Example-Based Machine Translation for Low-Resource Language Using Chunk-String Templates

Khan Md. Anwarus Salam
The University of Electro-Communications
Chofu, Tokyo, Japan
anwarus@ice.uec.ac.jp

Setsuo Yamada
NTT Corporation
Tokyo, Japan
yamada.setsuo@lab.ntt.co.jp

Tetsuro Nishino
The University of Electro-Communications
Chofu, Tokyo, Japan
nishino@ice.uec.ac.jp

Abstract

Example-Based Machine Translation (EBMT) for low resource language, like Bengali, has low-coverage issues, due to the lack of parallel corpus. In this paper, we propose an EBMT for low resource language, using chunk-string templates (CSTs) and translating unknown words. CSTs consist of a chunk in source-language, a string in target-language, and word alignment information. CSTs are prepared automatically from aligned parallel corpus and WordNet. To translate unknown words, we used WordNet hypernym tree and English-Bengali dictionary. If no translation candidate found, system transliterates the word. Proposed EBMT improved wide-coverage by 41 points and quality by 48.81 points in human evaluation.

1 Introduction

Bengali is the native language of around 230 million people worldwide, mostly from Bangladesh. According to “Human Development Report 2009”[1] 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, current commercial MT systems do not support Bengali language translation. However, there are several attempts in building English-Bengali MT system. The first available free MT system from Bangladesh was Akkhor Bangla Software[2]. The second available online MT system was apertium based Anubadok[3]. These systems used Rule-Based approach and did not consider about improving translation coverage by handling unknown words, in low-resource scenario.

In present, there are several ways of Machine Translation such as Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT) and Example-Based Machine Translation (EBMT).

RBMT require human made rules, which are very costly in terms of time and money, but still unable to translate general-domain texts.

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.

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.

[1] http://hdr.undp.org/en/reports/global/hdr2009/

[2] www.akkhorbangla.com

[3] anubadok.sourceforge.net

We considered achieving wide-coverage of EBMT by improving the translation quality. Currently, English has rich language resources like automated parser, tokenizer and WordNet. On the other hand Bengali is a low-resource language. In this scenario, we consider to use rich-resource source-language (SL) like English and low-resource target-language (TL) like Bengali.

In this paper, we propose an EBMT for low resource language, using chunk-string templates (CSTs) and translating unknown words.

CSTs consist of a chunk in the source language (English), a string in the target language (Bengali), and the word alignment information between them. CSTs are generated from the aligned parallel corpus and WordNet, by using English chunker. WordNet (Miller 2005) is a large lexical database of English, where nouns, verbs, adjectives and adverbs are grouped into clusters using <lexical filename> information. For clustering CSTs, we used <lexical filename> information for each words, provided by WordNet-Online[4].

To translate unknown words we used WordNet hierarchy of hypernym tree and an 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.

Based on the above methods, we built an English-to-Bengali MT system. Our proposed EBMT improved the wide-coverage of adequate determiners by 41 points and quality by 48.81 points. Currently 64.29% of the test set translations produced by the system were acceptable.

2 Related Works

Chunk parsing was first proposed by Abney (1991). Although EBMT using chunks as the translation unit is not new, it has not been explored widely for low-resource Bengali language yet. Kim et al. (2010) 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, Gangadharaiah et al. (2011) 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 available resources such as WordNet for clustering. That is why in our system, we used <lexical filename> information for each English words, provided by WordNet-Online for clustering the proposed CST.

Dasgupta et.al. (2004) proposed to use syntactic transfer. They converted CNF trees to normal parse trees and using a bilingual dictionary, generated output translation. This research did not consider translating unknown words.

Naskar et al. (2006a), 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.

Saha et al. (2005) reported an EBMT for translating news headlines. Their works showed that EBMT can be a good approach for Bengali language. Their approach only considered about news headlines.

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.

Salam et al. (2009) proposed EBMT for English-Bengali using WordNet in limited manner.

3 EBMT Architecture

The Figure 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[5] Chunker. The output 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 5. This process provides the CSTs candidates from the example-base. It also mark the unknown words. In Unknown Word

[4] http://wordnetweb.princeton.edu/perl/webwn

[5] opennlp.sourceforge.net

Translation step, using our proposed mechanism in section 6, we try to find translation candidates for those unknown words. Then in Generation process WordNet helps to translate determiners and prepositions correctly to improve MT performance (Salam et al. 2009). Finally using the generation rules we output the target-language strings. Based on the above MT system architecture, we built an English-to-Bengali MT system.

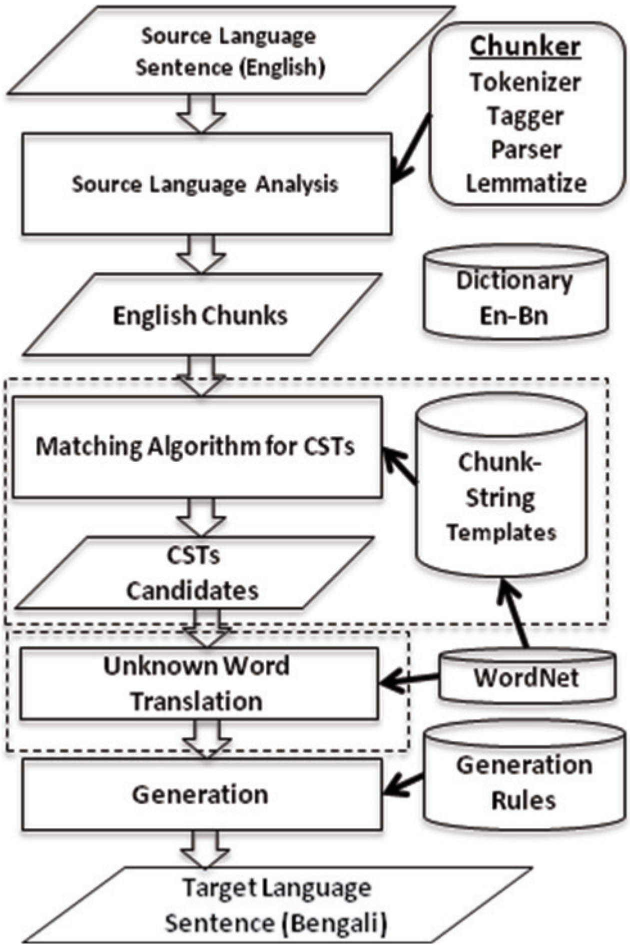

Figure 1: Proposed EBMT Architecture

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 to low-resource language. CST consists of a chunk in the source language (English), a string in the target 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 information 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.

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

Table 1: Example word-aligned parallel corpus

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 language (Bengali), and t is the word alignment information between them.

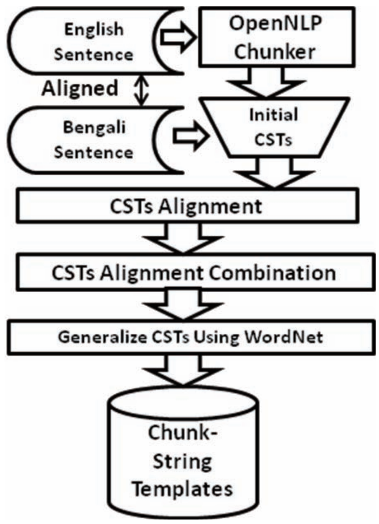

Figure 2: Steps of CSTs generation

Figure 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 and each CSTs alignment for complete sentences are generated using the parallel corpus. After that the system generate combinations of CSTs. Finally we generalize CSTs using WordNet to achieve wide-coverage.

4.1 Source Language Analysis

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, determiners, singular or mass noun. The third brackets "[]" define the starting and ending of a complete chunk.

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

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] ./."*

4.2 Initial CSTs

In this second step, we produced CSTs from the parallel corpus. Table 2 shows the initial CSTs for the parallel sentence given in Table1. In Table2 CST# is the CSTs number for reference, "C" is the individual English Chunks, "S" is the corresponding Bengali Words, "Align" is the same as "Align" in Table 1.

"Chunk-Start-Index" equals to the first word position of the chunk in original sentence, minus one. For example, from Table 1 we get:

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

The first word of this chunk is "around", which was in position 7. Subtracting 1, we get the CST5 chunk-start-index is 6.

"T" represents the alignment information inside the chunk. For calculating T, the system subtract the chunk-start-index from each original word alignment. In the above example, the system subtract the chunk-start-index 6 from each word alignment. Then we get final alignment'

T=[1,2,3,4]

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/DT 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 worl-wide/RB]	বিশ্বব্যাপী	1	11	10

Table 2: Example of initial CSTs

4.3 CSTs Alignment

CSTs alignment contains the original sentence alignment information. So that from the initial CSTs we can regenerate the original sentence in correct word order. The benefit of CSTs alignment is, later on we can use this to generate sentences in correct word order.

In this step, the system generates the alignment information from Initial CSTs as given in Table 2. For example, Table3 shows the chunk alignment information produced from Table 2. "CSTs" represent the original English chunks order and "Alignment" 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".

CCST#	CSTs	Alignment
CCST1	CST1 CST2 CST3 CST4 CST5 CST6	CST6 CST1 CST2 CST5 CST4 CST3

Table 3: Example of CSTs alignment

Figure3 visualize the CSTs alignment from Table3.

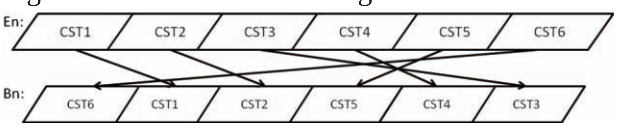

Figure 3: CSTs alignment

4.4 CSTs Alignment Combination

CSTs Alignment Combination generates new possible chunk orders. The goal is to match longer phrases to achieve wide-coverage. Without these combinations translation, the system coverage will be low. Advantage of this is to take less time during translation matching. Disadvantage of this is to take more memory. But as our EBMT is for low-resource, it will not affect much due to small parallel corpus.

From CSTs alignment as given in Table 3, system generates CSTs Alignment Combinations. It chooses any neighbor chunks where CSTs# difference is one in both English and Bengali. For example in Figure 4, circles identified the neighbors, CST1 and CST2 can be combined as CCST2, because they are neighbor in both source and target language.

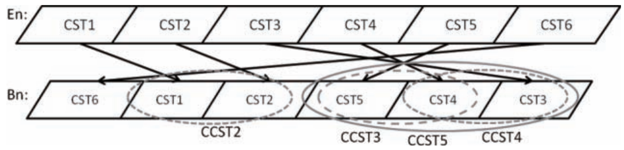

Figure 4: Chunk Alignment

Table 4 contains the Combined-CSTs (CCSTs) as shown in Figure 4.

CCST#	CSTs	Alignment
CCST2	CST1 CST2	CST1 CST2
CCST3	CST4 CST5	CST5 CST4
CCST4	CST3 CST4	CST4 CST3
CCST5	CST3 CST4 CST5	CST5 CST4 CST3

Table 4: CCSTs examples

4.5 Generalize CST Using WordNet

In this step CSTs are generalized by using WordNet to increase the EBMT coverage. To generalize we only consider nouns, proper nouns and cardinal number (NN, NNP, CD in OpenNLP tagset). For each proper nouns we search in WordNet. If available we replace that NNP with <lexical filename> returned from the WordNet. For example WordNet return <noun.communication> for "Bangla". For cardinal number we simply CDs together to <noun.quantity>. We show example generalized CSTs produced using WordNet in Table 5.

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]

Table 5 : Generalized CSTs

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.

5 Matching Algorithm for CSTs

Matching algorithm for CSTs has three components: search in CSTs, search in CCSTs and selecting CCSTs candidates. The Figure 5 shows the process of our proposed matching algorithm.

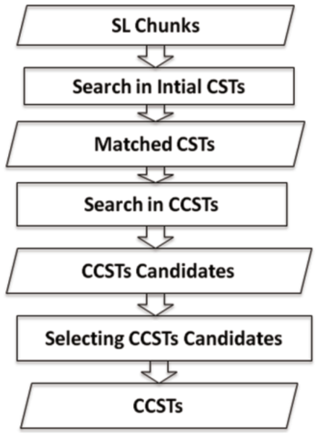

Figure 5: Matching Algorithm for CSTs

5.1 Search in CSTs

To search in CSTs our system first tries to find each chunk in initial CSTs. If it does not has 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.

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].

5.2 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] 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.

5.3 Selecting CCSTs candidates

Finally in this step using our selection criteria we choose the optimal 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.

For the above example, it chooses CCST1 as it has more CSTs match.

6 Unknown Word Translation

As the parallel corpus is small it is important to have a good method for translating unknown words. When the word has no match in the CST, it try to translate using dictionary. If it fails to find the English word in dictionary, it try to find related words from WordNet.

WordNet provide related word for nouns, proper nouns, verbs, adjectives and adverbs. For nouns and verbs WordNet provide hypernyms. Y is a hypernym of X if every X is a (kind of) Y. For example "canine" is a hypernym of noun "dog", 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. This process discovers the hypernym tree from WordNet in step by step.

As hypernyms can express the meaning, we can translate the hypernym of the unknown word. To do that, until any hypernym's Bengali translation found in the English-Bengali dictionary, we keep discovering upper level of hypernym's. Because, nouns and verbs are organized into hierarchies, defined by hypernym or IS-A-relationships in WordNet. So, we considered lower level concept is generally more suitable then the higher level words.

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, eutherian mammal
      => mammal
        => vertebrate, craniate
          => chordate
            => animal => ...
```

This process search in English-Bengali dictionary, for each of the entry of this hypernym tree. As we only had the entry for "animal", we translated "dog" as the translation of "animal", which is "পশু" [poshu] in Bengali.

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

However, when unknown word is not even found in WordNet nor in the dictionary, we transliterate the word to Bengali. For this we used transliteration mechanism of Akkhor Bangla Software.

For example, for the word "Muhammod" which is a popular Bengali name, we transliterated into "মুহাম্মদ" in Bengali.

As in our assumption, the main users of this EBMT will be monolingual people, they can not 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 Bengali language accepts foreign words, transliterating an English word into Bengali alphabet, makes that a Bengali foreign word.

7 Translation Generation

In this EBMT architecture we used Rule-Based generation method. Using dictionary and WordNet rules from example-base, we can accurately translate the determiners in Bengali. For translating determiner we adapted Salam et al.'s proposals (2009) to use WordNet.

Here WordNet provided required information to translate polysemous determiners 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. 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.

8 Experiments

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.

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 2000 word aligned English-Bengali parallel corpus, system generated 15356 initial CSTs, 543 Generalized CSTs and 12458 Combined-CSTs.

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

8.1 Wide-Coverage 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 6 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 effective for achieving wide-coverage in translating determiners.

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

Table 6: Wide-Coverage Comparison

8.2 Quality Evaluation

Quality evaluation measures the translation quality through human evaluation. Table 7 shows the human evaluation of current system.

Translation Quality	Grade	Baseline EBMT %	EBMT+ CSTs
Perfect Translation	A	10.12	25.60
Good Translation	B	7.44	38.69
Medium Translation	C	17.56	19.64
Poor Translation	D	64.88	16.07
Total		**100%**	**100%**

Table7: Human Evaluation using same testset

Perfect Translation means there is no problem in the target sentence, and exact match with test-set translation. Good Translation means not exact match with test-set reference, but still understandable for human. Medium means there are several problems in the target sentence, like wrong word choice and wrong word order. Poor Translation means there are major problems in the target sentence, like non-translated words, wrong word choice and wrong word order.

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

The identified main reason for improving the translation quality is our solution for translating unknown words. For example, even though "dog" was an unknown word, using our solution, it can be translated as "animal". As a result, during quality evaluation some test-set sentence improved from "poor" or "medium" to "good" translation.

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

On the other hand CSTs played a major role in sub-sentential match. As a result it helped to translate grammatically similar structured sentences as "perfect" or "good" translation. It also improved some test-set sentences from "poor" to "medium" or "good". Drawbacks of using CSTs are high computational complexity and big memory requirement for large parallel corpus.

9 Conclusion and Future Works

We proposed an EBMT system for low-resource language using CSTs in the example-base. Our EBMT system is effective for low resource language like Bengali. Using CSTs we improved the wide-coverage of our EBMT system for translating adequate determiners. We used WordNet to translate the unknown words which are not directly available in the dictionary. And then we used transliteration mechanism for the rest unknown words. To translate an English sentence, it is first parsed into chunks. Then the chunks matched with the CSTs to determine translation candidates. Finally the translation candidates are generated using a dictionary and Bengali generation rules to combine the target-language strings of the CSTs. Using this method, our proposed EBMT system improved the wide-coverage of adequate determiners by 41 points and quality by 48.81 points. Currently 64.29% of the test set translations produced by the system were acceptable.

Currently we used a small parallel corpus to generate CSTs. However to increase the performance we need a balanced parallel corpus (Salam et al. 2010). Although current system works well for small parallel corpus, the performance can decrease with larger parallel corpus. Because it will have many candidate CSTs. In future, we want to improve current CSTs selection mechanism.

We plan to use statistical language model for future improvement. It can improve the generation quality. In future we also want to evaluate the system using BLEU and other standard Machine Translation evaluation metrics.

References

Abney, Steven. 1991. Parsing by chunks. In Principle-Based Parsing, pages 257–278. Kluwer Academic Publishers.

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.

Diganta Saha, Sudip Kumar Naskar, Sivaji Bandyopadhyay. 2005. *A Semantics-based English-Bengali EBMT System for translating News Headlines,* MT Xummit X.

George A. Miller (1995). *WordNet: A Lexical Database for English.* Communications of the ACM Vol. 38, No. 11: 39-41.

Jae Dong Kim, Ralf D. Brown, Jaime G. Carbonell. 2010. *Chunk-Based EBMT.* EAMT, St Raphael, France.

Khan Md. Anwarus Salam, Mumit Khan and Tetsuro Nishino. 2009. *Example Based English-Bengali Machine Translation Using WordNetWordNet*. TriSAI, Tokyo.

Khan Md. Anwarus Salam, Yamada Setsuo and Tetsuro Nishino. 2010. *English-Bengali Parallel Corpus: A Proposal*. TriSAI, Beijing.

Md. Musfique Anwar, Mohammad Zabed Anwar and Md. Al-Amin Bhuiyan. 2009. Syntax Analysis and Machine Translation of Bangla Sentences, International Journal of Computer Science and Network Security, 09(08),317–326.

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.

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.

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).

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.

Sudip Kumar Naskar, Sivaji Bandyopadhyay. 2006a. *A Phrasal EBMT System for Translating English to Bengali.* Workshop on Language, Artificial Intelligence and Computer Science for Natural Language Processing applications (LAICS-NLP).

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