

# 1 A Systematic Approach to English to Bangla Sentence Translator

2 Joyassree Sen<sup>1</sup>, Bappa Sarkar<sup>2</sup>, Md. Shamim Hossain<sup>3</sup> and Md. Nazrul Islam<sup>4</sup>

3 <sup>1</sup> Islamic University, Bangladesh

4 *Received: 15 December 2019 Accepted: 2 January 2020 Published: 15 January 2020*

---

## 6 Abstract

7 This paper deals with the design and development of an expert sentence translation system.  
8 In this translator, the source language is English, and the target language is Bangla. The  
9 implemented translation system determines the relationship among different forms of English  
10 and Bengali sentences and makes appropriate correspondence between English and Bengali  
11 grammar. Here, we have been developing a top-down parsing program. The system  
12 incorporates itself with the dictionary and gives the corresponding Bengali meaning. The  
13 system performs translation procedure in three steps. The lexical analyzer reads the English  
14 sentence, tokenizes into words, and stores information into a stack. The lexical analyzer uses  
15 the English to Bangla dictionary and word morphology for finding lexical information. The  
16 parser parses the input sentence and identifies the types of it and finds tense, phrase, clauses,  
17 etc. The generator generates a Bangla sentence, which is equivalent to the given input English  
18 sentence. It uses the output of the lexical analyzer and the parser to make Bengali sentence.  
19 This system can translate all kinds of sentences. But the limitation is that it cannot handle  
20 semantic and contextual problems

---

22 *Index terms*— MT (machine translator), parser, lexical analysis, stack, syntax, dictionary

## 23 1 Introduction

24 translator is a program, which reads the source language as input and translates it into target language as output.  
25 The goal is to design and build software that will take English contexts as inputs and then analyze, understand  
26 them and finally generate Bengali languages, so that eventually we will be able to address our computer as though  
27 we were directing another person. By far, the utmost human linguistic communication occurs as speech. Written  
28 language is a recent invention and still plays a less central role than articulate sound in most activities. But  
29 processing of written is more facile than speech language [1]. For example, the pronunciation of a word differs  
30 with person to person, but the structure or component of a word doesn't vary with persons. So the translation of  
31 the written form of language can be efficiently programmed. Parser plays a vital role in this translation process.  
32 A parser for a grammar (G) is a program that takes a string (W) as an input and produces a parse tree as  
33 an output [8]. There are two basic types of parsers, topdown parser, and bottom-up parser. We use top-down  
34 parsing. The general parsing process is being illustrated in the following figure. The Fundamental Architecture  
35 of the Proposed Machine Translation (mt) System

36 The basic architectural block diagram of the proposed MT system is depicted in the following figure. The  
37 system works in three steps: lexical analyzing, parsing, and Bengali sentence generation. The Lexical analyzer  
38 reads the English sentence, separates the words, and populates it with lexical information. After lexical analysis,  
39 all the words of a sentence and the resulting facts are stored into a stack for the parser. The parser uses a  
40 rule-based top-down parser to parse the input sentence for syntactic correctness. Finally, the generator produces  
41 Bangla Sentence from the parser outputs and the dictionary. The Lexical analysis phase of MT is the first step,  
42 which is implemented by the Lexical analyzer. It can scan the whole document at a time or one by one. The  
43 later strategy is better for parsing, although it is a slow process. Thus sentence scanning and parsing technique

44 have been used in the proposed system. It is looking for the sentence delimiter. It gathers all the information  
45 about the words. It tokenizes and sends the information to the parser and subsequently to the generator. During  
46 the execution of the Lexical analysis phase, the Lexical analyzer reads the input sentences from the keyboard or  
47 a text file given by the user and separates the words. To find the word in the Dictionary, the Lexical analyzer  
48 uses the word morphology techniques. For example, in any human language, a word is used in different form,  
49 and the dictionary contains only one form. In the lexicon, only the singular form is presented. It can generate  
50 its plural by using morphological techniques. The word morphology uses the following strategies:

51 1. Read the whole word and search it in the dictionary, proceed to the next word if it is found. 2. If dictionary  
52 is failed to match the word then check it for the proper noun. If it is proper noun then proceeds for the next  
53 word. 3. Discard the last letter from the word and apply the above two steps to the remaining words, if succeeds,  
54 check whether the discarded letters are a valid suffix for the recognized word. If it succeeded, it moves to the  
55 next word. 4. Repeat steps (iii) each time discarding the one letter from the end of the word until the word is  
56 recognized or the length of the word becomes zero, declare the word is not in the dictionary or invalid.

57 The above procedure considers only the suffix morphology: it can also incorporate prefix morphology for  
58 slandering prefixes. For using morphology, it requires to envisage what will be the meaning of the processed word  
59 and its part of speech.

### 60 2 b) Parsing English Sentence (Syntactic Analysis)

61 This translator model uses the lexical analyzer. The parser obtains a string of token and verifies the source  
62 language which can be generated by the grammar. And one can use any efficient algorithm for parsing. We use  
63 top-down parsing technique for this purpose [4]. This parser reports any syntax error in an intelligible fashion.  
64 And it should also recover from commonly occurs errors so that it can continue processing of its input. There are  
65 two strategies of error recovery, and they are called panic mode and phraselevel recovery. Now the parser knows  
66 the format of the English sentence. While parsing, the system determines and keeps some significant information  
67 such as number and person of the subject, type of verb, the format of preposition etc. That will be useful during  
68 the formation of the Bengali sentence.

### 69 3 i. Top-Down Parsing

70 A top-down parser starts by hypothesizing a sentence and gradually predicting lower level element until individual  
71 terminal symbols have been written. In other words, top-down parser attempting to find a leftmost derivation  
72 for an input string. It trying to build a parse tree from the root of the input and creating the nodes of the parse  
73 tree in order. For example, consider the grammar  $S \rightarrow NP VP$ ,  $NP \rightarrow N|P$ ,  $VP \rightarrow V$ ,  $N \rightarrow \text{Babu} | \text{cow} | \text{mango}$ ,  $P \rightarrow \text{You}$   
74  $| \text{I} | \text{He}$ ,  $V \rightarrow \text{eat} | \text{drink} | \text{walk}$  and the input  $w = \text{I eat mango}$ . A top-down parser is used to construct a parse tree  
75 of this sentence, initially creates a tree consisting of a single node labeled S. An input pointer points I, the first  
76 symbol of w. Then use the first production of S to expand the tree and obtain

77 The leftmost symbol of the tree is a nonterminal, so expand it with the production rule for NP and obtain  
78 Again, the leftmost symbol is a nonterminal, so expand it with the production rule for P and obtain the following  
79 tree. Now the leftmost symbol of the tree is a terminal; compare it with the word pointed by the pointer, which is  
80 I and does not match. Then, go back to P and see whether there is another alternate for P, which might produce  
81 a match. Now, using the second alternative of P, replace the terminal you with terminal I, and find a match as  
82 shown. Then, forward the pointer to the next symbol of w, eat and go to the next leftmost nonterminal symbol  
83 of the tree, which is VP and expand it with its production rule. The last symbol inserted is not a terminal, which  
84 is a nonterminal V and expand it using its production rule and obtain following tree representation.

85 Then, forward the pointer to the next symbol of w, mango and go to the next leftmost nonterminal symbol of  
86 the tree, which is NP and expand it with production rule. The last symbol inserted is a nonterminal, so expand  
87 this using its production rule and obtain following tree representation. Now, the last inserted word is a terminal  
88 symbol, the pointer reaches at the end of the input sentence w, and complete parsing process.

### 89 4 c) Intermediate Representation

90 After parsing, the system knows the structure of the given English sentence. For each formation of grammars,  
91 the system also has the structure in the database that compares the Bengali pattern of the corresponding English  
92 structure. With the structure the system now translates the input English sentence into a converted form. It is  
93 the intermediate representation of the sentence.

### 94 5 d) Translating into Bangla

95 The next step is to perform the translation. In this phase, the system fetches the Bengali meaning for each token  
96 from the dictionary. The Bengali meaning for each noun or pronoun or adjective or adverb is replaced directly.  
97 But in the case of preposition and article, artificial intelligence is applied for the appropriate Bangla meaning.

---

## 6 i. Modification of meaning

We had to modify the meaning of the words according to different kinds of criterions such as verbs, articles, and prepositions [10]. In this section, we present different modifications to the meaning of the words.

### 7 a. Verb

The need for the verb table is due to the differences between the form of the verbs in English and Bengali languages. In English, there are four kinds of forms of each verb, but in Bengali, there are more than twenty-seven kinds of forms for each verb. So it becomes too difficult to find out the appropriate form of Bengali meaning. For this reason, we have to use a reliable method, which can create a direct link between the English and Bengali form of verb. For the present form of verb there are three categories in the meaning of the verb. Those are: Our first job for any language conversion is to design and build a database to work as a dictionary. For that, we first determine the properties of any word which are required to understand the use of it in any kind of sentence. We needed a primary key column that contains the characteristics if any word which will be unique. And that required property is the word itself. And now we need a few columns to store various meanings of the word; the maximum meaning can occur to only verbs. Verbs have various meanings for the different person of the subject. They may also have special meanings that are required after modals or as gerunds. So we created four columns to hold those meanings and took two columns for containing their parts of speech and their type. We also create a column named person to store the person of word if it is used as a subject. We also created two columns for holding the antonyms and synonyms of the word. And then, we inserted the words and their properties in the table. We use MySQL Query Browser to perform those actions.

We create a function named 'find' to return particular properties of certain words. While calling that function, we passed a string and two integers as parameters. The string contains the word to match with the dictionary. First integer specifies from which column we search the word to match. The other integer specifies contexts of which column will return to function. A different column holds different properties like meanings, parts of speech etc, of the word.

### 8 b) User input to the Database

While translating, we may find a word that doesn't exist in our dictionary. In that case, we ask the user to give the meaning and other properties of the function. We simply call another window page to take the input and have passed the inputs into the 'insert' function. The insert function takes the properties as parameters, builds a query statement, and then calls 'execute update' function passing that statement as a parameter. The function executes that query and inserts the properties.

IV.

## 9 Algorithm

Here we will see the general procedure of our translation task.

### 10 Algo\_Translation()

Step-1: Take and input English sentence from the user.

Step-2: Split the sentence into word.

Step-3: For each word do step 4

Step-4: From dictionary find the appropriate meaning, parts of speech, type(whether the subject, object or none), person (if subject or object)

Step-5: From above, find the subject, the verb, and the object of the sentence.

Step-6: Determine the structure of the sentence by the placement of subject, object, verb, person of subject etc.

Step-7: Put the word meanings as an order by which a corresponding Bangla Structure of the English Sentence Formed.

Step-8: Show the Bangla sentence at the user interface in any Bangla font} i. Dividing the sentence into words ii. Checking if any word doesn't exist in the Dictionary iii. Setting and saving parts of speech and types of words.

2. Finding the primary subject of the sentence.

3. Determining the type of sentence. 4. Putting the appropriate meaning in order.

### 11 a) Pre-processing of the sentence before actual translating

We divide the sentence using the split function and store the words in a string array. Then calling the 'find' function for each word we first see whether it exist in the dictionary or they are some processed form of a word that exists in the dictionary. We pass the word to a function named 'match' to check some modified form of an existing word (such as eats is a modified form of eat). By using this function we re-modify the word and match it with an existing word. If it exists then we get and save their parts of speech in 'POS' array of strings again with the help of 'find' function and save their types (whether it can be used as subject or object or both or none) in 'TYPES' array of string.

## 12 i. Finding the Subject

154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166

The most vital job to translate is to find the subject of the sentence. The form of verb varies with the person of subject, and positioning of words depends on it. So we create and use the function 'find Sub' for finding the subject. In that function, we check the previously construct 'Types' array. In a sentence, the subject is the first encountered word, which is either a noun or a pronoun or a gerund. So we search the Types array for the first word whose type is either 'subject' or 'both' until the occurrence of a verb or the end of sentence and place the meaning of the subject in a sentence. If we find an adjective before the occurrence of the subject then we place that before the subject. If we do not trace a subject then we assume the sentence as an imperative one, and the person of the subject is assigned. After finding the subject, we return from the function. Though we will return from the function if we find a verb assuming the sentence as imperative, however, we will continue even if we find auxiliary verb or a modal verb, but we will stop if we find 'let'. We also set the Boolean variable to have modal as true if we find a modal. We also set the Boolean used[i] as true where i+1 is the position of the subject or Auxiliary verb.

## 13 ii. Setting the type of sentence

167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177

Now one of our important jobs is to determine the type of sentence. We have divided it according to its meaning. We determine whether this is assertive, interrogative, imperative, optative, or exclamatory. We initially assume the sentence type as an assertive one. However, if the first word is a modal one but not 'let' or is an auxiliary verb or 'wh' pronoun, then sets the type as an interrogative sentence and calls the created function 'SetAs Question'. If it finds a verb as the first word, then it sets type as imperative. If the first word is 'may', then it is an optative one. Otherwise, we stick to our initial guess as assertive. If the first word is a modal, then the type is also modal, and then we set Boolean has modal as true. The function determines the questioning word, when the function 'setAsQuestion' is called. If the first word is 'wh' type, then the questioning word is the corresponding meaning. Otherwise the questioning word is the meaning of 'what' ("ki"). After determining the questioning word, we place that word after the subject.

## 14 iii. Putting the Bangla meaning together

178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189

Now we have done all the requirements of understanding the forms and meaning of a sentence. Now we will put the appropriate meanings in the sentence and thus will be building our translated sentence. For this, first, we use a for-loop to consider all the words in the given sentence. If the considering word is a preposition, then we will retrieve the corresponding meaning from the second column of the table dictionary from the database. Then we hold the meaning in a stack as it will use later after the object. If we find an "Adjective", then that will also be held in a stack. If we trace a word of type "negation", then we check if the previous word was an auxiliary verb or modal or do, does, did. If so, then we understand that the sentence is a negative one. When the type is negative, then we place the meaning of that word in the string "AfterVerb". If we find a verb, then we call function 'placeverb' and 'placeobject'. If we find a word, it can be used as an object. And if we find an auxiliary verb, we neglect that and move to the next word. And in any other case, we put the word straight into the sentence.

## 15 b) Process of getting output

190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203

Let the input sentence is, "I am eating mango and he is drinking milk." At first, the sentence is broken into two sentences, and those are:

1. "I am eating mango."
2. "He is drinking water."

Then the system takes the first sentence for processing. The sentence is read from left to right and grouped into words that are separated by space. Function split() is used to do this. The output of this function is a sequence of words constituting the sentence:

Then we search each word in the dictionary and obtain its parts of speech and its meaning. We rearrange words of the sentence by using the rules of part of speech. Here we have the following English structure: Pronoun + Aux + Verb +Noun ( I am eating mango) We get following sentence structure after re-arranging it. Pronoun + Noun + Verb (I mango eating) After this step, the words are arranged as After all other necessary modifications, the meaning of the sentence is . It is the output of the translator. Other part of the sentence is "he is drinking water", which follow the same process. After this, the conjunction meaning picked from the database and places it between two parts of the sentence. Finally, we will see the following output.

## 16 VI.

204  
205

Some Experimental Result VII.

## 17 Discussion

206  
207  
208  
209

The user can be able to append the required word and its necessary information in the dictionary. Suppose, the input sentence of our system is "Bangladesh is a beautiful country." And our dictionary does not contain the word "beautiful". So, for this input, the system responds with the message as follows:

210 The user can eliminate this problem by adding the bangla meaning of the word "beautiful" to the dictionary,  
 211 and then we get the following output In the case of a syntactically correct sentence, the system translates nicely.  
 212 However, when a sentence Now the meaning of "eating", i.e. is syntactically incorrect, or the delimiter of the  
 213 sentence does not match the sentence type, the system tries to find the nearest match and translate and finally  
 214 gives the closest Bangla output. This system can translate an assertive, interrogative, optative, imperative,  
 215 exclamatory, simple, compound, and complex sentences. There are more sophisticated compound sentences,  
 216 which may contain more than one clause. The system cannot provide any output for this type of sentences.  
 217 We can solve this problem by adding rules for parsing multi-casual sentence. This system can translate both  
 218 active voiced and passive voiced sentences. Besides this, the morphological techniques of English and Bangla are  
 219 implemented in the scanning and generation phases, respectively. But Bangla sentence has several construction  
 220 rules for personal pronouns. So, this system does not support all of them. This system supports one form only.  
 221 For example, consider the translation of the sentence, "You are a good girl.". The translation may be "?? ?? ????  
 222 ??? ??????" or "?? ? ??? ???? ??????" etc. However, the system gives only "?? ?? ??? ???? ??????" as output. The  
 223 word "You" has several meanings, and it is difficult to translate it in the correct form in the context of Bangla  
 224 grammar. Similarly, the meaning of "He" or "She" may be ?? or ????. This type of problem is common in  
 225 Bangla grammar. So, it is difficult to construct a Machine Translator for Bangla to handle all possible meaning  
 226 of a word. In English language, the same word can be used as different parts of speech in different sentences. As  
 227 a result, identify the correct form of a particular word in a sentence is a difficult task.

228 In most cases, the preposition does not maintain specific rules in English sentences. Therefore, it is difficult for  
 229 a rule-based parser to correctly identify prepositions for a particular meaning. For example, "to" is a preposition  
 230 and it can be used as-There is no precise grammar in English for determining cases of nouns and pronouns. But  
 231 it is an essential tool in Bangla to express something clearly. Therefore, cases of different nouns and pronouns in  
 232 an English sentence should be identified properly before translating a sentence into Bangla. Here, detecting the  
 233 relative position of the noun and pronoun with the verb and other words in English.

234 On the other hand, In Bangla, they are identified by considering the suffixes with nouns and pronouns.

235 Bangla is a relatively free word order language than English. So, sentence construction in Bangla has a  
 236 less specific rules. For example, let us consider the sentence "You have given him pen". This sentence can be  
 237 translated as Therefore, for Bangla sentences, parser design is very difficult. More complicated grammar should  
 238 be developed to avoid the problem.

239 VIII.

## 240 18 Conclusion

241 The system provides the user with the facility to append new words in the dictionary. Though the number of  
 242 the given words is a subset of the English language. The user can enrich the stock of words with the help of  
 243 an expert who has sufficient knowledge in both english and bangla language. Although the developed system is  
 244 successful in many aspects, Still have some limitations those are:

245 1. The knowledge base in this system is not selflearning. It cannot interfere the existing decision in the  
 246 knowledge base. 2. The system cannot handle the contextual and semantic problems. <sup>1 2</sup>

**Verb that ends with a meaning "ওয়া"(ooa). Example: Give="দেওয়া"**

**Verb that ends with a meaning "া"(akar). Example: Do="করা"**

**Verb that ends with a meaning "নো"(no). Example: Spin="ঘোরানো"**

1

Figure 1: Fig. 1 :

246

<sup>1</sup>© 2020 Global Journals

<sup>2</sup>© 2020 Global JournalsA Systematic Approach to English to Bangla Sentence Translator

Rule	English word	Bangla meaning	1 <sup>st</sup> person	2 <sup>nd</sup> person	3 <sup>rd</sup> person
"ওয়া"(ooa) changed to "ই" for 1 <sup>st</sup> person , "ও" for 2 <sup>nd</sup> person and "য়" for the third person.	Give	দেওয়া	দেই	দেও	দেয়
"া"(akar) changed to "ই" for 1 <sup>st</sup> person , "ও" for 2 <sup>nd</sup> person and "এ" for the third person.	Do	করা	করি	করো	করে
"নো"(no) changed to "ই" for 1 <sup>st</sup> person , "ও" for 2 <sup>nd</sup> person and "য়" for the third person.	Spin	ঘোরানো	ঘোরাই	ঘোরাও	ঘোরায়

Figure 2: Fig. 2 :

Input sentence (in English)	Translated Sentence (in Bangla)
It is his luck	এটা তার ভাগ্য।
It is a wrong word.	এটা একটি ভুল শব্দ।
The market is noisy	বাজারটি কোলাহলময়
Let me go.	আমাকে যেতে দাও।
Let him go.	তাকে যেতে দাও।
See the sky.	আকাশটি দেখ।
Read the book.	বইটি পড়।
Am I eating?	আমি কি খাচ্ছি?
Is he eating?	সে কি খাচ্ছে?
Do I eat rice?	আমি কি ভাত খাই?
Does he go to school?	সে কি স্কুলে যায়?
How are you?	তুমি কেমন আছো?
May I go?	আমি কি যেতে পারি?
When will you go?	তুমি কখন যাবে?
Which book do you like?	তুমি কোন বই পছন্দ কর?
We are enjoying song.	আমরা গান উপভোগ করছি।
This is my country.	এই আমার দেশ।
Shall I go?	আমি কি যাব?
You will drink milk.	তুমি দুধ পান করবে।
Walking is good.	হাটা ভালো।
I have asked him.	আমি তাকে জিজ্ঞাসা করেছিলাম।
He has added it.	সে এটা যোগ করেছিলো।
The sky is blue.	আকাশ নীল।
Reading book is good.	বই পড়া ভালো।
I run fast.	আমি দৌড়াই।
I drink milk and you see fish	আমি দুধ পান করি এবং তুমি মাছ দেখ।
I am weak but he is strong.	আমি দুর্বল কিন্তু সে শক্তিশালী।

Figure 3:

আমি আম খাচ্ছি

Figure 4:

“আমি আম খাচ্ছি”

Figure 5:

- 
- 247 [Masud et al. (2001)] *A Case Study on Syntactic Approach to Natural Language Processing*, M A Masud , M M  
248 Jorder , Tariq-Ul-Azam . November 2001. Bangladesh. Computer Science And Engineering Khulna University  
249 (Undergraduate Thesis Paper)
- 250 [Mukta et al. ()] ‘A Phrase-Based Machine Translation from English to Bangla Using Rule-Based Approach’.  
251 A P Mukta , A Mamun , C Basak , S Nahar , M F H Arif . *2019 International Conference on Electrical,*  
252 *(Bangladesh) 2019.* p. . Computer and Communication Engineering (ECCE), Cox’s Bazar
- 253 [Md and Muttalib ()] ‘A Review on Computer Based Bangali Processing’. Abdul Md , Muttalib . *NCCPB* 2004.
- 254 [Rhaman and Tarannum ()] ‘A Rule Based Approach for Implementation of Bangla to English Translation’. M  
255 K Rhaman , N Tarannum . *2012 International Conference on Advanced Computer Science Applications and*  
256 *Technologies (ACSAT)*, (Kuala Lumpur) 2012. p. .
- 257 [Aho et al. ()] A V Aho , R Shethi , J Ulman . *Compiler Principle Techniques and Tools*, 2001. Pearson Education  
258 Sixth Indian Reprint.
- 259 [Md and Shahidur Rahman ()] *Bangali Sorting Algorithm :A Linguistic Approach*, Md , Shahidur Rahman .  
260 2004. NCCPB.
- 261 [Lelin Mehedy et al. ()] *Bangali Syntax Analysis: A Comprehensive Approach*, S M Lelin Mehedy , Niaz , M  
262 Arifin , Kaykobad . 2003. ICCIT.
- 263 [Wren Martin ()] ‘English Grammar & Composition’. P C Wren & Martin . *& Company LTD* 2000.
- 264 [AhmedE ()] *English Grammar and Translation*, AhmedE . 1989. Globe Library.
- 265 [Haque and Hasan ()] ‘English to Bengali Machine Translation: An Analysis of Semantically Appropriate Verbs’.  
266 M Haque , M Hasan . *2018 International Conference on Innovations in Science, Engineering and Technology*  
267 *(ICISSET)*, (Chittagong, Bangladesh) 2018. p. .
- 268 [Chowdhury ()] ‘Modern Applied Grammar Translation Composition’. Z A Chowdhury . *Globe Library (PVT.)*  
269 *Limited Edition* 1997.
- 270 [Md. Musfique Anwar, Mohammad Zabed Anwar and Md. Al-Amin Bhuiyan (2009)] ‘Syntax Analysis and Ma-  
271 chine Translation of Bangla Sentences’. *IJCSNS International Journal of Computer Science and Network*  
272 *Security* Md. Musfique Anwar, Mohammad Zabed Anwar and Md. Al-Amin Bhuiyan (ed.) August 2009. 9  
273 (8) .
- 274 [Rahman ()] *Vchchotor Bangali Grammar*, M D Rahman , SH . 1989. Globe Library.
- 275 [Nur Hossain Khan et al. ()] ‘Verification of Bangla Sentence Structure using N-Gram’. Md Nur Hossain Khan ,  
276 Md Khan , Md Islam , Bappa Habibur Rahman , Sarker . *Global Journal of Computer Science and Technology:*  
277 *A Hardware & Computation* 2014. 14. (Issue 1 Version 1.0 Year)