ATLANTIS PRESS International Journal of Networked and Distributed Computing, Vol. 5, No. 3 (July 2017) 176–191

Improve Example-Based Machine Translation Quality for Low-Resource Language Using Ontology

Khan Md Anwarus Salam[1], Setsuo Yamada[2], Nishino Tetsuro[3]

[1] IBM Research Tokyo,
19-21 Nihonbashi, Hakozaki-cho, Chuo-ku,
Tokyo 103-8510
E-mail: kmanwar@gmail.com

[2] NTT Corporation,
NTT Hibiya Building 1-1-6 Uchisaiwai-cho, Chiyoda-ku
Tokyo 100-8019, Japan
E-mail: yamada.setsuo@lab.ntt.co.jp

[3] The University of Electro-Communications,
Graduate School of Informatics and Engineering
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585
E-mail: nishino@uec.ac.jp

Abstract

In this research we propose to use ontology to improve the performance of an EBMT system for low-resource language pair. The EBMT architecture use chunk-string templates (CSTs) and unknown word translation mechanism. CSTs consist of a chunk in source-language, a string in target-language, and word alignment in-formation. For unknown word translation, we used WordNet hypernym tree and English-Bengali dictionary. CSTs improved the wide-coverage by 57 points and quality by 48.81 points in human evaluation. Currently 64.29% of the test-set translations by the system were acceptable. The combined solutions of CSTs and unknown words generated 67.85% acceptable translations from the test-set. Un-known words mechanism improved translation quality by 3.56 points in human evaluation.

Keywords: Knowledge Engineering, WordNet, Example-Based Machine Translation;

1. Introduction

Example-Based Machine Translation (EBMT) for low-resource language pair, like English-Bengali, has low-coverage issues, due to the lack of parallel corpus. It also has high probability to handle unknown words.

In this research we propose to use ontology to improve the performance of an EBMT system for low-resource language pair. The EBMT architecture use chunk-string templates (CSTs) and unknown word translation mechanism. Using Word-Net [4] CSTs help to achieve wide-coverage and better quality in EBMT for low-resource language pair like English to Bengali language. For unknown word translation, we used related information such as synsets and hypernyms from WordNet.

CSTs consist of a chunk in source-language, a string in target-language, and word alignment information. CSTs are prepared automatically from word aligned parallel corpus. First the source-language chunks are auto generated by using OpenNLP chunker. Then initial CSTs are generated for each source-language chunks and each CSTs alignment for all target sentences are generated

using the parallel corpus. After that the system generates combinations of CSTs using the word alignment information. Finally, we generalize CSTs using Word-Net to achieve wide-coverage.

For unknown word translation, we used WordNet hypernym tree and English-Bengali dictionary. At first the system finds the set of hypernyms words and degree of distance from the English WordNet. Then the system tries to find the translation of hypernym words from the dictionary according to the degree of distance order. When no dictionary entry found from the hypernym tree, it transliterates the word.

We built an English-to-Bengali EBMT system using CSTs and un-known word translation mechanism. CSTs improved the wide-coverage by 57 points and quality by 48.81 points in human evaluation. Currently 64.29% of the test-set translations by the system were acceptable. Unknown words mechanism improved translation quality by 3.56 points in human evaluation. The combined solutions of CSTs and un-known words generated 67.85% acceptable translations from the test-set.

The rest of this article is organized as follows. In section 2 we explain the back-ground research of this paper. Section 3 gives a brief overview of our proposed EBMT Architecture. Then in Section 4 we explain the CSTs generation process and usage method in translation in details. For reporting the evaluation result we describe our findings in section 5. Then in section 6 we discuss our findings. Then finally we conclude our paper in section 7.

2. Background

Bengali is the native language of around 230 million people world-wide, mostly from Bangladesh. According to "Human Development Report 2009" of United Nations Development Program, the literacy rate of Bangladesh is 53.5%. So we can assume that around half of Bengali speaking people are monolingual. Since significant amount of the web contents are in English, it is important to have English to Bengali Machine Translation (MT) system. But English and Bengali form a distant language pair, which makes the development of MT system very challenging. Bengali is considered as low-resource language, due to the lack of language resources like electronic texts and parallel corpus. As a result, most of the commercial MT systems do not support Bengali language translation.

In present, there are several ways of Machine Translation such as Rule-Based Ma-chine Translation (RBMT), Statistical Machine Translation (SMT) and Example-Based Machine Translation (EBMT) which includes chunk-based and template-based approaches.

RBMT require human made rules, which are very costly in terms of time and money, but still unable to translate general-domain texts. There are several at-tempts in building English-Bengali MT system. The first available free MT sys-tem from Bangladesh was Akkhor Bangla Software . The second available online MT system was apertium based Anubadok . These systems used Rule-Based approach and did not con-sider about improving translation coverage by handling unknown words, in low-resource scenario. Dasgupta et.al. (2004) proposed to use syntactic transfer. They converted CNF trees to normal parse trees and using a bi-lingual dictionary, generated output translation. This research did not consider translating unknown words.

SMT works well for close language pairs like English and French. It requires huge parallel corpus, but currently huge English-Bengali parallel corpus is not available. The most widely used SMT system for Bengali language is Google translate. Google Translate is a free translation service that provides instant translations be-tween dozens of different languages. It can translate words, sentences and web pages be-tween any combination of our supported languages. However, the quality of the translation produced heavily suffer from unknown words problem. English to Bengali phrase-based statistical machine translation was reported by Islam et al. (2010). This system achieved low BLEU score due to small parallel corpus for English-Bengali.

EBMT is better choice for low-resource language, because we can easily add linguistic information into it. Comparing with SMT, we can expect that EBMT performs better with smaller parallel corpus. Moreover, EBMT can translate in good quality when it has good example match. However, it has low-coverage issues due to low parallel corpus. Another research [17] reported a phrasal EBMT for translating English to Bengali. They did not provide any evaluation of their EBMT. They did not clearly explain their translation generation, specially the word reorder mechanism. One research [3] reported an EBMT for translating news headlines. Another related

research [7] proposed EBMT for English-Bengali using WordNet in limited manner.

Chunk parsing was first proposed by Abney [1]. Although EBMT using chunks as the translation unit is not new, it has not been explored widely for low-resource Bengali language yet. [5] used syntactic chunks as translation units for improving insertion or deletion words between two distant languages. However, this approach requires an example base with aligned chunks in both source and target language. In our example-base only source side contains chunks and target side contains corresponding translation string.

Template based approaches increased coverage and quality in previous EBMT. Moreover, [15] showed that templates can still be useful for EBMT with statistical decoders to obtain longer phrasal matches. Manually clustering the words can be a time consuming task. It would be less time consuming to use standard avail-able resources such as WordNet for clustering. That is why in our sys-tem, we used <lexical filename> information for each English words, provided by Word-Net-Online for clustering the proposed CST.

For low-resource language [13] proposed source language adaptation approaches. Their approach needs large corpus which is similar to the low-resource language. However, Bengali has no such similar language which has large parallel corpus. Moreover, their approach requires a low-resource language as target language and a rich language as a source a language.

3. EBMT System Architecture

The Fig. 1 shows the proposed EBMT architecture. The dotted rectangles identified the main contribution area of this research. During the translation process, at first, the input sentence is parsed into chunks using OpenNLP Chunker. The out-put of Source Language Analysis step is the English chunks. Then the chunks are matched with the example-base using the Matching algorithm as described in Section 4.2. This process provides the CSTs candidates from the example-base and it also identify the unknown words in CSTs. In unknown word translation step, using our proposed mechanism in section 4.3, the system finds translation candidates for the identified unknown words from WordNet. Then in Generation process WordNet helps to translate determiners and prepositions correctly to improve MT quality [7]. Finally using the generation rules we output the target-language strings. Based on the EBMT system architecture in Fig. 1, we built an English-to-Bengali EBMT system.

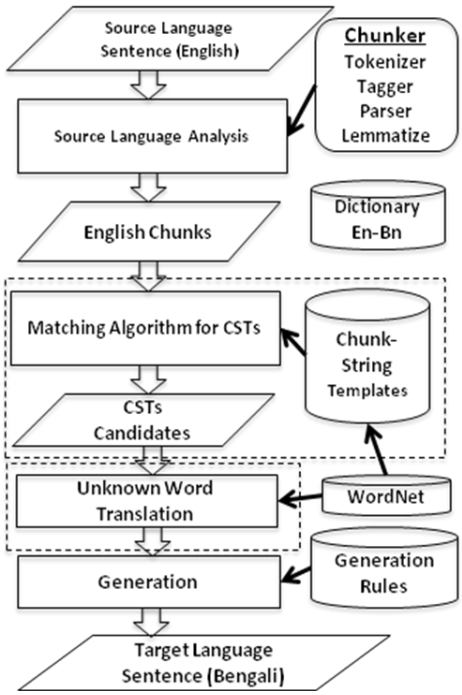

Fig. 1 Proposed EBMT System Architecture.

4. Chunk-String Templates

In this research we proposed EBMT based on chunk-string templates (CST), which is especially useful for developing a MT system for high-resource in source language to low-resource in target language. CST consists of a chunk in the source language (English), a string in the tar-get language (Bengali), and the word alignment information between them. From the English-Bengali aligned parallel corpus CSTs are generated automatically.

Table 1 shows sample word-aligned parallel corpus. Here the alignment infor-mation contains English position number for each Bengali word. For example, the first Bengali word "বিশ্বব্যাপী" is aligned with 11. That means "বিশ্বব্যাপী" is aligned with "worldwide", the 11th word in the English sentence. Although the last Bengali word "মাতৃভাষা" is aligned with 4, the word meaning includes "the native language". Therefore, the alignment information does not have 3rd and 5th words.

Table 1 Example word-aligned parallel corpus.

English	Bengali	Align
Bangla is the native language of around 230 million people worldwide 1 2 3 4 5 6 7 8 9 10 11	বিশ্বব্যাপী বাংলা হচ্ছে প্রায় ২৩০ মিলিয়ন মানুষ –এর মাতৃভাষা	11 1 2 7 8 9 10 6 4

The example-base of our EBMT is stored as CST. CST consists of <c;s;t>, where c is a chunk in the source language (English), s is a string in the target lan-guage (Bengali), and t is the word alignment information between them.

4.1. Generate CSTs

A chunk is a non-recursive syntactic segment which includes a head word with related feature words. In this paper OpenNLP has been used for chunking purpose. For example, "[NP a/DT number/NN]", is a sample chunk. Here NP, DT, NN are parts of speech (POS) Tag defined in Penn Treebank tag set as: proper noun, de-terminers, singular or mass noun. The third brackets "[]" define the starting and ending of a complete chunk.

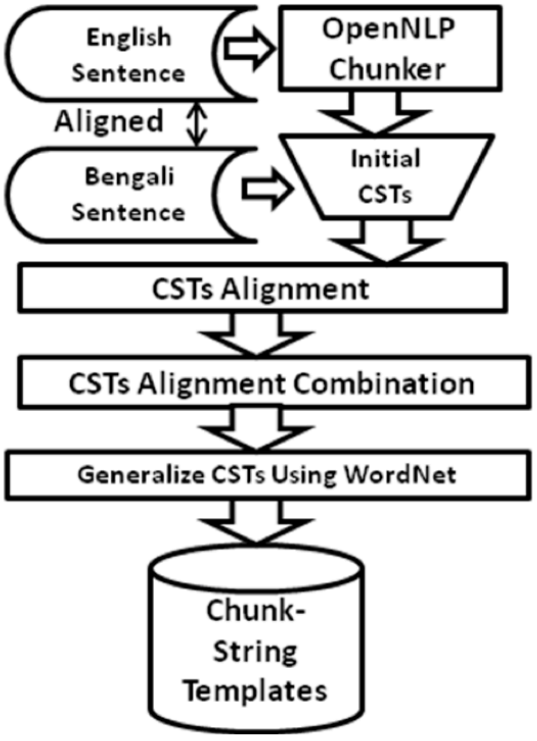

Fig. 2 Steps of CSTs generation

Fig. 2 shows the steps of CSTs generation. First the English chunks are auto generated from a given English sentence. Then initial CSTs are generated for each English chunks from the English-Bengali parallel corpus. Each CSTs alignment for all sentences are generated using the parallel corpus. After that the system gen-erate combinations of CSTs. Finally, the system produces CSTs by generalizing using WordNet to achieve wide-coverage.

4.1.1. OpenNLP Chunker

In the first step, using OpenNLP chunker, we prepare chunks of the English sentences from the word aligned English-Bengali parallel cor-pus.

Input of this step: *"Bangla is the native language of around 230 million people worldwide."*

Output of this step: *"[NP Bangla/NNP] [VP is/VBZ] [NP the/DT native/JJ language/NN] [PP of/IN] [NP around/RB 230/CD million/CD people/NNS] [ADVP worldwide/RB] ./."*

4.1.2. Initial CSTs

In this second step, initial CSTs are generated for each English chunks from the English-Bengali parallel corpus. Table 2 shows the initial CSTs for the word aligned parallel corpus given in Table1.

In Table2 "CST#" is the CSTs number for reference, "C" is the individual Eng-lish Chunks. "S" is the corresponding Bengali Words. "T" represents the align-ment information inside the chunk. "Align" is the same as "Align" in Table 1. "Chunk-Start-Index" equals to the first word position of the chunk in original sen-tence, minus one. For example, from Table 1 we get:

Align=[around,230,million,people]=[7,8,9,10]

Table 2 Example of initial CSTs

CST#	English Chunk (C)	Bengali (S)	T	Align	Chunk-Start-Index
CST1	[NP Bangla/NNP]	বাংলা	1	1	0
CST2	[VP is/VBZ]	হচ্ছে	1	2	1
CST3	[NP the/D native/JJ language/NN]	মাতৃভাষা	2	4	2
CST4	[PP of/IN]	-এর	1	6	5
CST5	[NP around/RB 230/CD million/CD people/NNS]	প্রায় ২৩০ মিলিয়ন মানুষ	1 2 3 4	7 8 9 10	6
CST6	[ADVP worlwide/RB]	বিশ্বব্যাপী	1	11	10

The first word of this chunk is "around", which was in word position 7 of English sentence. Subtracting 1, we get the CST5 chunk-start-index is word position 6. For calculating T, the system subtracts the chunk-start-index from each original word alignment. In the above example, the system subtracts the chunk-start-index 6 from each word alignment. Then we get final alignment, T=[1,2,3,4]

4.1.3. CSTs Alignment

CSTs alignment stores the English word order and Bengali word original sentence alignment information. So that from the initial CSTs the system can reorder the CSTs in Bengali word order.

In this step, the system generates the word order information from Initial CSTs as given in Table 2. For example, Table3 shows the word order information produced from Table 2. "English order" represent the original English chunks or-der and "Bengali order" represents the Bengali chunks order by using CSTs in Table 3. For example, [CST6 CST1 CST2 CST5 CST4 CST3] represents the Bengali sentence "বিশ্বব্যাপী বাংলা হচ্ছে প্রায় ২৩০ মিলিয়ন মানুষ–এর মাতৃভাষা".

Table 3 Example of CSTs alignment

CCST#	English order	Bengali order
CCST1	CST1 CST2 CST3 CST4 CST5 CST6	CST6 CST1 CST2 CST5 CST4 CST3

Fig. 3 visualize the CSTs alignment from Table3.

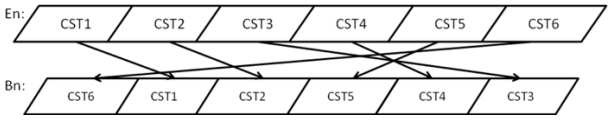

Fig. 3 CSTs alignment

4.1.4. CSTs Alignment Combination

In this step the system generates all possible chunk combinations. The goal is to match source language chunks with as many as possible CSTs. Without CSTs combinations, the system coverage will be low.
From CSTs alignment as given in Table 3, system generates CSTs Combinations. It combines all sequential CSTs. For example, in Fig. 4, circles identified the sequential CSTs combination in Bengali word order. Here CST1 and CST2 can be combined as CCST2, because they are sequential in target language word order.

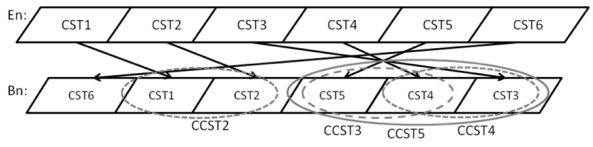

Fig. 4 Chunk alignment

Table 4 contains the whole sentence CCST and the Combined-CSTs (CCSTs) as shown in Fig. 4. The system also produces CSTs combination in source language correspond in target language.

Table 4 CCSTs examples

CCST#	English order	Bengali order
CCST1	CST1CST2 CST3 CST4 CST5 CST6	CST6 CST1 CST2 CST5 CST4 CST3
CCST2	CST1 CST2	CST1 CST2
CCST3	CST4 CST5	CST5 CST4
CCST4	CST3 CST4	CST4 CST3
CCST5	CST3 CST4 CST5	CST5 CST4 CST3

4.1.5. Generalize CSTs Using WordNet

In this step CSTs are generalized by using WordNet to increase the EBMT cover-age. To generalize we consider nouns, proper nouns and cardinal number (NN, NNP, CD in OpenNLP tagset) as our first step. For each nouns (NN) or proper nouns (NNP) the system search for the <lexical filename> in WordNet. If the sys-tem finds the noun or proper noun, the system replaces that with the <lexical file-name> in WordNet. For example, if the system finds "Bangla" in WordNet it re-place it with <noun.communication> . For each cardinal number (CD) we simply replace that cardinal number to <noun.quantity>.

Here we show some example of generalized CSTs produced using WordNet in Table 5.

Table 5 Generalized CSTs

CST#	English Chunk (C)	Generalized Chunk
CST1	[NP Bangla /NNP]	[NP <noun.communication> /NNP]
CST5	[NP around/RB 230/CD million/CD people/NNS]	[NP around/RB <noun.quantity> people/NNS]

Finally, we get the CSTs database which has three tables: initial CSTs, generalized CSTs and CCSTs. From the example word-aligned parallel sentence of Table 1, system generated 6 initial CSTs, 2 Generalized CSTs and 4 Combined-CSTs.

4.2. Matching Algorithm for CSTs

Matching algorithm for CSTs has three components: search in CSTs, search in CCSTs and selecting CCSTs candidates. The Fig. 5 shows the process of our proposed matching algorithm. The input is the English chunks from the source language sentence. At first the system finds candidate CSTs for each SL chunks from initial CSTs. Search for each chunks using initial CST. Until all chunks are matched the system generalizes the input chunks and search in generalized CSTs. Finally, the system selects best CSTs combination from all the CCSTs candidates.

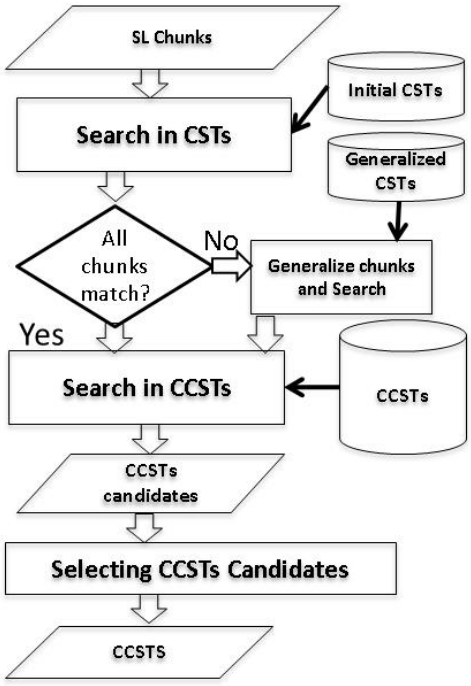

Fig. 5 Matching Algorithm for CSTs

4.2.1. Search in CSTs

To search in CSTs our system first tries to find each chunk in initial CSTs. If it does not have exact match, it tries to find the linguistically related matches in generalized CSTs. Linguistically relations are determined by POS tags given in source-language chunks and the information provided by WordNet. Finally this step provides a set of matched CSTs. All SL chunks can be matched with at CSTs, generalized CSTs; or marked as unknown word otherwise.

For example, we have 3 input chunks: [NP English/NNP][VP is/VBZ][NP the/DT native/JJ language/NN]. Second and third chunks are matched with CST2 and CST3 of initial CSTs in Table 2. But the first chunk [NP English/NNP], has no match. Then using WordNet the system generalized the input chunk "[NP English/NNP]" into "[NP <noun.communication>/NNP]". It matched with CST1 of Table 5. This step returns a set of matched CSTs [CST1, CST2, CST3] and match level (as described in section 4.2.3).

4.2.2. Search in CCSTs

The second step is to search the matched CSTs in CCSTs. The system performs all order CSTs combination search.

And it returns CCSTs candidates. For the above example, it returns [CCST1, CCST2, CCST4,CCST5] because these CCSTs include at least one matched CST in [CST1,CST2,CST3]. As this example if more than one CCSTs matches the CSTs, it returns all the CCSTs candidates, to select the best one in the next step.

4.2.3. Selecting CCSTs candidates

In this step using our selection criteria we choose the suitable CCSTs. From the set of all CCSTs candidates this algorithm selects the most suitable one, according to the following criteria:

1. The more CSTs matched, the better;
2. Linguistically match give priority by following these ranks, higher level is better:

Level 4: Exact match.
Level 3: <lexical filename> of WordNet and POS tags match
Level 2: <lexical filename> of WordNet match
Level 1: Only POS tags match
Level 0: No match found, all unknown words.

4.3. Unknown Word Translation

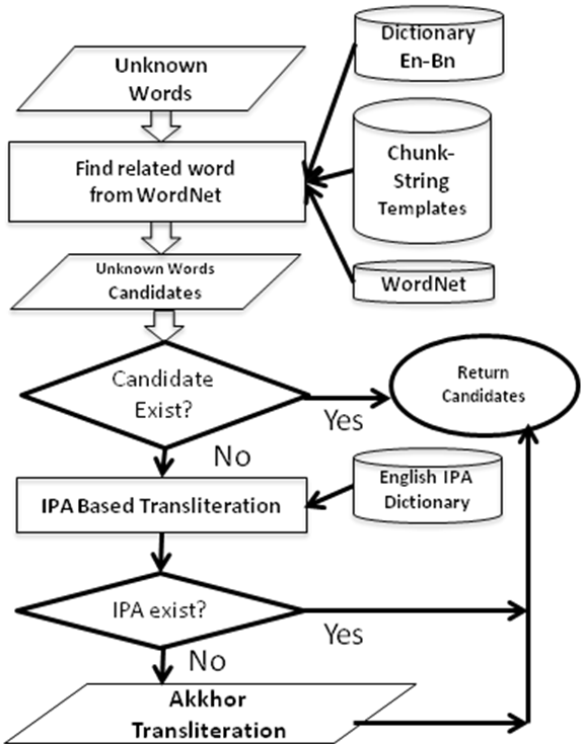

Fig. 6 Steps of handling unknown words

As in our assumption, the main users of this EBMT will be monolingual people; they cannot read or understand English words written in English alphabet. However, with related word translation using WordNet and Transliteration can give them some clues to understand the sentence meaning. As Bangla language accepts foreign words, transliterating an English word into Bangla alphabet, makes that a Bangla foreign word. For example, in Bangla there exist many words, which speakers can identify as foreign words. Fig. 6 shows the unknown words translation process in a flow chart. Proposed system first tries to find semantically related English words from WordNet for the unknown words. From these related words, we rank the translation candidates using WSD technique and English-Bangla dictionary. If no Bangla translation exists, the system uses IPA-based-transliteration. For proper nouns, the system uses transliteration mechanism provided by Akkhor Bangla Software.

4.3.1. Find Sublexical Translations

For sublexical matching our system divide the unknown word into sublexical units and then find possible translation candidates from these sublexical units. For this the system use following steps:

(1) Find the possible sublexical units of the unknown word. For example, the unknown word “bluebird” gets divided into “blue” and “bird”.
(2) Extract sublexical translations and restrain translation choices.
(3) Remove less probable sublexical translations
(4) Output translation candidates with the POS tags for the sublexical units of the unknown word.

From the set of all CSTs we select the most suitable one, according to the fol-lowing criteria:

1. The more exact CSTs matched, the better;
2. Linguistically match give priority by following these ranks, higher level is better:

•Level 4: Exact match.
•Level 3: Sublexical unit match, <lexical filename> of WordNet and POS tags match
•Level 2: Sublexical unit match, <lexical filename> of WordNet match
•Level 1: Only POS tags match.
•Level 0: No match found, all unknown words.

4.3.2. Find Candidates from WordNet

Due to small English-Bangla parallel corpus availability, there is high probability for the MT system to handle unknown words. Therefore, it is important to have a good method for translating unknown words. When the word has no match in the CSTs, it tries to translate using English WordNet and bilingual dictionary for English-Bangla. Input of this step is unknown words. For example, “canine” is an unknown word in our system.

Output of this process is the related unknown words translation.

Find Candidates from WordNet Synonyms

The system first finds the synonyms for the unknown word from the WordNet synsets. Each synset member becomes the candidate for the unknown word. WordNet provide related word for nouns, proper nouns, verbs, adjectives and ad-verbs. Synonymy is WordNet's basic relation, because WordNet uses sets of synonyms (synsets) to represent word senses. Synonymy is a symmetric relation be-tween word forms. We can also use Entailment relations between verbs available in WordNet to find unknown word candidate synonyms.

Find Candidates Using Antonyms

WordNet provide related word for nouns, proper Antonymy (opposing-name) is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs. For some unknown we can get the antonyms from WordNet. If the antonym exists in the dictionary we can use the negation of that word to translate the unknown word. For example, "unfriendly" can be translated as "not friendly". In Bengali to negate such a word we can simply add "না" (na) at the end of the word. So, "unfriendly" can be translated as "বন্ধুত্বপূর্ণ না" (bondhuttopurno na). It helps to translate unknown words like "unfriendly", which improves the machine translation quality.

Hyponymy (sub-name) and its inverse, hypernymy (super-name), are transitive relations between synsets. Because there is usually only one hypernym, this se-mantic relation organizes the meanings of nouns into a hierarchical structure. We need to process the hypernyms to translate the unknown word.

Find Candidates Using Hypernyms

For nouns and verbs WordNet provide hypernyms, which is defined as follows:

Y is a hypernym of X if every X is a (kind of) Y.

For example "canine" is a hypernym of noun "carnivore", because every dog is a member of the larger category of canines. Verb example, "to perceive" is an hypernym of "to listen". However, WordNet only provides hypernym(s) of a synset, not the hypernym tree itself. As hypernyms can express the meaning, we can translate the hypernym of the unknown word. To do that, until any hypernym's Bangla translation found in the English-Bangla dictionary, we keep discovering upper level of hypernym's. Because, nouns and verbs are organized into hierarchies, defined by hypernyms or is-a-relationships in WordNet. So, we considered lower level synset words are generally more suitable then the higher level synset words.

This process discovers the hypernym tree from WordNet in step by step. For example, from the hypernym tree of dog, we only had the "animal" entry in our English-Bengali dictionary. Our system discovered the hypernym tree of "dog" from WordNet until "animal". Following is the discovered hypernym tree:

```
dog, domestic dog, Canis familiaris
=> canine, canid
  => carnivore
    => placental, placental mammal, …
      => mammal
        => vertebrate, craniate
          => chordate
            => animal  => ...
```

This process search in English-Bangla dictionary, for each of the entry of this hypernym tree. So at first we used the IPA representation of the English word from our dictionary, then using transliterating that into Bengali. Then system produce "a kind of X" - এক ধরনের X [ek dhoroner X]. For the example of "ca-nine" we only had the Bengali dictionary entry for "animal" from the whole hypernym tree. We translated "canine" as the translation of "canine, a kind of animal", in Bangla which is "ক্যানাইন, এক ধরনের পশু" [kjanain, ek dhoroner poshu].

Similarly, for adjectives we try to find "similar to" words from WordNet. And for Adverbs we try to find "root adjectives".

Finally, this step returns unknown words candidates from WordNet which exist in English-Bangla dictionary.

Using the same technique described above, we can use Troponyms and Meronyms to translate unknown words. Troponymy (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower. Meronymy (part-name) and its inverse, holonymy (whole-name), are complex se-mantic relations. WordNet distinguishes component parts, substantive parts, and member parts.

4.4. Rank Candidates

To choose among the candidates for the unknown word, we need to rank all the candidates. Especially polysemous unknown words need to select the adequate WordNet synset to choose the right candidate. The system performs Google search with the input sentence as a query, by replacing the unknown word with each candidate words. We add quotation marks in the input sentence to perform phrase searches in Google, to find the number of in documents the sentence appear together. If the input sentence with quotation mark returns less than 10 results, we perform Google search with four and two neighbor chunks. Finally, the system ranks the

candidate words using the Google search hits information.

For example, the input sentence in SL is: This dog is really cool. The system first adds double quotation with the input sentence: “This dog is really cool”, which returns 37,300 results in Google. Then the system replaces the un-known word “dog” from discovered hypernym tree. Only for "This animal is really cool.", returned 1,560 results by Google. That is why “animal” is the second most suitable candidate for “dog”. However, other options "This domestic dog is really cool." or "This canine is really cool." etc. returns no results or less than 10 results in Google. So in this case we search with neighbor chunks only. For example, in Google we search with:

"This mammal is" returns 527,000 results;
"This canid is" returns 503,000 results;
"This canine is" returns 110,000 results;
"This carnivore is" returns 58,600 results;
"This vertebrate is" returns 2,460 results;
"This placental is" returns 46 results;
"This craniate is" returns 27 results;
"This chordate is" returns 27 results;
"This placental mammal is" returns 6 result;

Finally, the system returns the unknown word candidates: mammal, canid, canine, carnivore, vertebrate, placental, craniates, chordate, placental mammal.

4.4.1. Final Candidate Generation

In this step, we choose one translation candidate. If any of the synonyms or candidate word exist in English-Bangla dictionary, the system translates the un-known word with that synonym meaning. If multiple synonyms exist, then the entry with highest Google search hits get selected. English-Bangla dictionary also contains multiple entries in target language. For WSD analysis in target language, we perform Google search with the produced translation by the system. The sys-tem chooses the entry with highest Google hits as final translation of the un-known word. For example, for unknown word “dog”, animal get selected in our system. However, if there were no candidates, we use IPA-Based-Transliteration.

4.4.2. IPA-Based-Transliteration

When unknown word is not even found in WordNet, we use IPA-Based trans-literation using the English IPA Dictionary (Salam et. al., 2011). Output for this step is the Bangla word transliterated from the IPA of the English word. In this step, we use English-Bangla Transliteration map to transliterate the IPA into Bangla alphabet. From English IPA dictionary the system can obtain the English words pronunciations in IPA format. Output for this step is the Bengali word transliterated from the IPA of the English word. In this step, we use following English-Bengali Transliteration map to transliterate the IPA into Bengali alphabet. Fig. 7 shows our proposed English-Bengali IPA chart for vowels, diphthongs and consonants. Using rule-base we transliterate the English IPA into Bangla alpha-bets. The above IPA charts leaves out many IPA as we are considering about translating from English only. To translate from other language such as Japanese to Bangla we need to create Japanese specific IPA transliteration chart. Using the above English-Bangla IPA chart we produced transliteration from the English IPA dictionary. For examples: pan(pæn): প্যান; ban(bæn): ব্যান; might(maIt): মাইট.

However, when unknown word is not even found in the English IPA dictionary, we use transliteration mechanism of Akkhor Bangla Software as given in Fig. 7. For example, for the word “Muhammod” which is a popular Bangla name, Akkhor transliterated into “মুহাম্মদ” in Bangla.

বাংলা	অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
English	A	a/aa/ঁa	i/ঁi	I/ee/`I	u/ঁu	U/ঁU	ri/ঁri	e/ঁe	oi/`oi	o/ঁo	ou ou

বাংলা	ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ
English	k	kh	g	gh	Ng	ch	Ch	j	jh	Y
বাংলা	ত	থ	দ	ধ	ন	ট	ঠ	ড	ঢ	ণ
English	t	th	d	dh	n	T	Th	D	Dh	N
বাংলা	প	ফ	ব	ভ	ম	য	র	ল	শ	ষ
English	p	f/ph	b	bh/v	m	z	r	l	sh	S
বাংলা	গ	ক্ষ	হ্	ড়	ঢ়	য়	ং	ঃ	ঁ	
English	S	k-S	h	R	rh	y	ng	:	~	
বাংলা	১	২	৩	৪	৫	৬	৭	৮	৯	০
English	1	2	3	4	5	6	7	8	9	1
বাংলা	কা	কে	কি	কু	কো	ক্র	ক্রে	ক্রি	ক্রু	ক্রু
English	ka	ke	ki	ku	kO	kro	kre	kre e	kru	krU
বাংলা	কী	চী	মী	কূ	মূ	বূ	পূ	নূ	ক্য	ব্য
English	kI	chI	mI	kU	mU	bU	NU	nU	k-z	b-z

Fig. 7 Akkhor phonetic mapping for Bengali alphabets

Mouth narrower vertically	[i:]ই/ি sleep /sli:p/	[I] ই / ি slip /slIp/	[ʊ] উ / ু book /bʊk/	[u:]ঊ/ ু boot /bu:t/
	[e]এ/ে ten /ten/	[ə] আ / া after /a:ftə/	[ɜ:] আ / া bird /bɜ:d/	[ɔ:] র্ bored /bɔ:d/
Mouth wider vertically	[æ]এ্যা/ ্যা cat /kæt/	[^] আ /া cup / k^p/	[ɑ:] আ / া car / cɑ:r/	[ɒ] অ hot /hɒt/

English-Bengali IPA mapping for vowels

[Iə] ইয়া/িয়া beer /bIər/	[eI] এই/ েই say /seI/	
[ʊə] উয়া/ ুয়া fewer /fjʊər/	[ɔI] অয়/য় boy /bɔI/	[ə ʊ] ও / ো no /nəʊ/
eə ঈয়া/ ীয়া bear /beər/	[aI] াই / আই high /haI/	[aʊ] আউ /াউ cow /kaʊ/

English-Bengali IPA mapping for diphthongs

[p] প pan /pæn/	[b] ব ban /bæn/	[t] ট tan /tæn/	[d] ড day /deI/	[ʧ] চ chat /ʧæt/	[ʤ] জ judge /ʤ^ʤ/	[k] ক key /ki:/	[g] গ get /get/
[f] ফ fan /fæn/	[v] ভ van / væn/	[θ] থ thin /θIn/	[ð] দ than /ðæn/	[s] স sip /sIp/	[z] জ zip / zIp/	[ʃ] শ ship /ʃIp/	[ʒ] স vision /vIʒ^n/
[m] ম might /maIt/	[n] ন night /naIt/	[ŋ]ং/ঙ thing /θIŋ/	[h] হ height /haIt/	[l] ল light /laIt/	[r] র right /raIt/	[w] য় white/hwaIt/	[j]ইয়ে yes /jes/

English-Bengali IPA mapping for consonants

Fig. 8 English-Bengali IPA mapping

4.5. Translation Generation

In this EBMT architecture we used Rule-Based generation method. Using dictionary and WordNet rules, we can accurately translate the determiners in Bengali. For translating determiner, we adapted [7] proposals to use WordNet.

To reorder the CSTs for partial match in CCSTs, we remove the unmatched CSTs. Based on the morphological rules we change the expression of the words.

Here WordNet provided required information to translate polysemous deter-miners accurately. The system compared with the <lexical filename> of WordNet for the word NN. If the word NN is "<noun.person>", then determiner "a" will be translated as "ekjon". Otherwise "a" will be translated as "ekti".

For example "a boy" should be translated to "ekti chele" as boy is a person. "ekkhana chele" is a wrong translation, because "ekkhana" can be used only for NNs which is not a person.

For Bengali word formation we have created morphological generation rules especially for verbs. These rules are constructed by human.

5. Evaluation

We did wide-coverage and quality evaluations for the proposed EBMT with CSTs, by comparing with baseline EBMT system. Wide-coverage evaluation measures the increase of translation coverage. Quality evaluation measures the translation quality through human evaluation. Because of the unavailability of large parallel corpus for English-Bengali language pair, we could not evaluate the BLEU score.

Baseline system architecture has the same components as described in Figure 1, except for the components inside dotted rectangles. Matching algorithm of baseline system is that not only match with exact translation examples, but it can also match with POS tags. The Baseline EBMT use the same training data: English-Bengali parallel corpus and dictionary, but does not use CSTs, WordNet and unknown words translation solutions. Currently from the training data set of 2,000 word aligned English-Bengali parallel corpus, system generated 15,356 initial CSTs, 543 Generalized CSTs and 12,458 Combined-CSTs.

Table 6. Different category of test-set sentences

Sentence Type	Number of Sentences
Simple	136
Complex (Wh-Clause)	50
Complex (Infinitive Clause)	50
Unknown words	100
Total	336

The development environment was in windows using C Sharp language. Our test-set contained 336 sentences, which are not same as training data. The test-set includes simple and complex sentences, representing various grammatical phenomena. Table 6 shows the distribution of sentences in different categories. We have around 20,000 English-Bengali dictionary entries.

5.1. References

We calculated the rate of generalized CSTs usage to evaluate the achievement of wide-coverage. To match the English input chunks, baseline EBMT use translation examples and POS matching mechanism from the training data. On the other hand, proposed EBMT use CSTs to match the English input chunks. Table 7 shows the contribution of CSTs to achieve wide-coverage. Here wide-coverage = No. of Matched English chunks / No. of all English chunks in test-set. CSTs improved the wide-coverage by 57 points.

Table 7. Wide-Coverage Comparison

System Modules	wide-coverage
Baseline EBMT	23%
Proposed EBMT with CSTs	80%

5.2. Quality Evaluation

5.2.1. CSTs Evaluation

Quality evaluation measures the translation quality through human evaluation. Table 8 shows the human evaluation of the proposed EBMT system with CSTs only.

Table 8. Human Evaluation using same test-set

Translation Quality	Grade	EBMT+ CSTs	Google Translate
Perfect Translation	A	25.60	19.00
Good Translation	B	38.69	30.00
Medium Translation	C	19.64	27.00
Poor Translation	D	16.07	24.00
Total		**100%**	**100%**

Table 9 shows the explanation of translation quality used in our human evaluation process. Word selection means whether the system could choose a correct word candidate. Word order measures whether the words position in the translated sentence is grammatically correct. Functional word usage means whether the system could choose a correct functional word. Considering

these quality elements, we have evaluated the translation quality.
Perfect Translation means there is no problem in the target sentence. Good Translation means the target sentence is not grammatically correct because of wrong functional word, but still understandable for human.

Currently 64.29% of the test-set translations produced by the system were acceptable, produced by the system with proposed CSTs only. Around 48.81 points of poor translation produced by EBMT Baseline was improved using the proposed system with CSTs.

Table 9. Human Evaluation quality explanation

Translation Quality	Word Selection	Word Order	Functional Word Usage	Example Translation produced
Perfect Translation	YES	YES	YES	জাপানিজ হচ্ছে প্রায় ১২০ মিলিয়ন মানুষ–এর মাতৃভাষা
Good Translation	YES	YES	NO	প্রস্তুতি ম্যাচ সেঞ্চুরিটা কাজ লাগল ইমরুল কায়েসের
Medium Translation	YES	NO	YES/ NO	ইমরুল ফর্ম দেওয়া ফিরেছেন বাংলাদেশ ফেরার ইঙ্গিত
Poor Translation	NO	NO	NO	Roseland ওয়েষ্ট Pullman, এবং Riverdale আট ক্যাথলিক প্যারিশ সমন্বয়ে

For example, in Table 9 we graded the example translation produced in Good Translation category because the words ম্যাচ and কাজ is not in correct functional word form but still the sentence is understandable by human. Medium means there are several mistakes in the target sentence, like wrong functional word and wrong word order. In the ex-ample sentence we can see that the word order doesn't make any sense even though the word selection is correct. So human cannot understand the translated sentences in medium category. Poor Translation means there are major problems in the target sentence, like non-translated words, wrong word choice and wrong word order. In the example, the produced translation does not make sense due to wrong word selection with wrong word order.

Only perfect and good translations were "acceptable". Because even though the system chooses the correct word without generating the correct word order the translated sentence will be grammatically incorrect and may not be understandable.

The identified main reasons for improving the translation quality is our solution using CSTs generalization and sub-sentential match. Because of these contributions of CSTs some test-set sentence improved from "poor" or "medium" translation to "acceptable" translation.

Table 10 shows some good example translations comparison between EBMT+CSTs and Google Translate. It also shows the translation quality in bracket (A,B,C,D: Perfect, Good, Medium, Poor). In #1 translation, Google translate mistranslated the word "million" to "crore". Google translate could not translate #2 properly with wrong word choices and wrong word orders which results a poor translation quality. In #3 translation, it shows Google never translation the digits into Bangla digits and the word orders are not correct like other complex sentences. On the other hand, CSTs

performed well in all 3 examples, which demonstrate the goodness of using CSTs for low resource language.

Table 10. Comparison of CSTs output with Google Translate using same testset

#	English	EBMT+CSTs	Google Translate
	Japanese is the native language of around 120 million people	জাপানিজ হচ্ছে প্রায় ১২০ মিলিয়ন মানুষ–এর মাতৃভাষা (A)	জাপানি প্রায় 120 কোটি মানুষের মাতৃভাষা (C)
	The name Bangladesh was originally written as two words, Bangla Desh	বাংলাদেশ নামটি মূলত দুই শব্দে লেখা হতো, বাংলা দেশ (A)	নাম বাংলাদেশ মূলত দুই বোথ ওয়ার্ল্ডস, বাংলাদেশ হিসেবে লেখা হয়েছি (D)
	After high school, Obama moved to Los Angeles in 1979 to attend Occidental College.	হাই স্কুলের পরে, ওবামা লস এঞ্জেলেস এ ১৯৭৯ সালে অকসিডেন্টাল কলেজে যায় (A)	উচ্চ বিদ্যালয় পরে, ওবামা অক্সিডেন্টাল কলেজে যোগ দিতে 1979 সালে লস এঞ্জেলেস সরানো. (C)

We observed some drawbacks of using CSTs with generalization using WordNet as well. Sometimes our system chooses the wrong synset from the WordNet. As a result, some test-set still produced "poor" translation.

As we used same test-set, the result of Google MT is same for both Table 8 & 11. Our EBMT could translate better than Google because of our novel CSTs and unknown words translation mechanism.

5.2.2. Unknown Words Evaluation

Table 11. Human Evaluation of Unknown words using same test-set

Translation Quality	Grade	EBMT+CSTs+Unknown Words	Google Translate
Perfect Translation	A	30.95	19.00
Good Translation	B	36.90	30.00
Medium Translation	C	18.75	27.00
Poor Translation	D	13.39	24.00
Total		100.00	100.00

We also did quality evaluation for our unknown words solution. Table 11 shows the human evaluation of the EBMT system with CSTs and unknown word solution. Currently 67.85% of the test-set translations were acceptable, produced by the system with proposed CSTs and unknown words solutions. Comparing with EBMT+CSTs, unknown words mechanism improved translation quality by 3.56 points in human evaluation. We also compare our system with Google translate which is the most popular MT system for English-Bengali language pair.

Table 12 shows sample translation examples produced by EBMT+CSTS with unknown words solution compared with Google translate. It also shows the translation quality in bracket (A,B,C,D: Perfect, Good, Medium, Poor). As "aardvark" and "dog" are unknown words, Google translate produced medi-um translation for #1 and #2. As a result the translation quality improved to "good" quality. All these examples

Table 12. Human Evaluation of unknown words using same testset

#	English	EBMT+CSTs+Unknown Words	Google Translate
1	Are you looking for an aardvark?	আপনি কি আর্ডভার্ক, এক ধরনের পশু খুঁজছেন?(A)	আপনি একটি aardvark খুঁজছেন? (C)
2	This dog is really cool.	ডগ, এক ধরনের পশু আসলেই দারুন (A)	এই কুকুর সত্যিই শীতল হয়. (C)
3	WordNet is a lexical database for the English language.	শব্দজাল হচ্ছে ইংরেজি ভাষার জন্য একটি আভিধানিক ডাটাবেস (A)	WordNet ইংরেজি ভাষার জন্য একটি আভিধানিক ডাটাবেস. (B)
4	Sublexical units of a word are selected in parallel and are subsequently ordered.	শব্দের উপ-আভিধানিক অংশ নির্বাচন করা হয় সমান্তরাল ভাবে এবং অনুক্রম অনুসারে (A)	একটি শব্দের sublexical ইউনিট সমান্তরাল মধ্যে নির্বাচন করা হয় এবং পরবর্তীকালে আদেশ করা হয়(D)
5	The bluebird are a group of medium-sized, mostly insectivorous or omnivorous bird in the world	নীলপাখি হচ্ছে এক ধরণের পাখি যা বিশ্বের মাঝারি আকারের, সাধারণত কীটভক্ষক এবং সর্বভুক পাখির একটি গ্রুপ (A)	bluebirds বিশ্বের মাঝারি আকারের, বেশির ভাগ কীটভক্ষক এবং সর্বভুক পাখি একটি গ্রুপ (D)

demonstrate the effectiveness of our proposed solution for translating unknown words.

6. Conclusion and Future Works

6.1. Wide-Coverage of Adequate Determiner Evaluation

As we used WordNet to translate using adequate determiner, we measured the increase of translation coverage as following:

$$\text{wide} - \text{coverage} = \frac{\text{No. of system generate adequate determiner}}{\text{No. of all adequate determiner}}$$
(from example Human evaluation sentences)

Table 13 shows the EBMT system performance improvement for the test data of 336 sentences. In these test sentences we had 107 adequate determiners. The baseline EBMT produced 34 adequate determiners, which is 24% of all adequate determiners. The proposed EBMT produced 93 adequate determiners, which is 65% of all adequate determiners. Our proposed EBMT system improved the wide-coverage of adequate determiners by 41 points. We found generalized CSTs are also effective for achieving wide-coverage in translating determiners.

Table 12. Wide-Coverage Comparison

System Modules	wide-coverage
Baseline EBMT	24%
Proposed EBMT with WordNet	65%

6.2. Grammatical Structures of test-set sentences

English sentences in our test-set can be classified in four types: Declarative, Imperative, Interrogative and Exclamatory sentences. These sentences can also be classified using following complexity types: Simple, Compound, Complex and Compound-Complex. Current EBMT system performance depend on the quality of English chunker.

7. Conclusion and Future Works

We proposed to use ontology to improve the quality of EBMT system for low-resource language. Our EBMT system is effective for low resource language like Bengali. We used WordNet to translate the unknown words which are not directly available in the dictionary. To translate an English sentence, it is first parsed into chunks. Then the chunks matched with the CSTs to find translation candidates. Then the system determines translation candidates for the identified unknown words from WordNet. Finally using generation rules the target-language strings has been produced.

Using this method, our proposed EBMT system improved the wide-coverage by 57 points and quality by 48.81 points in human evaluation. Currently 64.29% of the test-set translations by the system were acceptable. Because our system can generate more general CSTs, and it increases the quality for low-resource language. Unknown words mechanism improved translation quality by 3.56 points in human evaluation. Currently 67.85% of the test-set translations were acceptable, produced by the system with proposed CSTs and unknown words solutions.

In future we would like to use statistical language model to improve the generation quality. We would like to evaluate the system with other high resource to low-resource language pair.

References

1. Abney, Steven. 1991. Parsing by chunks. In Principle-Based Parsing, Kluwer Academic Publishers. pages 257–278.
2. Diganta Saha, Sivaji Bandyopadhyay. 2006. A Semantics-based English-Bengali EBMT System for translating News Headlines. Proceedings of the MT Summit X, Second workshop on Example-Based Machine Translation Programme.
3. Diganta Saha, Sudip Kumar Naskar, Sivaji Bandyopadhyay. 2005. A Semantics-based English-Bengali EBMT System for translating News Head-lines, MT Summit X.
4. George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.
5. Jae Dong Kim, Ralf D. Brown, Jaime G. Carbonell. 2010. Chunk-Based EBMT. EAMT, St Raphael, France.
6. Karim, M. A., ed. Technical challenges and design issues in bangla language processing. IGI Global, 2013.
7. Khan Md. Anwarus Salam, Mumit Khan and Tetsuro Nishino. 2009. Example Based English-Bengali Machine Translation Using WordNet. TriSAI, Tokyo.
8. Khan Md. Anwarus Salam, Yamada Setsuo and Tetsuro Nishino. 2010. English-Bengali Parallel Corpus: A Proposal. TriSAI, Beijing.
9. Khan Md. Anwarus Salam, Setsuo Yamada and Tetsuro Nishino. 2011a. Example-Based Machine Translation for Low-Resource Language Using Chunk-String Templates, 13th Machine Translation Summit, Xiamen, China.
10. Khan Md. Anwarus Salam, Setsuo Yamada and Tetsuro Nishino, "Using WordNet to Handle the Out-Of-Vocabulary Problem in English to Bangla Machine Translation", Global WordNet Conference, Matsue, Japan, (Editors Christiane Fellbaum et. al., Tribun EU, Brno, 2012, ISBN 978-80-263-0244-5). Page 35-39. January 2012b.
11. Md. Musfique Anwar, Mohammad Zabed Anwar and Md. Al-Amin Bhuiyan. 2009. Syntax Analysis and Machine Translation of Bangla Sentences. Inter-national Journal of Computer Science and Network Security, 09(08),317–326.
12. Md. Zahurul Islam, Jörg Tiedemann & Andreas Eisele. 2010. English to Bangla phrase-based machine translation. Proceedings of the 14th Annual conference of the European Association for Machine Translation.
13. Pidong Wang, Preslav Nakov, and Hwee Tou Ng. 2016. Source language adaptation approaches for resource-poor machine translation. Comput. Linguist. 42, 2 (June 2016), 277-306.
14. Roy, Maxim. "Machine Learning Approaches for Bangla Statistical Machine Translation." Technical Challenges and Design Issues in Bangla Language Processing. IGI Global, 2013. 79-95.

15. R. Gangadharaiah, R. D. Brown, and J. G. Carbonell. Phrasal equivalence classes for generalized corpus-based machine translation. In Alexander Gelbukh, editor, Computational Linguistics and Intelligent Text Processing, volume 6609 of Lecture Notes in Computer Science, pages 13–28. Springer Berlin / Heidelberg, 2011.
16. Sajib Dasgupta, Abu Wasif and Sharmin Azam. 2004. An Optimal Way Towards Machine Translation from English to Bengali, Proceedings of the 7th International Conference on Computer and Information Technology (ICCIT).
17. Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2006b. Handling of Prepositions in English to Bengali Machine Translation. In the proceedings of Third ACL-SIGSEM Workshop on Prepositions, EACL 2006. Trento, Italy.
18. Sudip Kumar Naskar, Sivaji Bandyopadhyay. 2006a. A Phrasal EBMT Sys-tem for Translating English to Bengali. Workshop on Language, Artificial Intelligence and Computer Science for Natural Language Processing applications (LAICS-NLP).
19. Zhanyi Liu, Haifeng Wang And Hua Wu. 2006. Example-Based Machine Translation Based on Tree-string Correspondence and Statistical Generation. Machine Translation, 20(1): 25-41