Phrase-Based Statistical Machine Translator for English to Bangla Machine Translation

Anika Mahmud
Department of Computer Science
University of British Columbia
anikacs@cs.ubc.ca

Abstract

The work presented in this report is the implementation of a phrase-based statistical machine translator for translating English to Bangla. In this work word reordering technique, semi supervised learning and transliteration methods have been applied to improve the performance of the translator. This work also focuses on the parallel data scarcity for solving the current problem. Morphological analysis has been done and system is evaluated comparing with some similar systems.

1 Introduction

Statistical Machine Translation (SMT) requires enormous amount of parallel text in the source and target language to achieve high quality translation. However, many languages are considered to be low-density languages, either because the population speaking the language is not very large, or because insufficient digitized text material is available in a language. Bangla suffers from insufficient digitized data. Bangla is mostly spoken in Bangladesh and the Indian state of West Bengal. With nearly 230 million speakers, Bangla is one of the most spoken languages in the world, but only a very small number of tools and resources are available for Bangla. In this work I focus on the data scarcity for solving machine translation and present a phrase based SMT system for translating English to Bangla.

Scarcity in parallel data makes the task of building a SMT more complicated. For the research of English to Bangla machine translation there are three exiting corpus with parallel English and Bangla data. These are LDC corpus, EMILLE corpus and KDE corpus. All three corpus has around eleven thousand sentences in each. There is a monolingual corpus named Prothom Alo corpus with one million Bangla sentence. These data is very limited comparing to the other existing parallel corpus for other languages. In this paper I discuss some of the works done in English to Bangla and Bangla to English statistical machine translation. Some of of these works are partially implemented in the current work. Some new ideas are also exploited in this work. Word reordering and semi-supervised learning are two current approaches applied in Bangla to English machine translation. Those methods are implemented in this work. Transliteration module has been developed by some researchers for English to Bangla machine translation. Here I present my own module for transliteration. Prepositions are handled separately in most English to Bangla and Bangla to English machine translations because there is no preposition in Bangla. In this work I handled the prepositions and also handled different forms of second persons as Bangla has three forms of second person. All the methods applied in this current work are evaluated and compared with the related works. The contributions of this work are its transliteration module, reordering and semi supervised learning applied for the first time in this specific task for English to Bangla MT, different second person handler for Bangla. Though the comparing systems are not identical as this work because of different corpus and different domain but this system shows improvement over the baseline system.

The rest of the paper is arranged as related work in section 2, techniques applied to improve performance of the SMT in section 3 , handling data scarcity in section 4, data used in this system in section 5, results and evaluation in section 6 and conclusion and future work in section 7.

2 Related Work

Like the scarcity of the data the amount of research done in the English to Bangla and Bangla to English MT is also fairly little. (Dasgupta et al., 2004) proposed a Machine Translation for English to Bengali where they proposed a transfer architecture which is used in the syntactic transfer of English to Bengali with optimal time complexity. Their proposed transfer architecture has five stages—(1) Tagging, (2) Parsing, (3) Change CNF parse tree to normal parse tree, (4) Transfer of English parse tree to Bengali parse tree and (5) Generation with morphological analysis. In parsing stage, they used **Cockey-Younger-Kasami (CYK)** algorithm to minimize the parsing steps from exponential order to polynomial order.

(Saha and Bandyopadhyay,2005) proposed an English to Bangla EBMT system for translating news headlines . The translation from source to target headlines is done in three steps. In the first step, search is made directly in the example base; if it is not found there then it is searched in the generalized tagged example base. If a match is found in the second step, then it extracts the English equivalent of the Bangla words from the bilingual dictionary and applies synthesis rules to generate the surface words. If the second step fails, then the tagged input headline is analyzed and to identify the constituent phrases. The target translation is then generated from the bilingual example phrase dictionary, using heuristics to reorder the Bangla phrases.

(Naskar and Bandyopadhyay,2006) propose a Phrasal Example Based Machine Translation (EBMT) system from English to Bengali that identifies the phrases in the input through a shallow analysis, retrieves the target phrase using a Phrasal Example Base and finally combines the target language phrases employing some heuristics based on the phrase ordering rules for the Bengali. They also focus on the structure of noun, verb and prepositional phrases in English and how these phrases are realized in Bengali. Their study has an effect on the design of phrasal Example Base and recombination rules for the target language phrases.

(Anwar et al., 2009) proposed a method to analyze syntactically Bangla sentence using context sensitive grammar rules which accepts almost all types of Bangla sentences including simple, complex and compound sentences and then interpret input Bangla sentence to English using a NLP conversion unit. The grammar rules employed in the system allow parsing five categories of sentences according to Bangla intonation. The system is based on analyzing an input sentence and converting into a structural representation (SR). Once an SR is created for a particular sentence it is then converted to corresponding English sentence by NLP conversion unit. For conversion, the NLP conversion unit takes help of the corpus. The effectiveness of this method has been justified over the demonstration of different Bangla sentences with 28 decomposition rules and the success rates in all cases are over 90%.

There is also an open source rule-based MT system called Anubadok (*http://anubadok.sourceforge.net*) for translating English to Bangla . The first step of the three step translation process converts different kind of documents into xml format. Then it tokenizes tags and lemmatizes English sentences. At the beginning of the third step, it determines the sentence type, subject, object, verb and tense, and then translates English words to Bangla using a bilingual dictionary. Finally, it joins the subject, object and verbs in the SOV order.

(Roy 2009) proposes a semi-supervised approach for Bangla to English phrase-based MT where the baseline system was built using a limited amount of parallel training data. The system randomly selects sentences from a Bangla monolingual corpus, and translates them using the baseline system. Finally, source and translated sentences are added to the existing bilingual corpora. Acquiring parallel sentences is an iterative process until a certain translation quality is achieved. This can be a very useful procedure for a data sparse language like Bangla. In this current work I applied this method to deal with data scarcity for English to Bangal machine translation. Their work and this current work is based on the same LDC corpus and same test set.

(Islam et al., 2010) proposed a phrase-based Statistical Machine Translation (SMT) system that translates English sentences to Bengali. They added a transliteration module to handle OOV (Out-Of-Vocabulary) words. A preposition handling module is also incorporated to deal with systematic grammatical differences between English and Bangla. To measure the performance of their system, they used BLEU, NIST and TER scores. My current work is mostly related to this work because of the domain. In their work they used KDE and EMILLE corpus.

(Roy and Popowich, 2010) applied three reordering techniques namely lexicalized, manual and automatic reordering to the source and language in a Bangla-English SMT system and demonstrated that applying reordering approaches improves translation accuracy. In this current work I applied their lexicalized automatic reordering rules. I used the same data set used in their work.

3 Techniques for SMT

The baseline system is first created and different techniques are applied to improve the performance of the SMT.

3.1 Baseline System

The baseline system used here is almost similar to the baseline system of the workshop[1] . The translation system is a factored phrase- based translation system that

[1] http://www.statmt.org/wmt09/baseline. html

uses the Moses toolkit (Koehn et al., 2007) for decoding and training, GIZA++ for word alignment (Och and Ney, 2003), and SRILM (Stolcke, 2002) for language models. For comparing purposes some parameters of the baseline has been changed. In this case only up to 3-gram was calculated for training which is similar to (Roy, 2009). But in that work they omitted the unknown words. For transliteration purposes I retained those in this work.

Given English Sentence	Output Translation
After all this, no peace.	এ সব পর , শান্তি না ।
ABC's Gillian Findlay reports tonight from Palestinian Gaza.	এবিসি জোলাপের gillian ফিলিস্তিনের গাজা থেকে যেতে findlay মনে করে ।
India: Sino-Indian cordial relation will ensure peace and development in the region.	ভারত sino-indian : ঘনিষ্ঠ সম্পর্ক উন্নয়ন শান্তি ও নিশ্চিত হবে ।
Nicaraguan President Daniel Ortega may have accomplished over the weekend what his U.S. antagonists have failed to do: revive a constituency for the Contra rebels.	Nicaraguan র্পেসিডেন্ট Daniel Ortega মে উল্লেখয্ , তার সপ্তাহান্তের কি তাঁর মার্কিন antagonists বয্র্থ করার একটি : হাত ছাড়া বাণিজিয্ক শকিতর ।

Table 1. Outputs from baseline system

Table 1 shows some of the outputs of my baseline translation system. From the outputs we can clearly see there are lots of unknown words which the system didn't recognize.

3.2 Transliteration module

Most of the unrecognized words we get as an output of the baseline system are verbs and names. To solve the problem of those names in this work I implemented a transliteration module. The outline of the module is given in Figure 1.

In this transliteration module all the words those were not translated by the baseline and the file to be translated are fed in. The LingPipe[2] tool is used to identify the name entity by using its name entity identifier module. The name entities are extracted and compared with the not translated words. If they match the word is entered in the dictionary. These words are then fed into Google transliteration API. At this stage manual judgement is used to make the correct transliteration entry in the dictionary mostly because of the failure of the Google transliteration module. In some cases and there are some words which have their own form in Bangla as India is ভারত.

Is this module I have extracted 12 different transliteration word pairs and 819 name entities and their transliteration and created a dictionary out of these words. Because of the manual decision part this module cannot be used automatically but this step ensures the correct transliteration is added in the dictionary.

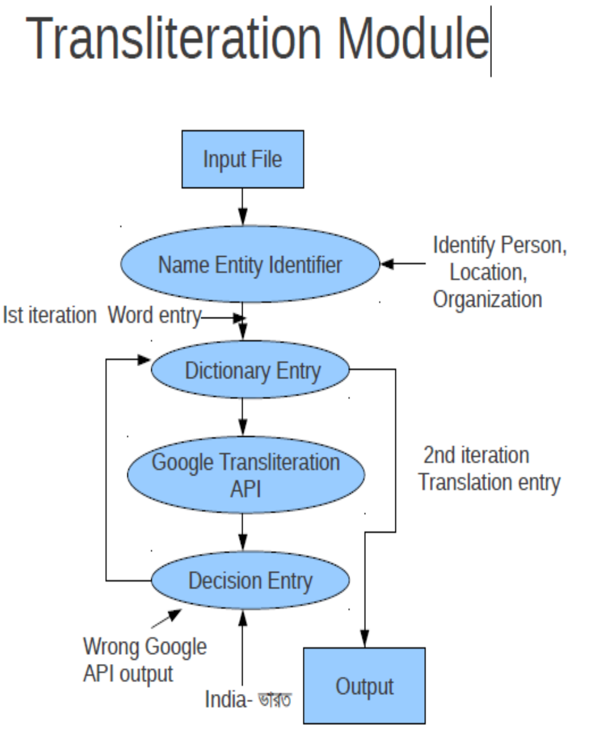


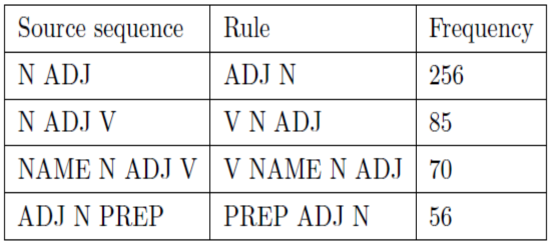
Figure 1. Transliteration module

3.3 Automatic Word Reordering

In most cases the translation does not look correct because of the wrong word reordering. In the baseline system used here, the reordering is done using GIZA++ tool and it depends on the statical data for reordering. To overcome the problem of reordering (Roy and Popowich, 2010) applied automatic reordering by using parts of speech (POS) tags. For this they needed POS tagged source language and word aligned target lan-

[2] http://alias-i.com/lingpipe/

guage. From these information they extracted the most frequent reordering rule and applied those when training. As I didn't have the aligned corpus I used the rules extracted from (Roy and Popowich, 2010) and applied in the reordering. Table 2 shows some rules from (Roy and Popowich, 2010).

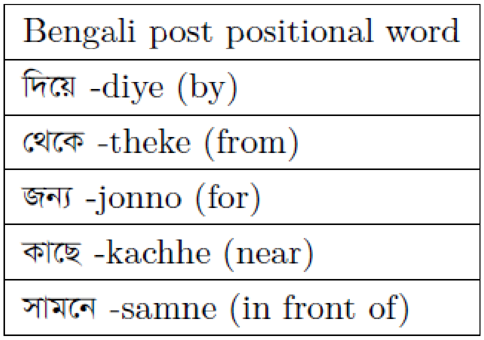


Source sequence	Rule	Frequency
N ADJ	ADJ N	256
N ADJ V	V N ADJ	85
NAME N ADJ V	V NAME N ADJ	70
ADJ N PREP	PREP ADJ N	56

Table 2. Some Automatic Reordering rules

3.4 Preposition Handling

One important feature of Bangla language is that there is no concept of preposition in Bangla. Bangla attaches inflection to the head noun or post positions word after the head noun. Table 3 is showing some post position words used in Bangla.



Bengali post positional word
দিয়ে -diye (by)
থেকে -theke (from)
জন্য -jonno (for)
কাছে -kachhe (near)
সামনে -samne (in front of)

Table 3. Post Position words used in Bangla

To deal with the prepositions used in English, in this work the rules mentioned in the previous section are used. For this we also need the POS data of the input set. The POS tag is used to extract prepositions from the output files generated by the baseline system for post processing. When the prepositions are identified the rules are used to put its corresponding Bangla word in correct position. If the non translated word is only a single English word preposition in the translated sentence then the prepositions are only replaced by their Bangla word. No reordering is Applied in this case.

3.5 Handling Second Person

Another interesting feature of Bangla is that for representing second person it has three different ways. Those are for single you in Bangla it can be আপনি ,তুমি , তুই. It also affects the form of verb depending on the subject of the sentence. If the subject is a person the verb takes one form if not then another form. Its really hard to identify which form of second person will be used without the context. In this work we use single sentence for translation there is no context present. So I did not deal with the form of second person. Rather I tracked the affect of the subject on the verb. For this POS tag data and the name entity identified in section 3.1 are used. LingPipe gives PERSON, LOCATION and ORANIZATION tags. For this particular purpose I use only the PERSON tagged words. The POS tag information is used to find the relative position of the PERSON tagged word with the verb to find whether its acting on the verb and change the translation accordingly.

4 Handling Data Scarcity

As mentioned before lack of parallel corpus is a huge barrier in getting good results from SMTs. To overcome this problem two methods can be applied. Either increase the amount of parallel data or apply a semi supervised method proposed in (Roy, 2010).

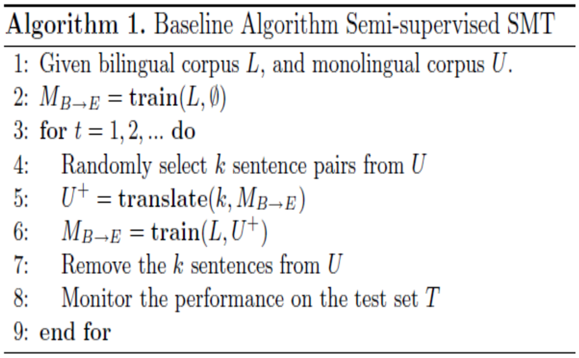
4.1 Adding more data

For adding more data to the current existing corpus I started combining two parallel corpus which are LDC corpus and EMILLE corpus. LDC corpus was prepared with parallel English and Bangla sentence combined in a single file. On the contrary the EMILLE corpus is distributed in 70 different small files. These files contain SGML tags. In some occasions there are multiple lines for a single sentence. As a whole this corpus required extensive cleaning. Moreover the language style used in this corpus does not match the language style used in LDC corpus. The language style used in EMILLE corpus is more formal and old style than the LDC corpus. Being a speaker of the language I could clearly see the difference and became skeptical about the result of adding these two corpus. So I ended up adding 10 files from the EMILLE corpus with LDC corpus with more than 600 lines added to the new corpus.

To increase the amount of parallel data from the beginning I started adding translated sentence to the Prothom Alo monolingual corpus. Social networking site is being used here for finding annotators. This is an on going process and so far 260 translated sentence have been added.

4.2 Semi Supervised SMT

(Roy, 2009) used a semi supervised method to increase the number of translated sentence in the corpus. Their baseline algorithm is shown in figure 2.



```
Algorithm 1. Baseline Algorithm Semi-supervised SMT
Given bilingual corpus $L$, and monolingual corpus $U$.
$M_{B \to E} = \text{train}(L, \emptyset)$
for $t = 1, 2, \ldots$ do
    Randomly select $k$ sentence pairs from $U$
    $U^+ = \text{translate}(k, M_{B \to E})$
    $M_{B \to E} = \text{train}(L, U^+)$
    Remove the $k$ sentences from $U$
    Monitor the performance on the test set $T$
end for
```

Figure 2. Baseline Algorithm for Semi Supervised SMT

For this project I implemented their baseline algorithm. In this work they repeated iteration until a certain accuracy was achieved. For this project I only used two iterations.

5 Data

As mentioned before in this project I used LDC parallel corpus, EMILLE parallel corpus and Prothom Alo mono lingual corpus. For the baseline system same LDC corpus and test set were used as used in (Roy, 2010) and (Roy , 2009).

The reordering, transliteration, preposition handling and second person handling were performed on the same data set used to develop the baseline system. Preposition handling , transliteration and second person handling were performed as post processing step.

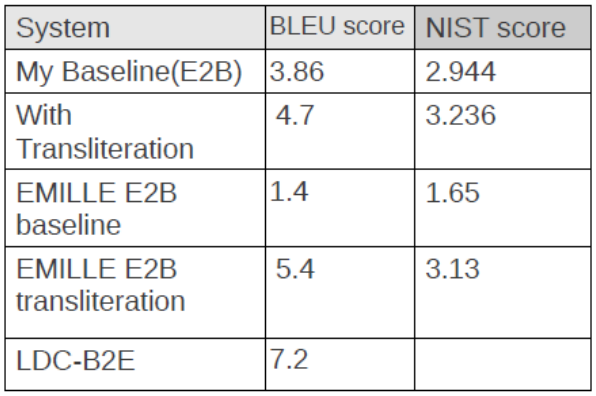
10 files from the EMILLE corpus were added to LDC corpus with around 600 more sentences. These dataset from the EMILLE corpus was again used as a training set in semi supervised SMT. Each of the methods applied here were evaluated with BLEU and NIST score as the related works were evaluated using these scores.

6 Evaluation

The methods applied in the current systems are evaluated using BLEU score and NIST score which are well known and widely used automatic evaluation metric used for MT. Both of these scores measure n-gram precision but BLEU gives same weight to all. On the other hand NIST does a weighted evaluation.

All the method applied in this work are compared with (Roy, 2009), (Roy ,2010) and (Islam, 2010). For clear understanding (Roy, 2009), (Roy ,2010) will be mentioned as B2E system and (Islam, 2010) as E2B system in the evaluation tables.

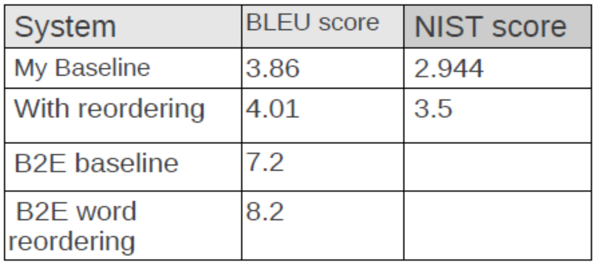
The transliteration module in my system was evaluated using BLEU score and NIST score. Table 4 contains the BLEU score of my system , E2B and B2E system. One aspect of this table is that the BLEU scores of the baseline systems for each of the system are different. Which is expected for E2B system because that uses a different corpus. The important thing to note here is even after using the same corpus the baseline system of B2E has larger BLEU score. Which implicates the complexity of English to Bangla translation. The table shows my transliteration module improved the BLEU score.



System	BLEU score	NIST score
My Baseline(E2B)	3.86	2.944
With Transliteration	4.7	3.236
EMILLE E2B baseline	1.4	1.65
EMILLE E2B transliteration	5.4	3.13
LDC-B2E	7.2	

Table 4. Transliteration module evaluation

After applying the reordering method for training with the baseline system some degree of improvement was noticed. The scores are shown in table 5. Reordering improved the performance from baseline but the percentage of improvement after applying reordering is only 3.89% for my system. The same for B2E is 13.9%

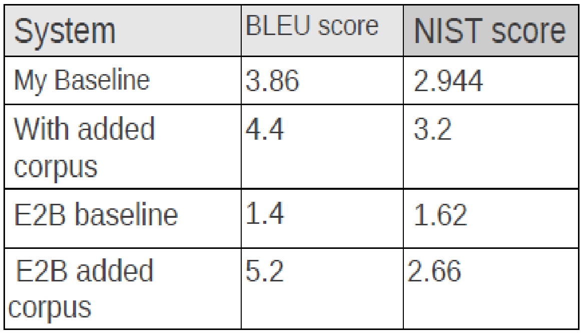


System	BLEU score	NIST score
My Baseline	3.86	2.944
With reordering	4.01	3.5
B2E baseline	7.2	
B2E word reordering	8.2	

Table 5. Evaluation of Reordering Method

This implicates that new rules should be generated from an aligned corpus specifically for English to Bangla machine translation. Clearly the existing rules for Bangla to English machine translation did not perform well enough.

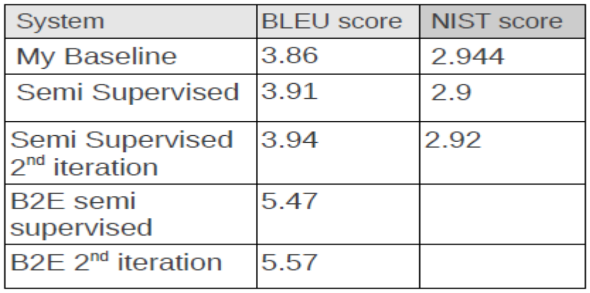
After adding some part of the EMILLE corpus with the existing LDC corpus the system was evaluated. It shows very little improvement. This suggest that more data has to be added with LDC corpus to verify whether the language style difference is having any effect on the performance. Because theoretically more data should give better performance. The results are shown in table 6. For E2B the added corpus is the EMILLE and KDE corpus.



System	BLEU score	NIST score
My Baseline	3.86	2.944
With added corpus	4.4	3.2
E2B baseline	1.4	1.62
E2B added corpus	5.2	2.66

Table 6. Added corpus evaluation

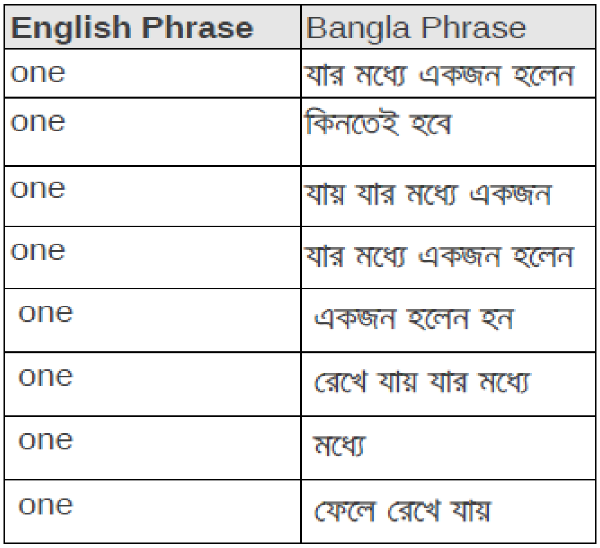
The semi supervised SMT was also evaluated using BLEU and NIST score. In this case the score decreased for my system from the baseline. The addition of previously translated sentence contain lots of unknown data as an output of the previous translation. This might affect the score. The results are shown in table 7.



System	BLEU score	NIST score
My Baseline	3.86	2.944
Semi Supervised	3.91	2.9
Semi Supervised 2nd iteration	3.94	2.92
B2E semi supervised	5.47	
B2E 2nd iteration	5.57	

Table 7. Semi-supervised SMT evaluation

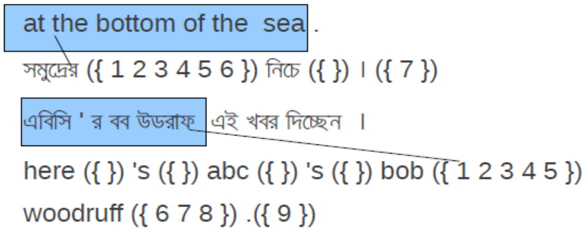
The overall performance of the current system is not satisfactory though it shows improvement from the baseline system. As the comparing systems are not exactly using the same corpus and not applied to the same domain the comparisons are non conclusive. During the analysis of the baseline system some aspects of the system were revealed. The phase table clearly shows the result of lack of data. For the word "one " it generates 23 different phrase table entry. Some entries are shown in table 8.



English Phrase	Bangla Phrase
one	যার মধ্যে একজন হলেন
one	কিনতেই হবে
one	যায় যার মধ্যে একজন
one	যার মধ্যে একজন হলেন
one	একজন হলেন হন
one	রেখে যায় যার মধ্যে
one	মধ্যে
one	ফেলে রেখে যায়

Table 8. Some phrase table entry for "one"

The phrase alignment also suffers from this data scarcity. Two examples from the phrase reordering table are shown in figure 3. Which clearly shows the misalignment of the phrases. Multiple words are grouped together on the contrary some phases are left as NULL.



at the bottom of the sea .

সমুদ্রের ({ 1 2 3 4 5 6 }) নিচে ({ }) । ({ 7 })

এবিসি ' র বব উডরাফ এই খবর দিচ্ছেন ।

here ({ }) 's ({ }) abc ({ }) 's ({ }) bob ({ 1 2 3 4 5 })

woodruff ({ 6 7 8 }) .({ 9 })

Figure 3. Misaligned Phrases

In this work the recent works in Bangla to English and English to Bangla have been explored. A new module for Transliteration have been used here. The POS tag data and name entity has been used to handle preposition and second person verbs . For covering more than one methods the work presented here does not go deep into each method. As a result the comparisons with the existing systems can not be analyzed properly. If the methods were implemented completely then the comparisons would provide more insight about the success of the methods in the current domain. If there is any effect on the outcome for the language that could have been answered too.

7 Future Work and Conclusion

As a future work the main goal is to complete the translation of Prothom Alo monolingual corpus and prepare a rich data set for SMT research. For the current work I did not had a POS tagger for both Bangla and English . Recently I came up with this POS tagger which can also tag Bangla named MALLET. In future work I will use this tagger for automatic alignment. This will also reduce the need of an aligned corpus.

The goal of this project was to implement a basic translator for English to Bangla translation and apply some current improvement techniques. That goal has been accomplished. The methods applied show that they improve the baseline system. To build a good translator for English to Bangla the data scarcity has to addressed properly. That can lay the ground for more sophisticated research in this domain.

Reference

Andreas Stolcke 2001. S*RILM–an extensible language modeling toolkit, Proceedings of the ICSLP.*

Diganta Saha and Sivaji Bandyopadhyay. 2005. *A semantics-based English-Bengali EBMT system for translating news headlines, Proceeding of MT Summit X second workshop on example-based machine translation.*

Franz Josef Och 2003. *Minimum error rate training in statistical machine translation, Proceedings of the 4th Annual Meeting of the Association for Computational Linguistics (ACL).*

Kishore Papineni, Salim Roukos, Todd Ward and We Jing Zhu. 2001. *BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.*

Maxim Roy 2009. *A Semi-supervised Approach to Bengali-English Phrase-Based Statistical Machine Translation, Proceedings of the 22nd Canadian Conference on Artificial Intelligence.*

Maxim Roy and Fred Popowich. *Word Reordering approaches for Bangla-English SMT, In Canadian Conference on AI 2010*

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 here Eisele: *English to Bangla phrase-based machine translation. EAMT 2010: Proceedings of the 14th Annual conference of the European Association for Machine Translation, 27-28 May 2010, Saint-Raphaël, France. Proceedings ed.Viggo Hansen and François Yvon; 8pp. [PDF, 601KB]*

Philipp Koehn, Hieu Hoang, Alexandra Birch, ChrisCallison-Burch, Marcello Federico, Nicola Bertoldi,Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst. *2007. Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL).*

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, 2006. *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 (LAICS–NLP).*

Appendix A

A.1

The tokenizer used in moses was changed to handle Bangla delimiters, specially Dari (u09F7)

A.2 Baseline scripts

```
/home/anika/scripts/tokenizer.perl -l en
</home/anika/work2/training/training.en>/home/anika/
work2/training/training.tok
```

```
/home/anika/scripts/tokenizer.perl -l
</home/anika/work2/training/training.bn>/home/anika/
work2/training/training.tok.bn
```

```
/home/anika/scripts/lowercase.perl
</home/anika/work2/training/training.tok.en>/home/ani
ka/work2/training/input
##no need to tokenize bangla##copy as reference
 /home/anika/moses-script/scripts-20101201-2301/training/clean-corpus-n.perl
/home/anika/work2/training/training.tok en bn
/home/anika/work2/training/training.l 1 40
/home/anika/srilm/bin/i686/ngram-count -order 5 -interpolate -kndiscount -text
/home/anika/work2/training/training.l.bn -lm /home/anika/work2/lm/b.lm
```

```
/home/anika/moses-script/scripts-20101201-2301/training/train-model.perl -scripts-root-dir
/home/anika/moses-script/scripts-20101201-2301/ -root-dir /home/anika/work2/training/ -corpus
/home/anika/work2/training/training.low -f en -e bn
-alignment grow-diag-final-and intersection -reordering
msd-bidirectional-fe -lm
0:5:/home/anika/work2/lm/b.lm:0
```

```
/home/anika/scripts/tokenizer.perl -l en
</home/anika/work2/training/dev.en>/home/anika/work
2/training/dev.tok
/home/anika/scripts/tokenizer.perl -l
</home/anika/work2/training/dev.bn>/home/anika/work
2/training/dev.tok.bn
```

```
/home/anika/moses-script/scripts-20101201-2301/training/mert-moses.pl
/home/anika/work2/training/tuning/input
/home/anika/work2/training/tuning/reference /home/anika/moses/moses-cmd/src/moses
/home/anika/work2/training/model/moses.ini --working-dir /home/anika/work2/ --rootdir /home/anika/moses-script/scripts-20101201-2301/ -mertdir
/home/anika/work2/mert/
```

```
/home/anika/scripts/reuse-weights.perl
/home/anika/work2/training/tuning/moses.ini <
/home/anika/work2/moses.ini >
/home/anika/work2/training/tuning/moses.weight-reused.ini
```

```
scripts/tokenizer.perl -l en < wmt08/devtest/devtest2006.en > working-dir/evaluation/devtest2006.input.tok
scripts/tokenizer.perl -l  <
wmt08/devtest/devtest2006.en > working-dir/evaluation/devtest2006.reference.tok
```

```
scripts/lowercase.perl < working-dir/evaluation/devtest2006.input.tok > working-dir/evaluation/devtest2006.input
```

```
/home/anika/moses-script/scripts-20101201-2301/training/mert-moses.pl
/home/anika/work2/training/tuning/input
/home/anika/work2/training/tuning/reference /home/anika/moses/moses-cmd/src/moses
/home/anika/work2/training/model/moses.ini --working-dir /home/anika/work2/ --rootdir /home/anika/moses-script/scripts-20101201-2301/ -mertdir
/home/anika/work2/mert/
```

```
/anika/scripts/reuse-weights.perl
/home/anika/work2/training/tuning/moses.ini <
/home/anika/work2/moses.ini >
/home/anika/work2/training/tuning/moses.weight-reused.ini
```

```
/home/anika/moses-script/scripts-20101201-2301/recaser/train-recaser.perl -train-script /home/anika/moses-script/scripts-20101201-2301/training/train-model.perl
-ngram-count /home/anika/srilm/bin/i686/ngram-count
-corpus /home/anika/work2/lm/b.lm -dir recaser
```

```
scripts/tokenizer.perl -l en < wmt08/devtest/devtest2006.fr > working-dir/evaluation/devtest2006.input.tok
scripts/tokenizer.perl -l < wmt08/devtest/devtest2006.en
> working-dir/evaluation/devtest2006.reference.tok
```

```
scripts/lowercase.perl < working-dir/evaluation/devtest2006.input.tok > working-dir/evaluation/devtest2006.input
```

```
/home/anika/scripts/wrap-xml.perl
/home/anika/work/training/testing/test.sgm bn <
/home/anika/work2/test/output >
/home/anika/work2/training/tuning/output.sgm
```

```
mteval-v11b.pl -r wmt08/devtest/devtest2006-ref.en.sgm -t working-dir/evaluation/devtest2006.output.sgm -s
wmt08/devtest/devtest2006-src.fr.sgm -c
```

A. 3 Script for POS and UTF-8

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Python

import sys
import re
import string
```

```
words= []

f =
open('/home/anika/output_with_unknown.txt')#.read().d
ecode('string-escape').decode("utf-8")

#regex = re.compile('[a-z|A-Z]+')

for line in f:
    for word in line.split(' '):
        #print word.decode("utf-8")
        if re.search("^[a-z|A-Z]+",word):
           w=word.rstrip('_NNP')
           w=word.rstrip('_NNPS')
           words.append(w)
           wDict = {}
           for w in words:
              if wDict.has_key(w):
                 wDict[w] += 1
              else:
                 wDict[w] = 1
f.close()

print 'There are a total of ' + str(len(wDict)) + ' words in
this text'

#print the distinct items
for i in range(len(wDict)):
        print wDict.items()[i]
```