

# An Approach Towards Multilingual Translation By Semantic-Based Verb Identification And Root Word Analysis

Md. Saidul Hoque Anik  
Department of CSE, BUET  
Dhaka, Bangladesh.  
onix.hoque@gmail.com

Md. Adnanul Islam  
Department of CSE, BUET  
Dhaka, Bangladesh.  
islamadnan2265@gmail.com

A. B. M. Alim Al Islam  
Department of CSE, BUET  
Dhaka, Bangladesh.  
alim\_razi@cse.buet.ac.bd

**Abstract**—Popular and widely available translators like Google Translator uses statistic based approach to build the multilingual translation system. This approach solely depends on the availability of a large number of samples. Which is why, Google translator performs interestingly well when it translates among the popular languages like English, French or Spanish, however, makes elementary mistakes when it translates the languages that are newly introduced or less known to the system. Most of the research found so far on natural language processing (NLP), have been performed keeping English as the target language. However, a good number of widely spoken potential languages remain nearly unexplored in the research fields which is quite unexpected in the era of global communication. In this study, we have tried to explore a generalized machine translation system, especially for the languages having insufficient availability in literature. This study basically focuses on Bengali Language as an example of such low resource languages. In this work, we have proposed different approaches for semantic based verb identification along with its translation, and hence, developed an algorithm for root word detection of a verb in any sentence which reflects significant improvement over Google Translator. Finally, we have shown a comparison among the different approaches in terms of accuracy, time complexity and space complexity.

**Keywords**—NLP, OpenNLP, Levenshtein, Wordnet, EBMT, SDL

## I. INTRODUCTION

Human beings have been communicating with various spoken languages since their earliest days on the Earth. Human languages can express thoughts on an unlimited number of topics e.g., the weather, the past, the future, gossip, etc. Every human language has a vocabulary consisting of hundreds of thousands of words which is initially built up from several dozen speech sounds. More remarkable point here to be noted is that every normal human child basically learns the whole system just from hearing others use it.

While many believe that the number of languages in the world is about 6500, there are actually 7097 living languages in the world [19]. Although this number might be the latest count, there is no one clear answer as to the exact number of languages that still exist. One statistics tells us that about 230 languages are spoken in Europe, whereas over 2000 languages are spoken across Asia. In the era of globalization, people often need to communicate in more than one language. As it is quite tough to learn and track multiple languages for

a single person, importance of machine translation follows. Machine translation has emerged as one of the top valuable technologies for localization and arguably even for global economies. It works reasonably well for most of the highly popular languages like English, French, Spanish, etc.

Bengali is considered as one of the low-resource languages for machine translation as it lacks different language resources like electronic texts and parallel corpus. Around 38% of Bengali speaking people are monolingual. Since significance of learning English is unavoidable at present, it is important to have a well developed Bengali to English translation system. Not only Bengali-English pair but also there are enormous numbers of different language pairs which thrive for a translation learning mechanism of their own like, Bengali-Arabic, Hindi-Bengali, Arabic-English, Arabic-Spanish, etc.

In this study, we take Bengali to English translation system as an example to propose a generalised skeleton for multilingual translation system. The main focus of this work is verb identification and optimization techniques using semantic analysis.

## II. MOTIVATION

Natural languages like English, Spanish, and even Hindi are rapidly progressing in processing by machines. While progress has been made in language translation software and allied technologies, the primary language of the ubiquitous and all influential World Wide Web is English. English is typically the language of latest-version applications and programs and new freeware, manuals, shareware, peer-to-peer, social media networks and websites. However, Bengali, being among the top ten languages in the world, lags behind in some crucial areas of research like parts of speech tagging, information retrieval from texts, text categorization, and most importantly, in the area of syntax and semantic checking [1].

Now-a-days, Google translator is one of the pioneer applications supporting a number of languages to translate from one to another. Although it works successfully for many languages, it is still in developing phase for Bengali to English translation. Google translator fails to detect the verbs in a sentence accurately. More importantly, it can not always retrieve necessary

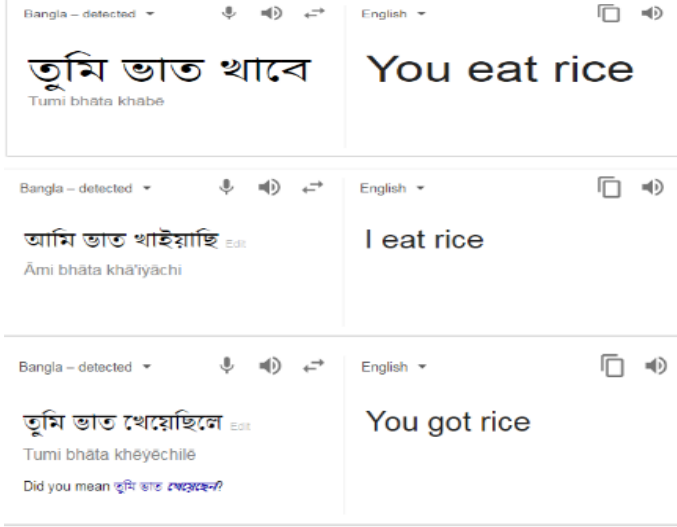


Fig. 1. Faulty translations of Google Translate

information like, person and number of the subject, tense of the verb, etc. correctly which are the pillars of a successful translation. Therefore, the resulting translation becomes faulty for a large set of sentences. Fig. 1 shows some examples of faulty translations by Google Translator for Bengali-English language pair. The correct translations of these sentences are respectively-

- You will eat rice
- I have eaten rice
- You ate rice

If we notice carefully, the source of these faults is mainly the misleading verbs since the detection of the tense from them is incorrect. This problem leads to faulty translations for a significant number of input sentences since the problem relates to the basic skeleton of a sentence construction. The corrections have been achieved in our proposed system by semantic analysis of the verbs which can be visualised in Fig. 16, discussed later in this paper. The other translators e.g., Bing, Yahoo Babel Fish, Systran Language Translation, SDL Free Translation, etc., cannot support Bengali and many other widely used languages like Bengali, Arabic, etc.

Therefore, the motive of our research is not only to efficiently translate one language to other using a generalised translation skeleton but also to teach the translation mechanism step by step. In this study, we mainly focus on the detection and the learning of the verbs semantically as semantic based verb identification and optimization is an attempt towards achieving that goal. We also show a comparative analysis of the results with Google Translator to point out the improvements achieved by our proposed mechanism.

### III. RELATED WORK

Bengali, being the native language of about 243 million people [20], still lacks significant research in the area of

natural language processing. Bangla to English translation was first proposed by Sk. Borhan Uddin, Dr. Md. Fokhray Hossain and Kamanashis Biswas using opennlp (OpenNLP) tool. They proposed a simple technique for synthesizing Bengali words. However, they used opennlp tool after translating the Bengali word to corresponding English word which caused erroneous Parts Of Speech (POS) tagging for different words and generated wrong outputs for very simple sentences.

Kim et al., [4] used syntactic chunks as units of translation for improving insertion or deletion of words between two distant languages. However, an example base with aligned chunks in both source and target language is missing in this approach.

Saha et al., [12] reported an EBMT (Example Based Machine Translation) for the translation of different news headlines. The work showed that EBMT can be a positive approach for Bengali language. However, their approach relied mostly on news headlines. Moreover, Gangadharaiah et al., [3] proposed that templates can be useful for EBMT to obtain longer phrasal matches if coordinated with statistical decoders. His study showed that it is a time consuming task to cluster the words manually and would be less time consuming to use standard available resources such as, WordNet for clustering.

Dasgupta et al., [6] proposed to use syntactic transfer. They converted CNF (Chomsky Normal Form) trees to normal parse trees and using a bilingual dictionary, generated output translation. However, this research did not consider translating the unknown words which did not appear in the bilingual dictionary.

### IV. PROPOSED MECHANISM

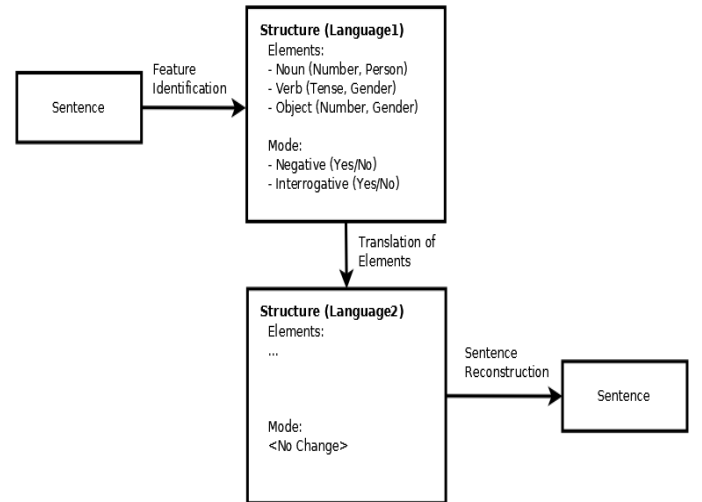


Fig. 2. Translation Methodology

Our proposed mechanism involves storing the gist or the concept of a sentence in a structure using semantic analysis.

A simple sentence can be basically broken down into its subject, verb and object in any order, corresponding to the language of the sentence. Each of them may have their own attributes such as, number, person, tense, etc. In addition, the overall sentence can have different modes e.g., negative form, interrogative form, etc. Fig. 2 shows the sequence of steps for translation.

During translation from one language to another, it is possible that direct translation of a word in destination language is not available. As our targeted translator system will contain a number of languages, we can use a chain of intermediate language translations to reach the destination language. Fig. 3 illustrates the process of this method.

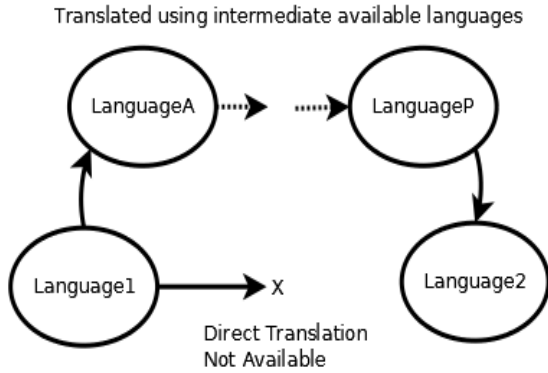


Fig. 3. Translation using intermediate languages

For example, we want to translate ‘word1’ from Bengali to English. When we look up on the vocabulary of the translator, we see that ‘word1’ does not exist in Bengali-English vocabulary. However, the word might be available on Bengali-Arabic vocabulary of the translator and the Arabic translation for ‘word1’ can be ‘word1-Arabic’. Now in the Arabic-English vocabulary, if ‘word1-Arabic’ is also available then we find that in English, ‘word1-Arabic’ stands for ‘word1-English’. Hence for the Bengali word ‘word1’, the appropriate English translation ‘word1-English’ is found by the proposed translator.

The proposed methodology discussed so far, may be appropriate for generic word translation only. However, this approach cannot be directly applied in translating verbs as they may appear in different forms depending on the tense of the sentence, number and person of the subject, etc. The scenario becomes more complex when some suffixes or prefixes are assimilated into the verbs. This is common in some languages such as Bengali or Arabic where a root form of verb changes into different modes depending on the subject and the tense of a sentence. In this scenario, a simple look-up table (for vocabulary) is not good enough for verb translation.

## V. VERB IDENTIFICATION & TRANSLATION METHODOLOGY

In our proposed translation system, the Bengali verbs are stored in a table along with the other words as a part of vocabulary for each language. However, we need to keep in mind that one verb may have multiple representations based on tense and subject of a sentence as shown in Fig. 4. The figure shows an example of different forms taken by each of the two different verbs, ‘eat’ and ‘play’ in Bengali. We have implemented three

খাওয়া → খেয়েছিলে, খেয়েছিলাম, খেয়েছিল, খাবে, খাবে, খাচ্ছ, খাচ্ছি, খাচ্ছিল, খাচ্ছিলাম, খাই, খাইতেছিল, খেয়েছি, খাচ্ছে, খাইতেছি, খেয়েছ, খেয়েছিলে, খায়, etc.  
 খেলা → খেলি, খেলে, খেল, খেলছে, খেলেছি, খেলেছে, খেলতেছিলে, খেলেছিলে, খেলেছিলাম, খেলেছ, খেলিছিলাম, খেলবে, খেলতেছি, খেলতেছ, খেলতেছে, খেলছে, খেলছ, খেলিয়াছে, খেলিয়াছ, etc.

Fig. 4. Multiple forms of verbs in Bengali

different approaches for translating the verbs efficiently which give us different results on performance and accuracy. One approach improves over another sequentially. After discussing them, we will show a comparative evaluation of these three different approaches.

### A. Naive Approach: Gigantic Database

This is the simplest approach (approach 1) to implement. Like all other words (nouns, pronouns, etc.), we simply insert all the different forms of a standard verb with their standard translation as separate entries in the database table for vocabulary. The following figure (Fig. 5) illustrates how multiple entries for a standard verb are incorporated in the database as vocabulary.

Word	Translation	Word	Translation
খাই	Eat	খায়	Eat
খাচ্ছি	Eat	খাচ্ছ	Eat
খেয়েছিলে	Eat	খেয়েছিলাম	Eat
খাইতেছি	Eat	খেয়েছ	Eat
খাচ্ছিলাম	Eat	খাবে	Eat

Fig. 5. Database table for translating verbs having different forms

Using this table we can find the standard translated verb (eat, go, play, etc.) which is then modified according to the tense and subject of the sentence applying semantic analysis (reported in our previous work, [1]). Here, we have shown an example of a translated verb ‘eat’. Now, it is processed based on the semantic analysis of the sentence e.g., is eating, ate, has eaten, etc. This approach guarantees 100 percent accuracy in terms of translating verbs. However, memory consumption becomes a major issue due to the repetitive insertions of one standard verb in various forms. We will come back to this point later with some statistical measures.

### B. Optimized Database with Semantic Analysis

We propose our next approach (approach 2) that reflects an immediate improvement over our previous approach. As discussed earlier, if we need to store the word translation for each form of the same verb then the database will become very large due to the repetitive insertions which leads to a massive memory consumption. However, we can avoid the multiple insertions of the same verb having different forms using this approach, database optimization technique with semantic analysis. We can store only the standard verb in

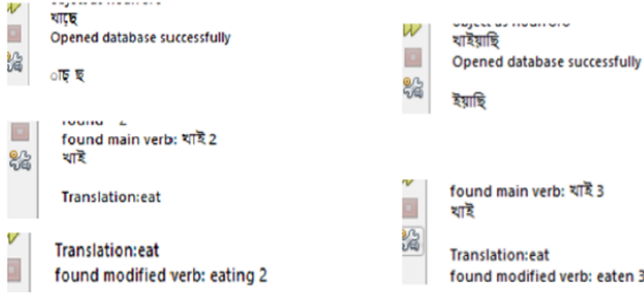


Fig. 6. Database optimization using semantic analysis on verbs

the vocabulary table and apply semantic analysis to detect the standard form from the other forms of the verb depending on number, person and tense as shown in Fig. 6. The figure shows how one word (standard verb) can take different forms and suggests insertion of only that particular standard word in the database table for vocabulary, not all of its different forms. This will avoid multiple insertions in the database for the same verb with multiple forms.

However, to detect that standard verb from its other different forms, we needed to concatenate all the different forms of the verb as a single large string and inserted it into another table with its standard form as a single entry as shown in Fig. 7.

Non Standard Forms	Standard form
খেয়েছিলেখোয়ছিলামখেয়েছিলথারবেখাচ্ছাচ্ছিখাচ্ছিলামখাইখাওখাইতছিলখেয়েছিখাচ্ছ ...	খাওয়া
খেলিখেলেনখেলছেনখেলিখেলেনখেলছেনখেলিখেলেনখেলছেনখেলিখেলেনখেলছেনখেলিখেলেনখেলছেন ...	খেলা

Fig. 7. Mapping between non-standard forms and standard form of verb

Although this approach improves the searching time and avoids overheads for multiple entries significantly, it offers no significant improvement in terms of overall memory required for actual data since all the forms of a standard verb is ultimately saved in database as a single string.

### C. Levenshtein Distance

Our latest approach (approach 3) emerges from the demand of sustainability to ensure green computation in terms of both the space complexity and the time complexity. The translation of a verb can be done using hash table which is implemented in this approach.

The key-value pair consists of only the standard form of verbs in two language. In order to effectively translate, we need to find a way to recognize the standard form of Verb from its non-standard form. For this purpose, we shall be using a modified version of a popular string similarity measurement algorithm, known as Levenshtein distance.

Levenshtein distance is the measurement to find out the minimum number of operations that are required to convert source string into destination string. The commonly used operations are:

- Insertion (of letter in source string)
- Deletion (of letter from source string)
- Substitution (of a letter with another letter in source string)

Each of these operations is associated with a cost. Whenever any operation is performed upon the source string, corresponding cost is taken into account. Higher cost refers to higher dissimilarity between the source and the destination string.

### D. Modified Levenshtein Distance

In a standard Levenshtein Distance algorithm, each of the operations has unit cost. We have modified the cost of these operations carefully with an algorithm to identify the root verb from a non-standard form of verb. A non-standard form of verb may have prefix and suffix assimilated into it based on tense and subject. Instead of directly trying to match a non-standard form of verb with a standard form, we are going to break down the non-standard form into its root word, and then try to match the root word with its standard form. The process can be visualized in Fig. 8.

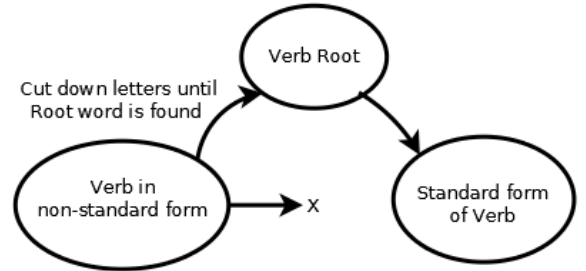


Fig. 8. Verb translation using modified Levenshtein distance algorithm

In order to convert into the root-verb, we need to cut down letters or characters from the non-standard form. So if we are deleting letters from the source, probably we are getting closer to the root word. This is why we have assigned the deletion cost to zero.

On the other hand, the set of letters in root-verb is almost always a subset of the letters in the non-standard form of that verb. An example is shown in Fig. 9. Therefore in the

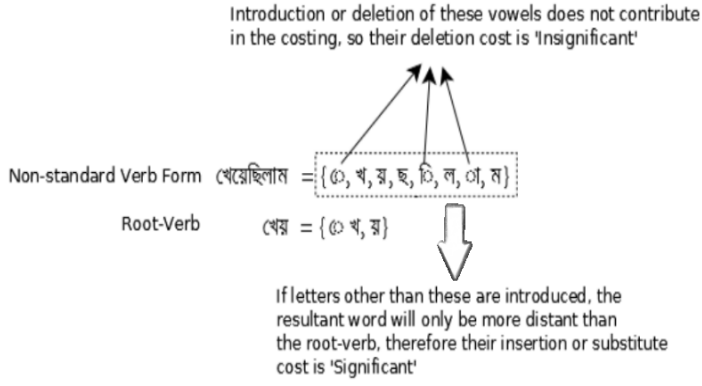


Fig. 9. Root-verb as a subset of the non-standard verb forms

process of modification, if we are introducing a new letter using insertion or substitution operation, we are most-likely deviating away from the destination root-verb. For this reason, the algorithm is modified in a way that introduction of a new letter or character penalizes the overall cost. We have denoted this cost as ‘Significant Cost’.

In order to improve the root-verb identification, we have introduced another concept into this algorithm. It is often seen that the root word contains a new vowel that is different from the non-standard form of that verb. To facilitate this process, we have considered insertion of these vowels as insignificant, and denoted the associated cost as ‘Insignificant Cost’. It improves the accuracy of root-word identification for languages such as Bengali or Arabic.

---

**Algorithm 1** Get the root word given other form of verb

---

```

procedure GETROOTWORD
Input : ModVerbFrom  $\leftarrow$  The modified verb from
Output : Root word of ModVerbFrom
  root_list  $\leftarrow$  List of root words
  matched_root  $\leftarrow$  root_list[0]
  min_dist  $\leftarrow$  GetDistance(ModVerbFrom, root_list[0])
  for root in root_list do
    temp  $\leftarrow$  GetDistance(ModVerbFrom, root)
    if temp < min_dist then
      min_dist = temp
      matched_root = root
  return matched_root

```

---

Algorithm 1, 2, and 3 demonstrates the complete algorithm of the modified Levenshtein distance calculation. First, from a given list of root words and a verb form, our system finds the root word that is closed to the given verb form using the ‘GetDistance’ procedure from Algorithm 1. Then Algorithm 2 calculates the ‘Min Distance’ to convert the verb form into the root word by inserting, deleting or replacing the characters. In case of insertion or replacement, it is considered whether the newly introduced character is an insignificant character

---

**Algorithm 2** Measure the weighted distance between a root word and another verb form

---

```

procedure GETDISTANCE
Input : lhs, rhs  $\leftarrow$  Two character sequences
Output : Cost difference between lhs and rhs
  len_lhs  $\leftarrow$  (Length of character sequence lhs) + 1
  len_rhs  $\leftarrow$  (Length of character sequence rhs) + 1
  cost  $\leftarrow$  Array of size len_lhs
  new_cost  $\leftarrow$  Array of size len_rhs
  for i := 0 to i < len_lhs step 1 do
    cost[i] := i
  for j := 1 to j < len_rhs step 1 do
    new_cost[0] := j
    for i := 1 to i < len_lhs step 1 do
      if lhs[i - 1] = rhs[j - 1] then match  $\leftarrow$  0
      else match  $\leftarrow$  GetCost(lhs[i - 1]) + GetCost(rhs[j - 1])
      cost_replace := cost[i - 1] + match
      cost_insert := cost[i] + GetCost(rhs[j - 1])
      cost_delete := new_cost[i - 1]
      new_cost[i] :=
        Min(cost_replace, cost_insert, cost_delete)
    Swap(cost, new_cost) //Swap the two arrays after
  each inner loop
  return cost[len_lhs - 1]

```

---



---

**Algorithm 3** Get Cost

---

```

procedure GETCOST
Input : c  $\leftarrow$  Character whose cost is to be calculated
Output : cost  $\leftarrow$  Cost of the character
  SignificantCost  $\leftarrow$  Significant change weight
  InsignificantCost  $\leftarrow$  Insignificant change weight
  insignificant_change_list  $\leftarrow$ 
    List of insignificant characters
  if c is in insignificant_change_list then return
    InsignificantCost
  return SignificantCost

```

---

(trivial characters like vowels that do not change the meaning of the verb significantly) or not. The comparison is done using the ‘GetCost’ procedure. Finally, procedure ‘GetCost’ returns ‘InsignificantCost’ if the input character is not significant, otherwise, returns ‘SignificantCost’ as shown in Algorithm 3.

**E. Finding Standard-form of Verb**

Our proposed translation system contains tables containing root verbs that are pointing towards their respective standard-form of verb. Using the algorithm demonstrated in the previous section, we shall be able to match the non-standard form of verb with the nearest matching root-verb. The root-verbs are mapped to standard verb forms using a hash-map. In our system, it is also possible that a single verb form may

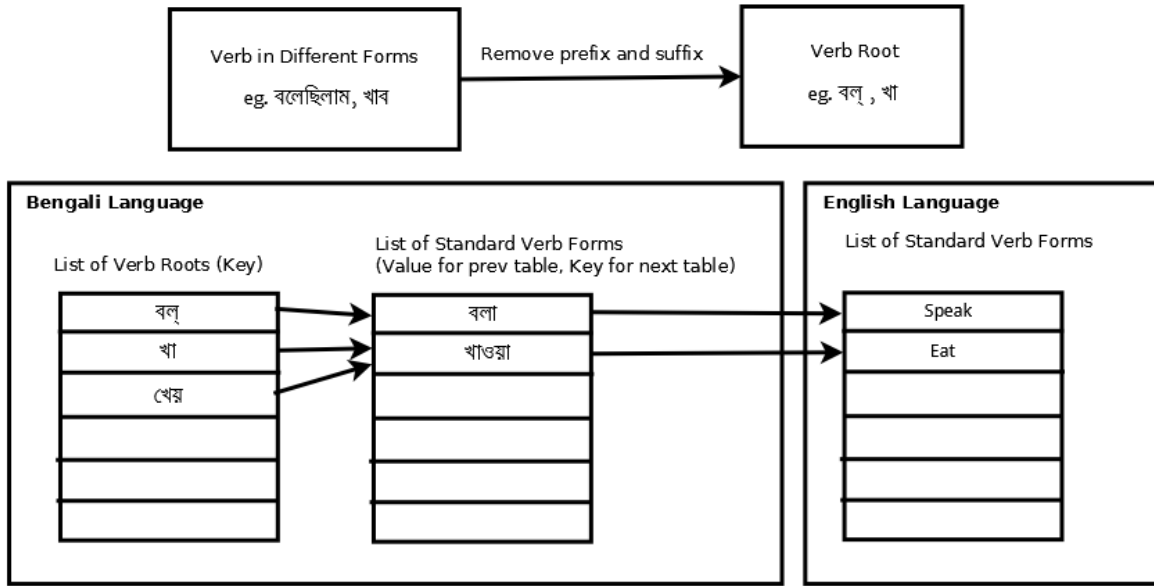


Fig. 10. Finding the standard form of verb

come from multiple root-verb. The complete process can be visualized in Fig. 10.

## VI. EXPERIMENTAL EVALUATION

The parameters and settings that were used to carry out is discussed in the following subsections.

### A. Tools and Settings

The proposed algorithm was implemented in Java Language, and was used to translate verbs from Bengali to English. Several forms of verbs were tested using the program. For experimentation, we used the following features in our implemented system:

- Language : JAVA
- Platform/ IDE : Netbeans
- Database : Sqlite
- Tool : Opennlp tools

A major issue arose while taking input and parsing Bengali texts in java. We set up text encoding to UTF-8 and also changed the font settings and some other settings to work successfully with Bengali texts in netbeans.

We used Sqlite with java for database in our system. Since we had to create a Bengali to English dictionary, we needed a database to retrieve the word translations. Therefore, we installed Sqlite and also added a jar file for Sqlite in our project.

To calculate the Levenshtein distance, the values of significant cost and insignificant cost were assigned two and zero respectively.

### B. Result

All the approaches gradually improve one over another. Specially, the approach of Levenshtein distance calculation shows significant improvement in terms of both the space and

Verb Form	Root Word	Standard Form	Translation	Remark
--[eat]--				
খেয়েছিলে	খেয়	খাওয়া	eat	OK
খেয়েছিলাম	খেয়	খাওয়া	eat	OK
খেয়েছিল	খেয়	খাওয়া	eat	OK
খাব	খা	খাওয়া	eat	OK
খাবে	খা	খাওয়া	eat	OK
খাস	খা	খাওয়া	eat	OK
খাচ্ছি	খা	খাওয়া	eat	OK
খাচ্ছিল	খা	খাওয়া	eat	OK
খাচ্ছিলাম	খা	খাওয়া	eat	OK
খাই	খা	খাওয়া	eat	OK
খাও	খা	খাওয়া	eat	OK
--[go]--				
যাই	যা	যাওয়া	go	OK
খেলেছি	খেল	খেলা	play	OK
খেলেছে	খেল	খেলা	play	OK
খেলেছিলেন	খেল	খেলা	play	OK
খেলেছিলে	খেল	খেলা	play	OK
খেলেছিলাম	খা	খাওয়া	eat	Incorrect
খেলব	খেল	খেলা	play	OK
খেলবে	খেল	খেলা	play	OK
--[study]--				
পড়ি	পড়	পড়া	study	OK
পড়	পড়	পড়া	study	OK
পড়ছি	পড়	পড়া	study	OK

Fig. 11. Output of modified Levenshtein distance algorithm



the time complexity. This algorithm was applied on several Bengali verbs to get the root words (verbs). The root words were then mapped to the standard form. The output generated by implementing this algorithm is summarized in Fig. 11.

First, we obtain the root-verb by calculating the Levenshtein distance accordingly. In the mean time, we can also identify the tense by extracting the suffixes from the verbs. Then after mapping the root-verb(s) to the standard form, we retrieve the raw translation of the verb. However, the detection of root verb by calculating the Levenshtein distance can be incorrect for some forms of verbs which can lead to wrong translation of the verb completely. In Fig. 11, we can notice one such faulty case where the verb has been erroneously translated to ‘eat’ in place of ‘play’. Fortunately, such erroneous cases have been handled successfully by slight preprocessing of the verbs, discussed in the next section. Nevertheless, we finally translate the verb by modifying its raw translated form after gathering other relevant information (POS tagging, person, number, etc.) from the input sentence as shown in Fig. 12.











 তারা ভাত খাইয়াছে। তারা in subject তারা	 আমি ভাত খাইয়াছি। আমি in subject আমি
 object as noun ভাত খাইয়াছে Opened database successfully	 খাইয়াছি Opened database successfully
 ইয়াছে Opened database successfully	 ইয়াছি
 found main verb: খাই 3 খাই	 found main verb: খাই 3 খাই
 Translation: eat found modified verb: eaten 3	 Translation: eat found modified verb: eaten 3

Fig. 12. Root-verb identification and translation

Then we generate the translation of the input sentence by applying necessary grammatical rules of the target language. The details of the complete translation mechanism has been discussed in our previous work [1].

### C. Findings

From the experimental result of approach 3, we can see that our algorithm is able to detect almost all of the root words successfully with some exceptions (Fig. 11). Afterwards, we found that if the Levenshtein distance can be calculated after preprocessing carefully the non-standard verbs a little. After removing the common suffixes which add to the verbs due to different tenses, we can get an optimized verb closer to the root-verb. It speeds up the Levenshtein distance calculation and also offers better accuracy in detecting the correct root-word.

Fig. 15 shows the improvement achieved (inside green box) due to the slight preprocessing of the verbs before Levenshtein distance calculation. It eliminates the incorrect detection of root verb shown earlier (in Figure 14) and ensures the correct root word detection for almost all the possible cases. The

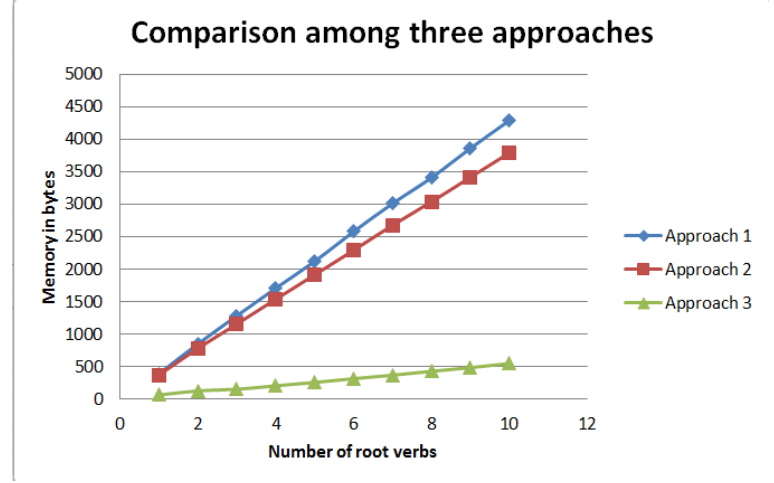


Fig. 13. Comparison of the proposed approaches in terms of memory consumption

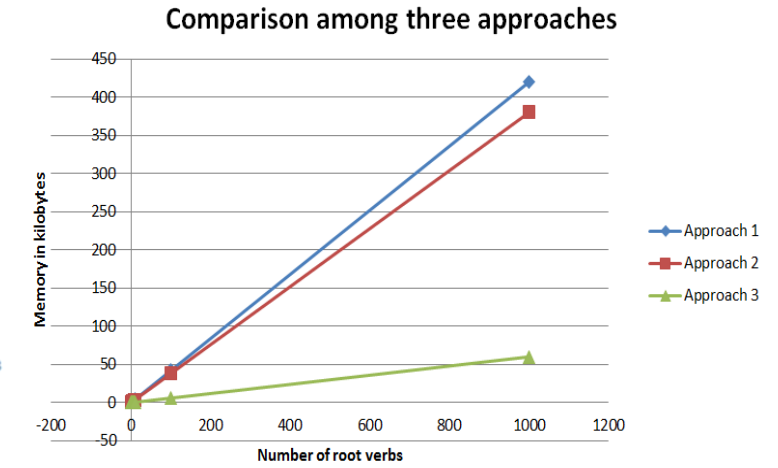


Fig. 14. Comparison of the proposed approaches in terms of memory consumption (larger number of verbs)

figure illustrates how the addition of the ‘Suffix Reduced Form’ improves the accuracy of the modified Levenshtein distance algorithm.

We found that both approach 1 and approach 2 generate almost 100 percent accurate result in translating different forms of verbs. Modified Levenshtein distance calculation approach generates around 90 percent accurate result which has been subsequently increased to almost 98-99 percent in the improved version of the algorithm (preprocessing of verbs before distance calculation).

However, the most statistical comparison among the three approaches can be shown in terms of space complexity. Fig. 13 and Fig. 14 graphically shows a comparative evaluation of these approaches for different number of verbs. Here in Fig. 14, we can see that approach 3 improves over approach 2

run:	Verb Form	Suffix Reduced Form	Root Word	Standard Form	Translation	Remark
	—[eat]—					
	খেয়েছিলেন	খেয়েছ	খের	খাওয়া	eat	OK
	খেয়েছিলেনাম	খেয়েছি	খের	খাওয়া	eat	OK
	খেয়েছিলেন	খেয়ে	খের	খাওয়া	eat	OK
	খাব	খাব	খা	খাওয়া	eat	OK
	খাবে	খাবে	খা	খাওয়া	eat	OK
	খাচ্ছ	খাচ্ছ	খা	খাওয়া	eat	OK
	খাচ্ছি	খাচ্ছি	খা	খাওয়া	eat	OK
	খাচ্ছিল	খাচ্ছি	খা	খাওয়া	eat	OK
	খাচ্ছিলেনাম	খাচ্ছি	খা	খাওয়া	eat	OK
	খাই	খাই	খা	খাওয়া	eat	OK
	খাও	খাও	খা	খাওয়া	eat	OK
	—[go]—					
	যাই	যাই	যা	যাওয়া	go	OK
	খেলছে	খেল	খেল	খেলা	play	OK
	খেলতেছিলেন	খেলতেছ	খেল	খেলা	play	OK
	খেলছিলেন	খেলছে	খেল	খেলা	play	OK
	খেলছিলেনাম	খেলছি	খেল	খেলা	play	OK
	খেলছে	খেলছে	খেল	খেলা	play	OK
	খেলব	খেলব	খেল	খেলা	play	OK
	খেলছিলেনাম	খেলছি	খেল	খেলা	olav	OK

Fig. 15. Improvement over modified Levenshtein distance algorithm due to preprocessing of the verbs

by saving around 130kB of memory. However, we have shown the result for only 1000 verbs from a single language. Our proposed translator should deal with hundreds of languages containing millions of verbs in each language. Considering this, the improvement achieved in terms of memory consumption can be around  $130\text{KB} \times 100 \times 1000 = 13\text{GB}$  keeping in mind that there are also other words in the vocabulary other than verbs which is significant for the mobile devices specially.

Now, we would like to show how our proposed translation system is improving over the most widely used translator, Google Translate, in Fig. 16. We carefully designed a dataset for Bangla-English translation so that more focus requires in the verbs in the input sentences. The figure illustrates some examples from a large dataset that how our proposed system achieves improvement over Google Translator by identifying the root-verbs efficiently.

It shows that Google Translator fails to identify the correct root-verb including the tense of the sentence (red coloured words) which leads to the incorrect translation for even different simple sentences. However, the accurate translations of those sentences have been generated by our system.

## VII. FUTURE WORK

One of the main challenges in Bengali to English text conversion remains in implementing its vast grammatical rules. If we can track the core rules to acquire a generalized format for all rules and exceptions then the translation task will be simpler and compact. Developing Opennlp tools for parts of

Counter example against Google Translator	Translation using proposed system	Translation using Google Translator	Does Google translate correctly? (Yes/No)
আমি ভাত খাইয়াছি	I have eaten rice	I <b>eat</b> rice	No
তুমি ভাত খাবে	You will eat rice	You <b>eat</b> rice	No
তুমি ভাত খেয়েছিলেন	You ate rice	You <b>got</b> rice	No
Etc., so on ...	...	...	No

Fig. 16. Improvement of our proposed translator over Google Translate

speech tagging of Bengali words in a sentence efficiently is one of the most crucial tasks in Bengali to English translation. We aim to extend our work on developing Opennlp tools for Bengali language.

There is a great deal of research opportunities in language processing. Grammars keep changing as the language builds its grammar. Therefore, we need to find a translation process to update new sentence making rules anytime. Machine Learning using Statistical Machine Translation can be one way to achieve it. We plan to experiment with it.

Besides, our initial motive was to build a translation model for Bengali to Arabic conversion. However due to lack of proficiency in Arabic language, we had to start with conversion



to English language. Therefore, we want to implement our proposed generalised translation skeleton for Arabic language soon so that we can help a large group of people to learn and understand Arabic language.

### VIII. CONCLUSION

In the era of technology and global communication, people generally thrive not only for translations between two languages but also for learning multiple languages equal effectively for sustaining. However, very general or primary grammatical rules of any language usually consist of a good number of exceptions. Keeping track of the wide varieties of possible cases is one of the most common features of a multi-lingual translation system which is a difficult task even for the most intelligent beings. Therefore, it massively demands wide and complex application of Artificial Intelligence to build up a near-accurate translator which on the contrary, may result in the degradation of the overall performance of the system drastically.

Our system currently focuses on Bengali to English translation. However, it has limited knowledge base and vocabulary till now. By increasing the vocabulary and the knowledge base we can improve its efficiency by testing over wide range of different cases for general purpose use. Considering the limitations of a machine translator, our preference is always towards making the learning of a language easier by implementing and teaching all the basic and necessary translation processes step by step.

### REFERENCES

- [1] M. Islam and A. Islam, Polygot: Going Beyond Database Driven And Syntax-based Translation, ACM DEV '16: Proceedings of the 7th Annual Symposium on Computing for Development, November 2016.
- [2] Z. Anwar, Developing a Bangla to English Machine Translation System Using Parts Of Speech Tagging: A Review, Vol. 1. No. 1, Journal of Modern Science and Technology, May 2013.
- [3] R. Gangadharaiah, R. D. Brown, and J. G. Carbonell., Phrasal equivalence classes for generalized corpusbased machine translation. In Alexander Gelbukh, editor, Computational Linguistics and Intelligent Text Processing, volume 6609 of Lecture Notes in Computer Science, pages 1328. Springer Berlin / Heidelberg, 2011.
- [4] S. Raphael, J. D. Kim, R. D. Brown, J. G. Carbonell, Chunk-Based EBMT. EAMT, 2010.
- [5] M. Roy, A Semi-supervised Approach to Bengali-English Phrase-Based Statistical Machine Translation, Proceedings of the 22nd Canadian Conference on Artificial Intelligence, 2009.
- [6] S. Dasgupta, A. Wasif, and S. Azam, An Optimal Way Towards Machine Translation from English to Bengali, Proceedings of the 7th International Conference on Computer and Information Technology (ICCIT), 2004.
- [7] M. Anwar and M. Bhuiyan, Syntax Analysis and Machine Translation of Bangla Sentences, International Journal of Computer Science and Network Security, 09(08),317326; 2009.
- [8] Sk. B. Uddin, Bangla to English Text Conversion using opennlp Tools; Daffodil International University Journal Of Science & Technology, Vol. 8, Issue 1, JANUARY 2013 .
- [9] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, A Study of Translation Edit Rate with Targeted Human Annotation, Proceedings of Association for Machine Translation in the Americas, 2006.
- [10] S.p K. Naskar and S. Bandyopadhyay, A Phrasal EBMT System for Translating English to Bengali, Proceedings of the Workshop on Language, Artificial Intelligence, and Computer Science for Natural Language Processing Applications (LAICSNLP), 2006.
- [11] D. Saha, S. K. Naskar, S. Bandyopadhyay, A Semantics-based English-Bengali EBMT System for translating News Headlines, MT Summit, 2005.
- [12] G. Doddington, Automatic Evaluation of Machine Translation Quality Using N-gram CoOccurrence Statistics, Proceedings of the second international conference on Human Language Technology Research, 2002.
- [13] N. Karamat, Verb Transfer For English To Urdu Machine Translation, FAST-Lahore, 2006
- [14] N. Chatterjee, S. Goyal, A. Naithani, Resolving Pattern Ambiguity for English to Hindi Machine Translation Using WordNet, Department of Mathematics, Indian Institute of Technology Delhi, Published in Workshop Modern Approaches in Translation Technologies, Borovets, Bulgaria, 2005.
- [15] Example Based English to Bengali Machine Translation Thesis work of Khan Md. Anwarus Salam completed in August 2009.
- [16] J. Tiedemann and L. Nygard, The OPUS corpus - parallel and free, Proceedings of LREC, 2004.
- [17] D. Melamed, A Geometric Approach to Mapping Bitext Correspondence, Proceedings of the First Conference on Empirical Methods in Natural Language Processing (EMNLP), 1996.
- [18] <https://www.ethnologue.com/guides/how-many-languages>
- [19] [https://en.wikipedia.org/wiki/World\\_language](https://en.wikipedia.org/wiki/World_language)