Journal of Computer Science

Original Research Paper

SuVashantor: English to Bangla Machine Translation Systems

Mahjabeen Akter, M. Shahidur Rahman, Muhammed Zafar Iqbal and Mohammad Reza Selim

Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh

Article history
Received: 13-06-2020
Revised: 24-07-2020
Accepted: 15-08-2020

Corresponding Author:
Mahjabeen Akter
Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh
Email: mahjabeen.sust@gmail.com

Abstract: This paper presents the system description of Machine Translation (MT) systems for English-Bangla language pair. Our goal was to create two benchmark MT systems that produce a better quality translation and comparatively higher evaluation score than existing MT systems for English to Bangla. In our experiments, we implemented two baseline MT systems using both statistical and neural methods for the said language pair. Our phrase-based statistical model and 2-layer LSTM neural model were trained and evaluated with a large dataset that is carefully pre-processed and contains unique training data to avoid biases from the cross-validation and test data. We achieved the highest scoring BLEU for our experiments with these setups. Furthermore, we improved the performance of the neural model using pre-trained embedding and synthetic monolingual data which are cutting-edge technology for neural models.

Keywords: Machine Learning, Machine Translation Systems, Statistical Machine Translation Systems, Neural Network, Neural Machine Translation Systems, Pre-trained Word Embedding, Synthetic Monolingual Data

Introduction

Machine translation systems have been adopted by many professionals over the last few decades to reduce human efforts, to break the linguistic barriers among the knowledgeable resources and for so many other reasons that are meant to put us at ease. Whereas there are many approaches available to achieve a computer-translated sentence or paragraph, statistical and neural network approaches are playing the lead role for a translation system. Statistical Machine Translation (SMT) was implemented by (Koehn *et al*., 2003). Neural Machine Translation (NMT) system was brought into action by (Sutskever *et al*., 2014; Bahdanau *et al*., 2014). Phrase-Based SMT (PBMT) is known to translate better the rare words while NMT is capable of producing fluent translations. But both systems need huge parallel data to acquire sufficient knowledge to generate acceptable translations. Bangla is considered as a resource-poor language. The amount of English-Bangla parallel data to conduct large scale machine translation experiments is notably insufficient. Most of the time, these data are domain-specific; hence the vocabulary range is limited which is a key problem for producing a fluent translation. Also, the morphological structure of both languages differs in a way, that it makes it difficult to meet up a major challenge of MT systems-alignment of words or phrases. Many pieces of research have been conducted to solve these issues, but the results were not so impressive. We discuss more details about different approaches to English-Bangla machine translation in the Related Works section.

In this study, we propose two baseline MT systems for English-Bangla using Statistical and Neural Network approach and two more systems with state of the art researches like pre-trained word embedding and synthetic monolingual data to explore the improvement over the baseline of Neural Network MT. Our contributions include:

- We carried out the experiments with a handpicked English-Bangla parallel corpus where the training, validation, and test data are pre-processed carefully to cover the problem of miserly vocabulary and biased output. We intend to make this corpus publicly available on Github to enable future research in this area
- We achieved a remarkable evaluation score in NMT compared to other available MT systems for English-Bangla language pair

The remainder of the paper is structured as follows. The ‘Related Works’ section presents the previous

approaches to English-Bangla machine translations and their results. The 'Background' section describes the statistical approach and neural network approach to machine translation in detail. We also describe how to incorporate pre-trained word embedding and synthetic monolingual data to a neural network in this section. The 'Methodology' section discusses the process of data preparation and the methodology for our experiments. In the 'Experiments' section, we illustrate the experimental process and discuss the evaluation metric, respectively. The 'Result and Analysis' section presents the results and sample translations. Finally, the 'Conclusion and Future Direction' section concludes the discussion by summarizing the experiments and by pointing out some scopes for future research.

Related Works

In this section, we discuss a few SMT and NMT based English to Bangla MT systems that have been developed in recent years. Islam *et al*. (2010) developed an SMT system, where they handled Bangla prepositions during translation. They also used a transliteration module to improve translation quality and accuracy. But their data was not sufficient enough for the system to perform well for all types of sentences. Their system works quite reasonable for short sentences only. Also, their system was unable to translate Out-Of-Vocabulary (OOV) words in many cases. Pal *et al*. (2013a; 2013b; Pal and Naskar, 2016) proposed a phrase-based SMT where a rule-based aligner is used to align Named Entities (NE), Multi-Word Expressions (MWE) and compound verbs. They showed improved performance in terms of translation quality for the various word alignment models. However, they trained their systems with a very small parallel corpus which contains only about 23,000 sentences. Also, their monolingual Bangla corpus is only the tourism domain-centric. Moshiul and Kamrul (2014) used an A* search algorithm in the statistical model to translate different types of English sentences into Bangla. They were able to reduce grammatical complexity and dependency on the structure of the sentence, but their system performs poorly in terms of execution time. This work also suffers from a data sufficiency problem. Therefore, their method works for only short sentences. Pal *et al*. (2014) incorporated a source chunks re-ordering method in English-Bangla phrase-based SMT. They showed that word alignment-based reordering of the source chunks is better than other reordering approaches for language pair with different word order like English and Bangla. However, they could not ensure their word alignment quality due to the unavailability of the gold-standard word alignment. Al Mumin *et al*. (2018) developed an SMT system using Neural Probabilistic Language Model (NPLM). They found that the system using NPLM is more capable of retaining the syntactic structure of the target Bangla text than a system that uses n-gram language model. But their system failed to retain subject-verb order of sentences in some cases. Dandapat and Lewis (2018) developed an NMT system with several different techniques to boost data store and tackle data sparseness, like crowd translation of selected monolingual data, back translation using synthetic monolingual data, data augmentation and early stopping. However, the quality of their synthetic data remains questionable, as they varied significantly across sentences. Ojha *et al*. (2018) developed both SMT and NMT systems for bidirectional English-Indic language which include Bangla. Their NMT system was based on Short Long Term Memory (LSTM) architecture and it outperformed their own SMT system which was trained with two different language models. But the NMT systems remained challenged in low-resource scenarios like Bangla language. Banerjee *et al*. (2018) developed a baseline bidirectional SMT system for English and Indic languages and an NMT system using multilingual transfer learning approaches including many-to-one, one-to-many or many-to-many translations. They were unable to provide any major improvement over the traditional PB-SMT system though.

The summary of all these systems is given in Table 1.

Table 1: Summary of previous English-Bangla MT systems

System reference	Type	Training data size (# parallel sentence)	Evaluation metric, best score
Islam *et al*. (2010)	SMT	10,850	BLEU = 11.4
Pal *et al*. (2013a; 2013b; Pal and Naskar 2016)	PB-SMT	23,492	BLEU = 20.87
Moshiul and Kamrul (2014)	SMT	2700	TFLD = 1.15
Pal *et al*. (2014)	SMT	22,176	BLEU = 13.17
Al Mumin *et al*. (2018)	SMT	3,330	BLEU = 5.7
Dandapat and Lewis (2018)	SMT, NMT	9,76,634	BLEU = 9.80
Ojha *et al*. (2018)	SMT, NMT	3,37,428	BLEU = 17~18
Banerjee *et al*. (2018)	PB-SMT	3,37,428	BLEU = 11.34

Background

In this section, we briefly describe the basic theory for Statistical Machine Translation and Neural Machine Translation method. We also describe the fundamentals of two approaches that can enhance the performance of a neural method when added with the original model; Pre-trained Word Embedding and Synthetic Monolingual Data.

Statistical Machine Translation (SMT)

The likelihood of a translation in an SMT is determined by Statistical probabilities. The parameters of the statistical model are analyzed by a bilingual text corpus. The basis for the SMT says that the probability distribution $p(e|f)$ implies, a string e in the target language is the translation of another string f in the source language. This probability distribution is modeled in the computer by accomplishing two sub-tasks. One generates the translation model $p(f|e)$ that gives the probability of the source language string as a translation of the target language string. Another creates a language model $p(e)$ that discovers the target language string in the bilingual text. These two models are combined as Bayes Theorem (Koehn *et al*., 2003) suggests $p(e|f) \approx p(f|e)\ p(e)$. Here, the $p(f|e)$ is decomposed with the help of a decoder, that generates all possible translations and chooses for the most probable one from among them. In a Phrase-Based SMT (PB-SMT), the source input sentence f is segmented into a sequence of phrases I during the decoding process. Then each source phrase f_i is translated into a target phrase e_i It is important to notice that, the target phrases may be re-ordered, which is modeled by a relative distortion probability distribution $d(a_i\text{-}b_{i\text{-}1})$, where a_i stands for the start position of the source phrase that is translated into the ith target phrase and $b_{i\text{-}1}$ stands for the end position of the source phrase translated into the i-1th target phrase.

State of the art PB-SMT decoders like Moses (Koehn *et al*., 2007) uses a Beam search algorithm inspired by (Jelinek, 1997). Here, the beam size denotes the maximum number of the hypothesis (probable target translations), is fixed to a certain number that is linear with the sentence length. Thus, the beam size not only limits the search space, but also the search quality. Hence, the proper trade-off between speed (low beam size) and performance (high beam size) has to be found carefully.

Along with the phrases or translation units, the modern PB-SMT systems uses a log-linear framework by (Och and Ney, 2002), that models the translation probability $p(x|y)$ as a log-linear combination of features as Equation 1:

$$p(e \mid f) = \exp\left\{\sum \lambda_k h_k f_1^n, e_1^m\right\} \tag{1}$$

Here, h_k is the feature function that can be added if necessary. Popular feature functions include a distortion model, a word penalty model, a phrase penalty model, a target language model, and phrase and lexical translation probabilities. The weights λ_k are optimized using an optimization function on a cross-validation data set that is used to fine-tune the translation model.

Figure 1 shows the basic PB-SMT system.

Neural Machine Translation (NMT)

While an SMT system has a separate language model, a translation model, and a re-ordering model, a neural network based MT system only has a single sequence model instead; to predict one word at a time. Kalchbrenner and Blunsom (2013; Sutskever *et al*., 2014; Cho *et al*., 2014) proposed the first neural network models, most of them belong to the encoder-decoders family, where an encoder reads a source sentence and encodes it into a fixed-length vector; from this encoded vector, the decoder than translates an output. But, the necessity to compress all the information of a source sentence into a fixed-length vector arises the issue of translating a long sentence correctly. Bahdanau *et al*. (2014) addressed this issue by adding an extension to the basic encoder-decoder model that learns to align and translate jointly, which has become very popular and is used by most of the modern NMT tools now. In this architecture, each conditional probability is defined by Equation 2:

$$p\left(y_i \mid y_1, ...y_{i-1}, X\right) = g\left(y_{i-1}, s_i, c_i\right) \tag{2}$$

where $X = \left(x_i, ..., x_{T_x}\right)$ is the sequence of a vector that the encoder reads as an input sentence, $y_{t'}$ is the next word to be predicted by the decoder, s_i is a hidden state for the time i, computed by $s_i = f(s_{i\text{-}1}, y_{i\text{-}1}, c_i)$. Here, for each target word y_i, c_i is a distinct context vector on which the probability is conditioned. From this equation, we further get an alignment model, which is given by Equation 3:

$$e_{i,j} = a\left(s_{i-1}, h_j\right) \tag{3}$$

where, $h_{1,.....T_x}$ is a sequence of annotations, to which the encoder maps the input sequence and $a_{i,j}$ is the weight of each annotation h_j.

This alignment model implements an attention mechanism in the decoder and thus relieves the encoder from having to encode all information in the source sentence into a fixed-length vector. Figure 2 depicts how this neural model predicts the t-th target word y_t, given a source sentence (x_1, x_2,……, x_T). Figure 2 shows an attention-based NMT system.

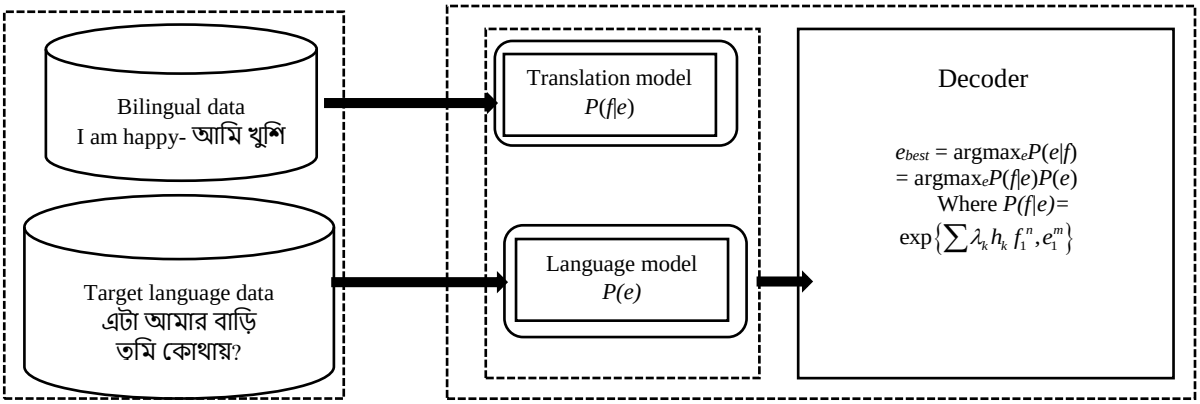

Fig. 1: PB-SMT system

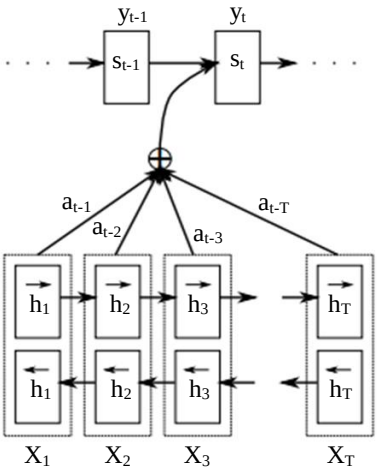

Fig. 2: Attention-based NMT System learning to align and translate jointly

For machine translation task, usually, the Recurrent Neural Network (RNN) is used, but a gated RNN like LSTM helps the model learns long-distance features by (Hochreiter and Schmidhuber, 1997), which we adapted.

NMT with Pre-trained Word Embedding

Although an NMT system is better than SMT for morphologically rich target language because of selecting more correct words and less re-ordering error (Bentivogli *et al*., 2016), for low-resource language like Bangla, it results in worse quality performance (Qi *et al*., 2018; Koehn and Knowles, 2017). In this case, where the availability of bilingual data is not sufficient enough, monolingual data plays a more effective role. Cheng *et al.* (2016) showed that there are several methods for using monolingual data in an NMT system. Among these, pre-trained word embedding has shown to improve BLEU score when integrated into the NMT system either as a standard translation system (Neishi *et al*., 2017) or as a method for learning translation lexicons in an unsupervised manner (Conneau *et al*., 2017). In our experiment, we initialize both encoder and decoder networks with pre-trained weights (embedding) of the source and target language models and then fine-tuned them with the bilingual corpus. We used the method proposed by (Ramachandran *et al*., 2017), where they observed, the improved generalization due to the pre-trained features provides the main advantage.

Pre-Trained word embedding brings in outside information and reduces the number of parameters that a neural network usually learns from scratch. For example, when a neural network encodes a word with a one-hot vector [0,0,,,,1,0,,,,0], it places 1 at the index corresponding to the appropriate vocabulary word and 0 everywhere else. This is the case where word embedding is missing. When v is the size of the vocabulary and h is the size of the hidden layer, the weight matrices connecting word-level inputs and the network's hidden layer would each be $v \times h$. When 100,000 words are fed into an LSTM layer with 100 nodes and 4 gates, the model needs to learn 400 million parameters in total (4 different weight matrices (for each gate of the LSTM), each with 100 million weights). But, with pre-trained word embedding, which maps each word onto a low-dimensional vector $w \in R^d$, where d is roughly 100, the number of parameters needed to be learned are effectively reduced. As the embedding is chosen based on the context in which words appear, it turns out, words that appear in a similar context, like 'bread' and 'butter' have similar embedding, while words that are not alike, like 'flower' and 'exam' have dissimilar embedding.

Figure 3 shows how pre-trained word-embedding is incorporated into the neural network architecture.

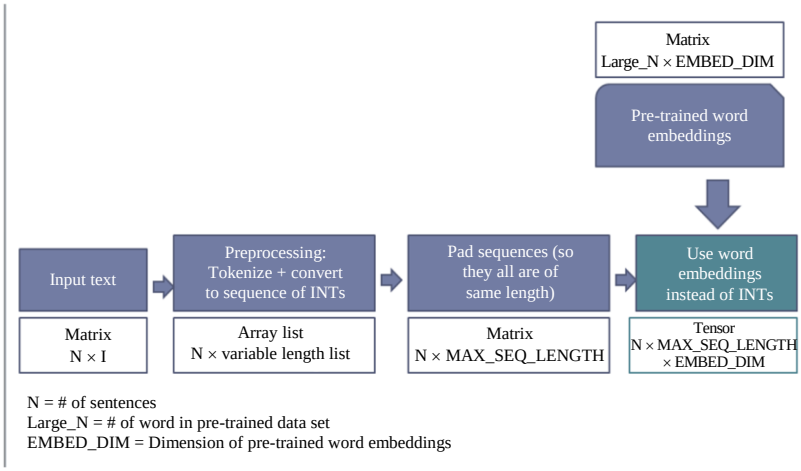

Fig. 3: Incorporating pre-trained word embedding into NMT

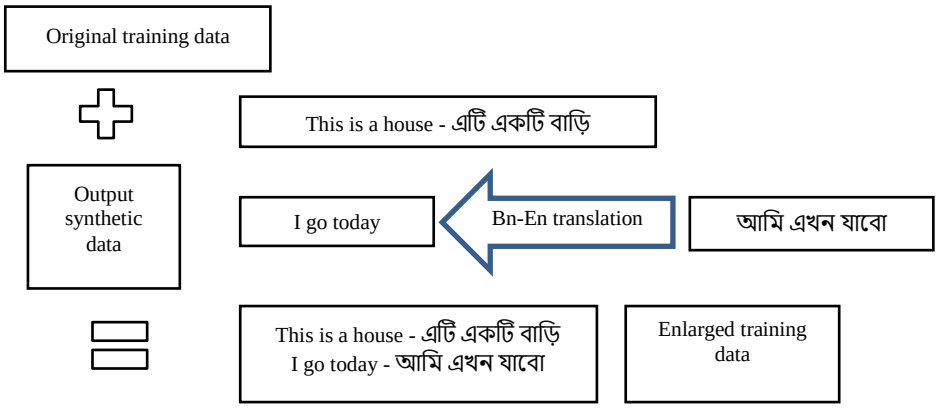

Fig. 4: Incorporating monolingual synthetic data into NMT

NMT with Synthetic Monolingual Data

SMT systems have been using target-side monolingual data to train language models from the beginning to boost fluency. An encoder-decoder NMT also can be trained with a language model from monolingual data without having to change the architecture of the neural network (Burlot and Yvon, 2018; Sennrich *et al*., 2015). To mix the monolingual target sentences into the training set, one can use dummy source sentences, or synthetic source sentences can be used, which is obtained via back-translation. We adapted the latter technology. Back translation refers to a process where an automatic translation of the monolingual target text into the source language is performed. Then the original parallel data (human translated or otherwise obtained) is mixed with the synthetic parallel data to train the system, while no network parameters are changed or discarded. Figure 4 shows the basic architecture for adding monolingual synthetic data into an NMT.

Methodology

In this study, we propose mainly two English to Bangla machine translation systems using a phrase-based statistical method and neural network method. We further exploit the neural method with two additional features that are unique for English-Bangla language pair; one with adding pre-trained word embedding for source and target data, another with adding synthetic monolingual data. All our systems are trained, tuned, and evaluated with a dataset, that is rich in vocabulary, diverse in a domain, and contains much larger data compared to other MT systems publicly available for English-Bangla language settings. We prepared our corpus in a way, that the training, validation, and test data do not contain duplicate sentences, rather the same context is present in all three sets of data. Thus we ensured the decoder can pick the most related vocabulary while translating. This method boosts the translation accuracy of our baseline systems. Also, by adding pre-

trained word embedding in the neural model, we reduce its workload. Furthermore, by adding synthetic data to our neural network, we ensured the decoder suffers less from the Out Of Vocabulary (OOV) problem. Our systems require some pre-processing, like corpus preparation, building the word embedding for the target language, generating synthetic data for the source language, etc., which are described in the next subsection. Then we trained both systems with the same data sets and finally, the decoders of the systems generated the translations. We describe our methodology through a flow chart in Fig. 5. The main components of this flow chart are described in the next subsection.

Corpus Preparation

We performed various pre-processing on our corpus. We made sure there are no HTML tags, English characters and non-Unicode characters are present in the Bangla data. We normalized our texts with the same set of punctuation marks on both source and target side, with an additional 'dari' for Bangla. Then we tokenized the text, true-cased them for English, and removed empty or overly long sentences. We created pre-trained embedding for Bangla and synthetic English data from monolingual Bangla data using back-translation.

Training

In this step, we used the pre-processed dataset to train both MT systems to produce translation models. The training period was different for statistical and neural methods due to their system architecture. For statistical method, a language model is generated from the target side data, which was not the case for the neural network, as we discussed before that, the neural model does not require a separate language model. We also tuned the SMT system with validation data after the training is done. For the neural network, the validation is done simultaneously while training is going on.

Decoding

In the final step, both systems generate Bangla translations from English data with their decoders. The translation model generated from the previous step is used for this purpose. In this study, we compare all our systems to gain the intuition of translation accuracy.

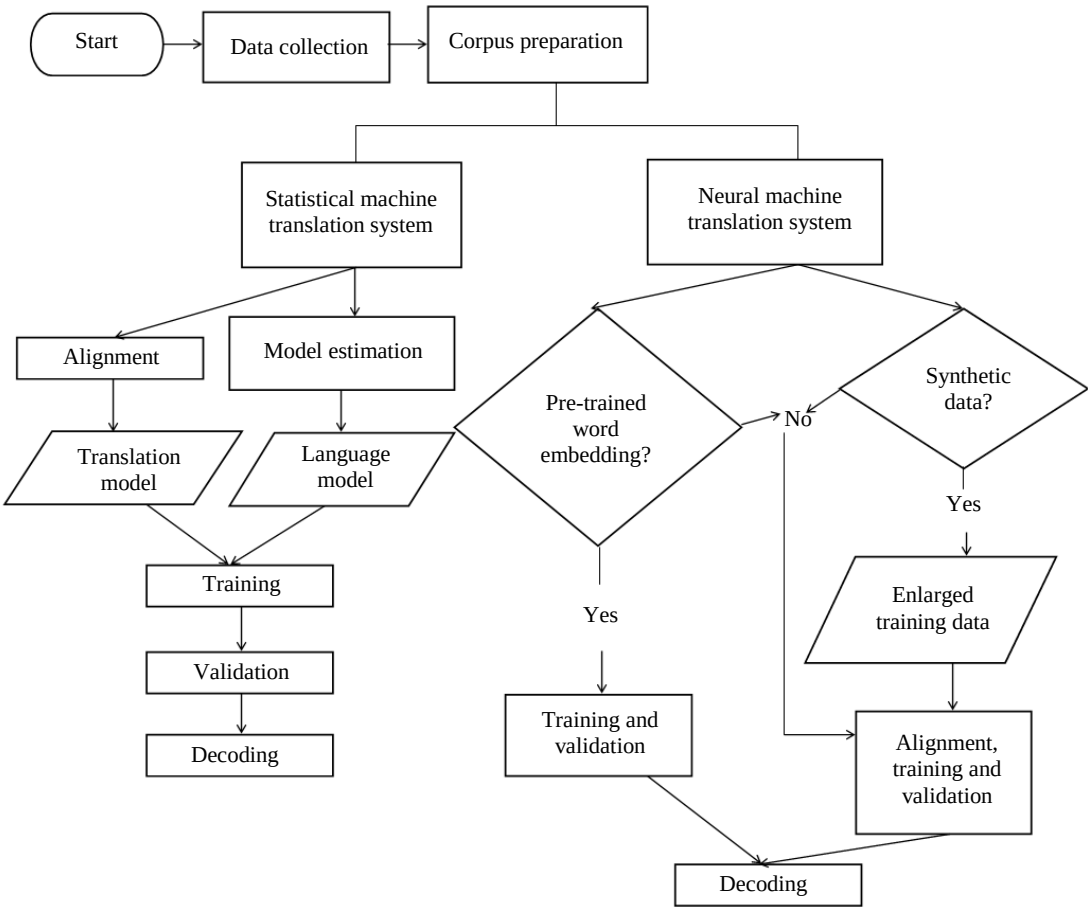

Fig. 5: Our methodology

Experiments

In this section, we briefly describe the experimental settings used to develop the PBSMT and NMT systems for English-Bangla language pair.

Datasets

We used a comparatively large parallel corpus for English and Bangla language pair in MT tasks. This parallel corpus is constructed by combining multiple open-source parallel corpora. The detailed statistics of these corpora are demonstrated in Table 2 which was used to train the MT systems. The parallel data were further divided into training, tuning, and testing sets among which 4,84,131 sentences were used for training, 2000 sentences for tuning, and 2000 sentences were used for evaluating the systems. The detailed information of the split is presented in Table 2. This corpus consists of data from various domains, like political news, history, movie subtitle, user manual of different tools and applications, literature, the scripture of the Holy Quran, etc.

Pre-Processing

We performed the following pre-processing steps for the scope of this work. Both types of corpora were tokenized using the Polyglot scripts. We true-cased the English representations of the corpora and cleaned (empty sentences are removed) the data using the Moses scripts. To avoid biases, we ensured that the training, tuning, and testing data do not contain duplicate sentences on the target side, although we kept the paraphrased translations to enrich our dataset with a larger vocabulary.

For our pre-trained embedding experiments, we extracted source (English) embedding from the Glove word embedding by (Pennington *et al*., 2014). For target (Bangla) embedding, we created our word embedding using gensim by (Rehurek and Sojka, 2010). For synthetic monolingual data, we trained our NMT to translate from Bangla to English with the original parallel corpus whose information is given above in Table 2 and then obtained 1,00,000 translated English sentences as synthetic data. We added this additional data with our original corpus.

Statistical Machine Translation System (SMT)

We built our phrase-based statistical MT systems using the Moses toolkit. To extract phrases from the corresponding parallel corpus, we used the GIZA++ toolkit by (Och and Ney, 2003) with the grow-diag-final-and heuristic. We used the IRSTLM toolkit by (Federico *et al*., 2008) to build a 5-gram language model. We used the Polyglot tokenizer by (Chen and Skiena, 2016) to tokenize the English and Bangla representations of our experiments.

Neural Machine Translation System (NMT)

We used OpenNMT-tf (the tensorflow port of OpenNMT toolkit) by (Klein *et al*., 2017) to build our Neural Machine Translation system. We used a 2-layer LSTM by (Hochreiter and Schmidhuber, 1997). This model is trained with mini-batches of 64 with 512 hidden units, a vocabulary size of 50,113 and 50,057 respectively for the source and target-side of the data. We maintained a static NMT-setup using the same hyper-parameters setting across the pre-trained embedding and synthetic monolingual data experiments.

Validation

Validation or tuning is done to improve translation quality and speed. For statistical approach, the decoder scores translation hypothesis using a linear model that measures the probabilities from language model, translation model, reordering model etc. Validation refers to find the optimal weights for this linear model. Here, optimal weights maximize translation performance on a small set of parallel data (Validation or Tuning data). In neural model, the validation is done in parallel with the training phase. To validate our systems, we used Minimum Error Rate Training (MERT) by (Och and Ney, 2003), which is a batch tuning algorithm.

Evaluation Metrics

We used Bilingual Evaluation Understudy (BLEU) by (Papineni *et al*., 2002) to evaluate the results of our systems. To calculate BLEU score, we need to compute precision for n-grams of size 1 to 4, then add the brevity penalty for too short translations.

Unigram precision is calculated as:

$$P = \frac{m}{w_t} \tag{4}$$

Here, m is the number of candidate words found in the reference and w_t is the total number of candidate words.

Table 2: Statistics of parallel corpora

Corpus name	Training	Tuning	Testing	Total parallel sentence	Author/Developer
SuPara	17,496	300	300	18,096	Al Mumin *et al*. (2012)
Indic Parallel	25,665	300	400	26,365	Post *et al*. (2012)
Open Subtitles	74,398	400	400	75,198	Tiedemann (2012b)
OPUS Ubuntu	5,111	400	300	5,811	Tiedemann (2012d)
OPUS Gnome	1,31,884	300	300	1,32,484	Tiedemann (2012a)
OPUS Tanzil	2,29,577	300	300	2,30,177	Tiedemann (2012c)

Then, the brevity penalty is calculated as:

$$BP = \begin{cases} 1 & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \le r \end{cases} \quad (5)$$

Here, c is the length of the candidate translation and r is the effective reference corpus length.

Finally, the BLEU score is calculated as:

$$BLEU = BP.\exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (6)$$

In our baseline, we used $N = 4$ and uniform weights $W_n = 1/N$.

Results and Analysis

In this section, we describe our experiment results in 2 ways: The automatic evaluation result and the human evaluation result. We also present some sample translations and analyze them.

Automatic Evaluation Result

We used the BLEU metric for our automatic evaluation. Equation 4 to 6 were used to calculate the BLEU score which gives the translation accuracy. The BLEU metric measures how many words overlap in a given translation when compared to a reference translation, giving higher scores to sequential words. BLEU score ranges from 0-100%. The higher the score, the better the translation accuracy is. Table 3 shows the overall results in terms of BLEU. We highlight the best system in bold and give progressive improvements in italic between consecutive systems.

From Table 3 we observe that our PB-SMT system scored the least BLEU, while the baseline NMT system scored a little higher. When provided with the pre-trained embedding for both source and target language, the NMT scored 0.16 higher BLEU than the baseline NMT. The monolingual synthetic data also outperformed the baseline NMT by 0.70 BLEU.

Human Evaluation Result

We performed a manual evaluation of the MT systems along with the automatic evaluation. We followed the guideline by (Brockett *et al*., 2002) to carry out the human evaluation. We assign values from a four-point scale to represent the translation quality on an absolute scale. The human evaluation scale is mentioned in Table 4.

We asked 5 independent evaluators to score 50 translation outputs that were randomly chosen from both the SMT and NMT (Baseline, +pre-trained embedding, +synthetic monolingual data) systems. The human evaluation scores lie in the possibly acceptable to an acceptable range (2~3) for all the systems.

Sample Translation

We present three sample translations for English to Bangla translated by our baseline SMT and NMT systems along with pre-trained word embedding and synthetic data experiments. We observe that our NMT system produces better translation than SMT. Table 5 represents the reference translations with the source sentences. Table 6 represents the translations from our SMT and NMT systems.

Table 3: BLEU score for SuVashantor: English to Bangla MT systems

System	BLEU
PB-SMT	24.08
Baseline NMT	26.76
NMT with pre-trained embedding	26.92 (+0.16)
NMT with synthetic monolingual data	**27.46** (+0.54)

Table 4: Human evaluation scale

Scale	Definition
1 = Unacceptable	Absolutely not comprehensible and/or little or no information transferred accurately.
2 = Possibly acceptable	Possibly comprehensible (given enough context and/or time to work it out); some information transferred accurately.
3 = Acceptable	Not perfect, but definitely comprehensible and with accurate transfer of all important information.
4 = Ideal	Not necessarily a perfect translation, but grammatically correct and with all information accurately transferred.

Table 5: English to Bangla source and reference translations

Source	Reference
Why they didn't do it in the first place?	তারা প্রথমে এটি কেন করলনা ?
See you later	পরে দেখা হবে
Image used with permission	ছবি অনুমতিক্রমে ব্যবহৃত হয়েছে

Table 6: English to Bangla sample translations from our systems

Baseline SMT	Baseline NMT	NMT with Pre-trained Embedding	NMT with Synthetic Data
তারা কেন n't এটা কি ?	কেন তারা এটা প্রথম পদক্ষেপ নিল না?	তারা কেন এটা আগে করলনা?	তারা কেন প্রথম পদক্ষেপ নিল না
পরে দেখা হবে।	পরে দেখা হবে।	পরে দেখা হবে।	পরে দেখা হবে।
অনুমতি সাথে ব্যবহৃত ছবি	ছবি অনুমতিক্রমে ব্যবহৃত।	ছবি অনুমোদিত	অনুমতিক্রমে ব্যবহৃত

We can see from Table 6, in sentence no. 1, for an interrogative sentence, out SMT performs the worst, NMT with monolingual data missed the punctuation mark '?', but baseline NMT and the enhancement with pre-trained embedding produce translations fairly closer to the reference. In sentence no. 2, for a simple affirmative sentence, all our systems produce the exact reference translation. In sentence no.3, the baseline NMT produces an extra punctuation mark '|', which was not present in the source sentence. Also, all the systems produce translation quite deviated from the reference translation.

Conclusion and Future Direction

In this study, we investigate the performance of Statistical Machine Translation systems and Neural Machine Translation systems for English to Bangla language pair with a substantially large parallel corpus. The overall results of our systems show that the system using synthetic monolingual data achieves an additional improvement of 0.7 BLEU score compared to the system using standard NMT language model. The resources used by our system are publicly available, which can be used to continue further research in English-Bangla machine translation.

Data Availability

The raw parallel English-Bangla text corpus data used to support the findings of this study have been deposited in online repositories, which are cited at Table 2. However, we used a pre-processed version of these data by HEQEP under license and so the final version cannot be made freely available.

Our baseline SMT and NMT systems are online for demonstration in this link: https://mt.sustbanglaresearch.org/.

To stimulate the baseline SMT and NMT systems, the official commands from Moses and OpenNMT-tf can be followed.

Funding Information

This work is supported and funded under the project of UGC and HEQEP (CP No: 3888: Development of Multi-Platform Speech and Language Processing Software for Bangla.)

Author's Contributions

Mahjabeen Akter: Background study, data collection and preparation, determining the methodologies, running experiments, preparing the final manuscript.

M. Shahidur Rahman: Background study, contributing to data collection, proof-reading of the manuscript.

Muhammed Zafar Iqbal: Contributing to data collection, investigating different phases of the experiments.

Mohammad Reza Selim: Supervising the research.

Ethics

It is testified by the authors that this article has not been published anywhere else and contains no ethical issues.

References

Al Mumin, A. A., Chy, M. H., Mollah, A., & Selim, R. (2018). English to Bangla Statistical Machine Translation using Neural Probabilistic Language Model.

Al Mumin, M. A., Shoeb, A. A. M., Selim, M. R., & Iqbal, M. Z. (2012). Supara: A balanced english-bengali parallel corpus. SUST Journal of Science and Technology, 46-51.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Banerjee, T., Kunchukuttan, A., & Bhattacharya, P. (2018). Multilingual Indian Language Translation System at WAT 2018: Many-to-one Phrase-based SMT. In WAT@ PACLIC.

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016). Neural versus phrase-based machine translation quality: a case study. arXiv preprint arXiv:1608.04631.

Brockett, C., Aikawa, T., Aue, A., Menezes, A., Quirk, C., & Suzuki, H. (2002). English-Japanese example-based machine translation using abstract linguistic representations. In COLING-02: Machine Translation in Asia.

Burlot, F., & Yvon, F. (2018). Using Monolingual Data in Neural Machine Translation: A Systematic Study. Proceedings of the Third Conference on Machine Translation (WMT), Research Papers, Belgium, Brussels, (pp. 144–155).

Chen, Y., & Skiena, S. (2016). False-friend detection and entity matching via unsupervised transliteration. arXiv preprint arXiv:1611.06722.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Semisupervised learning for neural machine translation. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, (pp. 1965–1974).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Conneau, A., Lample, G., Ranzato, M. A., Denoyer, L., & Jégou, H. (2017). Word translation without parallel data. arXiv preprint arXiv:1710.04087.

Dandapat, S., & Lewis, W. (2018). Training deployable general domain mt for a low resource language pair: English–Bangla.

Federico, M., Bertoldi, N., & Cettolo, M. (2008). IRSTLM: An open source toolkit for handling large scale language models. In Ninth Annual Conference of the International Speech Communication Association.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Islam, M. Z., Tiedemann, J., & Eisele, A. (2010, May). English to Bangla phrase-based machine translation. In Proceedings of the 14th Annual conference of the European Association for Machine Translation.

Jelinek, F. (1997). Statistical methods for speech recognition. MIT press.

Kalchbrenner, N., & Blunsom, P. (2013, October). Recurrent continuous translation models. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1700-1709).

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. arXiv preprint arXiv:1701.02810.

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions (pp. 177-180). Association for Computational Linguistics.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

Moshiul, H. M., & Kamrul, H. M. (2014). English to Bangla Statistical Machine Translation using A* Search Algorithm. Computer Sciences and Telecommunications, 1, 58-71.

Neishi, M., Sakuma, J., Tohda, S., Ishiwatari, S., Yoshinaga, N., & Toyoda, M. (2017, November). A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In Proceedings of the 4th Workshop on Asian Translation (WAT2017) (pp. 99-109).

Och, F. J., & Ney, H. (2002, July). Discriminative training and maximum entropy models for statistical machine translation. In Proceedings of the 40th Annual meeting of the Association for Computational Linguistics (pp. 295-302).

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. Computational linguistics, 29(1), 19-51.

Ojha, A. K., Chowdhury, K. D., Liu, C. H., & Saxena, K. (2018). The RGNLP machine translation systems for WAT 2018. arXiv preprint arXiv:1812.00798.

Pal, S., & Naskar, S. K. (2016). Hybrid word alignment. In Hybrid Approaches to Machine Translation (pp. 57-75). Springer, Cham.

Pal, S., Naskar, S. K., & Bandyopadhyay, S. (2013a, August). A hybrid word alignment model for phrase-based statistical machine translation. In Proceedings of the Second Workshop on Hybrid Approaches to Translation (pp. 94-101).

Pal, S., Naskar, S. K., & Bandyopadhyay, S. (2013b). MWE alignment in phrase based statistical machine translation. The XIV Machine Translation Summit, 61-68.

Pal, S., Naskar, S. K., & Bandyopadhyay, S. (2014, May). Word Alignment-Based Reordering of Source Chunks in PB-SMT. In LREC (pp. 3565-3571).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Post, M., Callison-Burch, C., & Osborne, M. (2012, June). Constructing parallel corpora for six indian languages via crowdsourcing. In Proceedings of the Seventh Workshop on Statistical Machine Translation (pp. 401-409).

Qi, Y., Sachan, D. S., Felix, M., Padmanabhan, S. J., & Neubig, G. (2018). When and why are pre-trained word embeddings useful for neural machine translation?. arXiv preprint arXiv:1804.06323.

Ramachandran, P., Liu, P. J., & Le, Q. V. (2017). Unsupervised Pre-training for Sequence to Sequence Learning. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, (pp. 383–391).

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

Tiedemann, J. (2012a). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012). http://opus.nlpl.eu/GNOME.php

Tiedemann, J. (2012b). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012). http://opus.nlpl.eu/OpenSubtitles.php

Tiedemann, J. (2012c). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012). http://opus.nlpl.eu/Tanzil.php

Tiedemann, J. (2012d). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012). http://opus.nlpl.eu/Ubuntu.php