B. Premjith*, M. Anand Kumar and K.P. Soman

Neural Machine Translation System for English to Indian Language Translation Using MTIL Parallel Corpus

https://doi.org/10.1515/jisys-2019-2510
Received February 8, 2018; previously published online March 20, 2019.

Abstract: Introduction of deep neural networks to the machine translation research ameliorated conventional machine translation systems in multiple ways, specifically in terms of translation quality. The ability of deep neural networks to learn a sensible representation of words is one of the major reasons for this improvement. Despite machine translation using deep neural architecture is showing state-of-the-art results in translating European languages, we cannot directly apply these algorithms in Indian languages mainly because of two reasons: unavailability of the good corpus and Indian languages are morphologically rich. In this paper, we propose a neural machine translation (NMT) system for four language pairs: English–Malayalam, English–Hindi, English–Tamil, and English–Punjabi. We also collected sentences from different sources and cleaned them to make four parallel corpora for each of the language pairs, and then used them to model the translation system. The encoder network in the NMT architecture was designed with long short-term memory (LSTM) networks and bi-directional recurrent neural networks (Bi-RNN). Evaluation of the obtained models was performed both automatically and manually. For automatic evaluation, the bilingual evaluation understudy (BLEU) score was used, and for manual evaluation, three metrics such as adequacy, fluency, and overall ranking were used. Analysis of the results showed the presence of lengthy sentences in English–Malayalam, and the English–Hindi corpus affected the translation. Attention mechanism was employed with a view to addressing the problem of translating lengthy sentences (sentences contain more than 50 words), and the system was able to perceive long-term contexts in the sentences.

Keywords: Neural machine translation, bidirectional RNN, LSTM, English–Indian languages parallel corpus, human evaluation.

1 Introduction

Machine translation (MT) is the process of translating text in one language to another language with the use of software by incorporating both computational and linguistic knowledge. Initially, the MT system finds the translation of a text in the source language by simply associating meaning of the words in the source language to the target language with the help of linguistic rules. However, such methods did not yield good results because of its inability to capture various sentence structures in the language. This translation process is highly time consuming and also required people who have proficiency in both languages. Then, corpus-based translation approaches like statistical machine translation (SMT) [21] and neural machine translation (NMT) [4, 8, 19, 41] were introduced with a vision to rectify the drawbacks of rule-based methods.

NMT is a recently formulated method for automatic translation with the help of deep neural networks. NMT has already shown promising results in translations of several language pairs. Unlike SMT, which requires separately trained sub-components for translation, NMT uses a single large neural network for training. This structure comprised of encoder and decoder networks where the encoder consumes the input

***Corresponding author: B. Premjith,** Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham 641112, India, e-mail: b_premjith@cb.amrita.edu
M. Anand Kumar and K.P. Soman: Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham 641112, India

sentences to produce a vector representation, and the decoder takes this vector and outputs the target language words. Generally, both encoder and decoder networks are designed using the recurrent neural networks (RNN) [14] or long short-term memory (LSTM) or gated recurrent unit (GRU) [9] or bidirectional RNN [32], which are the alternatives to RNN. Even though RNN, especially LSTM, is theoretically proven for handling long-term dependencies in the sentences, practically, translation of long sentences still remains an unsolved problem. This issue was tackled with a technique called attention mechanism [4, 24]. The basic working principle of attention mechanism is to look into different parts of the source sentences, which contributes maximum to the prediction of each target word and translate, instead of taking the entire sentence and translate. This, in turn, helps to align the source and target language words. However, the performance of an NMT system relied greatly on the size and quality of parallel corpus because unlike the SMT system, which has a separate language model, the NMT directly finds the probability of a target sentence for a particular source language sentence [19].

Unfortunately, automatic translations in several language pairs suffer less improvement mainly due to the unavailability of good-quality parallel sentences in abundance. This is the primary problem for the machine translation research in Indian languages not to take a big step. For this reason, the first objective of this work was to create a parallel corpus, which covers sentences from diverse areas and of good quality and large in number and model an NMT system with these sentences in four language pairs (English–Malayalam, English–Tamil, English–Hindi, and English–Punjabi). The motivation behind this was to support machine translation research in resource-poor languages such as Malayalam, Tamil, Hindi, and Punjabi. There are a few parallel corpora available in these language pairs. However, they are not readily usable for research because of the amount of impurities such as spelling mistakes, presence of unknown characters, and presence of bilingual sentences, which do not convey proper meaning in those corpus. To use such data for research, the researchers have to clean it or prepare a separate corpus, both of which are highly time consuming. Creating a parallel corpus is a herculean task especially for language pairs where the availability of bilingual sentences in digital form is very less. In order to avoid such unnecessary time delays, we prepared parallel corpus for four different language pairs by collecting numerous sentences from many resources, which spread across several domains and cleaned them to make good quality corpus. The second objective was to implement NMT for the above-mentioned language pairs by training the system with the prepared corpus. Not many research have been reported on the application of deep neural networks for the translation of language pairs containing Indian languages, and particularly, no other NMT system has been reported for English–Malayalam translation besides Google's NMT system [18]. The third goal of this research was to study the impact of attention mechanism in dealing with long sentences (sentences with the number of words more than 50). Though neural network architectures like the RNN and LSTM networks are theoretically proven capable of remembering long-term contexts; still, the deep learning architectures find it difficult to translate lengthy and complex sentences. Attention mechanism was introduced to address this difficulty in translation. However, still, there are problems in translating lengthy sentences. For this purpose, from the collected sentences, we keep longer sentences along with shorter ones without breaking them into short sentences. Analysis showed that when the number of words in the sentence is very large, the algorithm will not be able to capture the dependencies properly. This will affect the quality of the entire system. The results showed that the presence of a longer sentence affected the accuracy of translation in the English–Malayalam and English–Hindi systems. However, in English–Tamil and English–Punjabi, the number of sentences with a word length of more than 50 was very few. This is one of the reasons why those two systems achieved state-of-the-art results. In the English–Malayalam and English–Hindi corpus, there were a lot of lengthy sentences. To achieve a NMT system that produces quality translations, one of the important things to be taken into consideration is the size of the sentences in the corpus. So, in the corpus, the number of longer and shorter (word length less than four) sentences should be as minimum as possible.

All the four systems were trained with the same architecture and different corpus, and these systems were evaluated using the automatic evaluation method, BLEU (bilingual evaluation understudy) score [26] as well as the human evaluation metrics [11].

The structure of the paper is as follows: in the next section, related work in the field of machine translation in the English–Indian languages is discussed. The statistics and description about the parallel corpus

prepared for machine translation research are given in Section 3, and the methodology behind NMT is described in Section 4. The details about the experiments and he analysis of the results are discussed in Section 5, and finally the paper concludes in Section 6.

2 Related Work

Hindi, Malayalam, Tamil, and Punjabi are among the most widely used languages in India [44]. Also, Hindi and English are the official languages in India [30]. Most of the government documents are available in English and Hindi, and they should be translated into other widely spoken languages to reach the common people. Similarly, an English–Hindi or Hindi–English translation system can help government officials to prepare documents in either English or in Hindi and translate into other languages. Thus, translation from English into these four languages has very much importance in our country.

Much research works have been done on machine translation from English to Indian languages, mostly focusing on ruled-based methods due to the unavailability of good parallel corpora. In spite of the unavailability of sufficient parallel corpora, significant works were done using statistical as well as hybrid approaches to translate text from English to Indian languages. In this section, we will discuss the notable works done in English to Hindi, English to Malayalam, English to Tamil, and English to Punjabi machine translation.

Anglabharti [37] (IIT, Kanpur, India) is one of the oldest machine translation systems in India. It was a multilingual translation system that translates English sentences into Indian languages. This work followed a pseudo-target approach in which the source sentence was converted into an intermediate representation, and the target sentences were generated from this intermediate representation using a text generator. AnglaHindi [36] is an extension to AnglaBharti. In addition to the techniques used in Ref. [37], AnglaHindi (IIT, Kanpur, India) used an example-based approach to translate frequently occurring noun and verb phrases. Anusaaraka [5, 7] was another initiative put forward by the Indian Institute of Technology, Kanpur, which used the principles of Paninian Grammar for translation. Anusaaraka had two modules: the first module does the language-based analysis of the text, whereas the second module performs the statistical analysis.

MANTRA-Rajyasabha [6] developed by CDAC-Pune, India is a machine-aided translation tool primarily designed to translate government documents from English–Indian languages and vice versa. In this approach, both source and target language grammars were represented using the lexicalized tree-adjoining grammar (LTAG). CDAC-Mumbai, India developed a machine translation system called MaTra [25], which relied on a transfer-based approach for translation.

Dave et al. [10] proposed a machine translation approach based on interlingua. The method was designed in such a way that information extracted from sentences in source language are converted into a universal networking language (UNL), and target language sentences are generated from this UNL representation.

Two systems called Shiva [16] and Shakti [15] were developed by the Indian Institute of Science, Bangalore, India, and the International Institute of Information Technology, Hyderabad, India, in association with Carnegie Mellon University. The Shiva system is based on the example-based approach, while the principle behind Shakti is a hybrid approach, which incorporated rule based as well as SMT techniques.

Super Infosoft Pvt Ltd., Delhi, developed an English–Hindi machine translation system named Anuvaadak [3], which used built-in dictionaries for specific domains. This system had a provision to handle out-of-vocabulary (OOV) words by transliterating them. IBM India Research Lab, New Delhi, developed an SMT system [42] that used IBM models 1, 2, and 3. Other notable works in English–Hindi translation were done by Jadavpur University, IIT Delhi [38], and most of them followed the example-based statistical approach.

Some rule-based approaches toward the English–Malayalam automatic translations were proposed by Refs. [13] and [29] and Sunil et al. [40]. Apart from rule-based approaches, various studies were done on SMT in the English–Malayalam translation. In Ref. [28], Rahul et al. proposed a rule-based approach to modify the SMT output. The source language syntax was changed according to the syntax of the target language using some transformation rules and also using a morph analyzer for separating the suffixes from the root words in both languages. Sebastian et al. [33] used parts of speech (POS) information along with bilingual corpus

to improve the alignment model of an SMT system. Unnikrishnan et al. [43] incorporated morphological and syntactic knowledge to the SMT system to improve the quality of translation. They claimed that usage of morphological information in an SMT will reduce the size of parallel corpus to train the model substantially.

One of the initial researches in the English–Tamil machine translation was done by an NLP group [2], which was based on a hash array mapped trie (HAMT) approach. This translation system was constituted by three modules – an English morphological analyzer, a mapping unit, and a Tamil morphological generator. A web version of the English–Tamil translation system was developed by Renganathan [31]. It made use of a morphological tagger for translation. Anand Kumar et al. [1] proposed a factored approach for the English–Tamil translation, which used POS-tagged information and morphological information extracted from both languages to assist in the translation. In Ref. [46], authors applied certain rules to separate suffixes in both source and target languages and fed these information to a SMT system. Sridhar et al. [39] proposed a methodology, which used universal networking language as an intermediate representation in the English–Tamil translation. Ref. [12] is one of the latest research in English–Tamil machine translation in which the authors incorporated deep neural network for translation and further improved the system by segmenting the words morphologically. This improved the translation quality as well as reduced the target language vocabulary size by a factor of 8.

For the English–Punjabi translation, Gurleen Kaur Sidhu and Navjot Kaur proposed a hybrid approach [34] in which they used a word sense disambiguation module to identify the correct sense of words, which will, in turn, improve the translation. Parteek Kumar, in his PhD thesis, [23] proposed the UNL-based approach for the English–Punjabi translation. Another notable work in this area is Ref. [35], which is a statistical approach toward translation and the Anussaraka system [5].

Recently, the application of deep neural networks is making tremendous improvement in machine translation. However, not much studies have been done in English to Indian language translation, specifically in Malayalam. This is because the NMT requires relatively massive amount of good quality bilingual corpora. However, Google came up with a very good translation system called Google Neural Machine Translation (GNMT) system [18, 45], which used millions of samples for translation. The architecture of the Google NMT system consists of LSTM networks in which both the encoder and decoder network contains eight layers each. They also implemented attention mechanism to understand the contextual dependencies in the text and also divided words into a finite set of common word pieces to address the problem of occurrence of unknown words.

3 Parallel Corpus

The current research in machine translation demands huge parallel corpus because SMT and NMT systems are based on the probabilistic models generated using the features extracted from the parallel corpus. In order to derive better features, sentences in the corpus should be of good quality. Therefore, parallel sentences should convey the same meaning, and the presence of spelling mistakes and grammatical mistakes should be avoided. However, the SMT and NMT require a large number of good quality sentence pairs to deduce good feature representations. Unfortunately, such a corpus is not available for the English–Indian languages.

This inadequacy of bilingual corpora affected the machine translation research in Indian languages. Keeping this idea in mind, we decided to collect and clean parallel sentences from various sources for language pairs such as English–Malayalam, English–Hindi, English–Tamil, and English–Punjabi. The number of sentences collected and cleaned is listed in Table 1 and the vocabulary size in each language pair is shown in Table 2.

It is important to collect sentences from various domains while preparing a parallel corpus so that frequently used words in all those domains can be added to the vocabulary. This will further reduce the possibility of occurring out of vocabulary words when the system will be tested. On account of this reason, parallel corpora were prepared by collecting sentences from sources such as websites where bilingual texts are available (e.g. vikaspedia.in), story books, new websites, film subtitles, Bhagavat Gita Bible, Quran, and

Table 1: No. of Sentences, No. of Words, and Average Sentence Length in Four Language Pairs.

Language pair	English–Malayalam		English–Hindi		English–Tamil		English–Punjabi	
	English	Malayalam	English	Hindi	English	Tamil	English	Punjabi
No. of sentences	40,000	40,000	58,030	58,030	46,856	46,856	52,184	52,184
No. of words	718,488	449,289	947,913	1,053,084	619,132	446,281	674,241	760,016
Average words/sentence	18	11	16	18	13	10	13	15

Table 2: Vocabulary Size in Each Language in Four Language Pairs.

Language pair	English–Malayalam		English–Hindi		English–Tamil		English–Punjabi	
Language	English	Malayalam	English	Hindi	English	Tamil	English	Punjabi
Vocabulary size	48,659	106,707	41,352	92,729	79,532	68,567	28,195	27,751

freely available encyclopedias. The coverage of these sentences spread across various fields like film, sports, politics, short stories, agriculture, religion, health, education, and language.

Sentences that were collected from the above-mentioned sources, especially online resources, contained a lot of impurities. The objective of cleaning was to remove foreign characters and incomplete sentences. The sentences that did not convey the meaning properly were omitted, and foreign characters were removed using regular expression. In addition to online resources, parallel sentences were also collected from bilingual books. These books were scanned, and sentences were prepared using the approach described in Ref. [27]. In both cases, long sentences (sentences that contain more than 50 words) were not removed from the corpus with a view to study the performance of neural networks in handling long-term dependencies in the sentence.

This corpus was used in Shared task and workshop on Machine Translation in Indian languages (MTIL-2017) along with the corpus provided by the Technology Development for Indian Languages Programme (TDIL), India [17].

Analysis showed that English–Malayalam and English–Hindi contains a number of lengthy sentences (number of words in the sentence are large), whereas English–Tamil and English–Punjabi do not have sentences with a word length of more than 100. In the English–Tamil and English–Punjabi corpus, the percentage of sentences with a word length <20 is more than 85%, whereas that in the English–Malayalam and English–Hindi corpora is around 75%.

3.1 Test Data Set

Performance of the four NMT systems was tested using a data set containing 562 English sentences. This data set comprises sentences from different domains such as health, tourism, and entertainment. This test data set was prepared separately, and it did not include any sentence from the training data set.

4 Methodology

The fundamental idea behind an NMT system is to predict a sequence of words $Y = (y_1, \ldots, y_t)$ in the target language for a given source language sentence $X = (x_1, \ldots, x_s)$. This conditional probability distribution is modeled using the RNN-based encoder–decoder architecture. The encoder takes the variable length source language sentence and converts it into a fixed length vector. This fixed vector length vector (sentence embedding) contains the meaning of the input sentence. The decoder then takes this sentence embedding as the input and starts predicting the output words by taking the context of each word into consideration. Mathematically, this thought can be represented as,

$$\log P(y|x) = \sum_{k=1}^{t} \log P(y_k|y_{k-1}, \ldots y_1, x, c) \tag{1}$$

where $P(y|x)$ is the probability of obtaining a target language word y for a given source language word x, and c is the context of that particular word.

4.1 Encoder

Basically, an encoder transforms a sentence into a vector form, which represents the meaning of that sentence. Initially, word representations for both source and target language words are obtained. This word embedding is then fed into the encoder–decoder network. The encoder network transforms these word representations into a sentence embedding. This task is performed using a bi-directional recurrent neural network (Bi-RNN) [32] which contains two RNNs for computing rightward and leftward hidden state sequences. This allows us to capture both rightward and leftward contexts of each word.

$$\overrightarrow{h_i} = f(\overrightarrow{h_{i-1}}, W_e x_i) \tag{2}$$

$$\overleftarrow{h_i} = f(\overrightarrow{h_{i+1}}, W_e x_i) \tag{3}$$

where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ represents the rightward and leftward hidden state sequences, W_e is the word embedding matrix, x_i is the input (source) word, and f is a nonlinear function.

Now, each source language word, x_i can be represented using both rightward and leftward hidden state information, i.e. $h_i = (\overrightarrow{h_i}, \overleftarrow{h_i})$. This helps us to obtain more information about each word by incorporating the details of the surrounding words. This information is then fed into the input layer of the decoder.

For example, while translating an English sentence "you always work", it is first represented using a word-embedding mechanism and then fed into the encoder as input. The encoder is built with the LSTM/Bi-RNN networks and a zero vector as the starting state. The last hidden layer of the encoder network generates a vector, which carries the meaning of the input sentence.

The basic block diagrams of the neural network and NMT system (encoder–decoder architecture) are given in Figures 1 and 2, respectively.

4.2 Decoder

The decoder is responsible for predicting target language words by taking the sentence embedding obtained at the encoder, previously predicted target words, and the context of each word. From equation (1), $P(y_k|y_{k-1}, \ldots y_1, x, c)$ is computed as,

$$P(y_k|y_{k-1}, \ldots y_1, x, c) = g(y_{k-1}, f(s_{k-1}, y_{k-1}, c_k), c_k) \tag{4}$$

where g is a nonlinear function, y_{k-1} is the previously predicted target word, c_k is the context of each word, and s_{k-1} is the decoder hidden state at time $k-1$.

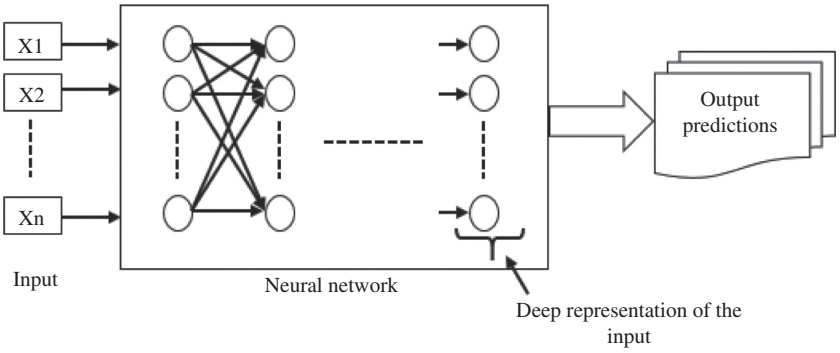

Figure 1: Block Diagram of Neural Network.

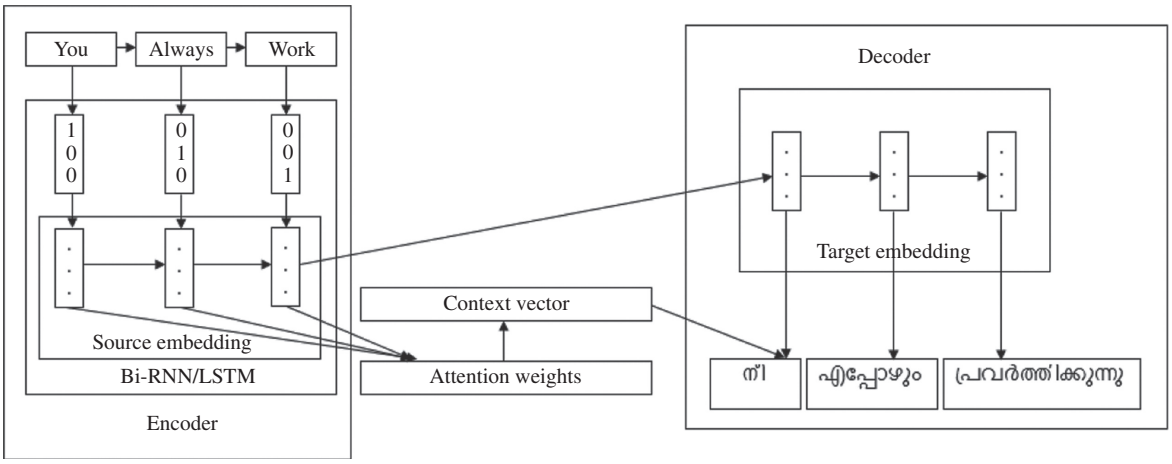

Figure 2: Neural Machine Translation Encoder–Decoder Architecture.

The context vector c_k plays an important role in the prediction of a target word. When the length of the source sentence is large, it is difficult for the machine to remember the context, and in turn, the probability of wrong predictions of target words increases. In order to alleviate this problem, while predicting the k^{th} target word, the machine is allowed to look into certain parts of the input sentence. This is achieved with the help of attention mechanism [4].

Attention mechanism simply finds the association between the target language words and the source language words. Therefore, prediction of each target word depends on the parts of the source language, which has more influence in generating that particular target word. This is executed by annotating each word with a context vector, which is computed using equation (5):

$$c_k = \sum_{i=1}^{s} \beta_{ki} h_i \tag{5}$$

$$\beta_{ki} = \frac{\exp(cs(s_{k-1}, h_i))}{\sum_{n=1}^{s} \exp(cs(s_{k-1}, h_n))} \tag{6}$$

where $cs(s_{k-1}, h_i)$ is the alignment score, which shows the association between the k^{th} output word and the i^{th} input word.

In the decoder, the network is initialized to the last layer of the encoder network because the decoder requires access to the input sentence. Therefore, the hidden layer at the source word "work" is shared with the decoder network, which then acts as the initial state to the decoder. Now, the decoder starts generating target words based on the hidden state information obtained from the encoder. This hidden state vector acts as the context vector, which embeds the knowledge of the source sentence. This vector may not be able to capture all the contexts associated with the source language sentence when the sentence is long. Attention mechanism is used to tackle such problems. In attention mechanism, attention weights are generated by comparing each target hidden state with the source hidden states. Based on the attention vector, a context vector is computed. Then, the attention vector is derived by combining the context vector and the attention weights. This attention vector is then fed as the input to the decoder. A graphical representation of these steps is shown in Figure 2.

5 Results and Discussion

We used the same NMT architecture for the English–Malayalam, English–Hindi, English–Tamil, and English–Punjabi language pairs. The parameters for the system architecture are shown in Table 3, and the

Table 3: System Specification.

Processor	Intel i7-4790 CPU @ 3.60 GHz
Number of core	4
Number of threads	8
Processor base frequency	3.60 GHz
Cache 8 MB	Smart Cache
RAM size	16 GB
RAM type	DDR3
Width	64 bits

configuration of the machine on which the NMT system was trained is shown in Table 4. The models were trained with a corpus described in this paper and bilingual sentences obtained from the TDIL database for training in the NMT system for all the four language pairs, which are listed in Table 5. A baseline system was implemented with the aforementioned architecture [20] using the data set of the size given in Table 5.

Before feeding into the network, sentences were tokenized using the tokenizer module given in the Moses tool kit [22]. The system was trained with different parameters such as the number of hidden layer units (200 and 500) and the neural network algorithm (LSTM and Bi-RNN) of which the set of parameters listed in Table 3 was found to be optimal. Details of the average speed taken by the machine while training is presented in the Table 6.

In order to capture the long-term dependencies in the sentences, we used attention mechanism explained in Ref. [24]. However, this mechanism failed when the source language sentences are too long. That is, contexts in the sentences were captured well when the number of sentences is less. From Table 7, it is understood that a number of sentences with more than 50 words are very high in English–Malayalam and English–Hindi corpus compared to English–Tamil and English–Punjabi. This is one of the reasons why the

Table 4: Training Corpus.

Language pair	Number of sentences
English–Malayalam	103 K
English–Hindi	162 K
English–Tamil	139 K
English–Punjabi	130 K

Table 5: Average Time Taken for Training.

Time taken per epoch	7661 s
Total time taken for training	99,604 s
Speed (tokens/s)	325 s

Table 6: Classification of Sentences in Each Language Pair Based on the Number of Words Present in the Sentence.

No. of words	No. of sentences							
	English–Malayalam		English–Hindi		English–Tamil		English–Punjabi	
	English	Malayalam	English	Hindi	English	Tamil	English	Punjabi
≥200	2	0	2	5	0	0	0	0
≥100 and <200	55	3	45	39	0	0	0	0
≥50 and <100	1051	196	677	1025	38	7	5	7
≥20 and <50	12,934	4410	14,490	18,469	6765	2001	4624	9839
≥10 and <20	13,953	14,613	30,115	27,315	24,970	17,508	32,735	32,234
≥5 and <10	9836	15,144	12,071	10,066	14,309	22,640	14,819	9834
<5	2169	5634	630	1111	774	4700	1	270

Table 7: Training Parameters.

Type of neural network used	Bidirectional RNN
Number of hidden layers in the encoder/decoder	2
Number of hidden units in each hidden layer	500
Word embedding size	300
Batch size	64
Number of training epochs	13
Optimization method	Stochastic gradient method
Initial learning rate	0.5
Drop out	0.3

Table 8: Human Evaluation.

Language pair	Algorithm	Adequacy and fluency (in percentage)	Rating (in percentage)
English–Malayalam	LSTM	42	44.4
	Bi-RNN	49.2	52.7
English–Hindi	LSTM	58.3	65.8
	Bi-RNN	62.13	70.5
English–Tamil	LSTM	27.23	25.95
	Bi-RNN	30.65	30.15
English–Punjabi	LSTM	64.5	69.3
	Bi-RNN	71.62	73.54

systems in Malayalam and Hindi show poor results even when Tamil and Punjabi show relatively good results. Another problem encountered while translating the test sentences was the occurrence of OOV words. Initially, the system was modeled in such a way that all OOV words in the test data set are replaced with a special character <UNK>. However, when the number of words in the source language sentence is large, the translated output will have many <UNK> characters in it. This problem was rectified using a transliteration module. When an OOV word is met during the decoding phase, instead of replacing it with <UNK>, a transliterated version of that particular word is used. This helps the translated sentence to be like a proper translation.

Evaluation of machine translation output is extremely important as it is the key factor that determines the quality of the translation, required level of post-editing, etc. Generally, MT outputs are evaluated both automatically and manually. One of the most common automatic evaluation metric is BLEU [26]. In automatic evaluation, metrics use gold-standard reference translations to which the translation outputs are compared. In human evaluation, annotators are asked to rate the translations on a 1–5 scale based on the source sentence and a gold-standard reference translation, where 1 is the lowest score, and 5 is the highest. Usually, three measures – adequacy, fluency, and rating – are used for human evaluation purposes [11]. Three annotators were assigned to evaluate the translation in each of the four languages. The final score was computed by taking the average of adequacy and fluency and converting into percentages, and the rating score was also converted into percentages. Human evaluation scores are shown in Table 8.

Automatic evaluation of machine translation results has equal importance as that of human evaluation. In this paper, we used the BLEU score to evaluate the translation. The BLEU score was computed for each of the eight architectures and are listed in Table 9. The LSTM and Bi-RNN algorithms were used to design the encoder network and trained the model with 200 and 500 hidden layer units in each of the hidden layers. It is evident from Table 9 that the Bi-RNN algorithm with 500 hidden layer units produces good translations compared to other models. Therefore, we can infer that, an increase in the hidden layer units will improve the translation. Similarly, the use of the Bi-RNN algorithm along with attention mechanism can capture the dependencies in the text well compared to the LSTM with attention mechanism.

Table 9: BLEU Scores of Translation.

Language pair	Algorithm	No. of hidden layers	BLEU score
English–Malayalam	LSTM	200	10.15
		500	12.62
	Bi-RNN	200	11.27
		500	12.75
English–Hindi	LSTM	200	0.98
		500	1.26
	Bi-RNN	200	1.17
		500	2.5
English–Tamil	LSTM	200	20.46
		500	23.78
	Bi-RNN	200	21.25
		500	24.01
English–Punjabi	LSTM	200	23.49
		500	26.54
	Bi-RNN	200	25.76
		500	27.12

6 Conclusion

Machine translation from English to Indian languages are always a difficult task due to the unavailability of a good quality corpus and morphological richness in the Indian languages. For an NMT system to produce better translations, the size of the corpus should be huge. In addition to that, the parallel sentences should convey similar meanings, and the sentences should cover different domains. Modeling the system with such a corpus can assure good translations while testing the model. This corpus is our contribution to the machine translation research community. Apart from the size and coverage of the corpus, the length of the sentences also plays a significant role in determining the quality of translation. The length should not be too short or too long because deep learning architectures cannot extract very long dependencies present in the sentences. As morphological richness in the English and Indian languages are in two extremes of the spectrum, adding linguistic features along with the sentences can improve the translation.

Bibliography

[1] M. Anand Kumar, V. Dhanalakshmi, K. P. Soman and S. Rajendran, Factored statistical machine translation system for English to Tamil language, *Pertanika J. Soc. Sci. Hum.* **22** (2014), 1045–1061.

[2] P. J. Antony, Machine translation approaches and survey for Indian languages, *Int. J. Comput. Linguist. Chinese Language Processing* **18** (2013), 47–78.

[3] Anuvaadak, Available from: http://www.mysmartschool.com/pls/portal/portal.MSSStatic.ProductAnuvaadak. Accessed 31 May, 2017.

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

[5] A. Bharati, V. Chaitanya, A. P. Kulkarni and R. Sangal, Anusaaraka: machine translation in Stages, *arXiv preprint cs/0306130* (2003).

[6] CDAC-MANTRA, Available from: https://www.cdacindia.com/html/aai/mantra.asp. Accessed 31 May, 2017

[7] S. Chaudhury, A. Rao and D. M Sharma, Anusaaraka: an expert system based machine translation system, in: *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, pp. 1–6, IEEE, Beijing, 2010.

[8] K. Cho, B. Van Merriënboer, D. Bahdanau and Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, *arXiv preprint arXiv:1409.1259* (2014).

[9] J. Chung, C. Gulcehre, K. H. Cho and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555* (2014).

[10] S. Dave, J. Parikh and P. Bhattacharyya, Interlingua-based English–Hindi machine translation and language divergence, *Mach. Transl.* **16** (2001), 251–304.

[11] M. Denkowski and A. Lavie, *Choosing the right evaluation for machine translation: an examination of annotator and automatic metric performance on human judgment tasks*, AMTA, 2010.
[12] K. Hans and R. S. Milton, Improving the performance of neural machine translation involving morphologically rich languages, *arXiv preprint arXiv:1612.02482* (2016).
[13] R. Harshawardhan, Rule based machine translation system for English to Malayalam language, *Diss. de mestrado*. Coimbatore: Amrita School of Engineering (2011).
[14] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9** (1997), 1735–1780.
[15] IIIT, Available from: http://shakti.iiit.net/. Accessed 31 May, 2017.
[16] IISC, Available from: http://ebmt.serc.iisc.ernet.in/mt/login.html. Accessed 31 May, 2017.
[17] G. N. Jha, The TDIL Program and the Indian Langauge Corpora Intitiative (ILCI), in: *LREC*, New Delhi, India, 2010.
[18] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, J. Dean, Google's multilingual neural machine translation system: enabling zero-shot translation, *arXiv preprint arXiv:1611.04558* (2016).
[19] N. Kalchbrenner and P. Blunsom, Recurrent continuous translation models, in: *EMNLP*, 3, p. 413, Seattle, WA, USA, 2013.
[20] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. M. Rush, OpenNMT: open-source toolkit for neural machine translation, *arXiv preprint arXiv:1701.02810* (2017).
[21] P. Koehn, *Statistical machine translation*, Cambridge University Press, Cambridge, UK, 2009.
[22] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, 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, 2007.
[23] P. Kumar and R. K. Sharma, *UNL based machine translation system for Punjabi language*, Ph.D. thesis, Thapar University, Patiala, India, 2012.
[24] M.-T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, *arXiv preprint arXiv:1508.04025* (2015).
[25] NCST-MATTRA, http://www.ncst.ernet.in/matra/. Accessed 31 May, 2017.
[26] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, Association for Computational Linguistics, 2002.
[27] B. Premjith, S. Sachin Kumar, R. Shyam, M. Anand Kumar and K. P. Soman, A fast and efficient framework for creating parallel corpus, *Indian J. Sci. Technol.* **9** (2016), 1–7.
[28] C. Rahul, K. Dinunath, R. Ravindran and K. P. Soman, Rule based reordering and morphological processing for English-Malayalam statistical machine translation, in: *Advances in Computing, Control, and Telecommunication Technologies, 2009. ACT'09. International Conference on*, pp. 458–460, IEEE, Trivandrum, India, 2009.
[29] R. Rajan, R. Sivan, R. Ravindran and K. P. Soman, Rule based machine translation from English to Malayalam, in: *Advances in Computing, Control, and Telecommunication Technologies, 2009. ACT'09. International Conference on*, pp. 439–441, IEEE, Trivandrum, India, 2009.
[30] Rajbhasha.nic.in Official languages in India, Available from: http://rajbhasha.nic.in/en/constitutional-provisions. Accessed 10 June, 2017.
[31] V. Renganathan, An interactive approach to development of English-Tamil machine translation system on the web, in: *The International Tamil Internet 2002 Conference and Exhibition (TI2002)*, University of Pennsylvania, Philadelphia, PA, 2002. Available at: https://www.sas.upenn.edu/~vasur/from_ccat/papers/english_tamil_mt.pdf.
[32] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Signal Process.* **45** (1997), 2673–2681.
[33] M. P. Sebastian, K. K. Sheena, G. Santhosh Kumar, English to Malayalam translation: a statistical approach, in: *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, p. 64, ACM, 2010.
[34] G. K. Sidhu, N. Kaur and IAEME Publication, English to Punjabi machine translation system using hybrid approach of word sense disambiguation and machine translation, *IJCET* **4** (2013), 350–357.
[35] G. Singh and P. G. Bhatia, English to Punjabi statistical based machine translation system, (2010).
[36] R. M. K. Sinha and A. Jain, AnglaHindi: an English to Hindi machine-aided translation system, *MT Summit IX*, New Orleans, *USA* (2003), 494–497.
[37] R. M. K. Sinha, K. Sivaraman, A. Agrawal, R. Jain, R. Srivastava and A. Jain, ANGLABHARTI: a multilingual machine aided translation project on translation from English to Indian languages, in: *Systems, Man and Cybernetics, 1995*. Intelligent Systems for the 21st Century, IEEE International Conference on, 2, pp. 1609–1614, IEEE, Vancouver, Canada, 1995.
[38] S. B. Sitender, Survey of Indian machine translation systems, *IJCST* **3** (2012), 47.
[39] R. Sridhar, P. Sethuraman and K. Krishnakumar, English to Tamil machine translation system using universal networking language, *Sā dhanā* **41** (2016), 607–620.
[40] R. Sunil, N. Manohar, V. Jayan and K. G. Sulochana, Development of Malayalam text generator for translation from English, in: *India Conference (INDICON), 2011 Annual IEEE*, pp. 1–6, IEEE, Hyderabad, 2011.
[41] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, in: *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

[42] R. Udupa and T. A. Faruquie, An English-Hindi statistical machine translation system, in: *International Conference on Natural Language Processing*, pp. 254–262, Springer, Hainan Island, China, 2004.
[43] P. Unnikrishnan, P. J. Antony and K. P. Soman, A novel approach for English to South Dravidian language statistical machine translation system, *IJCSE* **2** (2010), 2749–2759.
[44] Wikipedia List of languages by number of native speakers in India, Available from: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India. Accessed 10 June, 2017
[45] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. S. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: bridging the gap between human and machine translation, *arXiv preprint arXiv:1609.08144* (2016).
[46] L. R. O. B. ZdenekŽabokrtskỳ, Morphological processing for English-Tamil statistical machine translation, in: *24th International Conference on Computational Linguistics*, p. 113, Mumbai, India, 2012.