=S	QC_PAV1_NCS1_VC_IEND=S
NCB1_PAV1_NCB_END=S	QC_PAV2_NCS_VC_IEND=S
NCS2_PRV2_NCB2_END=S	QC_PRV1_NCB1_VC_IEND=S
NCS2_PRV2_NCO_END=S	QC_PRV2_NCB_VC_IEND=S
:	:

Detection accuracy is depended on the number of trained sentence patterns. The more trained sentence patterns can get better detection. Therefore, we have to be trained many sentence patterns in advance for different kinds of sentence types. If the sentence structure is grammatically incorrect, we analyze the errors by using Naïve Bayesian theory and rule-based model.

3.3.1. Naïve Bayesian Theory

Naive Bayesian theory is based on the simplifying assumption that the attribute values are conditionally independent given target value [17]. It can handle an arbitrary number of independent variables whether continuous or categorical. Given a set of n dimensional features, X = {x_1,x_2,...,x_n}, posterior probability is constructed for the event C_j among a set of possible outcomes (class) C = {c_1,c_2,...,c_i}.

$$P(c_i|x_1,x_2,...,x_n) = P(c_i) \times P(x_1,x_2,...,x_n|c_i) \quad (1)$$

$$P(C_i) = \frac{s_i}{s} \quad (2)$$

$$pc = arg_{c_i}\ max\ P(c_i|x_1,x_2,...,x_n) \quad (3)$$

Let $P(c_i|x_1,x_2,..,x_n)$ = Posterior probability
$P(c_i)$ = Prior probability
$P(x_1,x_2,\ldots,x_n|c_i)$ =Log Likelihood
pc = Maximum Posterior Probability
s_i =Number of training samples of class c_i
s =Total number of training samples

For analyzing chunk errors, we consider the features as consecutive previous two chunks and calculate the probability between possible chunks (c_1,c_2,...,c_i) and previous chunks (x_{i-2},x_{i-1}). s_i is the number of training chunk sequences of class C_i and *s* is the total number of training chunk types.

$$P(c_i|x_{i-2},x_{i-1}) = P(c_i) \times P(x_{i-1}|c_i) \times P(x_{i-2},x_{i-1}|c_i) \quad (4)$$

$$pc = arg_{c_i}\ max\ P(c_i|x_{i-2},x_{i-1}) \quad (5)$$

Possible chunks are disambiguated with the maximum number of the posterior probability (pc) by using equation (4) and (5). Bayesian classification is used for selecting the best possible chunk types to correct the sentence pattern.

3.3.2. Rule-based Model

Rule-based approach is more transparent: errors are easier to diagnose and debug. Grammatical rules describe sentence and phrase structures, and ensure the agreement relations between various elements in the sentence. English grammatical rules are developed to define precisely how and where to assign the various words in a sentence. In this system, different types of rules are used for refining POS tags, making chunks, detecting sentence patterns and suggesting possible words. The accuracy of the system can be increased by the product of the rule based model. Therefore, we need to construct more and more rules to get better performance.

Algorithm for detection and suggestion grammar errors of proposed system is shown in Figure 5.

Let TP =Translated chunk-based Sentence Pattern
RP = Reconstructed chunk-based Sentence Pattern
R={r_1,r_2, …,r_m}be the set of trained sentence rules
C ={ c_1,c_2, …,c_n}be the set of chunk types of TP
G ={ g_1,g_2, …,g_i}be the set of grammar rules
pc =possible chunk type

Detection:
for all rules r_m of R **do**
if R[r_m]= TP **then** return Correct
else return Incorrect
end if
end for

Suggestion for Incorrect:
for all chunk c_n of C **do**
$P(c_n|c_{n-2},c_{n-1}) = P(c_n) \times P(c_{n-1}|c_n) \times P(c_{n-2},c_{n-1}|c_n)$
end for
$pc = \arg_{c_n} \max P(c_n|c_{n-2},c_{n-1})$
reconstruct sentence RP with *pc*
for all rule r_m of R **do**
if R[r_m]= RP
then
return Correct
if G[g_1,g_2, ..,g_i]= *pc* **then** suggest possible word
end if
end if
end for

Figure 5. Algorithm of Proposed System

3.4. Evaluation of Proposed System

For example, after translating a Myanmar sentence "အလွန်လှပသော မိန်းကလေးတစ်ယောက်သည် ပံ

ပုံပြင်စာအုပ်တစ်အုပ်ကို ဖတ်နေသည်။”,“*A very beautiful girl are reading a story book.”* can be resulted for grammar checking. This sentence has subject -verb disagreement. It is transformed into chunk-based sentence pattern as shown in Figure 6.

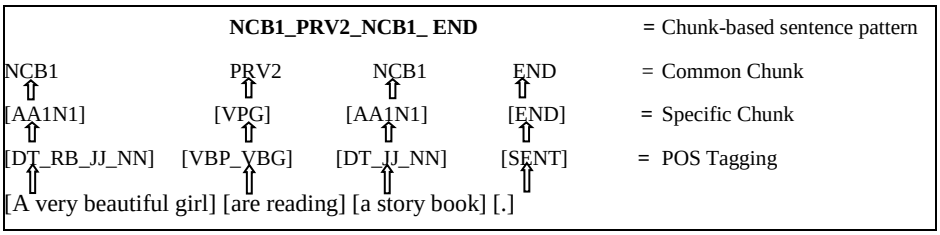

Figure 6. Making Chunk-based Sentence Pattern

After making chunk-based sentence pattern, we detect whether it is grammatically correct or not by using trained sentence rules. In this case, sentence pattern is incorrect. Therefore, we search for the chunks sequence P(PRV2/ none,NCB1) which has zero probability. Some prior probabilities for each chunk sequence are calculated as follows:

P(NCB1/none,none) = 0.14802
P(PRV1/none, NCB1) = 0.25758
P(PAV1/none, NCB1) = 0.22843
PAVC/none, NCB1) = 0.25342
P(FUVC/none, NCB1) = 0.26055
P(NCB/NCB1, PRV1) = 0.16597
P(NCB1/NCB1, PRV1) = 0.32780 …

Then we compute the likelihood $P(c_i/x_{i-2},x_{i-1})$ by multiplying $P(c_i)$, $P(x_{i-1}/c_i)$ and $P(x_{i-2},x_{i-1}/c_i)$ for each chunk:

P(NCB/NCB1,PRV1) =0.14802*0.25758*0.16597 =0.00632
P(NCB1/NCB1,PRV1) =0.14802*0.25758*0.3278 =0.01249
P(NCB2/NCB1,PRV1) =0.14802*0.25758*0.26970 =0.01028
P(NCO/NCB1,PRV1) =0.14802*0.25758*0.23651 =0.00901

After calculating the score of each chunk, we choose the possible chunk (PRV1) with maximum probability and fills up it in the sentence. Then, reconstructed sentence is checked whether the grammar is correct or not. If the sentence pattern is correct, we can suggest PRV1 is a correct chunk type. PRVI means Present Tense Verb chunk for Singular Noun according to common chunk rules. By this way, we can detect and suggest the grammar errors for translated English sentences.

4. Basic Sentence Structure

There are four basic different types of informational sentences in English according to the number of clauses they contain; simple, compound, complex and compound-complex.

A *simple sentence* consists of a single independent clause with no dependent clauses and expresses a complete thought. A *compound sentence* is made up of two or more independent clauses joined by a coordinator (*for, and, nor, but, or, yet, so*), conjunctive adverb (*however, therefore, thus, moreover, otherwise)* and semicolon. A *complex sentence* includes one or more independent clauses with at least one dependent clause, which cannot stand alone. A *compound-complex sentence* contains multiple independent clauses, at least two independent clauses and one subordinate clause.

In addition to the discipline, there are four kinds of sentences based on their purpose: declarative, exclamatory, interrogative and imperative: *A declarative sentence* or *declaration*, the most common type, commonly makes a statement. *An interrogative sentence or question* is commonly used to request information. *An exclamatory sentence or exclamation* is generally a more emphatic form of statement. *An imperative sentence or command* tells someone to do something [2].

This grammar checker analyzes the types of sentences: simple, compound and complex and also distinguishes the declarative, interrogative and imperative kinds of sentences.

5. Types of Errors

There are many English grammar errors to be corrected ungrammatical sentences. This grammar checker detects and provides the following errors based on the translated English sentences:

- **Chunk Omission-** If the sentence has missing chunks such as preposition (PPC), conjunction (COC) and existential (EX) then this system suggests the most appropriate chunk type. For example, if a sentence "*A lovely girl is drinking a cup water*." has missing preposition, possible chunk (PPC) can be suggested for "*A lovely girl is drinking a cup of water*." .
- **Subject-Verb Agreement**- is a syntactic constraint in English. In Subject-Verb agreement rule, verbs vary in form according to the person and number of the object. In this sentence "*They is my favorite authors*", subject and verb disagree in number. Therefore, possible verb chunk is suggested for "*They are my favorite authors*".
- **Inappropriate Determiner-** Translated English sentences can contain inappropriate determiner. For example, some nouns beginning with vowels (a,e,i,o,u), we have to change this determiner as "*an*".
- **Incorrect Verb Form -** Translated English sentences can have the incorrect verb form. The system has to memorize all of the commonly used tenses and suggest the possible verb form. For example, in the sentence "*She will gone to school*", "*will gone*" is wrong verb form. Therefore, the system suggests the correct verb form.
- **Missing Markers and Capital -** A sentence that makes a statement begins with a capital letter and ends with a period. If the input sentence has missing markers (./?), the system will add the full stop or question mark according to the sentence types such as interrogative or declarative sentences and change the words beginning with small letter.

6. Experimental Results

In our experiment, the system has classified the kinds of sentence such as simple, compound and complex and described whether the sentence type is interrogative, declarative or imperative. Tested sentences consist of simple, compound and complex sentence types for grammatically correct and incorrect. In simple sentences, the number of correct sentences is 350 and 300 for incorrect. Compound sentences consist of 320 for correct and 220 for incorrect. The number of correct sentences 290 and 230 incorrect sentences are tested for complex sentence type. Some correct and incorrect sample sentences for different sentence types are described in Table 3.

Performances of detection for all sentence types are fully correct except the input sentence patterns which are not contained in the training sentences. Detection accuracy mostly relies on trained sentence types and patterns. Suggestion accuracy of incorrect sentences for different error types are shown in Table 4. From these error types, chunk omission has the lowest accuracy for all sentence types because the trained sentence patterns may contain incorrect chunk sequence and it can be selected for suggestion. Suggestion accuracies of other error types are deal with the English grammar rules.

Table 3. Some Correct and Incorrect Sentences for Different Sentence Types

Sentence Types	Incorrect Sentences	Correct Sentences
Simple	A boy drinks a cup water.	A boy drinks a cup of water.
	is a book on the table.	There is a book on the table.
	They will gone to school.	They will go to school.
Compound	They was out but she is in.	They were out but she is in.
	He are at home but he is sick.	He is at home but he is sick.
	He is playing football and she is eating a apple.	He is playing football and she is eating an apple.
Complex	You walks quickly else you will not overtake him.	You walk quickly else you will not overtake him.
	I'd like to know why you attends English speaking class.	I'd like to know why you attend English speaking class.
	I don't know exactly when a man a woman came here.	I don't know exactly when a man and a woman came here.

Table 4. Suggestion Accuracy of incorrect sentences based on the types of Errors

Sentence Types	Types of Errors				
	Chunk Omission	Subject-Verb Agreement	Incorrect Verb Form	Inappropriate Determiner	Missing Capital and Markers
Simple	72%	78%	73%	80%	97%
Compound	65%	70%	68%	78%	94%
Complex	63%	68%	65%	76%	91%

7. Conclusion

This paper has presented what errors could be detected and suggested for each sentence types with experimental results. Context Free Grammar rules are used to identify the chunk-based sentence patterns. For checking sentence patterns, rule-based model is applied. In the past [13], we implemented a grammar checker by using Trigram language model. However, this approach cannot be able to suggest the most correct chunk type. Therefore, we use Naïve Bayesian which is an alternative method to disambiguate the task of finding an appropriate correct chunk type. By combining Naïve theory and rule-based model is more effective for detection and suggestion grammar errors. We expect that this system is a good complement for Myanmar-English machine translation system.

In the future, we will expand the sentence patterns and continue to study more English grammar rules to fully assess all sentence types.

References

[1] A.Sharma and N. Jaiswal, "Reducing Errors in Translation using Pre-editor for Indian English Sentences", *Proceedings of ASCNT*, CDAC, Noida, India, 2010, pp. 70-76.

[2] A. Turner, "English Solutions for Engineering and Science Research Writing: A guide for English learners to publish in international journals", Hanyang University, Seoul, Korea, December 2009.

[3] B. Crysmann, N.Bertomeu, P. Adolphs, D. Flickinger and T. Kluwer, "Hybrid processing for grammar and style checking", *Proceedings of the 22nd*

International Conference on Computational Linguistics (Coling 2008), Manchester, August 2008, pp. 153-160.

[4] B. Kundu, "Syntactic and Semantic Analysis of Bangla Language for Developing Grammar Checker System", CDAC Kolkata.

[5] C. Tschichold, F. Bodmer, E. Cornu, F. Grosjean and et al. "Developing a New Grammar Checker for English as a Second Language", Laboratory of Language Processing and Speech, University of Neuchatel, March 2000.

[6] D. Naber, *A Rule-Based Style and Grammar Checker* , Faculty of Engineering, University of Bielefeld, 2003.

[7] E.Izumi, K. Uchimoto and T. Saiga, "Automatic error detection in the Japanese learners' English spoken data", *ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Volume 2, ISBN:0-111-456789, 2003.

[8] E. Rich and K. Knight. *Artificial Intelligent*. Second edition. New York: McGraw Hill, Inc, 1991.

[9] F. R. Bustamante and F. S. Leon, "Gramcheck: A grammar and style checker", *Proceedings of the International Conference on Computational Linguistics* (COLING), 1996, pp.365-370.

[10] J. Alam, N. UzZaman and M. Khan, "N-gram based Statistical Grammar Checker for Bangla and English", Center for Research on Bangla Language processing, BRAC University, Dhaka, Bangladesh.

[11] K.D. Cooper, K. Kennedy and L. Torczon. *Bottom-up Parsing*, 2003.

[12] M. S. Gill and G. S. Lehal, "A Grammar Checking System for Punjabi", Department of Computer Science, Punjabi University, India, COLING 2008, Manchester, August 2008, pp. 149-152.

[13] N.Y. Lin, K.M.Soe and N.L.Thein, "Using a Grammar Checker for Detection Translated English Sentence", *9th International Conference on Computer Applications 2011*(ICCA 2011), Yangon, pp.237-243.

[14] R. Fox and M. Bowden, "Automatic Diagnosis of Non-native English Speaker's Natural Language", *14th IEEE International Conference on Tools with Artificial Intelligence* 2002 (ICATI 2002) , ISSN 1082-3409, 2002, pp. 301-306.

[15] S.Abney, *Parsing By Chunks*, Bell Communication Research, November 10, 1994.

[16] S. Abney, "Tagging and Partial Parsing". In: Ken, C., Steve, Y., Gerrit, B. (eds.) Corpus-Based Methods in Language and Speech. Kluwer Academic Publishers, Dordrecht, 1996.

[17] T. Mitchell. *Machine Learning*. McGraw-Hill, March 1, 1997.

Appendix

General Chunk Types	Common Chunk Types	Description	Specific Chunk Types	Example Words
NC	NCS1	Singular Noun Chunk for Subject only	PN1	He, She
	NCS2	Plural Noun Chunk for Subject only	PN2, PNC1	They, We all
	NCS	Singular and Plural Noun Chunk for Subject only	NEC	There
	NCB1	Singular Noun Chunk for both Subject and Object	ANN1, SN1, …	A boy, This book, his car
	NCB2	Plural Noun Chunk for both Subject and Object	ANN2, SN2, …	The boys, These girls, their cars
	NCB	Singular and Plural Noun Chunk for both Subject and Object	NDC, CD	This, These, Those, One
	NCO	Singular and Plural Noun Chunk for Object only	PN3,CDA, CDR,CDH	him, twelve years old, five o'clock
VC	PAVC	Past Tense Verb Chunk for both singular and plural noun	DV, VDN, VDNN	wrote, had given , had been taken
	PAV1	Past Tense Verb Chunk for singular noun only	VDZ,VDZG,VDZGN	was, was playing
	PAV2	Past Tense Verb Chunk for plural noun only	VDPG,VDP,VDPGN	were, were playing
	PRV1	Present Tense Verb Chunk for singular noun only	ZV,VZG,VZD	is, is writing, gives
	PRV2	Present Tense Verb Chunk for plural noun only	BV,PV,VPG,VPD	are, write, give
	FUVC	Future Tense Verb Chunk for both singular and plural noun	MVP,MVBG,MRVBN	will go, will be playing, will not be taken
TC	TC1	Time Chunk for Present Tense	ADV1	Now, Today
	TC2	Time Chunk for Past Tense	ADV2	Yesterday, last
	TC3	Time Chunk for Future Tense	ADV3	Tomorrow, next
COC	XC	Subordinated Conjunction Chunk	NPR,NDT	which, who, that
	CC	Coordinated Conjunction Chunk	CCOC,ACOC	and, but, or
INFC	INC	Infinitive Chunk with Noun	IAN1,IAN2,INN1	to the market, to the rivers, to school
	IVC	Infinitive Chunk with Verb	IBDV,IBNV,IBV,IPV	to be taken, to give
AC	AC	Adjective Chunk	R2A1,AC1,AC2,AC3	more beautiful, old, older, oldest
RC	RC	Adverb Chunk	R11,RC1	very hard, usually, quickly
PTC	PTC	Particle Chunk	PTC	up, down, back
PPC	PPC	Prepositional Chunk	PPC	at, on, in, under
QC	QC	Question Chunk	QDT, QP ,QP$,QRB	Which, Where, Whose, Who, What