Amarnath Pathak and Partha Pakray*

Neural Machine Translation for Indian Languages

https://doi.org/10.1515/jisys-2018-0065
Received January 30, 2018; previously published online June 18, 2018.

Abstract: Machine Translation bridges communication barriers and eases interaction among people having different linguistic backgrounds. Machine Translation mechanisms exploit a range of techniques and linguistic resources for translation prediction. Neural machine translation (NMT), in particular, seeks optimality in translation through training of neural network, using a parallel corpus having a considerable number of instances in the form of a parallel running source and target sentences. Easy availability of parallel corpora for major Indian language forms and the ability of NMT systems to better analyze context and produce fluent translation make NMT a prominent choice for the translation of Indian languages. We have trained, tested, and analyzed NMT systems for English to Tamil, English to Hindi, and English to Punjabi translations. Predicted translations have been evaluated using Bilingual Evaluation Understudy and by human evaluators to assess the quality of translation in terms of its adequacy, fluency, and correspondence with human-predicted translation.

Keywords: Machine Translation, Neural Machine Translation, OpenNMT, BLEU, Indian languages.

1 Introduction

Machine Translation (MT), an application area of Natural Language Processing (NLP) and a subfield of computational linguistics, facilitates automated translation of text or speech in a source natural language to corresponding text or speech in a different target natural language. Language incomprehensibility has wide-ranging adverse impacts on several aspects of human living, and the same can be reasonably alleviated with effective use of MT. Besides, the crucial idea of MT is to bridge communication barriers among people from different linguistic backgrounds.

Although MT-predicted translations differ from human-like translation, they are comprehensible and the translation process is free from human intervention. The effectiveness of the translation approach is manifested in its potential to ensure generation of semantically equivalent and grammatically sound target construct. An intellectual translation approach refrains from a word-for-word translation behavior and delves into conceptuality and crux of the languages, prior to translation.

Classical approaches of MT are broadly categorized into rule-based, corpus-based, and hybrid approaches. Rule-based approaches, namely transfer-based and interlingua-based approaches, rely on a set of predefined translation rules, and investigate the syntax, semantics, and morphology of the two languages to furnish target representation. Rule-based approaches resort to linguistic models to ensure comprehensibility of translation rules and to produce a syntactically and semantically sound translation that is well formed and less prone to grammatical errors [21]. However, interlingua-based approaches are inefficient, primarily because of their impractical and infeasible idea of using language-independent representation for translation.

***Corresponding author: Partha Pakray,** Department of Computer Science and Engineering, National Institute of Technology Mizoram, Mizoram, India, e-mail: parthapakray@gmail.com. https://orcid.org/0000-0003-3834-5154
Amarnath Pathak: Department of Computer Science and Engineering, National Institute of Technology Mizoram, Mizoram, India. https://orcid.org/0000-0002-1666-4464

Corpus-based or data-driven approaches, namely Example-Based Machine Translation (EBMT) and Statistical Machine Translation (SMT), dynamically build a translation model that is characterized by a set of translation rules learned from parallel corpus. A considerable number of instances in the corpus, and the ability of the translation approach to dynamically dig out excellent rules guide the result of translation to perfection. Given the test data (text to be translated), EBMT investigates the translation model to seek the optimal match using techniques of clustering or generalization. Parallel corpora, thesaurus for computing semantic similarity, bilingual dictionary, and syntactic parser are the key resources that are crucial to the functioning of EBMT. On the other hand, SMT employs Bayes' theorem to reformulate the translation problem as a probability maximization problem. Considering S to be the source language sentence and T to be the target language sentence, SMT attempts to find translation T that maximizes $P(T|S)$. Using Bayes' theorem, $P(T|S)$ can be rewritten as the right-hand side of Eq. (1):

$$P(T|S) = \frac{P(S|T) * P(T)}{P(S)}. \tag{1}$$

With $P(S)$ being fixed, the problem of translation boils down to finding a translation T, which maximizes $P(S|T)*P(T)$. Language model, $P(T)$, and translation model, $P(S|T)$, are learnt from the parallel corpora and constitute indispensable components of SMT. The two components are exploited by the decoding algorithm to predict target sentences. The inability of the SMT system to exploit context information of the source sentence, system complexity, and use of many different independently trained components are the key factors that add to the inefficiency of SMT systems.

Although conventional approaches of translation have served the purpose for years, their underlying demerits and the need for better-quality translation enforce exploration of new, better-performing techniques. Neural Machine Translation (NMT), a fairly new and proficient approach to translation, incorporates the use of adequately trained large neural networks in the translation process. Encoder and decoder, which are the networks of Long Short-Term Memory (LSTM) units, constitute key components of the NMT system architecture. In a baseline system, encoder encodes the source sentence, one symbol at a time, and stores the entire encoding in its last hidden state. Encoded representation is fed to decoder for translation prediction. The comprehensibility, adequacy, and fluency of predicted translation are largely determined by the implementation approach used for decoder. Recent past has witnessed the use of Convolutional Neural Networks (CNNs) as well as Recurrent Neural Networks (RNNs) for implementing the decoder network [4, 23]. However, carrying encoded representation along the decoder network is a primary requirement of the decoding process, and simple RNNs are bad at it. An improved implementation strategy for decoder replaces simple RNNs with LSTM, which memorizes encoded information and carries enough of it along the network. Use of attention mechanism [14] furthers the effectiveness of translation by allowing decoder to access the entire pool of encoder states for translation prediction. The NMT system endows scalability and a radical shift from phrase-based translation, as in the case of SMT, to sentence-based translation. Unlike classical MT systems, NMT systems are end-to-end systems that discard the use of additional components for translation. Decoder in NMT exploits a comparatively larger context, comprising source as well as partial target text, for accurate translation prediction. NMT is better at generating more fluent translations with less syntactic and semantic errors.

Motivated by the advantages of NMT over classical MT systems and the promising results produced by NMT in recent years, we have investigated its effectiveness in the context of Indian languages. In particular, we have trained and tested NMT systems for English to Tamil, English to Hindi, and English to Punjabi translations. Predicted translations have been evaluated by employing human evaluators and Bilingual Evaluation Understudy (BLEU) evaluation [18]. Besides, we have comprehensively analyzed the implications at the performance of the English-Hindi NMT system, given changes in training data, epochs, and length of sentences in the test set.

The rest of the paper is organized as follows: Section 2 reviews relevant literature on translation of Indian languages. Section 3 details the system architecture for NMT. Section 4 describes details of different experimental setups. Section 5 describes system results and their comprehensive analysis. Section 6 concludes the paper and points the direction for future research.

2 Related Works

The era of NMT emerged in 1987 when English to Spanish translation was attempted using backpropagation neural network and a highly limited vocabulary [1]. The translation process used backpropagation mechanism for mapping from one language to another. The NMT system architecture has been subjected to a number of modifications after its resurgence in 2013. An RNN encoder-decoder NMT system facilitates encoding of a variable-length source sentence into a fixed-length vector and decoding of fixed-length vectors to obtain the target sequence [3]. Gated Recurrent Units replaced CNNs [8] for implementing hidden units of decoder. LSTM [22] offers improved propagation of the encoded source sentence along the decoder network, resulting in improved translation quality of longer sentences.

Use of neural networks on phrase-based SMT has been limitedly explored in the context of Indian languages, such as Tamil and Punjabi. Neural networks have been used to learn sets of ordered rules for Hindi to English translation, and the idea can be extended to other Indo-European languages by changing the dictionary used for literal translation [2]. A feed-forward backpropagation Artificial Neural Network (ANN) architecture uses nine separate modules for translating simple sentences of the English language into Hindi [9]. Bilingual dictionary, and storing word meanings and word features of a language pair have been implemented using ANN. A quantum neural network-based approach for English to Hindi translation learns the pattern of the English-Hindi parallel corpus using part-of-speech information of each of the word in corpus, and later uses the gained knowledge to perform translation [16]. Use of quantum neural network for reordering of words for parts-of-speech tagging and their alignment during the MT has been found to increase the translation accuracy to a significant extent. Translation accuracy can be further enhanced by increasing the size of the training corpus through automatic extraction of parallel texts from comparable corpora and adding them to the training data [17].

A direct translation methodology for Punjabi to Hindi translation exploits the syntactic and semantic similarity between the two languages [7]. Using a number of lexicons and the word-for-word translation approach, words in the source language are replaced by their target language equivalents. An extended web-based Hindi to Punjabi MT system facilitates website and email translation [5].

Ambiguities of content as well as some function words pose challenges to MT systems. A supervised method for resolving prepositional ambiguity in English to Tamil translation performs disambiguation by exploiting collocation occurrences and linguistic information of words [12]. English to Tamil translation confronts issues of non-availability of parallel corpora and morphological difference between the two languages [11]. Such issues have been tackled by incorporating linguistic knowledge in SMT. Formalism-based MT approach for English to Tamil translation uses synchronous TAG pairs derived from XTAG English grammar [15]. Proposed tag systems can be well extended to other Indian languages as well. Furthermore, integrating rule-based MT systems with functionality for complex sentence simplification has been found to improve the quality of English to Tamil translation [20]. Complex sentences, connected using connectives, are split into simpler sentences before feeding them to the translation system.

Owing to the diversities of Indian languages and exceedingly better performance of NMT, Google has recently employed NMT in multilingual translation of nine Indian languages (http://indianexpress.com/article/technology/tech-news-technology/googles-neural-machine-translation-for-indian-languages-heres-what-it-means/), namely Hindi, Bengali, Marathi, Tamil, Telugu, Gujarati, Punjabi, Malayalam, and Kannada. Google's multilingual NMT system is characterized by simplicity, low-resource language improvement, and zero-shot translation features [6]. The zero-shot translation feature enables the system to do translation prediction for a previously unseen language pair. Google's multilingual NMT system does not affect the basic encoder-decoder architecture but instead uses a special token at the beginning of source sentence to specify the target language.

The Shared Task Cum Workshop on Machine Translation in Indian Languages (MTIL; http://nlp.amrita.edu/mtil_cen/) offers research infrastructure for the development, evaluation, and comparison of the MT systems [13]. The specific objectives of the workshop were to design high-quality parallel corpora of Indian languages, to explore the role of state-of-the-art MT techniques in the context of Indian languages,

and to cope with the issues of language divergence. A total of five teams participated in the workshop. CDAC-M was the top-ranked team in English-Malayalam, English-Tamil, and English-Hindi translation categories, whereas our NIT-M team was top ranked in the English-Punjabi translation category. Human evaluation score was preferred over BLEU score for ranking of the teams.

The factored SMT model of the CDAC-M team employs suffix separation (SS), source side reordering, and transliteration for all the four translation categories [19]. Reordering reorders the source side sentences as per the word order of target language. Reordering leads to better alignments and parallel phrase extraction, which eventually improves the translation quality. During SS, words are split into stem and suffixes, and a continuation symbol (@@) is added after the stem word. The continuation symbol helps combine suffixes after having performed the translation. Transliteration, a post-processing step, translates an out-of-vocabulary (OOV) word to the target language word. An unsupervised model based on expectation maximization has been used to train a transliteration model. Eventually, language modeling selects the best translation from the n-best transliterated output. Augmenting Moses (https://github.com/moses-smt)-based baseline system with pre-processing and post-processing steps have led to improvement in BLEU score for the English-Hindi and English-Tamil translation categories.

3 System Description

Data pre-processing, system training, and system testing/translation constitute key steps of system functioning, and the same have been elaborated in the following subsections. We have exploited the OpenNMT (http://opennmt.net/) system architecture, tuned its parameters, and trained and tested the system using corpora provided by MTIL organizers [13].

3.1 Data Preprocessing

MTIL corpora have been used to train and test the OpenNMT system (refer to Section 4.1 for corpora description). Raw data consist of parallel running source and target sentences, which are tokenized during the pre-processing step. Validation data, derived from training data provided by the organizers, has been used to evaluate convergence of training and to check the validity of system parameters. The remaining training data have been employed in system training. Thus, the training and validation data have no instances in common and are therefore disjoint. Using the entire training data for validation could result in an undesirable overfitted model. A usual practice enforces a maximum limit of 5000 over the number of sentences in validation files.

The pre-processor primarily aims at building dictionaries that index the words present in the training and validation datasets. Training and validation files are fed to the pre-processor module of OpenNMT to generate two human-readable dictionary files and a serialized torch file. Dictionary files list out all unique words of training data along with four extra words, namely *<blank>*, *<unk>*, *<s>*, and *</s>*. Each word is mapped to a unique index that serves as the system's internal representation for the word. The serialized torch file, which embeds dictionaries, training data, and validation data, is used to train the system.

3.2 System Training

We have trained a sequence-to-sequence recurrent neural network model, using attention mechanism, for translation prediction. Training data are shuffled and sorted prior to training. Shuffling ensures that instances in the training batch uniformly come from different parts of the corpus. Sorting ensures uniform length of instances in the training batch. Training performance can be escalated with use of multiple Graphics Processing Units (GPUs) to train different batches of training data in synchronous or asynchronous fashion.

An epoch in training refers to one forward and one backward pass over all the training instances. The system is trained for some fixed number of epochs. An epoch comprises iterations, and in each iteration

one forward pass and one backward pass are performed over a set of training instances. The system has been trained for 15 epochs in the first experimental setup and 19 epochs in the other three setups (refer to Section 4.2 for details of the experimental setups). A validation score, dynamically computed using validation data, helps in checking the convergence of training. The learning rate of the network decays by a factor of 0.7 if validation score improvement falls below 0.

The primary components of the system architecture are discussed next.

3.2.1 Encoder

A unidirectional sequencer has been used as encoder for encoding variable-length input sequence of the source language into fixed-size vectors. The architecture of encoder is characterized by a two-layer LSTM recurrent neural network having 500 hidden units in each layer. LSTM, unlike conventional backpropagation neural networks, remembers encoded source representation and carries enough of it along the decoder network.

To convert the input sequence into word embedding, encoder splits the input sequence into an array of words. Each word in the array is mapped to its index in the vocabulary, and the index of the <*unk*> word is used for OOV words. Sorting of the input sequences in a batch, prior to training, eliminates the need for zero padding. Thereafter, each index value is transformed into a vector of fixed length and different word vectors are combined to give a single fixed-length vector that is representative of the complete input sequence.

3.2.2 Decoder

Similar to encoder, a two-layer LSTM decoder, having 500 hidden units in each layer, has been used to decode a fixed-size source vector using input feeding and global attention mechanism. A decoder using global attention mechanism consults the entire pool of source states at each step of decoding. Not only the last hidden state but also the entire hidden states of source are considered as representatives of sentence meaning. Score function takes the current hidden state of decoder h_t and the source vector $\overline{h}_s$ as argument to determine the attention score of each source state. The score function used by the system is given by Eq. (2):

$$\text{score}(h_t, \overline{h}_s) = h_t^T W_a \overline{h}_s. \tag{2}$$

The probabilistic attention score of a state is its score divided by sum of all the attention scores. Variable-length weight alignment vector a_t uses probabilistic attention scores and gives an estimate of the amount of attention to pay at different places in the source. A context vector c_t is then computed using source vector $\overline{h}_s$ and weight alignment vector a_t. The decoder uses its current hidden state information h_t and context vector c_t to predict attentional vector $\overline{h}_t$, which eventually predicts the current word y_t. Figure 1 describes working of decoder in a nutshell [14].

Moreover, the system uses the input feeding approach to feed attentional vector $\tilde{h}_t$ to current hidden state, a mechanism that keeps the system informed about past alignment decisions.

Figure 2 illustrates the system architecture. Attention mechanism and input feeding are used to transform input sequence “A B C D” into target sequence “X Y Z” [14].

3.3 System Testing/Translation

System testing uses a trained model to predict translation for test sentences, which are fed to the system in batches. The translation process makes use of beam search, a heuristic-based optimized version of best first search, to search for the best or the list of best translations. The effectiveness of the search mechanism is manifested in its ability to facilitate trade-off between translation time and search accuracy, which is ensured by setting beam size to a relatively small value. Besides, the translator uses the <*unk*> symbol when it is uncertain about the target word.

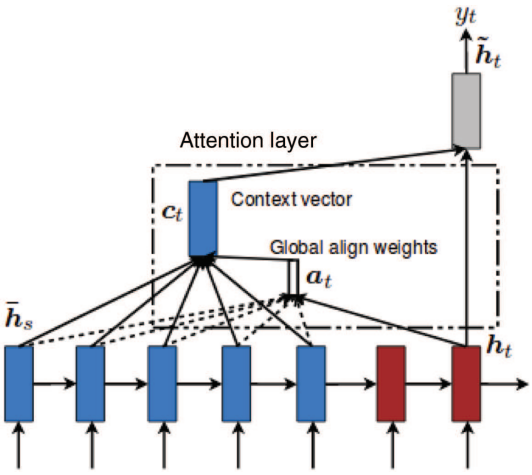

Figure 1: LSTM Decoder Using Context Vector and Current Hidden State for Translation Prediction.

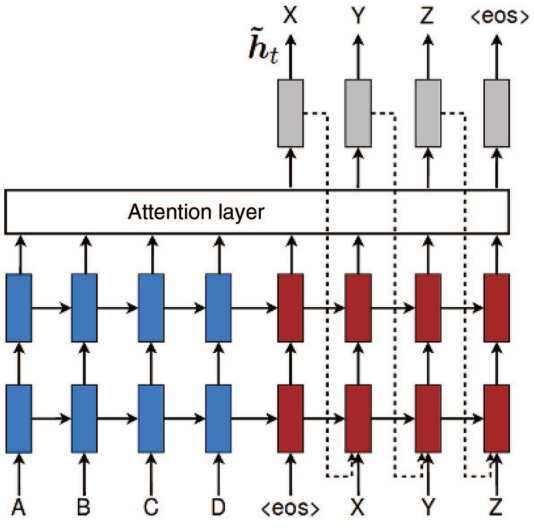

Figure 2: System Architecture for NMT.

4 Experimental Design

This section contains detailed description about the experimental setup and corpora used for training and testing the translation effectiveness of the NMT system. OpenNMT, an open source toolkit, facilitates the required experimental framework, and provides a platform for training and deploying NMT models [10].

4.1 Corpora Description

The NMT system has been trained using Hindi, Punjabi, and Tamil MTIL training corpora, which comprise parallel running source-target sentence pairs, with English being the source language in all the three corpora. System training imposes certain constraints on the corpora format, which necessitate pre-processing of data prior to training (refer to Section 3.1). Validation data, a subset of training corpus containing 4000 instances, are used for checking the convergence of training. The MTIL test corpus containing 562 English sentences has been used for testing the translation effectiveness of trained and validated models. Table 1 summarizes the nature of corpus, the name of the corresponding corpus, and the number of instances present therein [11, 12, 20].

Table 1: Corpora Description.

Nature of corpus	Name of corpus	Number of instances
Training	Hindi_MTIL2017-Training	160,758
	Punjabi_MTIL2017-Training	129,022
	Tamil_MTIL2017-Training	139,033
Test	MTIL-Test	562
Validation (Hindi)	Hindi_MTIL2017-Training	4000
Validation (Punjabi)	Punjabi_MTIL2017-Training	4000
Validation (Tamil)	Tamil_MTIL2017-Training	4000
Gold	MTIL-Hindi_Gold	562

4.2 Experimental Setup

We have used the following different experimental setups to train, test, and analyze the system's performance from different perspectives.

(i) Initially, we trained the NMT system using English-Hindi, English-Punjabi, and English-Tamil parallel training corpora. The three different trained models were tested using MTIL-Test corpus. Result sets containing predicted translations were provided to MTIL organizers for human and BLEU evaluation.

(ii) We have re-trained the NMT system using Hindi_MTIL2017-Training corpus and saved the trained model obtained at 19 different epochs. Each of the 19 models has been tested using MTIL-Test corpus, and prediction results have been subjected to BLEU evaluation using Gold Data provided by the organizers. Such a setup helps in analyzing the change in translation behavior of the NMT system with increase in the number of epochs.

(iii) Besides, we have re-trained the NMT system for 19 epochs using 80 k instances of Hindi_MTIL2017-Training corpus, which is approximately half of the original corpus size. The prediction results from each of the 19 models have been subjected to BLEU evaluation. Such a setup helps in analyzing change in the translation behavior of the NMT system with change in the number of instances in the training data.

(iv) Furthermore, we have created four different test sets from the original test data, each test set containing 100 sentences. The average length of sentences in the four test datasets is 10, 15, 20, and 25, respectively. The best epoch model, one having the highest BLEU score, is tested using the four test datasets, and prediction results have been evaluated using BLEU evaluation. Such a setup helps in assessing the relationship between translation performance and the average length of sentences in the test dataset.

The results of all these experimental setups have been detailed and analyzed in Section 5.

5 Results and Analysis

5.1 Evaluation Results Provided by MTIL Organizers

The prediction results of our first experimental setup were provided to MTIL organizers for human and BLEU evaluation [13]. Human evaluators assessed the quality of translation with respect to adequacy, fluency, and overall rating. They compared the prediction results and Gold Data to rate the three parameters on a scale of 1 to 5, with 1 being the least and 5 being the maximum parameter value.

The adequacy of translation is a measure of the amount of meaning expressed in a reference translation that is also expressed in a translation sentence. Table 2 summarizes the adequacy ratings provided by three independent human evaluators for all target languages and for all participating teams. Among all the participating teams, our NIT-M team attained the highest adequacy rating of 3.38 and the least adequacy rating of 1.59 for the English-Punjabi and English-Tamil language pairs, respectively. In Tables 2–5, different scores of our NIT-M team are highlighted in bold.

Table 2: Adequacy Table.

Language	Team				Adequacy
		Evaluator 1	Evaluator 2	Evaluator 3	Average
Malayalam	CDAC-M	2.20	1.36	2.20	1.92
Tamil	CDAC-M	3.34	1.82	2.69	2.62
	NIT-M	**1.53**	**1.69**	**1.54**	**1.59**
	Hans	3.14	1.83	1.51	2.16
Hindi	JU	1.97	1.51	1.97	1.81
	IIT-B	2.44	2.66	2.55	2.55
	CDAC-M	3.92	3.71	3.82	3.82
	NIT-M	**3.49**	**2.59**	**3.72**	**3.27**
Punjabi	IIT-B	2.03	3.62	2.29	2.65
	NIT-M	**3.30**	**3.45**	**3.38**	**3.38**
	CDAC-M	2.42	3.44	3.28	3.05

CDAC-M, Centre for Development of Advanced Computing, Mumbai; IIT-B, Indian Institute of Technology, Bombay; NIT-M, National Institute of Technology Mizoram; JU, Jadavpur University; HANS, SSN College of Engineering.

Table 3: Fluency Table.

Language	Team				Fluency
		Evaluator 1	Evaluator 2	Evaluator 3	Average
Malayalam	CDAC-M	1.76	1.34	1.90	1.67
Tamil	CDAC-M	2.95	1.81	2.94	2.57
	NIT-M	**1.51**	**1.72**	**1.72**	**1.65**
	Hans	3.09	1.76	1.50	2.12
Hindi	JU	1.80	1.54	1.81	1.72
	IIT-B	2.93	3.52	3.23	3.23
	CDAC-M	3.61	3.65	3.63	3.63
	NIT-M	**3.94**	**2.97**	**3.76**	**3.56**
Punjabi	IIT-B	2.10	3.60	2.43	2.71
	NIT-M	**3.84**	**3.64**	**3.74**	**3.74**
	CDAC-M	2.48	3.49	3.10	3.02

Table 4: Rating Table.

Language	Team				Rating
		Evaluator 1	Evaluator 2	Evaluator 3	Average
Malayalam	CDAC-M	1.83	1.29	1.68	1.60
Tamil	CDAC-M	2.95	1.82	2.43	2.40
	NIT-M	**1.51**	**1.71**	**1.52**	**1.58**
	Hans	2.96	2.06	1.50	2.17
Hindi	JU	1.61	1.51	1.60	1.57
	IIT-B	2.37	2.81	2.59	2.59
	CDAC-M	3.27	3.59	3.43	3.43
	NIT-M	**3.94**	**2.37**	**3.45**	**3.26**
Punjabi	IIT-B	1.94	3.62	2.30	2.62
	NIT-M	**2.85**	**3.65**	**3.25**	**3.25**
	CDAC-M	2.28	3.40	3.07	2.92

The fluency of translation concerns the well formedness of a translation sentence in the target language, irrespective of sentence meaning. A fluent translation will be flawless, syntactically correct, and comprehensible, but need not necessarily be a semantically correct translation. Table 3 summarizes the fluency ratings provided by three independent human evaluators for the three language pairs. The highest fluency rating of

Table 5: Percentage Measures and BLEU Score.

Language	Team	Adequacy and fluency in %	Rating in %	BLEU score
Malayalam	CDAC-M	35.85	31.94	2.6
Tamil	CDAC-M	51.8	48	6.15
	NIT-M	**32.37**	**31.64**	**1.31**
	Hans	42.8	43.5	1.93
Hindi	JU	35.28	31.5	3.57
	IIT-B	57.81	51.87	21.01
	CDAC-M	74.53	68.64	20.64
	NIT-M	**68.27**	**65.14**	**23.25**
Punjabi	IIT-B	52.93	52.4	11.38
	NIT-M	**67.55**	**65.05**	**9.24**
	CDAC-M	60.91	58.34	8.68

3.74 and lowest fluency rating of 1.65 were scored by our NIT-M team for the English-Punjabi and English-Tamil language pairs.

Overall rating expects annotators to rate the predicted translations on a scale of 1–5, with the least value of 1 referring to completely wrong and worst translation whereas the maximum value of 5 refers to excellent translation. The overall ratings provided by evaluators for the three language pairs are summarized in Table 4 The lowest overall rating of 1.58 and highest overall rating of 3.26 were attained by our NIT-M team for the English-Tamil and English-Punjabi translations, respectively.

Further, Table 5 summarizes the percentage measures and BLEU score for the three language pairs. Our NIT-M team attained the highest BLEU score of 23.25 and lowest BLEU score of 1.31 for the English-Hindi and English-Tamil language pairs, respectively.

As can be seen from Tables 2–5, among all the teams, our team ranked first in English to Punjabi and English to Hindi translations from the human evaluation and BLEU evaluation perspectives, respectively. Moreover, for all the language pairs and among all the three human evaluation parameters, the highest measures have been recorded for fluency. This is attributed to the fact that NMT systems are well known for producing fluent translations.

5.2 Performance Comparison of NIT-M and CDAC-M Systems

Figures 3–5 show the comparison of adequacy, fluency, and BLEU scores of NIT-M and CDAC-M systems for the three translation categories.

The three scores of the NIT-M team are lower than those of CDAC-M for the English-Tamil translation. This owes to the agglutinative nature of Tamil and the morphological divergence between the English and Tamil languages. The majority of Tamil words are formed by a combination of multiple words, a phenomenon referred to as agglutination. Agglutination and morphological divergence lead to generation of more unknown (*<unk>*) words, which lowers the quality of translation. As discussed in the Section 2, the

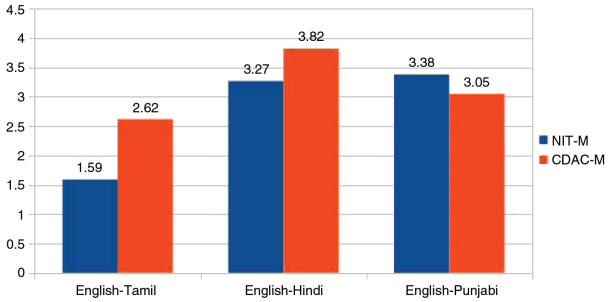

Figure 3: Comparison of Adequacy Scores of NIT-M and CDAC-M Systems.

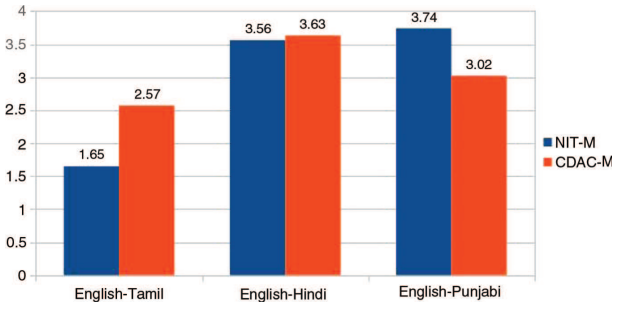

Figure 4: Comparison of Fluency Scores of NIT-M and CDAC-M Systems.

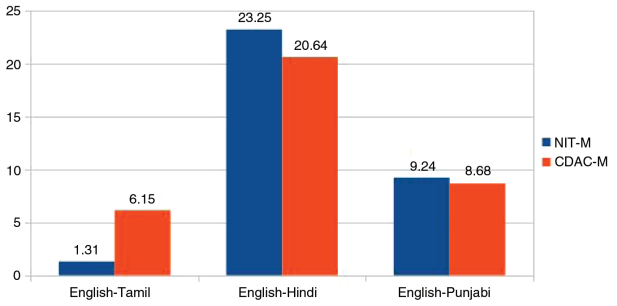

Figure 5: Comparison of BLEU Scores of NIT-M and CDAC-M Systems.

CDAC-M system employs SS, source side reordering, and transliteration to cope with the challenges of agglutination and morphological divergence in the English-Tamil translation. However, our NIT-M system makes no attempt to handle such linguistic issues, hence the poor performance.

The remarkable performance of the NIT-M system for the English-Hindi and English-Punjabi language pairs owe to a large-sized corpus. Training using a large corpus ensures effective tuning of system parameters and generation of a sound translation model, which eventually leads to better predictions. Moreover, the performances of two teams are comparable in these two translation categories. Although the Tamil training corpus is also fairly large sized, agglutination and morphological divergence cause the NIT-M system to lag behind in this translation category.

Furthermore, for English-Punjabi translation, the human and BLEU evaluation scores are negatively correlated, with low BLEU score and high manual evaluation scores. This is attributed to the underlying working principle of BLEU, which relies on the precision of n-grams in the reference and candidate translation. Even minor lexical differences can cause a huge difference in n-gram precision, which eventually affects the BLEU score. However, such minor lexical differences are often considered insignificant from the perspective of human evaluation.

Moreover, the comparatively lower evaluation scores of the CDAC-M system for English-Punjabi translation can be attributed to the inability of preprocessing and post-processing steps to handle the language-specific constructs of Punjabi language.

5.3 Evaluation Results of Different Experimental Setups

Furthermore, we have examined different experimental setups, as mentioned in Section 4.2, to analyze a system's translation performance with respect to number of epochs, size of training data, and average length of sentences in the test dataset. We have used a multi-BLEU (http://www.statmt.org/moses/?n=Moses.SupportTools) evaluator, using 1-g precision, to compare Gold Data and predicted translations.

Figure 6 shows the BLEU score versus epoch plot for English-Hindi MT. The highest BLEU score of 52.54 is attained at epoch 18 and the BLEU score curve converges after epoch 16 (say, epoch_converge). The decreasing curve between epoch 6 and epoch 7 is presumably because of decay in the learning rate of the network, which

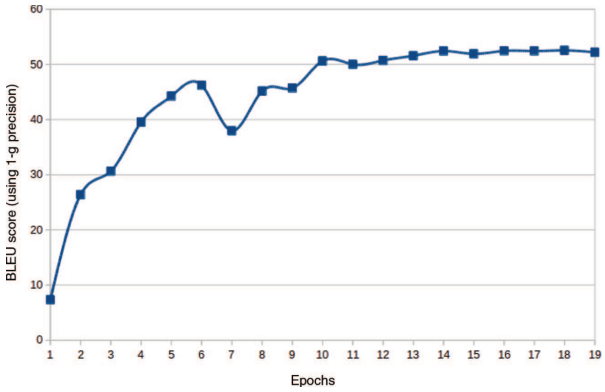

Figure 6: BLEU Score Achieved by NMT System at Different Epochs.

leads to poor translation prediction. The learning rate of network is decayed if the validation score improvement falls below zero. As the BLEU score versus epoch plot converges after a specific epoch (epoch_converge), the plot can be helpful in selecting the maximum number of training epochs. The number of training epochs should preferably not exceed the value *epoch_converge*.

Furthermore, Figure 7 shows a performance comparison between the NMT system trained using complete training data (NMT-I) and the one trained using half the training data (NMT-II). The high BLEU scores for the former indicate that the system's performance considerably improves with increase in the number of instances in the training data. As NMT-I has seen more number of sentence pairs during training, it predicts correct Hindi translation of many English words and generates less <*unk*> words in comparison to NMT-II.

To further analyze the system, we have selected the best-trained model, the one obtained at epoch 18 of the second experimental setup, and tested it using four distinct test sets. The four test sets (each of size 100) have been derived from the original test set provided by organizers, in such a way that the average lengths of sentences in these sets are 10, 15, 20, and 25. Sentences longer than 25 words have been accommodated in the fourth test set. Also, sentences having an average length of less than 10 words have been dropped. Figure 8 shows the BLEU scores achieved by the NMT system for the four test sets.

It has been interesting and uncommon to observe that the performance of the NMT system (in terms of BLEU score) improves with increase in the length of test sentences. It probably owes to the context-analyzing ability of the NMT system. Use of attention mechanism and context vectors facilitate improvement in translation performance with increase in length of test sentences. At each translation step, decoder uses global attention to investigate the entire pool of source states. Context vector generated from source states, current hidden state of decoder, and partial target text are then used by decoder for translation prediction of the current word. Thus, the entire decoding process in NMT relies on the context of the source sentence. As the larger source (test) sentences are context rich, their predicted translations are better in quality.

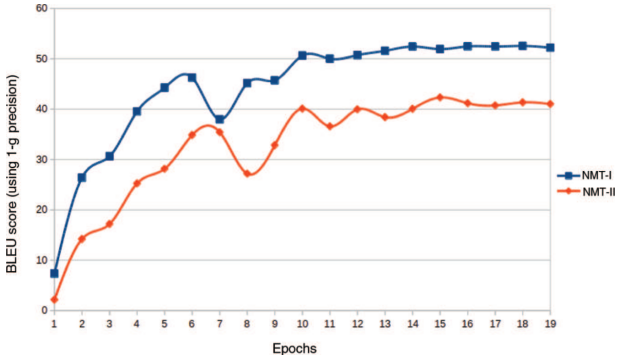

Figure 7: BLEU Score Achieved by NMT-I and NMT-II at Different Epochs.

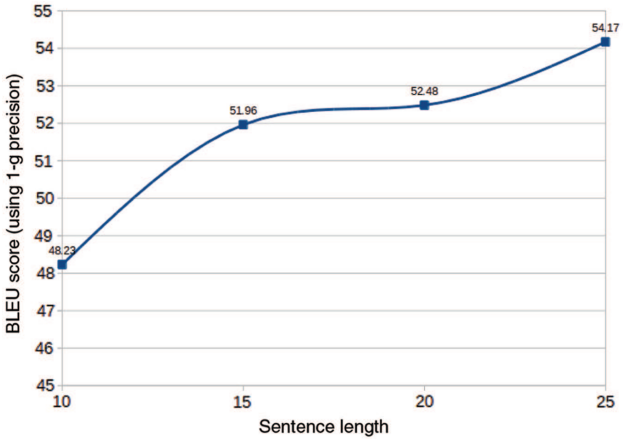

Figure 8: BLEU Score Achieved by NMT System for Different Sentence Lengths.

6 Conclusion and Future Scope

NMT, a fairly new approach to MT, uses an adequately trained end-to-end neural network system for translation prediction. The ability to produce fluent translation, better context-analyzing abilities, and improved performance over the SMT system are some of the key benefits that motivated us to explore the usage of NMT in the context of Indian languages. In particular, we have trained and tested NMT systems for English-Tamil, English-Hindi, and English-Punjabi language pairs. Predicted translations were provided to MTIL organizers for human and BLEU evaluations whereby translation quality was evaluated on the grounds of adequacy, fluency, and an overall rating. Besides, different experimental setups have been designed to analyze the change in the translation performance of the English-Hindi NMT system with change in the number of epochs, training data, and length of test sentences. A close analysis of predicted translations guide us to the conclusion that NMT systems produce fluent translations and their performance improves with increase in training data and length of test sentences. Moreover, a translation performance versus epoch plot can be helpful in checking the convergence of system training.

The underlying working idea of NMT relies heavily on the size of the training corpus, which directs us to increase the number of instances in the training corpus. The effectiveness of translation is largely determined by attention mechanism and the score function used for computing the attention of each source state. The score function can be modified to increase the interaction between the source state vector $\overline{h}_s$ and the current hidden state of decoder h_t. Besides, a better grasp over the crux of target language constructs can help improve the comprehensibility, adequacy, and fluency of translation. Furthermore, a skillful and careful selection of values for system parameters such as number of epochs, hidden layers, GPUs, etc., can also add to translation quality.

Acknowledgment: The work presented here falls under Research Project Grant No. YSS/2015/000988, Funder Id: 10.13039/501100001843 and partially supported by the Department of Science & Technology (DST) and Science and Engineering Research Board (SERB), Government of India. The authors would like to acknowledge the Department of Computer Science & Engineering, National Institute of Technology Mizoram, India, for providing infrastructural facilities and support.

Bibliography

[1] R. B. Allen, Several studies on natural language and back-propagation, in: *Proceedings of the IEEE First International Conference on Neural Networks*, 2, IEEE Piscataway, NJ, pp. 335–341, San Diego, California, 1987.

[2] A. Chandola and A. Mahalanobis, Ordered rules for full sentence translation: a neural network realization and a case study for Hindi and English, *Pattern Recogn.* **27** (1994), 515–521.

[3] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1724–1734, Doha, Qatar, 2014.
[4] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, Convolutional sequence to sequence learning, in: *Proceedings of the 34th International Conference on Machine Learning*, pp. 1243–1252, Sydney, Australia, 2017.
[5] V. Goyal and G. S. Lehal, Web based Hindi to Punjabi machine translation system, *J. Emerg. Technol. Web Intell.* **2** (2010), 148–151.
[6] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes and J. Dean, Google's multilingual neural machine translation system: enabling zero-shot translation, *Trans. Assoc. Comput. Linguist.* **5** (2017), 339–351.
[7] G. S. Josan and G. S. Lehal, A Punjabi to Hindi machine translation system, in: *22nd International Conference on Computational Linguistics: Demonstration Papers*, COLING '08, pp. 157–160, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008.
[8] N. Kalchbrenner and P. Blunsom, Recurrent convolutional neural networks for discourse compositionality, in: *Proceedings of the 2013 Workshop on Continuous Vector Space Models and their Compositionality*, pp. 119–126, Sofia, Bulgaria, 2013.
[9] S. Khan and R. B. Mishra, A neural network based approach for English to Hindi machine translation, *Int. J. Comput. Appl.* **53** (2012), 50–56.
[10] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. Rush, OpenNMT: open-source toolkit for neural machine translation, in: *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72, Association for Computational Linguistics, Vancouver, Canada, 2017.
[11] M. A. Kumar, V. Dhanalakshmi, K. P. Soman and S. Rajendran, Factored statistical machine translation system for English to Tamil language, *Pertan. J. Soc. Sci. Hum.* **22** (2014), 1045–1061.
[12] M. A. Kumar, S. Rajendran and K. P. Soman, Cross-lingual preposition disambiguation for machine translation, *Proc. Comput. Sci.* **54** (2015), 291–300.
[13] M. A. Kumar, B. Premjith, S. Singh, S. Rajendran and K. P. Soman, An overview of the shared task on machine translation in Indian languages (MTIL) – 2017, *J. Intell. Syst.* **28** (2019), 455–464.
[14] M.-T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, 2015.
[15] V. K. Menon, S. Rajendran and K. P. Soman, A synchronised tree adjoining grammar for English to Tamil machine translation, in: *International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015*, pp. 1497–1501, Kerala, India, 2015.
[16] R. Narayan, S. Chakraverty and V. P. Singh, Quantum neural network based machine translator for Hindi to English, *Appl. Soft Comput.* **38** (2016), 1060–1075.
[17] S. Pal, P. Pakray and S. K. Naskar, Automatic building and using parallel resources for SMT from Comparable Corpora, in: *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, pp. 48–57, Gothenburg, Sweden, 2014.
[18] 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 of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 311–318, Philadelphia, PA, 2002.
[19] Raj Nath Patel, Prakash B. Pimpale and M. Sasikumar, Machine translation in Indian languages: challenges and resolution, *J. Intell. Syst*. **28** (2019), 437–445.
[20] C. Poornima, V. Dhanalakshmi, K. M. Anand and K. P. Soman, Rule based sentence simplification for English to Tamil machine translation system, *Int. J. Comput. Appl.* **25** (2011), 38–42.
[21] K.-Y. Su and J.-S. Chang, Why corpus-based statistics-oriented machine translation, in: *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, pp. 249–262, Montreal, Canada, 1992.
[22] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, in: *Proceedings of Advances in Neural Information Processing Systems*, pp. 3104–3112, Montreal, Canada, 2014.
[23] H. Xiong, Z. He, X. Hu and H. Wu, Multi-channel encoder for neural machine translation, *arXiv preprint arXiv:1712.02109* (2017).